

Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Сибирский государственный аэрокосмический  
университет имени академика М.Ф.Решетнева»

На правах рукописи



Казаковцев Лев Александрович

**МЕТОД ЖАДНЫХ ЭВРИСТИК  
ДЛЯ СИСТЕМ АВТОМАТИЧЕСКОЙ ГРУППИРОВКИ ОБЪЕКТОВ**

05.13.01 – Системный анализ, управление и обработка информации  
(информатика, вычислительная техника, управление)

**ДИССЕРТАЦИЯ**

на соискание ученой степени доктора технических наук

Научный консультант  
доктор технических наук, профессор  
Антамошкин А.Н.

Красноярск – 2016

## Оглавление

ВВЕДЕНИЕ .....	6
Глава 1. Современное состояние развития алгоритмов автоматической группировки объектов с большим объемом входных данных .....	14
1.1 Общая постановка задач автоматической группировки объектов и сферы их применения.....	14
1.2 Основные подходы к решению задач группировки объектов и данных.....	18
1.3 Задачи автоматической группировки объектов и теория размещения.....	31
1.4 Эволюционные и иные стратегии глобального поиска .....	39
1.5 Жадные эвристические процедуры в эволюционных алгоритмах .....	45
Выводы к Главе 1 .....	48
Глава 2. Применение жадных эвристических алгоритмов к дискретным задачам автоматической группировки объектов и монотонной псевдодобулевой оптимизации .....	51
2.1 Постановка задачи автоматической группировки на сети .....	51
2.2 Известные алгоритмы группировки узлов сети.....	54
2.3 Алгоритм метода изменяющихся вероятностей .....	59
2.4 Результаты метода изменяющихся вероятностей, настройка параметров .....	66
2.5 Сравнительные результаты метода изменяющихся вероятностей .....	69
2.6 Результаты комбинированных методов.....	71
2.7 Параллельная версия алгоритма метода изменяющихся вероятностей .....	76
2.8 Жадная агломеративная эвристическая процедура для задач псевдодобулевой оптимизации и некоторые свойства задач размещения .....	79
2.9 Постановка задачи составления расписаний загрузки производственных мощностей литейно-прокатного и химического производства .....	85
2.10 Алгоритм с жадной эвристической процедурой для задачи о составлении расписания.....	92
2.11 Результаты вычислительных экспериментов для задачи о составлении расписания.....	100
Результаты и выводы к Главе 2. ....	105

Глава 3. Метод жадных эвристик для непрерывных и дискретных задач автоматической группировки объектов .....	107
3.1 Общая постановка непрерывных задач автоматической группировки объектов .....	107
3.2 Известные методы.....	114
3.3 Модификация жадной эвристической процедуры – частичное объединенное решение .....	121
3.4 Дальнейшие модификации жадной эвристики .....	126
3.5 Вычислительные эксперименты с модификациями жадных эвристик в составе генетического алгоритма .....	133
3.6 Применение новых модификаций жадных эвристических процедур к дискретным задачам размещения .....	138
3.7 Детерминированный алгоритм с жадной эвристической процедурой.....	148
3.8 Адаптивный метод жадных эвристик .....	153
3.9 Комбинация жадных эвристических алгоритмов с альтернативными алгоритмами локального поиска.....	160
3.10 Модификации для решения серии задач .....	165
3.11 Общая схема метода жадных эвристик .....	168
Результаты и выводы к Главе 3. ....	175
Глава 4. Алгоритм и система автоматической группировки электрорадиоизделий космического применения по производственным партиям .....	178
4.1 Общая постановка задачи .....	178
4.2 Типы испытаний электрорадиоизделий .....	181
4.3 Особенности комплектации космических аппаратов электрорадиоизделиями зарубежного производства .....	185
4.4 Проблема создания специальных партий ЭРИ .....	188
4.5 Методы машинного обучения и анализа данных для задачи автоматической группировки объектов электрорадиоизделий.....	194
4.6 Проблема сопоставимости точности входных данных и результата .....	197
4.7 Влияние выбора меры расстояния на оценку состава производственных партий .....	202
4.8 Способы оценки числа групп и обоснования результатов группировки ЭРИ	204

4.9 Проблема неустойчивости испытаний ЭРИ и проверка результатов автоматической группировки.....	211
4.10 Экспериментальные исследования .....	216
4.11 Общая схема метода принятия решений по комплектации электронных узлов КА.....	222
Заключение к Главе 4 .....	230
Глава 5. Задачи автоматической группировки.....	232
и размещения со специальными метриками и мерами расстояния .....	232
5.1 Теория размещения и функции расстояний .....	233
5.2 Классы функций расстояния.....	238
5.3 Свойства задачи Вебера с прямоугольной метрикой.....	247
5.4 Алгоритм для задачи с метрикой лифта .....	252
5.5 Алгоритм для задачи с метрикой французской столицы (French Metro).....	256
5.6 Применение и примеры для моделей размещения с метриками лифта и French metro, оценка вычислительной сложности алгоритмов.....	263
5.7 Обобщение метрик лифта и французской столицы, алгоритм для множественной задачи Вебера.....	268
5.8 Алгоритм для простейшей множественной задачи размещения .....	274
5.9 Задачи размещения с метриками, основанными на угловых расстояниях .....	280
5.10 Алгоритм для задачи Вебера с метрикой подъемного крана .....	284
5.11 Алгоритм для задачи Вебера с метрикой Москвы .....	290
5.12 Алгоритм для задачи размещения с метрикой британской железной дороги .....	296
5.13 Задача Вебера с мерой расстояния, включающей минимальную стоимость транспортировки.....	297
5.14 Задача Вебера на области, ограниченной дугами.....	310
Выводы к Главе 5 .....	318
Глава 6. Задачи автоматической группировки и размещения с произвольной мерой расстояния.....	320
6.1 Постановка задач .....	320
6.2 Существующие методы.....	323
6.3 Постановка задачи в дискретных координатах .....	324

6.4 Последовательная реализация алгоритма и его OpenMP-параллелизация.....	326
6.5 Результаты экспериментов.....	330
6.6 Практический пример: задача размещения точек доступа беспроводной сети .....	332
6.7 Математическая модель размещения точек доступа .....	336
6.8 Настройка параметров модели, результаты .....	343
Выводы к Главе 6 .....	346
ЗАКЛЮЧЕНИЕ .....	348
СПИСОК ЛИТЕРАТУРЫ .....	351
Приложение А. Исходные данные задач автоматической группировки электрорадиоизделий.....	393
Приложение Б. Сравнение работы различных алгоритмов для задачи автоматической группировки электрорадиоизделий.....	395
Приложение В. Описание программы классификации элементной базы космического применения .....	425
Приложение Г. Акты об использовании результатов .....	427

## ВВЕДЕНИЕ

Актуальность настоящей работы обусловлена ростом и бурным развитием систем искусственного интеллекта, использующих, в частности, методы автоматической группировки данных, а также повышением общих требований к экономической эффективности, достигаемой, в том числе, за счет применения моделей оптимального размещения и автоматической группировки объектов по территориальному принципу или по сходству характеристик.

Прогресс в развитии методов автоматической группировки объектов и данных, наиболее популярные из которых основаны на решении  $p$ -медианной и подобных ей задач ( $k$ -средних,  $k$ -медоид) в основном направлен на повышение их быстродействия. Поиск универсального и при этом точного метода решения подобных задач с большим объемом входных данных давно признан бесперспективным. Усилия исследователей сосредоточены на развитии компромиссных эвристических алгоритмов, дающих достаточно быстрое решение с минимальной ошибкой. Под эвристическим алгоритмом или процедурой, называемыми в специализированной литературе также словом «эвристика», будем понимать алгоритм (или его часть), не имеющий строгого обоснования, но дающий приемлемое решение задачи для большинства практических случаев. Наиболее распространены рандомизированные (использующие генератор случайных значений) процедуры, в связи с чем их результат зависит от значений случайных величин, от порядка следования данных и малых колебаний значений входных данных. В частности, рандомизированными алгоритмами, как правило, являются эволюционные (генетические) алгоритмы, основанные на применении эволюционных операторов – процедур, реализующих принципы биологической эволюции: селекции, скрещивания (рекомбинации), мутации. Эвристическими являются и так называемые «жадные» («пожирающие») алгоритмы, в которых на каждой итерации выбирается наилучшее решение из некоторого подмножества промежуточных решений. В то же время, некоторые из практических задач автоматической группировки требуют решения, которое не только было бы близко к точному решению задачи, но и являлось бы стабильным при многократных запусках алгоритма, воспроизводимым и, следовательно, проверяемым. Задачи должны решаться в интерактивном режиме при ограниченном времени работы с большим объемом входных данных. К таким задачам

можно отнести, например, задачу формирования специальных партий электрорадиоизделий космического применения, где необходимость получения стабильных результатов обусловлена требованием воспроизводимости и проверяемости результатов расчетов, являющихся частью производственного процесса, в который вовлечены две стороны с не совпадающими интересами.

Наибольший вклад в теорию размещения и автоматической группировки объектов внесли Дрезнер, Бримберг, Младенович (задачи размещения), Хакими (задачи на сети), Трубин (задачи Вебера в прямоугольной метрике), Весоловский (широкий круг задач), Лав (непрерывные задачи с различными метриками). Наиболее полные монографии по известным моделям и методам теории размещения составлены под редакцией Дрезнера и Хамахера (2004), Фарахани и Хекматфара (2009). Значительный вклад внесли Чен, Станимирович, Даскин, Питсулис, Пардалос, Резенде и др. Одними из первых в СССР модели размещения предприятий исследовали Черенин, Хачатуров. Модели размещения использовались и как основа моделей стандартизации и унификации, схожих с моделями, применяемыми в задачах автоматической группировки объектов, например, Институтом математики им. С.Л. Соболева СО РАН. Работы Береснева, Гимади, Колоколова а позже Кочетова, Забудского, Еремеева и др. создали фундамент для разработки программно-математического аппарата решения этих задач.

Детерминированные алгоритмы автоматической группировки и размещения существуют с середины XX века (работы Хакими). В то же время, вычислительная сложность этих алгоритмов, использующих, в частности, жадные агломеративные и диссоциативные эвристические процедуры, экспоненциально зависит от объема входных данных. Использование агломеративных жадных эвристических процедур в составе эволюционных алгоритмов для задач размещения позволяет получить хорошие по точности и стабильности результаты, но при этом эффективность алгоритма ограничена объемом входных, и их применение является обоснованным лишь для относительно узкого круга задач. Авторы исследований в этой области предлагают использование жадных эвристических процедур либо исключительно в виде самостоятельного алгоритма или самостоятельного эволюционного оператора (для дискретных задач размещения), либо в комбинации с отдельными известными методами локального поиска для данных задач. Тем не менее, точность и стабильность

результатов получаемых жадными эвристическими процедурами, дает обоснованную надежду на построение эффективных алгоритмов, обладающих свойством стабильности получаемых результатов, пригодных к применению в интерактивных автоматизированных системах, предназначенных для решения задач автоматической группировки и размещения.

Для непрерывных задач размещения разработаны алгоритмы лишь для наиболее распространенных метрик и мер расстояния. В то же время, мера расстояния, используемая в практических задачах автоматической группировки объектов в географическом/геометрическом пространстве зависит от особенностей пространства и средств транспортировки. Этим обусловлена необходимость расширения арсенала используемых моделей и построения универсальных методов решения задач размещения с различными мерами расстояния (например, заданными алгоритмически), а также необходимость усовершенствования методов решения задач группировки с большим объемом входных данных.

Идея настоящей диссертации состоит в разработке метода эффективного комбинирования алгоритмов локального поиска, стратегий глобального поиска, методов агрегирования данных, алгоритмов решения задачи Вебера с жадными агломеративными эвристическими процедурами.

Объектом диссертационного исследования являются задачи автоматической группировки и размещения объектов с большим объемом входных данных, предметом исследования – методы и алгоритмы их решения.

Цель исследования состоит в повышении точности и стабильности результатов при потоковом решении дискретных и непрерывных задач автоматической группировки и размещения с различными мерами расстояния при большом объеме входных данных.

Поставленная цель достигается путем решения следующих задач.

1. Провести анализ и систематизацию существующих методов локального поиска, стратегий глобального поиска, эвристических алгоритмов, в том числе эволюционных операторов, а также их комбинаций, применяемых для решения задач автоматической группировки объектов и задач размещения.

2. Исследовать применимость методов, включающих жадные эвристические процедуры, изначально разработанных для решения дискретных задач размещения, к



решению классов задач оптимизации в псевдодобулевой постановке и разработать соответствующие алгоритмы.

3. Исследовать применимость и эффективность методов с жадной эвристической процедурой, изначально разработанных для решения дискретных задач автоматической группировки и размещения объектов, для потокового решения задач автоматической группировки данных в непрерывном пространстве при различных постановках задач и мерах расстояния.

4. Исследовать и систематизировать свойства функций расстояния, используемых в моделях размещения и автоматической группировки объектов, разработать алгоритмы решения задач размещения и автоматической группировки объектов для моделей с нестандартными мерами расстояния.

5. Разработать эффективный эвристический метод потокового решения задач автоматической группировки, размещения объектов и аналогичных задач оптимизации, позволяющий использовать широкий диапазон стратегий глобального поиска (мультистарт, ГА, случайный поиск с адаптацией, детерминированные методы) в комбинации с известными методами локального поиска, обеспечивающий стабильный результат с минимальной ошибкой за приемлемое время.

Методы исследования. Основные теоретические и прикладные результаты получены с применением методов системного анализа, исследования операций, теории оптимизации, теории вероятностей, аналитической геометрии, параллельного программирования.

Новые научные результаты и положения, выносимые на защиту:

1. Впервые предложен метод потокового решения непрерывных и дискретных задач автоматической группировки и размещения объектов с различными мерами расстояния и монотонной псевдодобулевой оптимизации – метод жадных эвристик. Метод основан на комбинации жадных агломеративных эвристических процедур (как известных, так и новых модификаций), различных стратегий глобального поиска (эволюционный алгоритм, МИВЕР – метод изменяющихся вероятностей, детерминированные методы, мультистарт), методов локального поиска и методов решения задачи поиска центра группы (задачи Вебера с соответствующей мерой расстояния). Разработанный метод позволяет синтезировать алгоритмы,

обеспечивающие стабильный и достаточно точный результат за приемлемое время для задач с большим объемом входных данных.

2. Разработанные новые модификации жадных агломеративных эвристических процедур, эффективны в составе оптимизационных алгоритмов метода жадных эвристик, для решения непрерывных и дискретных задач автоматической группировки, размещения объектов и псевдобулевой монотонной оптимизации.

3. Новая жадная агломеративная эвристическая процедура с вещественным алфавитом для потокового решения непрерывных задач автоматической группировки, может эффективно применяться как в качестве компонента метода жадных эвристик, так и в качестве самостоятельного алгоритма.

4. Разработанный новый алгоритм МИВЕРа для решения задач автоматической группировки объектов и задач размещения на сети в качестве средства инициализации эволюционных оптимизационных алгоритмов позволяет ускорить работу последних.

5. Впервые предложенные алгоритмы решения задачи Вебера с метриками, основанными на угловых расстояниях (метрики Москвы, FrenchMetro и др.), метрикой лифта и аналогичных, а также новый алгоритм решения задачи Вебера с мерой расстояния на основе минимальной фиксированной стоимости транспортировки, эффективны в качестве самостоятельных средств решения соответствующих задач размещения, а также в качестве компонента метода жадных эвристик при решении задач автоматической группировки.

6. Разработанный новый алгоритм метода жадных эвристик для потокового решения задач оптимальной загрузки мощностей литейно-прокатных и химических производств позволяет решать поставленные задачи в интерактивном режиме.

7. Впервые предложенный метод автоматической классификации электрорадиоизделий позволяет в условиях специализированных тестовых центров эффективно решать задачи выделения однородных производственных партий изделий, что в свою очередь позволяет запустить производство новой продукции – спецпартий с повышенными требованиями к эксплуатационным характеристикам, в том числе космического применения.

8. Предложенный новый алгоритм МИВЕРа для приближенного решения задач автоматической группировки и размещения объектов с произвольной мерой расстояния, в частности, заданной алгоритмически, позволяет решать практические задачи с точностью, приемлемой для инженерных расчетов.

Значение для теории. Результаты исследования дополняют арсенал эффективных эвристических методов решения NP-трудных задач автоматической группировки и размещения с широким кругом используемых мер расстояния. Разработаны новые виды жадных эвристических процедур и способы их комбинации со стратегиями глобального и локального поиска. Пополнен арсенал точных методов теории размещения, расширен круг используемых при этом моделей с нестандартными мерами расстояния, создана основа для синтеза новых моделей и методов. Кроме того, результаты систематизируют знания в области применения жадных агрегативных эвристических процедур в комбинации с другими известными методами оптимизации для решения задач автоматической группировки объектов, задач размещения и монотонной псевдодобулевой оптимизации.

Практическая ценность методов решения задач автоматической группировки и задач размещения обусловлена широтой их применения как непосредственно в практических задачах об оптимальном пространственном размещении и группировке объектов в географическом/геометрическом смысле (элементов городской инфраструктуры, точек доступа и базовых станций беспроводных сетей, коммутационного оборудования проводных сетей, складов, товаров на складах, точек сервиса, аварийных служб, датчиков мониторинга окружающей среды и др.), так и опосредованно в задачах автоматической группировки данных, кластерного анализа, теории оценивания и др. Разработанный метод позволяет повысить точность решения таких задач с большим объемом входных данных за ограниченное время, а также повысить стабильность получаемых результатов.

Для некоторых классов задач с нестандартными мерами расстояния разработан единый подход, основанный на декомпозиции в серию задач с прямоугольной метрикой. Дальнейшее обобщение задачи Вебера, а также решение ее особых случаев существенно расширяет возможности использования моделей и методов непрерывной теории размещения в практически важных задачах размещения и автоматической группировки.

Показано, что алгоритмы метода жадных эвристик применимы и в других областях, что продемонстрировано на практическом примере задачи оптимальной загрузки производственных мощностей литейно-прокатных производств или производств изделий из пластмасс с разнородной выпускаемой продукцией. Эффективность алгоритма при этом позволяет решать задачи в реальном времени, тогда как предложенные ранее алгоритмы не обеспечивают этой возможности.

Практическая реализация результатов: алгоритм решения задачи оптимальной загрузки производственных мощностей литейно-прокатных и химических производств лег в основу программы для ЭВМ, позволяющей решать задачи в реальном времени, в отличие от предыдущих разработок, что позволило реализовать данный алгоритм в составе автоматизированной подсистемы составления производственного расписания производства полипропиленовых изделий с повышенными эксплуатационными характеристиками ООО «ИСТ» (г.Красноярск). Метод оптимального размещения на сети использован ФГБУ ВНИИ ГОЧС МЧС России при разработке математических моделей для оптимального размещения пунктов экстренного реагирования МЧС на территории Красноярского края. Метод автоматической классификации электрорадиоизделий по производственным партиям лег в основу СППР автоматической классификации ОАО ИТЦ – НПО ПМ (г.Железногорск).

Апробация. Основные положения и результаты работы докладывались и прошли всестороннюю апробацию на международных и всероссийских научных и научно-практических конференциях и семинарах. В их числе:

- IV, VI International Congress on Ultra Modern Telecommunications and Control Systems (2012, 2014гг., г.Санкт-Петербург),
- Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation (EMS, 2012г., Мальта),
- UKSim-AMSS Seventh European Modelling Symposium on Computer Modelling and Simulation (EMS, 2013г., г.Манчестер, Великобритания),
- ICTP Workshop on Optimization Technologies for Low-Bandwidth Networks (2006г., г.Триест, Италия),
- Relevance of Low Cost Wireless ICT Solutions for Development: an Awareness Workshop (2008г., г.Триест, Италия),
- ICTP Second Workshop on Open Source and the Internet for Building Global Scientific Communities (2011, г.Триест, Италия),
- Пятая Международная конференция «Системный анализ и информационные технологии» (САИТ-2013, 2013г., г.Красноярск),
- International Workshop on Mathematical Models and their Applications (IWMMMA, 2014г., г.Красноярск),
- Southern African Southern Mathematical Association International Conference (SAMSA 2012, г.Лилонгве, Малави),

- International Conference on Frontiers of Mathematical Sciences with Applications (ICFMSA-2012, г.Калькутта, Индия),
- XIII Международная научно-практическая конференция «Перспективы развития информационных технологий (2013г., г.Новосибирск),
- VI Международная конференция «Проблемы оптимизации и экономические приложения» (2015, г.Омск),
- Шестая Международная конференция «Системный анализ и информационные технологии» (САИТ-2015, 2015г., г.Светлогорск) и др.

Научные и научно-технические результаты были получены в рамках тематических планов НИР СибГАУ (2012-2014гг.), хозяйственного договора между СибГАУ и ОАО «ИСС- НПО ПМ». Работа была шесть раз поддержана грантами Международного центра теоретической физики им.Абдуса Салама (ICTP Abdus Salam, г.Триест, Италия) при ЮНЕСКО и МАГАТЭ. Основные результаты исследований были отмечены Правительством и Законодательным собранием Красноярского края Государственной премией Красноярского края в области профессионального образования в 2014 году.

Диссертационная работа в целом обсуждалась на семинаре “International Workshop on Mathematical Models and their Applications” (IWMMMA, 2014г., г.Красноярск), на семинаре «Математические модели принятия решений» (г.Новосибирск, Институт математики им. С.Л.Соболева, 2015 г.), на международной конференции «Проблемы оптимизации и экономические приложения» (2015, г.Омск).

Публикации. По материалам диссертации опубликовано 50 работ, в том числе 19 статей в российских рецензируемых периодических изданиях, рекомендуемых ВАК РФ для опубликования основных научных результатов диссертационных исследований, и 19 статей в зарубежных изданиях, включенных в международные базы цитирования Web of Science, Scopus и Mathematical Reviews. В Роспатенте зарегистрированы 5 программ для ЭВМ.

Структура работы. Диссертация состоит из введения, шести глав, заключения, приложений и списка литературы из 547 наименований.

## **ГЛАВА 1. СОВРЕМЕННОЕ СОСТОЯНИЕ РАЗВИТИЯ АЛГОРИТМОВ АВТОМАТИЧЕСКОЙ ГРУППИРОВКИ ОБЪЕКТОВ С БОЛЬШИМ ОБЪЕМОМ ВХОДНЫХ ДАННЫХ**

В Главе 1 проведен анализ современного уровня развития методов автоматической группировки. Показана тесная взаимосвязь задач автоматической группировки объектов в непрерывном либо дискретном пространстве признаков с задачами теории размещения, проанализировано современное состояние теории размещения. Обозначены проблемы, возникающие при потоковом решении задач автоматической группировки объектов в системах, требующих получения достаточно точного и стабильного решения.

### **1.1 Общая постановка задач автоматической группировки объектов и сферы их применения**

Проникновение в сферу анализа данных некоторых идей из теории искусственного интеллекта способствовало появлению и бурному развитию такого направления в обработке информации, как интеллектуальный анализ данных (Data Mining), моделирующий поведение человека, занятого решением новых для него сложных задач обобщения, выявления закономерностей и ассоциаций [1].

Достижения в технологиях автоматического сбора данных, передачи и хранения информации, а также взрывной технологический рост в таких отраслях, как технологии поиска информации в сети Интернет, передача цифровых изображений и видеонаблюдение, привели к появлению гигантских массивов многомерных данных. Человечество, по скромным оценкам, потребило порядка 1000000 терабайт данных в 2011 году [2, 3], и это число неуклонно растет в геометрической прогрессии. Большинство данных хранится в цифровой форме (текст, изображение и видео, данные телеметрии и т.п.), таким образом обеспечивая огромный потенциал для развития автоматического анализа данных, их автоматической группировки и поисковых методов. Электронные письма, блоги, учетные данные оперативной деятельности предприятий и огромное количество веб-страниц ежедневно создают терабайты новых данных, как структурированных, так и слабоструктурированных [3].

Увеличение объема и разнообразия данных требует новых достижений в методологии восприятия, обработки и обобщения данных. Методы анализа данных можно разделить на два основных типа [4]: исследовательский или описательный, предполагающий, что исследователь не пользуется предварительно определенными моделями и гипотезами, стремясь понять общие характеристики или структуру многомерных данных, и подтверждающий (логически выведенный) тип методов, при котором исследователь подтверждает адекватность модели или справедливость предположений на имеющихся данных.

Для анализа данных было предложено множество статистических методов, таких как дисперсионный анализ, линейная регрессия, дискриминантный анализ, анализ корреляции, многомерное шкалирование, факторный анализ, кластерный анализ [5]. В задачах распознавания образов анализ данных имеет целью построение прогнозной модели: по части данных следует предсказать поведение остальной части данных. Эта задача также называется обучением. Иногда выделяют два больших класса задач: обучение с учителем (классификация - в этом случае имеется обучающая выборка, в которой данные заранее отнесены к тому или иному классу) и без учителя (объединение в кластеры - заранее принадлежность к той или иной группе данных неизвестна даже для части данных, зачастую неизвестно и количество групп) [6]. В принципе, задачи автоматической группировки могут относиться к любому из этих классов: и в том, и в другом случае ставится задача разбиения множества объектов на сравнительно однородные группы, но объединение в кластеры - задача более трудная и сложная. Растет интерес к гибриднему подходу - полуконтролируемому обучению [7], при котором лишь для части обучающей выборки известна принадлежность к той или иной группе объектов (маркировка), а решение об отнесении остальных элементов к той или иной группе выносится с учетом близости объектов к этим маркированным объектам. При полуконтролируемой кластеризации вместо того, чтобы четко разбивать объекты на классы (маркировать объекты), определяются попарные ограничения, являющиеся более "слабым" способом закодировать предварительные знания. Попарные ограничения [8] сводятся к требованиям о том, что два элемента (объекта данных) обязаны быть объединены в одну группу (кластер), либо наоборот не могут входить в одну группу (кластер).

Цель автоматической группировки данных, также известной как кластерный анализ, состоит в том, чтобы обнаружить естественную группировку ряда образцов, пунктов или объектов. Словарь Вебстера [9] дает социологическое определение кластерного анализа как статистического метода классификации для определения того, попадают ли люди в различные группы путем количественного сравнения множества характеристик.

Целью решения задачи автоматической группировки является разработка алгоритма и/или автоматизированной системы, способных обнаруживать эти естественные группировки в не маркированных предварительно данных.

Более точно задачу можно определить следующим образом: учитывая представление  $N$  объектов, найти в них  $k$  групп (т.е. разбить их на  $k$  непересекающихся подмножеств), основываясь на некоей мере подобия таким образом, чтобы объекты, принадлежащие одной и той же группе, были подобны друг другу (обладали схожими характеристиками), а объекты, принадлежащие различным группам, были не схожи по своим характеристикам.

Такая формулировка порождает как минимум два вопроса: Первый — о количестве групп объектов  $k$ , второй — об используемой мере подобия.

Если мы используем метрическое определение подобия - расстояние в некотором пространстве признаков между объектами, то группы могут отличаться с точки зрения своей формы, размера и плотности (рис. 1.1). Присутствие шума в данных делает обнаружение групп еще более трудным. Идеальная группа может быть определена как ряд точек (объекты или элементы данных представлены точками в некотором пространстве характеристик), который компактен и изолирован. В действительности группа - сущность, восприятие которой зачастую субъективно, определение которой может требовать знаний в соответствующей области. Человек способен превосходно выделять группы в двух- и трехмерных данных, но число измерений данных может быть поистине колоссальным. Например, данные, группировке которых посвящена Глава 4 настоящей работы, имеют размерность от нескольких десятков до тысяч измерений, что отнюдь не является пределом для других более сложных задач, которые продолжают появляться по мере прогресса в технологиях автоматической группировки. Развитие технологий группировки всегда являлось предметом междисциплинарных



исследователей таксономистов, социологов, психологов, биологов, медиков, математиков, программистов.

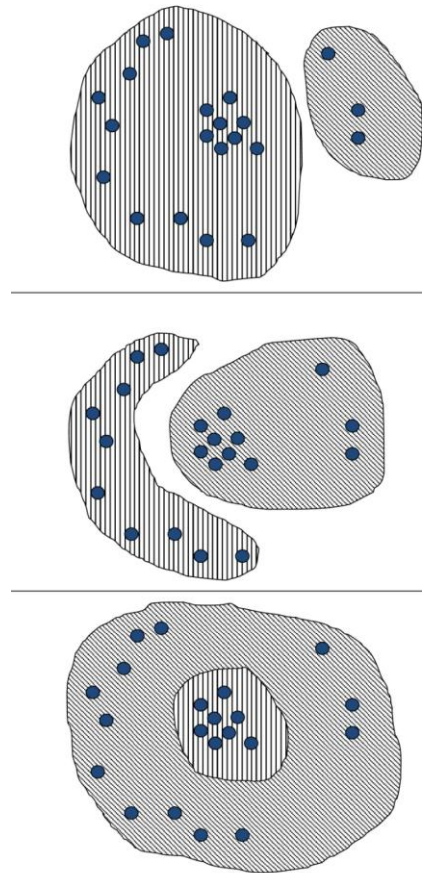


Рисунок 1.1 – группы (кластеры) различной формы

Автоматическая группировка может упоминаться как Q-анализ, типология, таксономия, объединение в группы и т.п., хотя эти понятия не являются полностью синонимами.

Задачи автоматической группировки могут возникать практически в любой дисциплине, предполагающей анализ многомерных данных. Трудно исчерпывающе перечислить многочисленные научные области, которые использовали методы группировки, а также тысячи изданных алгоритмов. Поиск в системах "Академия Google", РИНЦ (elibrary), Web of Science по ключевым словам "автоматическая группировка", "группировка данных" ("data grouping"), "кластеризация", "кластерный анализ" ("clustering", "cluster analysis") выдает тысячи наименований научной литературы, начиная с 1954 года [10]. Например, к автоматической группировке сводятся такие важные задачи, как сегментация изображения (в компьютерном зрении) [11-14], группировка документов для их эффективного поиска, быстрого доступа и

эффективного использования памяти при хранении [15-18], разбиение клиентов (потребителей) предприятия на группы в CRM-системах для организации эффективной маркетинговой деятельности [19, 20], группировка потенциальных клиентов сервисных точек по географической/геометрической близости для эффективной организации обслуживания, управления трудовыми ресурсами и планирования [21], задачи из области биологии [22, 23], задачи распознавания печатного [24] и рукописного [25] текста.

Автоматическая группировка может требоваться при естественной классификации (например, организмов в живой природе или неживых объектов, таких как производственные партии электрорадиоизделий, рассматриваемые в Главе 4), при структурировании данных, выделении аномалий в данных (например, при выявлении некачественной продукции в производственном процессе), для сжатия данных путем замены одинаковых либо очень близких по характеристикам объектов данных единственным объектом, являющимся их обобщенным (усредненным) представлением.

## **1.2 Основные подходы к решению задач группировки объектов и данных**

Анализ и классификация подходов, предложенных для решения задач автоматической группировки объектов и данных и кластерного анализа, является трудной задачей ввиду чрезвычайного их разнообразия. В различных подходах используются всевозможные меры сходства (обзор различных метрик и мер расстояния, а также попытка их классификации приведены в Главе 5), различные целевые функции (минимизация суммарного расстояния между объектами, минимизация суммарного расстояния до центров кластеров, минимизация максимальных расстояний в кластере и т.д.), различные эвристики и схемы локального и глобального поиска, необходимость применения которых обусловлена невыпуклостью целевых функций большинства таких задач (обзор и обобщение стратегий глобального и локального поиска, применяемых совместно с жадными агломеративными эвристическими алгоритмами представлены в Главе 3).

Группы можно определить как области высокой плотности в пространстве признаков (характеристик), разделенные областями низкой плотности. Алгоритмы, непосредственно основанные на этом определении, занимаются поиском связанных

областей высокой плотности в пространстве признаков, при этом различные алгоритмы используют различные определения связности. Алгоритм Джарвиса-Патрика и подобные ему алгоритмы определяют подобие между парой точек как число общих соседей, которых они разделяют, где соседи - точки, находящиеся внутри круга заданного радиуса вокруг точки, либо используют метод окна Парзена для определения плотности [26, 27]. Простейшими оценками плотностей можно считать гистограмму и полиграмму первого и более высокого порядков [28, 29]. Вклад в методы непараметрического оценивания плотности внесли Е. Парзен [30, 31], В.А. Епанечников [32], А.В.Медведев [33]. В работах этих и других авторов вводятся новые классы оценок, обобщающие гистограмму. Так, один из этих классов, называемых "ядерными оценками", был предложен Розенблатом и Парзенем. В [33-36] решается задача обеспечения несмещенности таких оценок, исследуется скорость сходимости отклонений, изучается влияние формы ядра на качество приближения оценки к функции плотности, а также рассматриваются вопросы определения параметров алгоритмов восстановления плотности — параметра размытости (сглаживания) непараметрических оценок плотности, например — с гауссовым ядром. Разработаны принципы построения систем автоматической группировки, в том числе — для задач с большим объемом входных данных, позволяющие обрабатывать данные за один проход [27].

В том или ином варианте, алгоритмы, использующие методы восстановления плотности, зависят от выбранного масштаба, в котором измерены расстояния, от количества точек, попадающих в окрестность друг друга, достаточного для того, чтобы определить такое скопление точек как группу, и от максимального расстояния, на которое должны быть удалены точки в группе. Выбор перечисленных параметров - отдельная, весьма сложная задача, решением которой определяется точность метода и адекватность модели автоматической группировки. Весьма существенная проблема всех подобных методов - «проклятие размерности» [37], связанное с экспоненциальным увеличением количества данных, необходимых для эффективной работы алгоритмов восстановления плотности, с ростом размерности пространства. Если число признаков велико и все они учитываются при относительно небольшом объеме данных, то все объекты оказываются примерно на одинаковом расстоянии друг от друга, вследствие этого подобные подходы практически неприменимы к многомерным данным.

Тем не менее, было предложено множество подходов, оперирующих вероятностными моделями плотности [38-40], таких как модель размещения Патинко [41] (PAM – Pachinco Allocation Model, не путать с PAM-процедурой для задачи  $k$ -медоид — Partition Around Medoid, рассматриваемой в Главе 3) и ненаправленная графическая модель [42]. Хотя такие методы так же чувствительны к выбору параметров, требовательны к вычислительным ресурсам и имеют ограничения на размерность данных, эти методы, в особенности основанные на непараметрических моделях плотности [40], достаточно популярны из-за их способности выделять группы (кластеры) произвольной формы. Если пространство признаков имеет очень высокую размерность, сравнимую с количеством группируемых объектов, данные (точки, объекты) в таком пространстве удалены друг от друга, как правило, весьма значительно, что делает практически невозможным построение более-менее универсальной процедуры выбора параметров этих алгоритмов, при которых они обеспечивали бы эффективную группировку. Существуют непараметрические алгоритмы, комбинированные с другими популярными методами, например, с методом  $k$ -средних [43], применимые, однако, лишь для задач с небольшим объемом входных данных (в [43] рассматриваются примеры с 2 кластерами).

Метод  $k$ -средних (альтернативное наименование — MSSC — Minimum Sum of Squares Clustering – группировка по минимуму суммы квадратов [44]) — один из классических, хорошо исследованных алгоритмов автоматической группировки (обучения без учителя). Как и другие методы, данный алгоритм пытается найти возможные классы (группы) объектов данных, организованных групп, элементы которых подобны в некотором роде [43]. Группа соответствует коллекции объектов, которые до некоторой степени "эквивалентны" друг другу, а также некоторому виртуальному объекту — центру (центроиду) кластера (группы). Задача автоматической группировки может быть сформулирована как модель распределения объектов в метрическом  $d$ -мерном пространстве между  $k$  группами так, чтобы в каждой группе суммарное расстояние от элементов группы до ее центра достигало минимума. Метод  $k$ -средних можно считать самым важным методом обучения без учителя [45, 43]. Метод обладает следующими потенциальными преимуществами: (1) позволяет учитывать признаки (характеристики) различных типов, (2), обнаруживать группы произвольной формы, (3) обрабатывать данные при минимальных требованиях к знанию предметной области, к

которой относятся входные параметры (характеристики группируемых объектов), (4) использование данных с шумами и выбросами. Вследствие этого метод находит широчайшее применение практически во всех сферах автоматической группировки и кластерного анализа [46].

Алгоритм включает всего два чередующихся шага [47, 48]: разбиение на группы (кластеры) вокруг известных центров (объект относится к той группе, центр которой является к нему ближайшим) и переопределение центров групп. Аналогичная методика используется также в алгоритме FOREL [49] и алгоритме *N*-means (*N*-средних).

Таким образом, *k*-средних — простой и быстрый алгоритм, применимый к широчайшему классу задач. Алгоритм *k*-средних имеет некоторые ограничения [50, 51], в частности — нужно заранее задать число групп *k*, на которые разбиваются объекты. В Главе 3 рассмотрены методы решения данной проблемы. Отметим, что метод жадных эвристик, которому в основном посвящена настоящая работа, позволяет решить одновременно серию задач с различными значениями *k*, частично данную проблему снимая. Отдельным направлением можно считать развитие методов разбиения объектов на две группы (более узкая по сравнению с темой настоящего исследования задача бикластеризации). Алгоритмы для решения этой задачи исследованы, в частности, А.В.Кельмановым [52, 53].

Группировка, основанная на теории графов, иногда называемая спектральным объединением в кластеры, представляет объекты (точки) данных как узлы во взвешенном графе. Ребра, соединяющие узлы, имеют длину (вес), равную или обратно пропорциональную мере сходства между объектами (точками). Идея состоит в том, чтобы разделить узлы на два подмножества *A* и *B* таким образом, чтобы величина разреза, т.е., сумма весов ребер, соединяющих узлы, находящиеся в множествах *A* и *B*, была минимальной. Алгоритмы, изначально предлагавшиеся для решения таких задач, часто приводили к разбиению на группы сильно различающегося (несбалансированного) размера, из-за чего в дальнейшем было предложено использовать ограничение на число точек в группе [54] и более сложные ограничения [13].

В [55] Ёном и др. был предложен подход, в котором новое представление данных получается из нормализованных собственных векторов ядра матрицы графа. Хофманом и Буманом [56] предложен детерминированный алгоритм имитации отжига для группировки данных, представленных с использованием мер сходства между объектами

данных. В [57] сформулирована задача попарного объединения в группы на основе максимальных доминирующих множеств [58], которые являются непрерывным обобщением клик в графе.

Методы на основе методов изменяющихся вероятностей и жадных агломеративных эвристик для задач автоматической группировки на сетях (графах) рассмотрены в Главе 2 настоящей диссертации. Заметим также, что метод и задача  $k$ -медоид ( $k$ -medoids), которая также может быть представлена как задача автоматической группировки или множественного размещения на графе, рассмотрены в Главе 3 настоящей диссертации.

Советские и российские исследователи традиционно формулируют задачи автоматической группировки и размещения как на сетях, так и в непрерывном пространстве, в виде задач целочисленного линейного программирования [59], причем, несмотря на NP-трудность задач, разработан широчайший арсенал в основном точных эффективных методов решения [60-66, 49], в том числе — для двухуровневых задач, в которых оценка значения целевой функции предполагает решение вложенной оптимизационной задачи. Тем не менее, для алгоритмов, эффективных при решении, например, экономических задач, NP-трудность решаемых задач становится проблемой при решении больших задач автоматической группировки с учетом взрывного роста объемов данных, собираемых и обрабатываемых в автоматизированных системах. Под NP-трудными задачами в теории вычислительной сложности алгоритмов понимают класс задач, которые «как минимум так же трудны, как самая трудная из задач класса NP». Иными словами, NP-трудными называют задачи, к которым можно свести любую задачу из класса NP за полиномиальное время (полиномиально зависящее от объема входных данных). В свою очередь, к классу NP относят задачи, решения которых могут быть проверены на недетерминированной машине Тьюринга за полиномиальное время. Такие задачи, как  $p$ -медианная на графе, кроме NP-трудности, обладают тем свойством, что за полиномиальное время в общем случае невозможно получить приближенное решение с константной оценкой точности [67, 68].

Методы иерархической группировки могут осуществлять группировку объектов в непрерывном пространстве характеристик, строя при этом древовидную модель взаимоотношений объектов, что сближает их в какой-то мере с методами автоматической группировки на графах [69]. Эти методы могут быть использованы для группировки при

достаточно больших объемах данных. Иерархические алгоритмы автоматической группировки рекурсивно выделяют вложенные группы агломеративно (начиная с рассмотрения каждого объекта в качестве отдельной группы, затем последовательно объединяя эти группы попарно, формируя таким образом иерархию этих групп) или применяя аналитический (нисходящий, диссоциативный) способ, начинающийся со всех объектов данных в одной группе и рекурсивно делящий каждую группу на меньшие группы. По сравнению с иерархическими алгоритмами объединения в группы, алгоритмы, подобные методу  $k$ -средних, находят все группы одновременно, не налагая иерархическую структуру. Входные данные иерархического метода группировки должны быть представлены матрицей подобия размера  $N \times N$ , где  $N$  - число группируемых объектов. Для методов, подобных  $k$ -средних, объекты располагаются в  $d$ -мерном пространстве характеристик, хотя некоторые разновидности алгоритмов, такие как  $k$ -медоид, могут оперировать и матрицей подобия. Матрица подобия может быть легко получена по координатам объектов в пространстве характеристик, обратное преобразование требует применения сложнейших методов, таких как многомерное шкалирование, которые сами по себе могут включать в себя многократное решение задач автоматической группировки [3].

Метод  $X$ -means [70] включает в себя агломеративные и диссоциативные приемы - попарное объединение или разъединение групп, позволяя таким образом находить оптимальное значение числа групп  $k$  на основе оценок такими критериями как информационный критерий Акаике (AIC - Akaike Information Criterion) или Байесов информационный критерий (BIC - Bayesian Information Criterion). В методе  $k$ -медоид [71] группировка осуществляется вокруг центров групп, в качестве которых могут выбираться исключительно группируемые объекты (не произвольные точки в пространстве характеристик).

Как и метод  $X$ -means, метод жадных эвристик, являющийся главной темой настоящей работы, включая в себя агломеративную стратегию последовательного (но не попарного) объединения групп, не строит при этом иерархической структуры, характерной для иерархических методов автоматической группировки. Метод  $X$ -means, позволяя одновременно с решением задачи  $k$ -средних определять оптимальное значение числа групп  $k$  для задач с относительно небольшим объемом входных данных, снижает точность результата и ни в коей мере не добавляет стабильности получаемым

результатам [27]. Использование таких критериев, как ВИС и АИС, учитывающих расстояния между объектами внутри групп, но никак не учитывающих расстояния между объектами принадлежащими разным группам, в практических задачах может давать неадекватную оценку числа групп (вследствие этого в Главе 4 для задачи группировки электрорадиоизделий по производственным партиям выбран оказавшийся более информативным критерий силуэта). Метод жадных эвристик, в то же время, позволяет решить одновременно серию задач с различными значениями  $k$ , результаты которых могут быть оценены в дальнейшем с применением любых критериев.

Некоторые алгоритмы автоматической группировки, такие, как метод минимальной энтропии [72] имеют формулировку, оперирующую понятиями информационной теории. Метод информационного узкого места или бутылочного горлышка (IBC - Information Bottleneck Clustering) [73-75] был предложен как обобщение теории зависимости искажения информации от скорости передачи и оперирует понятиями, характерными для алгоритмов сжатия данных с потерями. Например, этот метод применяется при группировке документов по ключевым словам [75]. IBC-методы начинают группировку, рассматривая каждый объект как отдельную группу (кластер), а затем исключают одну за другой группы таким образом, чтобы мера сходства, измеряемая в данном случае с применением понятий, характерных для информационной теории, оставалась максимальной для текущего числа групп. Такие методы чрезвычайно требовательны к вычислительным ресурсам.

Заметим, что методы, основанные на жадных агломеративных эвристических процедурах, имеют общие черты с методами информационного бутылочного горлышка, и в предельных случаях вырождаются в детерминированные методы, аналогичные либо тождественные методам информационного бутылочного горлышка (показано в Главе 3).

Кроме упомянутых выше вопросов о применяемой мере сходства, применяемой целевой функции (и связанного с ними вопроса о форме групп – кластеров) и вопроса о числе групп (кластеров), перед исследователем, сталкивающимся с задачей автоматической группировки, могут возникать и другие вопросы.

Так, представление данных - один из наиболее важных факторов, определяющих работу алгоритма группировки. Если представление данных (выбор обрабатываемых характеристик объектов) выбрано правильно, можно применить какой-либо из простейших алгоритмов, получив при этом компактные и изолированные друг от друга



группы [55]. Отсутствие важных характеристик делает разбиение на группы невозможным либо бессмысленным, присутствие "лишних" характеристик, несущественных с точки зрения конкретной задачи автоматической группировки, приводит к разбиению на более мелкие группы, не позволяя достичь требуемого уровня обобщения и структурирования данных. Если же используются характеристики, каждая из которых, будучи информативной, является зависимой от других характеристик, это, не искажая результата автоматической группировки, может весьма существенно повысить вычислительную сложность большинства алгоритмов автоматической группировки.

Хотя сам по себе алгоритм, часто называемый стандартной процедурой  $k$ -средних, является достаточно высокопроизводительной процедурой, для дальнейшего улучшения производительности этого алгоритма предложено множество различных методик [76-]. Алгоритм чувствителен к начальному выбору центров групп (кластеров), может сходиться к местным оптимумам [79]. Кроме того, когда число группируемых объектов очень велико (десятки тысяч и более), требуется большое количество времени для нахождения глобального оптимума или близкого к нему решения [80].

Хотя существуют «глобальные» методы решения задачи [77], претендующие на нахождение глобально оптимального решения (практически неприменимые к очень большим задачам и фактически точное решение не гарантирующие), основное направление современных исследований - развитие эвристических алгоритмов, дающих субоптимальные решения, близкие к истинному оптимуму задачи.

В [81] задействованы возможности генетического алгоритма (ГА) для улучшения результата  $K$ -средних. Как правило, в генетическом алгоритме промежуточные решения ("хромосомы") кодируются в виде множеств объектов данных, относящихся к той или иной группе (кластеру) или в виде множества объектов данных, выбранных в качестве начальных центров групп [82, 83], которые затем уточняются алгоритмом  $k$ -средних (стандартной процедурой  $k$ -средних).

В [45, 84] также описан алгоритм автоматической группировки, основанный на ГА. В данной стратегии глобального поиска промежуточное решение кодируется вещественными числами (ГА с вещественным алфавитом). Промежуточное решение представляет собой множество центров групп в  $d$ -мерном пространстве. Отметим, что в Главе 3 настоящей работы представлена версия ГА с вещественным алфавитом,

использующая жадную агрегативную эвристику в качестве процедуры кроссинговера (скрещивания), характерной для генетических алгоритмов. Существуют и иные схемы глобального поиска.

В [85] представлен модифицированный алгоритм К-средних (PSOKM) с использованием метода оптимизации роя частиц. Результаты этой работы показали возможность построения высокоточных классификаторов с более низким уровнем ошибочной классификации для весьма небольших по объему наборов данных (несколько сотен объектов).

Багировым и др. [86] предложено развитие модифицированного глобального алгоритма к-средних Ликаса и др. [77, 87], позволяющее получать точное решение быстрее предыдущих алгоритмов глобального поиска. На тестовых примерах доказана применимость к задачам с несколькими тысячами группируемых объектов, при этом для решения задачи требуется очень большое время счета и огромные объемы памяти. Тем не менее, NP-трудность задачи не позволяет применять данный алгоритм к задачам с очень большим объемом данных. Кроме того, несмотря на присутствие слова «глобальный» в названии метода, он не гарантирует нахождение истинного глобального минимума, хотя и демонстрирует сходимость к «очень хорошему» результату.

Метод К-средних, как правило (и по умолчанию), используется с квадратом евклидовой метрики для вычисления расстояния между центрами группы и объектами [3]. В результате К-средних находят круглые (шарообразные) группы в данных (т.е. имеющие форму круга с той метрикой или мерой расстояния, которая применяется [88]). Метрика Махаланобиса позволяет обнаружить гиперэллипсоидальные группы [89], что сказывается весьма существенно на вычислительных затратах. Вариант метода к-средних, использующий расстояние Итакуры-Саито [3] используется для векторной квантизации в речевой обработке [90]. Достаточно широко используется метод к-средних с расстоянием  $L_1$  (иначе называемым прямоугольной или манхэттенской метрикой) [88], позволяющий весьма существенно снизить влияние шумов на результат по сравнению с использованием квадратичных евклидовых расстояний практически без повышения вычислительной сложности. Группы (кластеры) при этом получаются менее компактными, что продемонстрировано на реальных данных в Главе 4. Используются и многие другие специальные метрики и меры расстояния, например, расстояния Брегмана [91]. Отметим, что, если в качестве меры расстояния используется метрика, то

постановка задачи  $k$ -средних в полной мере соответствует постановке одной из классических задач теории размещения – непрерывной задаче о  $p$ -медиане. Отметим, что квадратичное евклидово расстояние метрикой не является: не выполняется правило треугольника – квадрат длины отрезка  $AC$  может быть больше суммы квадратов длин отрезков  $AB$  и  $BC$ . В то же время, если считать, что в качестве меры расстояния в  $p$ -медианной задаче могут использоваться и другие меры расстояния (не метрики), то постановку задачи  $k$ -средних и  $p$ -медианной задачи следует считать эквивалентными. В случае, если выполняется группировка некоторых реальных объектов по мере близости друг к другу в реальном географическом/геометрическом пространстве, как правило, используются евклидовы расстояния [92] (в этом случае задача  $k$ -средних в полной мере соответствует классической непрерывной  $p$ -медианной задаче), хотя при группировке объектов в очень обширных географических регионах переходят к расстояниям на сфере. Метод  $k$ -средних предполагает получение относительно равновесных кластеров, хотя такое свойство не характерно для некоторых его разновидностей, иногда рассматриваемых как отдельные методы. Так, метод ISODATA [93] состоит в том, что, задавшись центрами  $K$  классов, размещают все объекты по группам (классам), используя вместо расстояния до центров групп внутриклассовый критерий минимальности. После отнесения всех элементов к группам центры пересматриваются, далее эти два шага чередуются [94, 93]. Иной способ отнесения объектов к группам предусмотрен в [95]. Здесь для определения ближайшего к объекту центра группы строится  $kd$ -дерево. Методы Kernel  $K$ -means ( $k$ -средних с выделением "ядра" группы) [96] были предложены для обеспечения возможности обнаружить группы произвольной формы, при этом используется соответствующая "ядерная" функции подобия. Подобные модификации метода вносят изменения исключительно в способ распределения объектов между группами, вводя дополнительные внутриклассовые коэффициенты или более сложные модификации. Различные модификации метода  $k$ -средних, стратегии глобального поиска и иные его усовершенствования, разработанные для применения с задачами  $k$ -средних в «классической» постановке могут применяться (возможно, после адаптации) и с модификациями, подобными ISODATA [93, 97, 98].

Встречаются задачи автоматической группировки, в которых фигурируют большое число группируемых объектов (иногда - миллионы), каждый из которых характеризуется большим числом измерений (характеристик). Число измерений

(размерность пространства характеристик) может достигать до нескольких тысяч (такова, например, задача автоматической группировки электрорадиоизделий, рассматриваемая в Главе 4). К большим задачам автоматической группировки относится, например, задача контекстного поиска изображений (Content Based Image Retrieval, CBIR), цель которой состоит в том, чтобы найти визуально подобные изображения по заданному в запросе изображению. Хотя данная тематика изучается более 15 лет, достигнуты достаточно скромные успехи в данном направлении. В ранних версиях методы CBIR сосредотачивались на определении подобия изображений по вычислительным оценкам особенностей цвета, формы, и текстуры изображения [99], более поздние версии [100] основываются на определении ключевых точек изображения. Тем не менее, такие подходы малоприменимы на более-менее крупных библиотеках изображений.

С другой стороны, поиск текстов требует гораздо меньше времени. Так, 0.1 секунды требуется для поиска в более чем 10 миллиардах документов Google [3]. К задаче поиска текстов может сводиться и задача поиска изображений: ключевые точки изображений сначала разбиваются на большое количество групп (решается задача автоматической группировки), называемых визуальными словами. Представляя каждое изображение гистограммой визуальных слов, задача превращается в простую задачу текстового поиска. Здесь также важной проблемой является определение числа выделяемых групп.

Для крупных наборов данных разработано большое число алгоритмов автоматической группировки, среди которых можно выделить основные категории:

А) Поиск эффективного ближайшего соседа. Одна из основных операций в любом алгоритме автоматической группировки состоит в определении членства в группе для каждого из группируемых объектов, которое может определяться в соответствии с отношением ближайшего соседнего объекта к той или иной группе (кластеру) [101, 102]. Метод ближайшего соседа сам по себе – достаточно грубый метод автоматической группировки, что справедливо и для методов, производных от него.

Б) Суммирование данных: цель подхода состоит в том, чтобы повысить эффективность алгоритма группировки, вначале выполнив подведение итогов на большом наборе данных, заменив близкие друг к другу объекты единственным объектом (первый этап группировки) и получив таким образом относительно небольшое подмножество, а затем применяя к этому подмножеству какие-либо алгоритмы

автоматической группировки (второй этап группировки). К таким алгоритмам относятся BIRCH [103], алгоритмы Штайнбаха [104], методы огрубления [105]. Как следует из названия последнего из приведенных подходов, применение таких двухуровневых технологий неизбежно приводит к "огрублению" - снижению точности результата.

В) Нарастающее объединение в кластеры. Эти алгоритмы [106] направлены на обработку потоковых данных таким образом, чтобы достичь максимальной эффективности (в плане вычислительной производительности) обработки массива данных за один проход (обработка в единственном цикле) в отличие от большинства подходов, требующих многократного обращения к данным. Как правило, используются методы иерархической классификации [107].

Г) Подходы, основанные на выборках (методы сэмпинга). Такие алгоритмы, как CURE [108, 109], CLARA/CLARANS [71, 110] и др [111] выбирают некое случайное подмножество из большого набора данных, выполняют автоматическую группировку этого набора, а результат ее используют как исходное решение при выполнении группировки исходного большого набора данных.

Нетрудно заметить, что все четыре подхода ставят своей главной целью повышение вычислительной производительности. В принципе, за исключением нарастающей кластеризации, все эти подходы могут применяться для решения задачи  $k$ -средних и  $p$ -медианной задачи. Однако и методы сэмпинга, вначале решающие упрощенную задачу, и методы суммирования данных, и методы, основанные на поиске ближайшего соседа, разбивают процесс автоматической группировки как минимум на два этапа - решение множества локальных задач автоматической группировки на первом этапе различными способами для получения упрощенной ("огрубленной") задачи, затем - ее решение, а затем (не обязательно) - решение исходной задачи с использованием решения «огрубленной» задачи в качестве начального решения. Таким образом, все подобные методы "заточены" на получение быстрого решения упрощенной, "огрубленной" задачи. Соответственно, применение подобных подходов неизбежно ведет к снижению точности результата, а тот факт, что как минимум первый этап решения (получение "огрубленной" задачи) является рандомизированной процедурой, снижает стабильность получаемого решения.

Модели и методы автоматической группировки находят многочисленные успешные применения в области анализа данных [3]. Тем не менее, построение

алгоритмов систем автоматической группировки связано с решением некоторых возникающих проблем, например:

А) Желательно достигнуть более тесной интеграции между алгоритмами автоматической группировки и прикладными задачами. Например, в некоторых приложениях необходимо выделить несколько связанных групп (часть данных - менее связанные группы - могут быть проигнорированы), в то время как другие приложения могут требовать обязательного разделения всех данных. Часто важен не выбор наилучшего алгоритма разбиения на группы, а именно наиболее адекватной модели группировки.

Б) Независимо от принципа автоматической группировки и соответствующей целевой функции, большинство методов автоматической группировки в конечном счете сводятся к комбинаторным задачам оптимизации (требуется применение различных стратегий глобального поиска: задачи, как правило, NP-трудны). В результате при решении задач с большим объемом данных на первый план выходит проблема вычислительной сложности. Следовательно, важно выбрать принципы построения алгоритма автоматической группировки, которые обеспечивают эффективные в вычислительном отношении решения для широкого круга задач в той или иной предметной области.

В) Важнейшей проблемой, связанной с автоматической группировкой, является стабильность результата. Хорошим подходом к созданию алгоритмов и систем автоматической группировки следует считать тот, который стабильно приводит к разбиению на одни и те же группы (кластеры), либо к очень близким результатам, и практически не зависит от малых колебаний в исходных данных. Хотя точность решения, безусловно, всегда важна, во многих практических приложениях на первый план выходит именно проблема создания стабильного алгоритма [112]. Существующие детерминированные алгоритмы, во-первых, не достаточно точны (показано в Главе 3), а во-вторых, требуют весьма высоких вычислительных затрат. Алгоритмы, предназначенные для обработки больших объемов данных [112], основаны на декомпозиции исходной задачи и решении ее в два этапа: создание "огрубленной" аналогичной задачи с использованием рандомизированной процедуры, ее решение и использование полученного решения в качестве начального решения исходной задачи. Такой двухуровневый подход неизбежно приводит к снижению стабильности результата (как и к снижению точности).

Первая из трех приведенных проблем связана с учетом особенностей прикладной задачи. Тем не менее, разрабатываемые методы автоматической группировки должны быть достаточно гибкими, позволяя эти особенности учитывать. В решении второй проблемы – вычислительной сложности в связке с решением третьей проблемы – повышения стабильности получаемого результата, - как показывают результаты настоящего исследования, существенный прогресс может быть достигнут путем выработки достаточно универсального подхода (т.е. позволяющего учитывать характерные особенности конкретной задачи и ее предметной области), суть которого состоит в комбинировании различных схем (стратегий) глобального поиска, таких как эволюционные алгоритмы, метод изменяющихся вероятностей, случайный мультистарт, детерминированный алгоритм, с особыми видом эвристических алгоритмов – жадными агломеративными эвристическими алгоритмами (процедурами).

В дальнейшем будем называть эти жадные агломеративные алгоритмы и процедуры жадными эвристиками.

Сами по себе используемые жадные агломеративные эвристики являются детерминированными процедурами, последовательно улучшающими промежуточный результат от шага к шагу. Характерной их особенностью является то, что в промежуточном решении задачи выделяется количество групп (кластеров) большее, чем требуется по условиям задачи, и это число последовательно уменьшается. Такие эвристики комбинируются с иными схемами локального поиска, которые хорошо себя зарекомендовали при решении той или иной конкретной задачи автоматической группировки объектов. Кроме того, при решении задач автоматической группировки объектов в непрерывном пространстве характеристик, таких как задача  $k$ -средних или непрерывная  $p$ -медианная задача, жадные эвристики и построенные с их применением алгоритмы могут применяться для решения задач с различными мерами расстояния.

### **1.3 Задачи автоматической группировки объектов и теория размещения**

Приведенное в начале данной главы определение задачи автоматической группировки объектов предполагает наличие некой меры подобия (сходства), либо наоборот — меры различия. Мера различия по сути является расстоянием между объектами в некотором дискретном либо непрерывном пространстве характеристик.

Мерой сходства может являться, например, обратная ей величина. Поскольку задачи оперируют расстояниями и положениями некоторых точек (объектов) в пространстве, очевидной является связь таких задач с задачами теории размещения, которые определяются большинством авторов как задачи, основными параметрами которых являются положения каких-либо объектов в пространстве и расстояния между ними [113-117].

Теория размещения зародилась в рамках экономической теории [118], при этом ее взаимосвязь с задачами автоматической группировки отмечена достаточно давно [119, 120, 121]. Так, формулировка задачи группировки объекта в виде множественной задачи Вебера встречается в [122, 121] (фактически имеется в виду поиск центраида, то есть решение разновидности  $p$ -медианной задачи).

Задачи размещения широко применяются при застройке городов, в архитектуре, транспортировке, индустриализации и т.д. [123-126, 113]. Существует также множество опосредованных их применений в таких областях, как стандартизация [60, 61, 127]. Исторически, начиная с 1960-х годов, в СССР (например, в Институте математики им. С.Л.Соболева) задачи размещения формулировались в терминах состава технических систем, оптимального ряда изделий и формально с размещением связаны не были [67, 63, 128]. В дальнейшем, после выявления связи этих двух направлений исследования все более концентрировались именно на задачах размещения.

Существует множество разновидностей задач размещения, и следовательно, множество разделов данной теории. Если объект можно разместить в любой точке пространства, то данную задачу рассматривают как непрерывную задачу размещения; если размещение возможно лишь в определенных точках, то задача классифицируется как дискретная; также выделяются задачи на сети. Примечательно, что в течение долгого времени теория размещения и кластерный анализ развивались параллельно, используя одинаковые либо очень схожие методы. Так, например,  $p$ -медианная задача — одна из основных задач теории размещения [129] — и наиболее популярная модель кластерного анализа —  $k$ -средних [130] не только имеют очень схожие постановки, но и традиционные подходы к ним — ALA-процедура (Alternating Location-Allocation — чередующееся распределение-размещение) [129] и стандартная процедура  $k$ -средних [48] — построены по одной и той же схеме. Подобную же параллель можно проследить между агломеративными эвристическими процедурами, применяемыми в кластерном



анализе (например, метод информационного бутылочного горлышка [74, 73]), в которых изначально каждый объект рассматривается как отдельный кластер, а затем кластеры один за другим объединяются, и методами, предложенными еще в 1963 году для задачи размещения складов [131], где изначально при каждом потребителе размещается склад, а затем они один за другим удаляются.

Классификация задач размещения [125] основана на характеристиках задач:

- Количество размещаемых объектов (один или более объектов — задачи с единственным объектом и множественные задачи размещения);
- Множество, на котором размещаются объекты размещаемых объектов (непрерывные и дискретные задачи);
- Тип размещаемых объектов (медианы, центры и т.д.);
- Вид алгоритма, применяемого для решения задачи (точный/эвристический, иные виды классификации);
- Число целевых функций для оценки положения объекта (одна или более).

Для задач автоматической группировки можно применять ту же классификацию, за исключением первого пункта (всегда предполагается разбиение объектов более чем на одну группу, то есть решается множественная задача размещения).

Задача Ферма — вероятно, первая и простейшая из задач размещения — в оригинале (на латыни) звучит так: «*datis tribus punctis, quartum reperire, a quo si ducantur tres rectae ad data puncta, summa trium harum rectorum sit minima quantitas*», или по-русски: «для данных трех точек найти четвертую, от которой до остальных трех сумма расстояний будет минимальна» [132]. Пьер де Ферма в семнадцатом веке в письме к итальянскому физическому и математику Эвангелисте Торричелли [133] поставил задачу о нахождении точки, обладающей следующим свойством: сумма расстояний от точки до вершин заданного треугольника минимальна. Первое решение задачи было предложено Торричелли, с тремя равносторонними треугольниками на каждой стороне заданного треугольника и тремя описанными около равносторонних треугольников окружностями. Тогда точка пересечения данных окружностей будет являться решением задачи Ферма и данная точка называется точкой Ферма-Торричелли. Данная точка в литературе известна как пятая замечательная точка треугольника [134] и это единственная замечательная точка треугольника, открытая после античности [124].

Следующий шаг в исследовании задачи Ферма сделал Б.Кавальери, доказавший, что отрезки от вершин до точки Ферма-Торричелли образуют углы 120 градусов каждый [135]. Винченцо Вивиани, ученик Торричелли и последователь Галилео, опубликовал решение Торричелли [136], которое было дано в современной терминологии в [137].

Веком позже английский математик Т.Симпсон упростил решение Торричелли, соединив вершины равносторонних треугольников с противоположащими сторонами исходного треугольника. Точка пересечения трех таких линий является точкой Ферма-Торричелли [135].

Верхняя граница суммы расстояний от вершин исходного треугольника дана в следующем выражении:

$$RA+RB+RC < \max\{a+b, b+c, c+a\}.$$

Данное выражение было выведено немецким математиком Виссчерзом и доказано румынским математиком Джозефом Шандором [138].

Позднее задача Ферма была развита австрийским математиком Альфредом Вебером, где требовалось найти точку минимума суммы расстояний не для трех, а для произвольного числа точек. Кроме того, Вебер включил в задачу весовые коэффициенты точек. Задача была сформулирована как задача теории размещения производства [139]. Эту задачу, называемую задачей Вебера, Ферма-Вебера, 1-медианной задачей [140] или задачей Штайнера, принято считать отправной точкой современной теории размещения. Геометрическим аспектам данной задачи посвящена, например, работа [141].

Алгоритмы для некоторых практически важных задач Вебера, в которых используются неевклидовы метрики и функции расстояния, рассмотрены в Главе 2. Отметим, что решение задачи Вебера является составной частью многих методов решения задач автоматической группировки, в частности — методов на основе жадных агломеративных эвристических алгоритмов, которым посвящена настоящая работа.

Примечательна сама по себе история создания очень простого метода решения задачи Вебера (история рассматривается, например, в [142], в обзоре [143] и во второй части книги [123]) 16-летним венгерским математиком Эндре Вайсфелдом (в то время носившим фамилию Вайсфелд) [144], опубликованного в 1937 году. Сам автор [145] ставил своей задачей не создание алгоритма (само понятие «алгоритм» было неизвестно большинству математиков), а доказательство теоремы, сформулированной Штурмом [146] в 1884. В одном из доказательств Вайсфелд определил последовательность,

которая сходилась к оптимальному решению задачи Ферма-Вебера. Фактически процедура получения такой последовательности [147] является простым алгоритмом градиентного спуска, хотя само понятие «градиентный спуск» появилось много позже. Данный алгоритм наряду с его более совершенными модификациями [144, 148] и сегодня широко применяется при решении задач размещения.

Вероятно, самое обширное описание норм, метрик и иных функций расстояния, используемых в различных научно-технических задачах, собрано в «Энциклопедии расстояний» и других работах М.Дезы и Е.Дезы [88, 149, 150]. Наиболее популярны модели с метриками, основанными на  $l_p$ -нормах Минковского:

$$L(X, Y) = l_p((x_1, x_2, \dots, x_d), (y_1, y_2, \dots, y_d)) = \left( \sum_{i=1}^d |x_i - y_i|^p \right)^{1/p}, p \geq 1$$

и ее частных случаях – евклидовой  $l_2$  (при  $p=2$ ), прямоугольной или манхэттенской  $l_1$  ( $p=1$ ), чебышевской  $l_\infty$  ( $p = +\infty$ ), а также с квадратичными евклидовыми расстояниями (такая мера расстояния метрикой не является):

$$l_2^2(X, Y) = \sum_{i=1}^d (x_i - y_i)^2.$$

Используются и другие обобщения евклидовой метрики – например, метрика Махаланобиса, а также метрика Жаккара (удобна как мера расстояния между векторами булевых значений), метрика Канберры (применяется в интеллектуальном анализе), метрика Соресена (например, в ботанике, экологии и др.)

Популярность квадратичных евклидовых расстояний объясняется, в том числе, простотой решения задачи Вебера с данной мерой расстояния. Решением является точка

$$X^* = (x_1^*, \dots, x_d^*); x_r^* = \frac{\sum_{i=1}^N w_i a_{i,r}}{\sum_{i=1}^N w_i} \quad \forall r = \overline{1, d}.$$

Широко применяются в задачах размещения и другие функции расстояния (меры), метриками не являющиеся, например, косинусная мера:

$$L(X, Y) = 1 - \frac{\sum_{k=1}^d x_k y_k}{\sqrt{\sum_{k=1}^d x_k^2} \sqrt{\sum_{k=1}^d y_k^2}}.$$

Другое возможное обобщение задачи Вебера – множественная задача Вебера – формулируется как задача о нахождении  $m$  точек:

$$\begin{aligned} & \arg \min f(X_1, X_2, \dots, X_m) = \\ & = \arg \min \sum_{i=1}^N \sum_{j=1}^m w_{i,j} L(A_i, X_j) + \sum_{j=1}^{m-1} \sum_{k=j+1}^m v_{j,k} L(X_j, X_k). \end{aligned}$$

Здесь каждая известная точка  $A_i$  характеризуется  $m$  весовыми коэффициентами  $w_{i,1} \dots w_{i,m}$  для каждой искомой точки  $X_j$  (этот коэффициент, например, характеризует потребность точки  $A_i$  в товарах  $j$ -го типа), для каждой пары искомых точек  $X_j$  и  $X_k$  вводится весовой коэффициент  $v_{j,k}$ , характеризующий стоимость перевозок между ними.

Детальное исследование метрик и мер расстояния, применяемых в задачах размещения и автоматической группировки, дано в Главе 5.

Непрерывные задачи размещения в евклидовой метрике ( $l_2$ ), а также в метриках  $l_1$  (прямоугольной или манхэттенской) и  $l_\infty$  (чебышевской) изучены и созданы универсальные алгоритмы, в том числе – на основе процедуры Вайсфельда, исследована их вычислительная сложность [151, 152]. Решение тех же задач, например, с запретными зонами [153] уже не является тривиальным. В практических задачах при использовании евклидовой или прямоугольной метрики такие модели являются достаточно грубым приближением [154, 155], поскольку не учитывают свойства пространства и средств транспортировки, в частности, наличие и качество дорог, барьеры, рельеф и т.п. Нередко функция расстояния является решением другой (вложенной) оптимизационной задачи, и задается в этом случае алгоритмически [155]. Таким образом, зачастую целесообразной является дискретизация задачи [156, 157] (подходы рассмотрены в Главе 6). Такие задачи можно рассматривать как  $p$ -медианные задачи на полном графе. Компромиссные подходы, сочетающие в формулировке задачи элементы дискретного и непрерывного подходов, рассмотрены в [158-160].

Такие задачи автоматической группировки объектов, как  $k$ -средних,  $k$ -медоид, в терминах теории размещения являются задачами размещения-распределения (LAP, Location-Allocation Problems) [114]: следует найти оптимальное положение  $k$  центров групп таким образом, чтобы распределение объектов между центрами (и соответствующими этим центрам группами) было оптимальным. Очень близкие по своей постановке задачи теории размещения – UFLP (Uncapacitated Facility Location Problem – задача размещения неограниченных производственных мощностей, иначе называемая задачей о многомерном размещении [161]) и CFLP (Capacitated Facility

Location Problem – задача размещения ограниченных производственных мощностей) [114, 162]. Постановки UFLP и  $p$ -медианной задачи фактически равнозначны [67] – каждая из производственных мощностей, местоположения которых требуется найти, потенциально может обслуживать неограниченное число потребителей, местоположение которых известно. В терминах, более привычных для задач автоматической группировки, это можно сформулировать так: размер группы, для элементов которой данный конкретный центр является ближайшим, неограничен. Отметим при этом, что задачи размещения, как правило, имеют гораздо меньший объем данных по сравнению с задачами автоматической группировки. Так, например,  $p$ -медианные задачи с числом потребителей, равным 1938, классифицируются как «очень большие» задачи [163]. Более сложная в своей постановке задача CFLP также имеет аналог среди задач автоматической группировки. В CFLP производственные мощности имеют ограниченную производительность и могут обслуживать, соответственно, ограниченное число потребителей. В задачах автоматической группировки этому соответствует требование об ограничении сверху количества объектов, относимых к группе. Обе задачи UFLP и CFLP находят широкое применение в различных сферах, таких как размещение складов и розничных торговых точек крупных торговых сетей [164], промышленных объектов [165], элементов транспортной инфраструктуры [166], электроэнергетических и телекоммуникационных узлов [167-169], пунктов реагирования в чрезвычайных ситуациях [170].

В теории размещения для таких задач также может применяться процедура с двумя чередующимися шагами распределения и размещения, аналогичная процедуре  $k$ -средних, и такой весьма хорошо зарекомендовавший себя подход в теории размещения иногда называют подходом, основанным на группировке (Clustering-Based Approach) [162]. Если в задачах  $k$ -средних,  $k$ -медоид,  $p$ -медианной, UFLP распределение объектов между найденными центрами является простой очевидной процедурой: объект приписывается к ближайшему центру, то в задаче CFLP и аналогичных постановках распределение является решением вложенной оптимизационной задачи, при решении которой применяются методы, основанные на определении минимального потока в сети с различными целевыми функциями [171, 172]. Для решения перечисленных задач существуют методы их представления в виде задач линейного программирования и целочисленного линейного программирования [173-175]. Тем не менее, вследствие NP-

трудности задач [176] точные методы, разработанные для экономических приложений, практически неприменимы [162] для задач с большим объемом входных данных, характерных для автоматической группировки, и основные усилия исследователей направлены на поиск эвристических алгоритмов, дающих хорошие субоптимальные решения [177, 178]. Такие эвристические методы, как CAPCLUST [179], непосредственно основанные на методе k-средних, для более сложных задач размещения-распределения дают незначительные преимущества по производительности по сравнению с линейным программированием [162]. При этом дальнейшие модификации методов, основанных на k-средних, показывают приемлемые результаты [162] на относительно небольших наборах данных (до нескольких сотен объектов при числе групп  $k < 10$ ).

В противоположность применению при решении задач размещения методов, характерных для задач кластерного анализа и автоматической группировки, предложено достаточно большое количество подходов, предусматривающих применение традиционных для теории размещения методов к решению задач группировки объектов. Так, в работе [180] для непрерывной p-медианной задачи предложен комбинированный метод, состоящий из трех последовательно выполняемых этапов: метод прямого поиска – перестановка центров (группируемых объектов, выбираемых в качестве центров) [181], метод лагранжевых релаксаций с мультистартом [182], метод ветвей и границ. Каждый последующий этап берет за основу лучшее решение, полученное на предыдущем этапе. Такой подход позволяет получить точное решение задачи с числом группируемых объектов до 318.

Важным аспектом при применении p-медианной модели в задачах автоматической группировки [183, 182] является то, что эта модель может применяться для группировки реальных объектов в реальном 1-, 2- или 3-мерном пространстве. Кроме того, для некоторых структур данных p-медианная модель может обеспечить более качественную группировку (меньшую долю ошибочно классифицированных объектов), чем иные процедуры объединения в кластеры [183], такие как k-средних с квадратичными евклидовыми расстояниями.

Дискретная p-медианная задача (фактически – задача k-медоид) решена в [184]. Метод, предложенный в данной работе, основанный на идее поиска с чередующимися окрестностями (VNS – Variable Neighborhood Search) – популярном методе решения

задач дискретной оптимизации Н.Младеновича [185, 186, 187], позволяет находить хорошие субоптимальные решения достаточно больших задач автоматической группировки (максимальные рассмотренные задачи из библиотеки примеров BIRCH предполагают группировку 89000 объектов в 2-мерном пространстве) без применения технологий агрегирования, таких как сэмплинг, получая результат, не превышающий лучшее известное решение более чем на 1%. В то же время, авторы указывают на желательность совместного применения методов решения данной задачи с методами сэмплинга, такими как CLARA [71].

#### 1.4 Эволюционные и иные стратегии глобального поиска

Целевая функция практически любой задачи автоматической группировки (за исключением частных случаев, не представляющих практического интереса) является нелинейной [1], обладающей множеством локальных минимумов. Для точного решения подобных задач необходимо решить сложную комбинаторную задачу поиска оптимального варианта разбиения на группы. В принципе, возможно получить решение любой подобной задачи полным перебором: число различных вариантов разбиения  $N$  объектов на  $k$  групп равно [188]

$$M(N, k) = \frac{\sum_{i=1}^k (-1)^i \binom{k}{i} (k-i)^N}{k!}.$$

Очевидно, что вычислительная сложность алгоритма полного перебора экспоненциально зависит от объема данных в задаче (от числа группируемых объектов  $N$ ).

Классические алгоритмы кластерного анализа, такие как стандартная процедура  $k$ -средних, используя различного рода ограничения (например, на число и форму получаемых групп), выполняют направленный поиск в относительно небольшом подмножестве пространства возможных решений, при этом не гарантируя нахождение строго оптимального решения. Данные алгоритмы являются алгоритмами локального поиска в том смысле, что они последовательно улучшают заранее известный промежуточный результат (хотя не являются оптимизационными методами локального поиска в строгом смысле: поиск следующего решения осуществляется не обязательно в

окрестности предыдущего), и, соответственно, результат таких алгоритмов зависит от выбранного начального решения. Для поиска оптимального решения может применяться мультистарт данных методов с рандомизированными процедурами выбора начальных решений и/или рандомизированной процедурой локального поиска [189, 190] или более сложные подходы [1], многие из которых основаны на идеях, позаимствованных из живой природы. К таковым можно отнести генетические (эволюционные) алгоритмы [191], нейронные сети [192], методы имитации отжига [193] и т.д. Существуют экспериментальные исследования, подтверждающие преимущества эволюционных алгоритмов перед классическими алгоритмами [194].

Эволюционные алгоритмы используют принципы и терминологию, связанную с природной эволюцией. Например, используется понятие популяции – набора промежуточных решений (различных вариантов группировки), называемых также «хромосомами» или «особями», и специальных операторов – процедуры селекции – отбора «особей» (например, отбора лучших или случайного отбора), процедуры рекомбинации (скрещивания, кроссинговера) и процедуры мутации – случайной замены некоторых элементов решения на другие, позволяющих из одной или нескольких (обычно – двух) родительских «особей» получить одну или несколько «особей» – потомков.

Генетический алгоритм, предложенный в [195], имитирует адаптацию популяции «особей» к некому аналогу окружающей среды, причем алгоритм вводит специальную меру приспособленности (fitness function, иначе – функция полезности), которая, как правило, определяется на основе целевой функции задачи. По умолчанию мы в данном исследовании будем считать понятия «целевая функция» и «функция приспособленности» эквивалентными. Процесс работы алгоритма представляет собой последовательную смену популяций, состоящих из  $N_{POP}$  «особей» [196]. Отметим, что популяция может сменяться целиком или частично, размер популяции может оставаться неизменным (по умолчанию будем считать его неизменным) либо меняться динамически по ходу работы алгоритма [197, 198].

Генетический алгоритм, как правило, рассматривают в качестве стратегии глобального поиска. Более того, для многих версий генетического алгоритма характерен следующий парадокс [67]: в то время, как классические методы локального поиска легко находят собственно локальные оптимумы задачи, генетический алгоритм часто



предъявляет в качестве ответа глобальный оптимум. Само понятие глобального оптимума задачи конструктивным не является [67]: даже получив его, трудно проверить и доказать его глобальную оптимальность.

Основные шаги типичного генетического алгоритма решения задачи автоматической группировки описываются в обзоре [1] таким образом.

Алгоритм 1.1. Общая схема генетического алгоритма для задачи автоматической группировки.

1. Формируется случайная популяция «особей» - группировочных решений. Каждый вариант группировки при этом представлен в виде последовательности из  $N$  целых чисел, представляющих номера групп. Для каждой «особи» оценивается значение целевой функции.

2. Генерируется следующая популяция, при этом используются так называемые эволюционные операторы. Оператор селекции служит для случайного отбора «родительских» особей, наилучших с точки зрения критерия качества. Рекомбинация (скрещивание, кроссинговер, кроссовер) служит для образования из отобранных «особей», представленных последовательностями номеров групп, новых последовательностей ( в простейшем случае – путем перестановки одного или нескольких сегментов в этих последовательностях). Оператор мутации – практически всегда рандомизированная процедура – случайным образом меняет последовательность. Например, она может заменить в случайно выбранном элементе один номер группы другим. Для новой популяции вычисляются значения целевой функции. Этот шаг повторяется, пока не будет выполняться заданное условие остановки.

Отметим, что могут использоваться и иные способы кодирования «особей», например – в виде множества индексов группируемых объектов, выбранных в качестве центров групп в  $k$ -медоидной задаче [82],  $p$ -медианной задаче на сети или множества группируемых объектов, выбранных в качестве начальных центров, которые затем изменяются процедурой  $k$ -средних или иными процедурами локального поиска [199]. Реже используется упоминавшееся выше кодирование вещественными числами – в виде множества точек в пространстве характеристик, выбранных в качестве центров групп [45, 84].

Описанный простейший оператор рекомбинации обладает некоторыми недостатками. Например, группы одного и того же состава могут иметь различные номера. Данный недостаток может быть устранен достаточно затратной процедурой [200]. Также могут возникать ситуации, когда «особь»- потомок имеет в своем составе меньше групп, чем требуется по условиям задачи (некоторые номера групп в полученной последовательности отсутствуют). Отметим, что данный недостаток является менее существенным: такие «особи» имеют значение целевой функции большее, чем аналогичные «особи», в которых единственный элемент изменен на отсутствующий номер группы. Соответственно, такие «особи» постепенно должны выбывать в результате селекции. Данное свойство задач автоматической группировки, следующее из свойства монотонности целевой функции, представленной в виде псевдо-булевой функции, доказано для задачи  $k$ -медоид в Главе 2.

При использовании кодирования «особей» в виде множества индексов объектов, выбираемых в качестве центров групп, могут возникать более серьезные проблемы: в полученном в результате рекомбинации решении может быть число групп (фактически – число центров) большее, чем требуется условиями задачи. В этом случае множество индексов, которым представлена «особь», может быть «урезано» до фиксированной мощности путем упорядочения его элементов и выбора  $k$  элементов из начала и конца полученной последовательности [82], случайным выбором  $k$  элементов или с применением жадных эвристических алгоритмов [67].

Могут применяться комбинированные схемы глобального поиска, включающие элементы генетического подхода и классических процедур размещения-распределения, таких как стандартная процедура  $k$ -средних. Так, в [162] общая схема генетического алгоритма, включающего в себя элементы «кластерного» подхода, описывается следующим образом.

#### Алгоритм 1.2. «Кластерный» генетический алгоритм.

1.Инициализация «особей», представленных множествами центров групп. Для каждой «особи» выбираются (случайным образом)  $k$  объектов в качестве центров.

2.Выполнить процедуру распределения объектов по группам. В задаче  $k$ -средних или  $p$ -медианной объекты приписываются к группе, центр которой является ближайшим.

3. Выполнить процедуры локального поиска. В [162] предлагается произвести попытку замены части элементов в решении для задачи CFLP. Такой же подход применим и для других задач. Шаг 3 предлагается выполнять до тех пор, пока он приносит результаты.

4. Получить новое поколение «особей» (см. Шаг 2 Алгоритма 1.1).

5. Проверить условия останова, перейти к Шагу 2.

Аналогичные способы включения стандартных для задач автоматической группировки процедур размещения-распределения предложены в [199]. Процедуры размещения-распределения могут быть включены аналогично Шагу 3 Алгоритма 1.2 либо выполнены в самом конце алгоритма, после выполнения условий останова, для уточнения полученного генетическим алгоритмом результата (второй способ – быстрый, но менее точный).

Современная литература предлагает множество вариантов эволюционных операторов применительно к задачам автоматической группировки. Наиболее критичным и в наибольшей степени зависимым от конкретной постановки задачи является оператор скрещивания (кроссинговера, рекомбинации).

Первое применение генетических алгоритмов для решения р-медианной задачи на сети, близкое к Алгоритму 1.1, было предоставлено Хосаджем и Гудчайлдом [201].

В генетическом алгоритме Чиоу и Ляна [202] эволюционные операторы позволяют выделить узлы сети, наиболее подходящие в качестве элементов решения (центров), тогда как прочие узлы могут становиться центрами группы, если это улучшает целевую функцию.

Бозкая и др. [203] предложили генетический алгоритм с кодированием "особей" в виде множества индексов узлов сети, выбираемых в качестве центра группы. При этом используются сразу три оператора кроссинговера. Альпом, Эркутом и Дрезнером [204] дано описание простого и достаточно быстрого генетического алгоритма с особой процедурой рекомбинации - жадной (агломеративной) эвристикой, которая и взята за основу в настоящей работе. Данная эвристика вместо перестановки последовательностей, которыми представлены родительские "особи", производит объединение родительских множеств индексов узлов, выбранных в качестве центров групп. В результате получается в общем случае недопустимое решение (в дочернем

решении центров получается больше, чем требуется по условиям задачи). Затем лишние центры удаляются один за другим, пока не будет достигнуто допустимое решение. Каждый раз удаляется тот элемент решения, удаление которого дает наименьший прирост целевой функции. В [205] предложена мера состоятельности родительских "особей", помогающая достигать лучших предположений о выборе "правильных" родительских "особей" на этапе отбора. Эта техника позволяет рассматривать родителей-претендентов совместно, а не сравнивать их попарно.

Лим и Сю [83] использовали подмножество фиксированной мощности в качестве промежуточного решения кандидата. Однако существенные вычислительные результаты в работе не представлены. Аналогичный подход для задачи р-медиан был предложен в [82], при этом получены хорошие результаты для задач достаточно большого объема данных. Использование динамической популяции было предложено в [206]. В этом алгоритме рассматриваются две различные целевые функции, и задача группировки формулируется как двухкритериальная задача оптимизации.

Генетические алгоритмы, в том числе применяемые к задачам автоматической группировки и размещения, могут обходиться без включения в них методов локального поиска, но эффективность таких алгоритмов, как правило, сравнительно невысока. Необходимость частичной перестройки решения, построенного оператором кроссигнера, методами локального спуска объясняется тем, что [207] эволюционный алгоритм в этом случае сосредотачивается только на локальных оптимумах, множество которых несравнимо меньше по мощности, чем множество всех допустимых решений задачи (хотя число локальных оптимумов также может расти экспоненциально с ростом объема исходных данных). Экспериментальные исследования показывают высокую вероятность сосредоточения множества локальных оптимумов вблизи глобального оптимума [208], при этом очередное сгенерированное генетическим алгоритмом решение на основе двух локальных оптимумов имеет более высокие шансы найти глобальный оптимум [207].

В [209] описана иная эвристика, оперирующая локальными оптимумами: для решения непрерывной р-медианной задачи локальным поиском поочередно решается то непрерывная задача, то дискретная, причем дискретная задача решается на множестве точек, являющихся локальными оптимумами непрерывной задачи, найденными на одном из предыдущих шагов. Множество локальных оптимумов, участвующих в

решении дискретной задачи при этом пополняется, а решения дискретной задачи затем используются как начальные решения непрерывной задачи. Такой подход показал эффективность при решении сравнительно небольших задач (до сотен группируемых объектов) на плоскости.

### 1.5 Жадные эвристические процедуры в эволюционных алгоритмах

Как отмечалось выше, идея жадных эвристических алгоритмов не нова. Такие методы, как метод информационного бутылочного горлышка, агломеративные методы автоматической группировки, один из старейших (1963 года) алгоритм для задачи размещения складов Куна и Гамбургера [131], являются жадными эвристическими методами. Основная отличительная особенность жадных (в старой терминологии – «пожирающих», англ. «greedy») методов состоит в том, что они, являясь методами, последовательно улучшающими известный результат (методами локального поиска), в некоторой окрестности известного решения, выбирают в качестве следующего решения тот вариант, который дает наибольший прирост целевой функции (наибольшее уменьшение значения – в случае минимизации) или функции полезности. Для классической задачи о  $p$ -медиане (на сети) метод локального поиска в направлении наилучшего элемента с асимптотической сходимостью  $O(N^2p)$  предложен в [181] и улучшен в [210] до асимптотической сходимости  $O(N^2)$ , где  $N$  – число узлов сети,  $p$  – число размещаемых на сети центров групп.

Генетические алгоритмы и более широкий класс – эволюционные алгоритмы – задают лишь общую схему организации поиска, реализация которой зависит от выбора процедур селекции, мутации и особенно – рекомбинации (кроссинговера, скрещивания), которая изначально задумывалась как простая рандомизированная процедура [195], а затем была адаптирована для конкретных задач в виде иногда очень сложных алгоритмов [211]. Практика решения многих NP-трудных задач показывает эффективность перехода от рандомизированных процедур рекомбинации к поиску наилучшего способа рекомбинации [212, 213]. В [213] рассматривается задача построения наилучшей «особи» - потомка от двух известных «особей» - родителей. При этом предусмотрено кодирование «особей» в виде последовательностей нулей и единиц, причем при совпадении значения цифры в родительских последовательностях она

наследуется «особи» - потомку, при несовпадении решается задача выбора такого значения несовпадающей цифры, при котором потомок имеет наилучшее значение целевой функции. Отметим, что наилучшее значение целевой функции у потомка в первом поколении не гарантирует наилучших значений в последующих поколениях – в данном случае алгоритм выбора наилучшего потомка в первом поколении тоже является в некотором смысле жадным. Но и такая ограниченная задача построения наилучшего потомка от двух известных родительских «особей» оказалась NP-трудной, например, применительно к  $r$ -медианной задаче (при этом для других NP-трудных задач оптимальный потомок может быть найден за полиномиальное время). Это позволяет сделать предположение о том, что правильный выбор процедуры кроссинговера может быть задачей не менее трудной, чем собственно нахождение оптимального решения задачи оптимизации. Под «правильным» выбором будем понимать выбор процедуры, позволяющей получить лучший результат за установленное время.

Различные методы локального поиска, в том числе некоторые жадные алгоритмы, для дискретных задач размещения (в том числе для задачи о  $r$ -медиане) исследованы в диссертации Ю.А.Кочетова [67]. В частности, исследуются два вероятностных жадных рандомизированных алгоритма поиска в достаточно простых окрестностях для многостадийной задачи размещения. Отмечается, что рандомизированный характер поиска позволяет сократить погрешность результата. Один из данных алгоритмов, аналогично рассматриваемой в настоящей работе жадной агломеративной эвристике, построен на последовательном сокращении мощности множества, которым представлено промежуточное решение, при этом удаляемый из множества элемент (предприятие в многостадийной задаче размещения предприятий) выбирается случайно из некоторого подмножества потенциально удаляемых объектов. При решении дискретных задач размещения используется локальный поиск в различных окрестностях: от простых окрестностей, состоящих из узлов сети, смежных с узлами текущего решения, SWAP-окрестностей, получаемых заменой одного элемента решения на любой другой элемент (узел сети),  $k$ -SWAP-окрестностей, получаемых заменой до  $k$  элементов решения, до более сложных, например, Лина-Кернигана [214]. Эффективность конкретных методов локального поиска и применяемой окрестности сильно зависит от задачи. Тем не менее, поиск в простых окрестностях, как правило, достаточно эффективен для большинства практических дискретных задач

автоматической группировки. При этом в [67] доказано (в предположении, что класс задач NP не эквивалентен классу полиномиально разрешимых задач), что для задачи о р-медиане не существует такой полиномиально проверяемой окрестности, при которой любой локальный оптимум является глобальным. Полиномиально проверяемой называется окрестность, которую можно обойти (проверить все решения в ней) за полиномиальное время. Отметим, что настоящая работа сосредоточена на построении эффективных алгоритмов автоматической группировки для реальных практических задач, в ней не рассматриваются методы решения наиболее сложных задач, специально разработанных в качестве «ловушек» для различных методов поиска [67].

Важным параметром генетического или любого эволюционного алгоритма является размер популяции – число особей  $N_{POP}$ . Сильно ограниченная по размеру популяция приводит к тому, что в отсутствие процедуры мутации (либо при незначительном влиянии мутации) алгоритм начинает генерировать одни и те же решения (популяция «вырождается») [215]. В [216] авторы указывают, что наилучшие результаты, как правило, достигаются при приблизительно одинаковом числе сменяющихся поколений «особей» и числе самих «особей». При вычислительно сложной процедуре кроссинговера, однако, сложно предсказать время ее выполнения, и, следовательно, сложно предсказать число сменяющихся поколений за установленное время. В [196] исследована зависимость времени достижения локального оптимума от размера популяции и размера турнира (размер турнира  $s$  – число «особей», среди которых выбираются лучшие в качестве родительских «особей», порождающих «особи» следующего поколения). Если  $h$  – число неоптимальных решений в окрестности,  $L$  – минимальная вероятность достижения локального оптимума в очередном поколении,  $E$  – вероятность того, что хотя бы одна «особь»-потомок будет по значению целевой функции не хуже родительской,  $r$  – доля «особей» популяции, участвующих в турнирной селекции, то размер популяции и размер турнира оцениваются как

$$N_{POP} = 2 \left\lceil \frac{1 + \ln h}{L \cdot E(1/e^{2r})} \right\rceil, s = \lceil rN_{POP} \rceil, \text{ где } \lceil \cdot \rceil - \text{ символ округления вверх.}$$

Такой подход к оценке размера популяции требует решения сложных задач – определения значений  $h$ ,  $L$ ,  $E$ . В [196] оценки этих значений даны для простых процедур кроссинговера. Если же в процедуру кроссинговера включен локальный поиск, то выбор любых двух родительских «особей» дает в результате локальный оптимум в качестве

потомка, то есть для достижения локального оптимума достаточно  $N_{POP}=2$ , что ни в коей мере не отражает размер популяции, позволяющей получить решения, близкие к глобальному оптимуму.

Поскольку оптимальный размер популяции, необходимый для эффективной работы эволюционного алгоритма, установлен лишь для генетических алгоритмов с простыми способами рекомбинации, в настоящей работе оценка оптимального размера популяции выполнялась исключительно экспериментальным путем. В связи с этим перспективным представляется развитие методов, использующих динамические популяции.

### **Выводы к Главе 1.**

Приведенный анализ литературных источников показывает, что методы решения задач автоматической группировки объектов, несмотря на сходство и даже эквивалентность их с задачами размещения, развивались во многом отлично от методов теории размещения. Алгоритмы автоматической группировки объектов основаны в основном на эвристических эвристических локального поиска и стратегиях глобального поиска. В то же время, многие методы автоматической группировки, несмотря на существенное различие в применяемой терминологии, имеют общие черты с методами решения задач размещения. Так, общим широко применяемым методом локального поиска является процедура с двумя чередующимися шагами размещения-распределения, причем задача определения центра группы объектов (центра, центроида) на шаге размещения фактически является задачей Вебера – одной из ключевых задач теории размещения.

Методы решения задач автоматической группировки можно условно разделить на два больших класса. Первый класс составляют методы, дающие достаточно точный результат (иногда – в строгом смысле точный), требующие значительных вычислительных затрат, и применимые, соответственно, лишь для группировки относительно небольшого числа объектов. Методы этого класса часто имеют наибольшее число общих черт с методами теории размещения. Второй класс – специфические методы, нацеленные на быстрое решение задач с очень большим объемом данных. Эти методы часто дают весьма неточное решение задачи. Кроме того –



и в некоторых случаях это является их наиболее существенным недостатком – эти методы дают нестабильный результат, сильно зависящий от значений случайных величин (большинство методов – рандомизированные, т.е. использующие случайные величины), от порядка расположения данных и т.д. Кроме того, методы как первого, так и второго класса, как правило, «заточены» под использование той или иной специфической метрики или меры сходства (например, квадратичных евклидовых расстояний или прямоугольной метрики).

Отмечая некоторый дефицит компромиссных по времени счета и качеству результата (под качеством будем понимать точность – близость значения целевой функции к глобальному оптимуму и стабильность – близость получаемых значений друг к другу при множественных запусках алгоритма), в качестве цели настоящего исследования ставится задача разработки универсального метода (методов) потокового решения практических задач автоматической группировки объектов, который бы удовлетворял следующим критериям:

1. Задачи должны решаться за приемлемое время, позволяющее построение интерактивных автоматизированных систем для решения подобных задач.

2. Метод должен позволять решать задачи автоматической группировки десятков и сотен тысяч объектов как на сетях, так и в многомерном пространстве характеристик (размерность – до нескольких тысяч измерений)

3. Метод должен позволять решать задачи с применением различных моделей группировки, основанных на поиске центров групп (k-средних, k-медоид, p-медианная).

4. Для непрерывных задач автоматической группировки метод должен быть применим с различными метриками и мерами расстояния, которые могут применяться в практических задачах из различных областей знания.

5. Метод должен комбинироваться с методами локального поиска, применяемыми для конкретных решаемых задач, и предоставлять возможность использовать различные стратегии глобального поиска.

6. Метод должен давать стабильные результаты при многократных его запусках (в идеале – быть детерминированным), при этом точность результатов не должна уступать другим известным методам при сопоставимом времени счета.

7. Метод, кроме решения задачи автоматической группировки с заранее известным количеством групп, должен либо давать оценку числа групп, либо решать сразу серию задач с различным количеством групп.

Выбор жадных агломеративных эвристических алгоритмов в качестве средства локального поиска обусловлен тем, что эти методы позволяют получить результаты высокой точности (хотя и требуют при этом значительных, иногда – неприемлемых вычислительных затрат), сами по себе являются детерминированными процедурами, что в некоторой степени дает надежду на получение более стабильных результатов при применении таких эвристик в составе различных стратегий глобального поиска, в том числе рандомизированных.

Кроме того, исследование ставит целью изучение применимости жадных эвристик к более широкому классу задач, в частности – к задачам оптимизации псевдобулевых функций (связь которых с задачами размещения известна), а также развитие методов решения задач размещения с нестандартными метриками и мерами расстояния, которые могут применяться при решении соответствующих задач автоматической группировки.

## ГЛАВА 2. ПРИМЕНЕНИЕ ЖАДНЫХ ЭВРИСТИЧЕСКИХ АЛГОРИТМОВ К ДИСКРЕТНЫМ ЗАДАЧАМ АВТОМАТИЧЕСКОЙ ГРУППИРОВКИ ОБЪЕКТОВ И МОНОТОННОЙ ПСЕВДОБУЛЕВОЙ ОПТИМИЗАЦИИ

Исследуется дискретная задача автоматической группировки на сети. Поскольку NP-трудность данной задачи доказана, применение точных алгоритмов для задач большой размерности (скажем,  $p > 10$  при  $N > 1000$ ), за исключением особых случаев (например, размещение на деревьях), в интерактивных автоматизированных системах исключено, усилия исследователей сосредоточены на поиске эвристических алгоритмов, предлагающих субоптимальные решения, достаточно близкие к оптимальному.

В настоящей главе представляется новый метод решения задачи автоматической группировки объектов (узлов) на сети, представляющий собой комбинацию особого алгоритма на базе метода изменяющихся вероятностей и модифицированного генетического алгоритма с жадной эвристикой. Кроме того, метод, основанный на применении жадной агломеративной эвристической процедуры в составе генетического алгоритма, применяется к более широкому классу задач условной оптимизации псевдодобулевых монотонных функций, в частности – к задаче составления оптимального расписания загрузки производственных мощностей литейно-прокатного или химического производства.

### 2.1 Постановка задачи автоматической группировки на сети

Задачу автоматической группировки узлов сети можно в общем виде сформулировать следующим образом. Требуется разбить узлы некоторой сети на  $p$  групп таким образом, чтобы близкие друг к другу узлы оказались в одной группе. Более точные математические постановки задачи могут быть различными, например:

- минимизировать сумму расстояний от каждого из узлов до ближайшего узла своей группы:

$$\arg \min_{C(i), i=1, n} \sum_{i=1}^n \min_{X \in C(i) \setminus \{i\}} L(X, A_i);$$

Здесь  $n$  – число узлов сети,  $C(i)$  – множество (группа) узлов, к которому принадлежит  $i$ -й узел (всего не более  $p$  таких групп),  $A_i$  –  $i$ -й узел,  $L(A_i, A_j)$  – расстояние между двумя узлами;

- максимизировать сумму расстояний от каждого из узлов до ближайшего из узлов не своей группы:

$$\arg \max_{C(i), i=1, n} \sum_{i=1}^n \min_{X \in \{1, N\} \setminus \{i\}} L(X, A_i);$$

- минимизировать взвешенное расстояние до центра каждой из групп. Каждому из узлов сети ставится в соответствие действительное число – весовой коэффициент. В качестве центров выбирается любой из узлов:

$$\arg \min_{X_1, \dots, X_p} F(X_1, X_2, \dots, X_p) = \sum_{i=1}^n w_i \min_{j \in \{1, p\}} L(X_j, A_i) \quad (2.1)$$

Здесь  $X_1, \dots, X_p$  – узлы, выбранные в качестве центров.

Данная задача – задача о  $p$ -медиане (она же –  $p$ -медианная или  $k$ -медианная задача) – является одной из ключевых задач как дискретной, так и непрерывной теории размещения.

Цель  $p$ -медианной задачи на сети [217], являющейся одной из четырех основных задач теории размещения, является нахождение  $p$  узлов сети, таких, чтобы сумма взвешенных расстояний от всех узлов сети до ближайшего из  $p$  выбранных узлов достигало минимума. В общем случае задача является NP-трудной [151] и не может быть решена за полиномиальное время, алгоритм с полиномиальным временем известен только для деревьев [218].

Ввиду высокой вычислительной сложности предложено множество эвристических подходов к решению задачи.

Несмотря на высокую сложность точного решения задачи, различные эвристические алгоритмы позволяют получать хорошие результаты для многих практических задач за приемлемое время. Простейшие подходы состоят в локальном поиске, при котором в качестве окрестности рассматривается множество узлов сети, смежных узлам текущего промежуточного решения задачи [219, 220]. Раббани [221] предлагает алгоритм, основанный на теории графов, который, однако, подходит лишь для очень небольших задач. Отдельно рассмотрены задачи на двудольных графах [222], но и здесь приемлемые результаты достигаются лишь для небольших задач.

Использование лагранжевых релаксаций позволяет получать приближенное решение задач достаточно большой размерности [223, 224], до 90000 узлов в сети. Тем не менее, ”хорошие” решения [224] достигнуты подобной с применением подобной техники лишь для задач до  $n=3795$ , причем такие задачи ( $n$  – число вершин графа) авторами названы задачами большой размерности.

Способам эффективного решения этой последней задачи в основном и посвящена настоящая глава. Для задачи автоматической группировки в такой либо подобной постановке предложен алгоритм на базе метода изменяющихся вероятностей, а также генетический алгоритм с жадной агломеративной эвристикой, область применения которого расширена до задач псевдобулевой оптимизации с монотонной целевой функцией и монотонными левыми частями ограничений. Эффективность алгоритмов продемонстрирована на тестовых примерах  $p$ -медианных задач на сети большой размерности ( $n>500$ ), а также на задаче теории расписаний – задаче календарного планирования загрузки производственных мощностей литейно-прокатных и химических производств.

Задачи размещения — класс задач оптимизации, в которых основными параметрами выступают координаты (местоположения) точек в некотором пространстве (непрерывном либо дискретном) и расстояния между ними [225, 226]. Задачи находят непосредственное практическое применение как при размещении производств, сетей, объектов инфраструктуры, датчиков и т.п., так и при решении задач кластерного анализа, распознавания образов, теории оценивания [227].

Как отмечалось в Главе 1, задача Вебера (Ферма-Вебера) [225] — задача о поиске точки, сумма взвешенных расстояний от которой до известных точек (точек требования) минимальна, является обобщением простейшей задачи Ферма о нахождении точки, сумма расстояний от которой до трех известных точек минимальна, и имеет, в свою очередь, множество обобщений, главные из которых — множественная задача Вебера [228] и  $p$ -медианная задача [217] в непрерывном пространстве или на сети. Собственно задача Вебера – задача (2.1) при  $p=1$ . В контексте задач автоматической группировки задача Вебера – поиск центра некоторой группы объектов, например, сети (в качестве таковой может выступать некоторая подсеть большой сети, узлы которой разбиваются на группы). Центром (в некоторых постановках задачи используются термины «центроид»,

«медоид», «медиана» и др.) будем называть собственно решение 1-медианной задачи или задачи Вебера:

$$F(X) = \sum_{i=1}^n w_i L(X, A_i) \rightarrow \min.$$

В задачах на сети решение в общем случае требует перебора элементов множества  $\{A_i\}$  в качестве решений-кандидатов. На практике при решении задач на сети более актуальной (по сравнению с дискретной задачей Вебера) является обратная задача – определение подмножества (группы) узлов сети, относящихся к  $j$ -му центру. Узел относится к центру, если данный центр является для него ближайшим. Если расстояния до двух центров равны, могут выстраиваться различные системы предпочтений. Например, узел относится к центру с наименьшим номером.

При создании алгоритма для  $p$ -медианных задач с большим объемом входных данных попутно должны быть решены задачи оптимального использования вычислительных мощностей. В связи с повсеместным распространением многоядерных систем, разработка параллельных версий алгоритмов для систем с общей памятью может существенно снизить временные затраты на решение подобных задач. Прогнозируемое появление и распространение в ближайшем будущем персональных кластеров делают оправданной разработку параллельных алгоритмов размещения для систем с распределенной памятью. Эффективность параллельных алгоритмов на базе метода изменяющихся вероятностей, описанных в [156, 229, 230], для задач большой размерности доказана экспериментально.

В настоящей Главе рассматривается применение жадных агломеративных эвристических алгоритмов и метода изменяющихся вероятностей (МИВЕР) в комбинации с особым генетическим алгоритмом к задачам автоматической группировки узлов сети, а также к более широкому классу задач псевдобоулевой оптимизации. Эффективность применяемых алгоритмов показана экспериментально для  $p$ -медианных задач большой размерности, а также для задач составления расписаний.

## 2.2 Известные алгоритмы группировки узлов сети

В  $p$ -медианной задаче на сети [217] поиск решений (медиан) осуществляется в известном конечном множестве узлов графа (сети). В данной задаче требуется выбрать  $p$

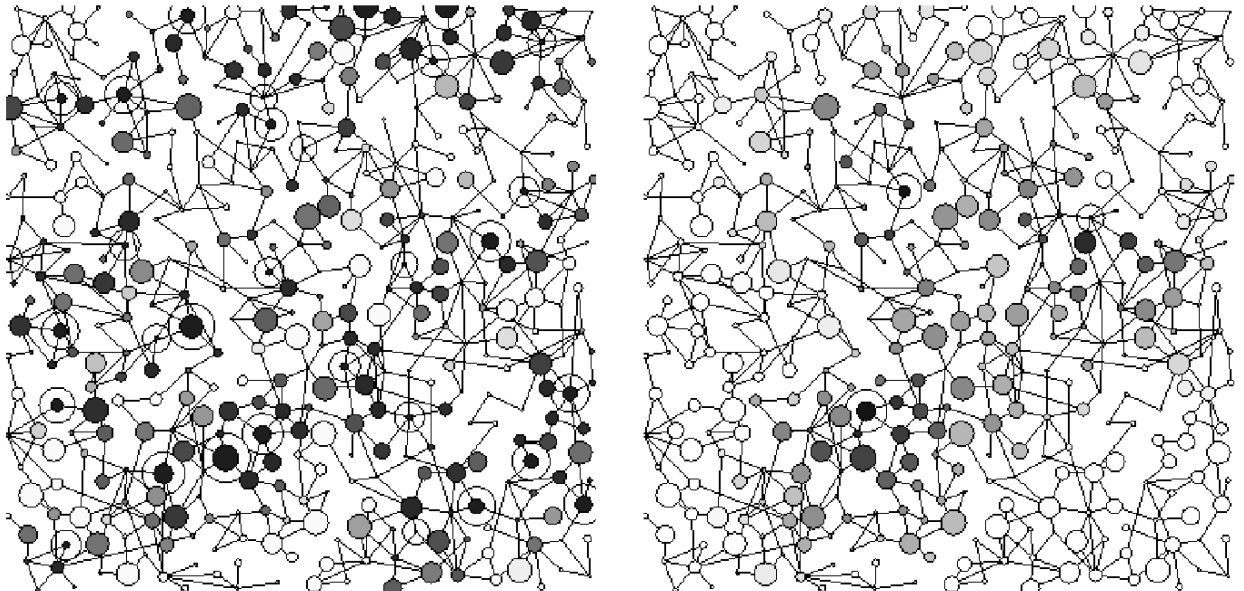
вершин таким образом, чтобы сумма взвешенных расстояний от каждой из вершин графа до ближайшей из выбранных вершин была минимальной. Такое множество будем называть  $p$ -медианой, а элементы этого множества – медианами.

Пусть  $G = (V, E)$  - неориентированный граф (сеть),  $V = \{v_1, \dots, v_n\}$  - множество вершин (рис. 2.1),  $E = \{e_i | i = \overline{1, m}\}$  - множество ребер,  $e_i = (v_j, v_k)$ ,  $j, k \in \{1, n\}$ ,  $i \in \{1, m\}$  без петель ( $e_i \neq (v_j, v_j) \quad \forall i = \overline{1, m}, j = \overline{1, n}$ ). Для каждого ребра  $e_i$  его длина  $l_i$  определена,  $l_i \geq 0 \forall i = \overline{1, m}$ . Для ребра  $e_i = (v_j, v_k)$ , обозначим  $l_{j,k} = l_i$ . Вес  $w_j \geq 0$  определен для каждой вершины  $v_j$ . Для каждой пары вершин  $(v_j, v_k)$  функция расстояния  $L(j, k)$  определена как длина кратчайшего пути из  $v_i$  в  $v_j$ :

$$L(j, k) = \sum_{q \in P_{j,k}^*} l_q = \min_{P \in P_{j,k}} \sum_{q \in P} l_q.$$

Здесь  $P_{j,k}^*$  - множество ребер кратчайшего пути между  $v_j$  и  $v_k$ ,  $P_{j,k}$  - множество всех возможных путей между этими вершинами. Можно сформулировать  $p$ -медианную задачу так:

$$\arg \min_{m_1, \dots, m_p \in \{1, n\}} f(m_1, \dots, m_p) = \arg \min_{m_1, \dots, m_p \in \{1, n\}} \sum_{i=1}^n w_i \min_{i=1, p} L(m_j, i), \quad p < n.$$



а) Решение для  $p = 32$       б) Решение для  $p = 3$

Рисунок 2.1 – Схема задачи и ее решения,  $n = 400$

Задача заключается в выборе  $p$  вершин таким образом, чтобы сумма взвешенных расстояний от каждой вершины до ближайшей из выбранных вершин была минимальной.

Пусть

$$S_i = \{j \mid \exists e_j = (v_i, v_k), j \in \overline{1, m}, k \in \overline{1, n}\}$$

- множество ребер, инцидентных  $i$ -й вершине;

$$C_i = \{k \mid \exists e_j = (c_i, v_k), j \in \overline{1, m}, k \in \overline{1, n}\}$$

- множество индексов вершин, смежных с  $i$ -й;

$$l_i^* = \min_{j \in \{1, p\}} L(m_j, i)$$

- расстояние (длина кратчайшего пути) от  $i$ -й вершины до ближайшей из  $p$  вершин  $m_1, \dots, m_p$ .

Для вычисления расстояний до ближайших выбранных вершин и целевой функции  $f(m_1, \dots, m_p)$  мы используем следующий алгоритм, основанный на алгоритме Дейкстры [231, 232] (фактически на сети строится диаграмма Вороного одновременно с подсчетом расстояний [233]):

Алгоритм 2.1. вычисление целевой функции  $p$ -медианной задачи на сети.

Дано:  $p$  индексов выбранных вершин  $m_1, \dots, m_p$ .

1:  $l_i^* = +\infty \forall i = \overline{1, n}$ ;

2:  $l_{m_i}^* = 0 \forall i = \overline{1, p}$ ;

3:  $V^* = \{m_1, \dots, m_p\}$ ;  $V^{**} = V^*$ ;

4: пока  $|V^{**}| \neq 0$  выполнять:

4.1:  $V' = \emptyset$ ;

4.2: для  $i \in V^{**}$  выполнять:

4.2.1: для  $j \in C_i$  выполнять:

4.2.1.1: если  $v_j \notin V^*$ , то  $V' = V' \cup \{j\}$ ;  $l_j^* = l_i^* + l_{i,j}$ ;

4.2.1.2: иначе если  $l_j^* = l_i^* + l_{i,j}$ , то  $l_j^* = l_i^* + l_{i,j}$ ;

4.2.1.3: следующая итерация 4.2.1;

4.2.2: следующая итерация 4.2;



4.3:  $V^{**} = V'$ ;  $V^* = V^* \cup V'$ ;

4.4: следующая итерация 4;

5: возврат  $f(m_1, \dots, m_p) = \sum_{i=1}^n w_i l_i^*$ .

Для сравнения мы использовали алгоритм локального поиска [220] со случайным порядком выбора вершин (Алгоритм 2.2) в качестве одного из простейших, но эффективных алгоритмов.

Алгоритм 2.2. Локальный поиск в окрестности, состоящей из смежных вершин

Дано: массив индексов вершин (начальное решение)  $M = \{m_1, \dots, m_p\}$ , значение целевой функции  $f^* = f(m_1, \dots, m_p)$ .

1: выстроить в случайном порядке (перетасовать) элементы  $M$ ;  $r = 0$ ;

2: для каждого элемента  $m$  массива  $M$  выполнять:

2.1: для каждой вершины  $m^*$ , смежной с  $m$  выполнять:

2.1.1:  $f^{**} = f(m_1, \dots, m^*, \dots, m_p)$  (здесь вершина  $m$  замещена вершиной  $m^*$ );

2.1.2: если  $f^{**} < f^*$ , то заместить  $m$  вершиной  $m^*$  в множестве  $M$ ;  $f^* = f^{**}$ ;

$r = 1$ ;

2.1.3: следующая итерация 2.1;

2.2: следующая итерация 2;

3: если  $r = 1$ , то перейти к 1;

5: возврат нового решения  $(m_1, \dots, m_p)$ .

Генетический алгоритм с жадной эвристикой, предложенный в [204], включает в себя специальную процедуру кроссинговера (Алгоритм 2.3). "Хромосомами", кодирующими «особи» этого алгоритма, являются множества вершин (промежуточные решения р-медианно задачи). Будем считать понятия «хромосома» и «особь» эквивалентными.

Алгоритм 2.3. Оператор кроссинговера (скрещивания) для ГА с жадной эвристикой

Дано: Набор индексов вершин («хромосомы»)  $M_1 = \{m_{1,1}, \dots, m_{1,p}\}$ ,  
 $M_2 = \{m_{2,1}, \dots, m_{2,p}\}$ .

1:  $M = M_1 \cup M_2$ ;  $p_M = |M|$ ;

2: пока  $p_M > p$ , выполнять:

2.1:  $f^* = +\infty$ ;

2.2: для каждой вершины  $m^*$  в множестве  $M$  выполнять:

2.2.1:  $M^* = M \setminus \{m^*\}$ ;  $f^{**} = f(M^*)$ ;

2.2.2: если  $f^{**} < f^*$ , то  $m^{**} = m^*$ ;

2.1.3: перейти к 2.2;

2.3:  $M = M \setminus \{m^{**}\}$ ;  $p_M = p_M - 1$ ;

2.3: перейти к 2;

3: возврат нового решения («хромосомы»)  $M$ .

Данный метод использует оригинальную детерминированную процедуру создания начальной популяции [204]. Метод не включает в себя процедур мутации, характерных для генетических алгоритмов.

Метод изменяющихся вероятностей (МИВЕР) – это метод псевдо-булевой оптимизации, целевая функция для его применения должна быть функцией от булевых переменных.

Введем новые переменные  $x_1, \dots, x_n$ :

$$x_i = \begin{cases} 1, & i \in \{m_1, \dots, m_p\} \\ 0, & i \notin \{m_1, \dots, m_p\} \end{cases} \quad (2.2)$$

В данном случае задача имеет ограничение

$$\sum_{i=1}^n x_i = p. \quad (2.3)$$

Обратное преобразование к задаче с целочисленными переменными осуществляется следующим образом:

$$\{m_1, \dots, m_p\} = \{i \mid x_i = 1, i = \overline{1, n}\}.$$

Задача с псевдобулевой целевой функцией формулируется следующим образом:

$$f_b(x_1, \dots, x_n) = f(\{j \mid x_j = 1, j = \overline{1, n}\}) = \sum_{i=1}^n w_i \min_{j \mid x_j = 1, j = \overline{1, n}} L(i, j)$$

с ограничением (2.3).

В данной форме задача может быть решена множеством методов [234-237, 229], включая метод изменяющихся вероятностей.

### 2.3 Алгоритм метода изменяющихся вероятностей

При решении дискретных задач размещения применяется генетический алгоритм и другие методы [238, 204].

Изначально разработанный для решения безусловных задач псевдодобулевой оптимизации, метод изменяющихся вероятностей (МИВЕР) [237, 239, 234, 240] — это метод случайного поиска, который может также рассматриваться как специфический вариант генетического алгоритма [241].

#### Алгоритм 2.4. Метод изменяющихся вероятностей

Дано:  $F(X)$  – псевдодобулева целевая функция – вещественная функция  $N$  булевых переменных,  $R$  – предельное число итераций. Здесь  $X=(x_1, \dots, x_N)$  – вектор булевых переменных.

1.  $k=0$ . устанавливаются начальные значения вероятностей  $P_k=\{p_{k1}, p_{k2}, \dots, p_{kN}\}$ , где  $p_{kj}=P\{x_j=1\}$ . Здесь и далее индекс  $k$  – номер итерации. Корректный выбор начальных значений  $P_0$  — важная проблема, особенно при переходе к условной оптимизации.

2. В соответствии вектором распределений  $P_k$ , генерируем набор из  $N_{POP}$  векторов  $X_{ki}$ ,  $i \in \{1, \dots, N_{POP}\}$ , каждый элемент  $p_{kj}$  вектора  $P_k$  задает математическое ожидание генерации единицы в  $j$ -й компоненте векторов  $X_{ki}$ . Число генерируемых на каждом шаге векторов  $N_{POP}$  называется размером популяции по аналогии с генетическим алгоритмом.

3. Вычисляется значение  $F(X_{ki})$  для каждого из сгенерированных векторов.

4. Выбираются некоторые значения целевой функции  $F(X_{ki})$  и соответствующие векторы  $X_{ki}$ , например, экземпляры сгенерированных векторов с максимальным и минимальным значением  $F(X_{ki})$ .

5. На основе результатов шага 4 модифицируется вектор  $P_k$ .

6.  $k=k+1$ , если  $k < N_{STEPS}$ , то переход к 2. Возможны другие условия останова. Здесь  $N_{STEPS}$  – установленное предельное число итераций.

## 7. Иначе останов.

Проведем некоторые аналогии с непрерывными задачами размещения.

Для применения в задачах условной оптимизации метод должен быть изменен. Модификации, предложенные в [242], позволяют решать задачи размерности до миллионов переменных.

Непрерывные задачи размещения, такие как множественная задача Вебера с евклидовой, прямоугольной или чебышёвской метриками, поддаются анализу и решаются аналитическими методами. Процедура Вайсфелда [144] и метод Трубина для задач с прямоугольной метрикой [243] основаны на том, что частные производные целевой функции определены, т.е., малое изменение координат одной из искомым точек допустимого решения ведет к некоторому малому изменению значения целевой функции. На том же свойстве основаны и методы для непрерывной  $p$ -медианной задачи. Если  $X = (x_1, x_2)$  – решение некоторой задачи размещения на плоскости, то, за исключением некоторых особых точек,

$$\frac{\Delta f(X)}{\Delta X} = \frac{\Delta f(x_1, x_2)}{\Delta \sqrt{\Delta x_1 + \Delta x_2}} < const.$$

Пусть  $L_{max}$  - максимальное расстояние между двумя вершинами графа:

$$L_{max} = \max_{i, j \in \{1, n\}} L(i, j),$$

$l_{avg}$  – средняя длина ребра:

$$l_{avg} = \sum_{j=1}^m l_j / m. \quad (2.4)$$

Наш алгоритм основывается на двух гипотезах:

Гипотеза 2.1. Если  $l_{avg}/L_{max}$  мало ( $l_{avg}/L_{max} \rightarrow 0$ ), то  $p$ -медианная задача на сети имеет свойства, близкие к непрерывной задаче. В частности, если  $\{m_1, \dots, m_p\}$  – решение  $p$ -медианной задачи, то, заменив любую  $m_i$ -ю вершину в этом решении на любую  $j$ -ю вершину, такую, что  $L(j, m_i)$  мало, мы получим малое изменение целевой функции:

$$\exists l_{\Delta}, \Delta_{Fmax} > 0: (j, m_i) < l_{\Delta} \Rightarrow |f(m_1, \dots, m_i, \dots, m_p) - f(m_1, \dots, j, \dots, m_p)| < \Delta_{Fmax}$$

Гипотеза 2.2. Если решение  $\{m_1, \dots, m_i, \dots, m_j, \dots, m_p\}$  содержит две вершины –  $v_i$  и  $v_j$  такие, что  $L(m_i, m_j)$  мало, то высока вероятность (математическое ожидание) того, что для решения  $\{m_1, \dots, m_i, \dots, k, \dots, m_p\}$  с вершиной  $v_j$ , замененной на другую случайным образом выбранную вершину  $v_k$ , значение целевой функции будет лучше (меньше), чем для исходного решения:

$$\begin{aligned} & \exists l_{\min} : L(m_i, m_j) < l_{\min} \wedge L(m_i, k) > l_{\min} \Rightarrow \\ & \Rightarrow \mu\{f(m_1, \dots, m_i, \dots, m_j, \dots, m_p) \geq f(m_1, \dots, m_i, \dots, k, \dots, m_p)\} > 0.5 \\ & \text{и } \lim_{l_{\min} \rightarrow 0} \mu\{f(m_1, \dots, m_i, \dots, m_j, \dots, m_p) \geq f(m_1, \dots, m_i, \dots, k, \dots, m_p)\} \rightarrow 1. \end{aligned}$$

Очевидно, что метрические графы с относительно малой длиной ребра этим условиям соответствуют. Но и в иных случаях данные утверждения справедливы (например, рис. 2.1).

Докажем состоятельность гипотез для специальных (предельных) случаев.

Лемма 2.1. Пусть  $L(m_i, j) = 0$ . Тогда  $f(m_1, \dots, m_i, \dots, m_p) = f(m_1, \dots, j, \dots, m_p)$ .

*Доказательство.* Выберем произвольно вершину (пусть это будет  $i^*$ -я вершина),  $i^* \in \overline{\{1, n\}}$ .

Если  $i^* \in \{m_1, \dots, m_i, \dots, m_p\}$ , то, очевидно,  $\min_{i'' \in \{m_1, \dots, m_i, \dots, m_p\}} L(i^*, i'') = 0$ .

$$\min_{i'' \in \{m_1, \dots, j, \dots, m_p\}} L(i^*, i'') = \min \left\{ \min_{i'' \in \{m_1, \dots, m_i, \dots, m_p\}} L(i^*, i'') + L(m_i, j); L(i^*, j) \right\} = \min\{0 + 0; L(i^*, j)\} = 0.$$

Для случая  $i^* \notin \{m_1, \dots, m_i, \dots, m_p\}$  введем следующие обозначения:

$P_{i^*, j}$  - множество всех возможных путей из  $i^*$ -й вершины к  $j$ -й,

$P_{i^*, m_i}$  - множество всех возможных путей из  $i^*$ -й вершины в  $m_i$ -ю,

$P_{i^* (m_i) j}$  - множество всех возможных путей из  $i^*$ -й вершины к  $j$ -й через  $m_i$ -ю вершину,

$P_{i^* (\overline{m_i}) j}$  - множество всех возможных путей из  $i^*$ -й вершины в  $j$ -ю, не включающих  $m_i$ -ю вершину.

$$\begin{aligned} L(i^*, j) &= \min_{P \in P_{i^*, j}} \sum_{e_k \in P} l_k = \min \left\{ \min_{P \in P_{i^* (m_i) j}} \sum_{e_k \in P} l_k; \min_{P \in P_{i^* (\overline{m_i}) j}} \sum_{e_k \in P} l_k \right\} = \\ &= \min \left\{ \min_{P \in P_{i^*, m_i}} \sum_{e_k \in P} l_k + \min_{P \in P_{m_i, j}} \sum_{e_k \in P} l_k; \min_{P \in P_{i^* (\overline{m_i}) j}} \sum_{e_k \in P} l_k \right\} = \end{aligned}$$

$$= \min\{ \min_{P \in P_{i^*, m_i}^*} \sum_{e_k \in P} l_k + 0; \min_{P \in P_{i^*}(\overline{m_i})_j} \sum_{e_k \in P} l_k \} = \min\{L(i^*, m_i); \min_{P \in P_{i^*}(\overline{m_i})_j} \sum_{e_k \in P} l_k \} \leq L(i^*, m_i).$$

$$L(i^*, m_i) = \min_{P \in P_{i^*, m_i}^*} \sum_{e_k \in P} l_k = \min\{ \min_{P \in P_{i^*}(\overline{j})_{m_i}} \sum_{e_k \in P} l_k; \min_{P \in P_{i^*}(\overline{j})_{m_i}} \sum_{e_k \in P} l_k \} =$$

$$= \min\{ \min_{P \in P_{i^*, j}^*} \sum_{e_k \in P} l_k + \min_{P \in P_{j, m_i}} \sum_{e_k \in P} l_k; \min_{P \in P_{i^*}(\overline{j})_{m_i}} \sum_{e_k \in P} l_k \} =$$

$$= \min\{ \min_{P \in P_{i^*, j}^*} \sum_{e_k \in P} l_k + 0; \min_{P \in P_{i^*}(\overline{j})_{m_i}} \sum_{e_k \in P} l_k \} = \min\{L(i^*, j); \min_{P \in P_{i^*}(\overline{j})_{m_i}} \sum_{e_k \in P} l_k \} \leq L(i^*, j).$$

$$L(i^*, m_i) \leq L(i^*, j) \wedge L(i^*, j) \leq L(i^*, m_i) \Rightarrow L(i^*, m_i) = L(i^*, j).$$

Лемма 2.2. Пусть  $\{m_1, \dots, m_i, \dots, m_j, \dots, m_p\}$  - решение  $p$ -медианной задачи на сети с  $n$  узлами,  $w_i > 0 \forall i = \overline{1, n}$ ,  $L(m_i, m_j) = 0$ , при этом  $\exists k \in \overline{1, n} : L(m_q, k) > 0 \forall q = \overline{1, p}$ .

$$\text{Тогда } f(m_1, \dots, m_i, \dots, m_j, \dots, m_p) > f(m_1, \dots, m_i, \dots, k, \dots, m_p).$$

Доказательство. Введем обозначения  $M_0 = \{m_1, \dots, m_i, \dots, m_j, \dots, m_p\}$ ,  $M_1 = \{m_1, \dots, m_i, \dots, k, \dots, m_p\}$ .

Определим функцию

$$f_m(i', S) = \min_{i'' \in S} L(i'', i').$$

Ее значения для множеств, обозначенных выше, равны

$$f_m(i', M_0) = \min\{ f_m(i', \{m_i\}); f_m(i', \{m_j\}); \\ f_m(i', M_0 \setminus \{m_i\}) \} \forall i' \in \overline{1, n}.$$

Принимая во внимание, что  $L(m_i, m_j) = 0$ , согласно Лемме 2.1, для множества вершин решения

$$f_m(i', \{m_1, \dots, m_i, \dots, m_j, \dots, m_p\}) = f_m(i', \{m_1, \dots, m_i, \dots, m_i, \dots, m_p\}) = f_m(i', M_0) \quad \forall i' \in \overline{1, n}.$$

Далее,

$$f_m(i', M_1) = \min\{ f_m(i', \{m_i\}); f_m(i', \{k\}); f_m(i', M_1 \setminus (\{m_i\} \cup \{k\})) \} \quad \forall i' \in \overline{1, n}; \\ f_m(i', M_0) = \min\{ f_m(i', \{m_i\}); f_m(i', M_0 \setminus \{m_i\}) \} = \min\{ f_m(i', \{m_i\}); f_m((M_1 \setminus \{k\}) \setminus \{m_i\}) \} = \\ = \min\{ f_m(i', \{m_i\}); f_m(M_1 \setminus (\{k\} \cup \{m_i\})) \} \geq \min\{ f_m(i', \{m_i\}); f_m(i', \{k\}); \\ f_m(i', M_1 \setminus (\{m_i\} \cup \{k\})) \} = f_m(i', M_1) \quad \forall i' \in \overline{1, n};$$

Таким образом,

$$f(m_1, \dots, m_i, \dots, k, \dots, m_p) = \sum_{i'=1}^n f_m(i', M_1) \leq \sum_{i'=1}^n f_m(i', M_0) = f(m_1, \dots, m_i, \dots, m_j, \dots, m_p).$$

Докажем состоятельность Гипотезы 2.2 экспериментально.

Алгоритм 2.5. Определение оценки вероятности выполнения условия Гипотезы 2.2 при заданном предельном расстоянии  $L_0$ .

Дано: граф  $G$  с  $n$  вершин, количество вершин в решении  $p$ , предельное расстояние  $L_0$ .

1. Присвоить  $N=0, N_>=0$ .
2. Выбрать случайным образом  $p$  индексов вершин из  $n$ :  $m_1, \dots, m_p \in \overline{\{1, n\}}$ .
3. Если не существуют  $i, j \in \overline{\{1, p\}}: L(m_i, m_j) < L_0$ , то перейти к Шагу 2.
4. Выбрать случайным образом  $k \in \overline{\{1, n\}}$ . Если  $L(k, m_j) < L_0$ , то повторить Шаг 4.
5. Если  $f(m_1, \dots, m_i, \dots, m_j, \dots, m_p) \geq f(m_1, \dots, m_i, \dots, k, \dots, m_p)$ , то  $N_>=N_>+1$ .
6. Присвоить  $N=N+1$ ; если  $N < 1000$ , то перейти к Шагу 2.
7. ОСТАНОВ, возврат  $P_> = N_> / 1000$ .

Результаты работы Алгоритма 2.5 для автоматически сгенерированных задач (см. Алгоритм 2.7) и для задач pmed22 и pmed39 из набора тестовых задач ORLIB [244] представлены на рис.2.2. Оценка вероятности дана в процентах. Для задач pmed22 и pmed39, благодаря специфике их генерации, соотношение  $l_{avg} / L_{max}$  велико, и условие Гипотезы 2.2 выполняется лишь при очень малых значениях  $L_0$  (много меньших, чем  $l_{avg}$ ).

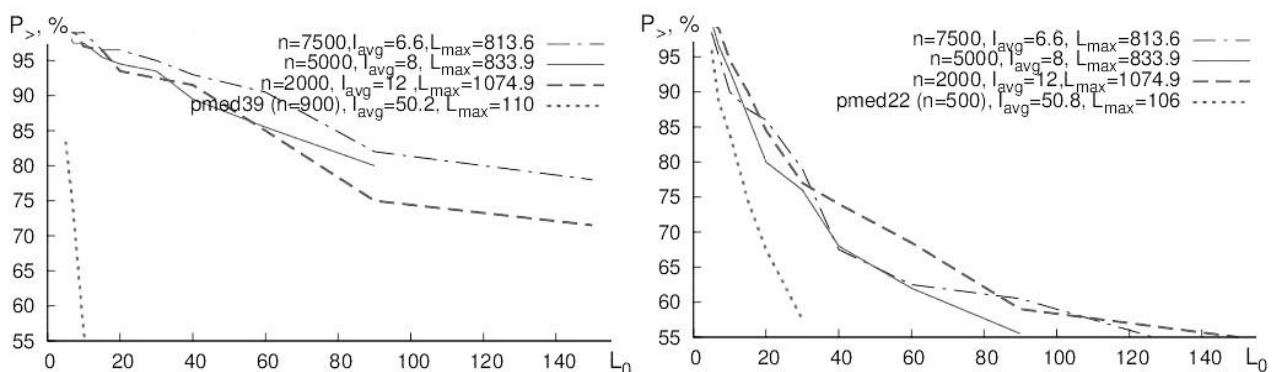


Рисунок 2.2 – Соотношение оценки вероятности выполнения условия Гипотезы 2.2 и предельного расстояния  $L_0$ .

Для применения к р-медианной задаче, которая, будучи сформулированной в виде задачи псевдодулевой оптимизации, является задачей условной оптимизации, метод изменяющихся вероятностей должен быть соответствующим образом адаптирован. В

нашей версии Алгоритма 2.4 Шаги 2 и 5 модифицированы. На Шаге 2 (генерация экземпляров векторов  $X$ ) следует принимать во внимание условие (2.3). Приведенный ниже алгоритм генерирует только допустимые решения.

Алгоритм 2.6. Шаг 2 Алгоритма 2.4

Дано: Вектор вероятностей  $P = (p_1, \dots, p_n)$ .

- 1: Инициализировать множество  $\chi = \emptyset$ ;
- 2: Для каждого  $i \in \overline{\{1, p\}}$  выполнять
  - 2.1: Присвоить  $r = \text{Random}() \cdot \sum_{j=1}^n p_j$ ;  $S = 0$ ;
  - 2.2: Для каждого  $j \in \overline{\{1, n\}}$  выполнять
    - 2.2.1: Присвоить  $S = S + p_j$ ;
    - 2.2.2: Если  $S \geq r$ , то присвоить  $\chi = \chi \cup \{j\}$ ; перейти к 2.3;
    - 2.2.3: Следующая итерация 2.2;
- 2.3: Следующая итерация 2;
- 3: Возврат  $\chi$ .

Результатом данного алгоритма является множество  $\chi$ . Соответствующий вектор булевых переменных  $X$  может быть рассчитан в соответствии с (2.2). Здесь,  $\text{Random}()$  - генератор случайных величин с равномерным распределением ( $\text{Random}() \sim U[0,1)$ ).

Пусть  $L_0$  будет максимальным расстоянием, считающимся «малым» в Гипотезе 2.1 и Гипотезе 2.2. В соответствии с Гипотезой 2.2,

$$\mu\{\exists m_i, m_j \in \chi : L(m_i, m_j) < L_0\} < \mu\{L(m_i, m_j) \geq L_0 \forall m_i, m_j \in \chi\};$$

$$\lim_{L^* \rightarrow 0} \mu\{\exists m_i, m_j \in \chi : L(m_i, m_j) < L^*\} = 0.$$

Предложим дальнейшую модификацию Алгоритма 2.4.

Алгоритм 2.7. Шаг 2 Алгоритма 2.4, версия.2

Дано: Вектор вероятностей  $P = (p_1, \dots, p_n)$ .

- 1:  $\chi = \emptyset$ ;
- 2: для каждого  $i \in \overline{\{1, p\}}$  выполнять



2.1: Присвоить  $P^* = P$ ;

2.2: Присвоить  $r = \text{Random}() \cdot \sum_{j=1}^n p_j^*$ ;  $S = 0$ ;

2.3: Для каждого  $j \in \overline{\{1, n\}}$  выполнять

2.3.1: Присвоить  $S = S + p_j^*$ ;

2.3.2: Если  $S \geq r$ , тогда присвоить  $\chi = \chi \cup \{j\}$ ;  $j' = j$ ; перейти к 2.3;

2.3.3: Следующая итерация 2.3;

2.4: Для каждого  $j \in \{k \mid k \in \overline{\{1, n\}} \wedge L(j', k) < L_0\}$  выполнять:  $p_j^* = p_j^* \cdot L(j, j') \cdot L_0$  ;

следующая итерация 2.4;

2.5: Следующая итерация 2;

3: Возврат  $\chi$  .

Шаг 5 Алгоритма 2.4 (адаптация вектора вероятностей  $P$ ) на его  $k$ -й итерации должен быть выполнен в соответствии с Гипотезой 2.2. Мы используем мультипликативную адаптацию.

$$P_{k,i} = p_{(k-1),i} \cdot d_{k,i}^b / d_{k,i}^w, \quad (2.5)$$

$$d_{k,i}^b = \begin{cases} 1 + \frac{L_0}{1 + L(i^b(i), i)}, & L(i, i^b(i)) < L_0, \\ 1, & L(i, i^b(i)) \geq L_0 \end{cases}, \quad (2.6)$$

$$d_{k,i}^w = \begin{cases} 1 + \frac{L_0}{1 + L(i^w(i), i)}, & L(i, i^w(i)) < L_0, \\ 1, & L(i, i^w(i)) \geq L_0 \end{cases}. \quad (2.7)$$

Здесь

$$i^b(i) = \arg_{i'} \min_{i' \in \chi^b} L(i, i'),$$

$$i^w(i) = \arg_{i'} \min_{i' \in \chi^w} L(i, i'),$$

$\chi^b$  и  $\chi^w$  – лучший и худший экземпляры множеств индексов узлов  $\chi$ , сгенерированных Алгоритмом 2.6 или Алгоритмом 2.7. В простейшем случае,

$$\chi^b = \arg_{\chi'} \min_{\chi' \in X} f(\chi'),$$

$$\chi^w = \arg_{\chi'} \max_{\chi' \in X_k} f(\chi').$$

Здесь  $X_k$  - множество всех экземпляров множества  $\chi$  на  $k$ -й итерации Алгоритма 2.4.

Отметим, что дискретизированная обобщенная задача Вебера, описанная в [156] является особым случаем  $p$ -медианной задачи, рассматриваемой в данном параграфе.

## 2.4 Результаты метода изменяющихся вероятностей, настройка параметров

Для целей тестирования мы использовали  $p$ -медианные задачи, сгенерированные автоматически специальным Алгоритмом 2.8.

Алгоритм 2.8. Генерация сетей для тестовых задач.

Дано: число узлов будущей сети  $n$ .

1: Для  $i \in \overline{1, n}$  выполнять

1.1: Присвоить  $c_x^i = \text{Random}() \cdot 500$ ;  $c_y^i = \text{Random}() \cdot 500$ ;  $w_i = 1 + 9\text{Random}()$ ;

1.2: Если  $\exists j \in \overline{j, n-1} : \sqrt{(c_x^i + c_x^j)^2 + (c_y^i + c_y^j)^2} < 10$ , то перейти к 1.1;

1.3: Следующая итерация 1;

2. Заполнить матрицу смежности  $A$  нулевыми значениями; присвоить  $E = \emptyset$ ;

3: Для каждого  $i \in \overline{1, n}$  выполнять

3.1: Присвоить

$$D_i = \begin{cases} 1, & i \in \overline{[0.6n+1], n}, \\ 2, & i \in \overline{[0.4n]+1, [0.6n]}, \\ 3, & i \in \overline{[0.25n]+1, [0.4n]}, \\ 4 + [\text{Random}() \cdot 4], & i \leq [0.25n]. \end{cases}$$

3.2: Для  $d \in \overline{1, \sum_{j=1}^n A_{i,j}}$  выполнять

3.2.1: Присвоить  $j = \arg \min_{j \in \overline{1, n}, A_{i,j} > 0} \sqrt{(c_x^i - c_x^j)^2 + c_y^i - c_y^j)^2}$ ;

$A_{i,j} = 1$ ;  $A_{j,i} = 1$ ;  $E = E \cup \{(i, j)\}$ ;  $l_{i,j} = \sqrt{(c_x^i - c_x^j)^2 + c_y^i - c_y^j)^2}$ ;

3.2.2: Следующая итерация 3.2;

4: Возвратить матрицу смежности  $A$ , множество ребер  $E$ , длины ребер  $\{l_{i,j}\} \forall (i, j) \in E$  и их веса  $w_i \forall i \in \overline{1, n}$ .

Схема подобной сети показана на рис.2.1. Длины ребер пропорциональны длинам линий, представляющих их на схеме, Диаметры окружностей, которыми представлены узлы, пропорциональны весовым коэффициентам узлов. Кроме того, на рис. 2.1 показаны решения (выбранные узлы), при  $p = 32$  и  $p = 3$ . Узлы, выбранные в качестве решения, обведены дополнительной окружностью. Для каждого из остальных узлов цветом обозначено расстояние до ближайшего из выбранных узлов. Узлы, ближайшие к выбранным, показаны темным, более удаленные – светлым.

Для наших экспериментов мы использовали вычислительную систему Depo X8Sti (6-ядерный ЦПУ Xeon X5650 2.67 GHz, 12Гб ОЗУ, технология hyperthreading отключена) и компилятор ifort с полной оптимизацией и неявным параллелизмом (опция -O3). Сравнение результатов, достигнутых вычислительной системой данной конфигурации с результатами маленькой вычислительной системы с 2-ядерным ЦПУ Atom N2701.6GHz, 1Гб ОЗУ, показаны на рис. 2.3 (комбинированный метод "МИВЕР+ГА" ("Prob.chang.+GA") описан ниже.

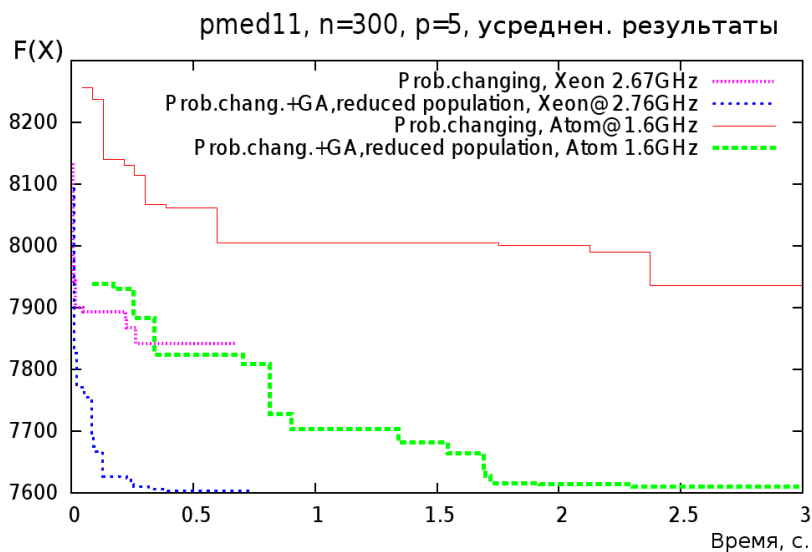


Рисунок 2.3 – Сравнение производительности алгоритма на высокопроизводительной системе (ЦПУ Xeon) и на малой вычислительной система (ЦПУ Atom)

Рис. 2.4 иллюстрирует динамику изменения вектора вероятностей для задачи с  $p = 3$ . Узлы, которым соответствуют компоненты вектора вероятностей с большими (близкими к 1) значениями, показаны светлым, узлы с меньшими значениями вероятностей - темным.

Диаметр окрестности  $L_0$ , для которой соблюдаются условия Гипотезы 2.1 и Гипотезы 2.2 – важный параметр лгоритма. Для задачи с  $p=12$  сравнение эффективности алгоритма при различных значениях параметра  $L_0$  показаны на Рис. 2.5.

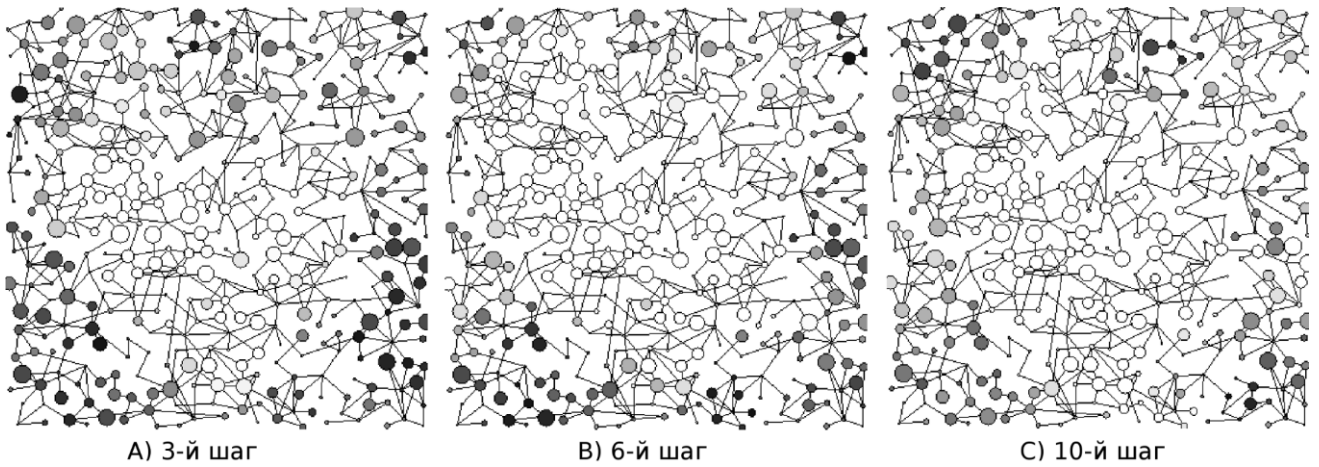


Рисунок 2.4 – Изменение значений вероятностей.  $L_0 = 100$ ,  $p = 3$ .

Результаты запуска алгоритма с использованием Алгоритма 2.6 в качестве генерирующей процедуры обозначены "L0=0". Наилучшие результаты для данной задачи достигаются при  $L_0 = 90$ . Оптимальное значение  $L_0$  зависит от  $p$ . При  $p = 32$  наискорейшая сходимость наблюдается при  $L_0 = 60$ . Тем не менее, алгоритм в широком диапазоне изменения параметра  $L_0 \in [10, 350]$  демонстрирует результаты, лучшие, чем "классический" метод изменяющихся вероятностей (случай  $L_0 = 0$ ).

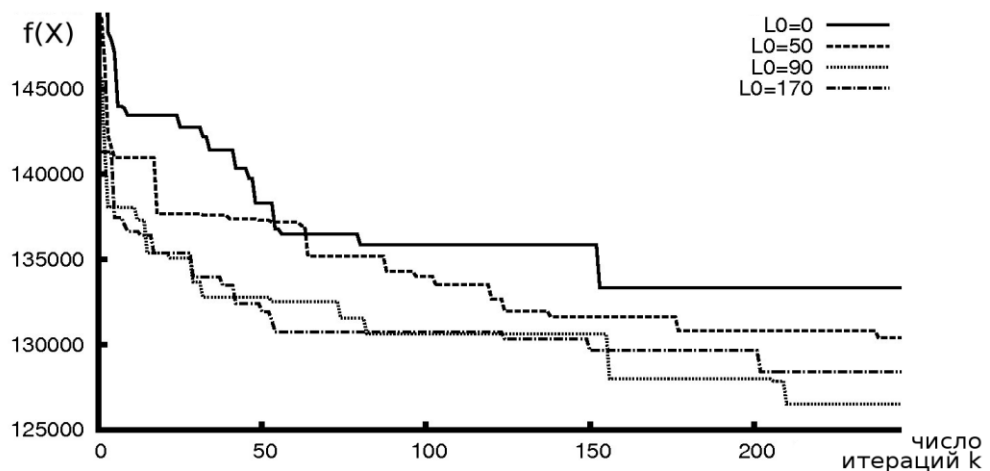


Рисунок 2.5 – Сравнение эффективности алгоритма на основе метода изменяющихся вероятностей при различных значениях  $L_0$ ,  $p = 12$

В дальнейших экспериментах использовано значение  $L_0 = L_{avg}/3$ , где  $L_{avg}$  – оценка математического ожидания расстояния от случайно выбранного узла до ближайшего узла, входящего в случайным образом сгенерированное решение. (случайным образом выбираются  $p$  узлов):

$$L_{avg} = \mu \left\{ \sum_{i=1}^n \min_{j \in \{1, p\}} L(i, j) / n \right\}. \quad (2.8)$$

В качестве оценки  $L_{avg}$  использовано расстояние от 10 случайным образом выбранных узлов до ближайшего узла в 10 случайным образом сгенерированных множествах из  $p$  узлов.

Вычисление значения  $L_{max}$ , которое используется в условиях Гипотез 2.1 и 2.2, занимает значительное количество времени. Вместо этого использовано  $l_{avg}/L_{avg} \rightarrow 0$  в качестве условия применимости алгоритма, см. (2.8).

## 2.5 Сравнительные результаты метода изменяющихся вероятностей

Точные методы решения  $p$ -медианной задачи на сети, предложенные Хаками [217], имеют экспоненциальную асимптотическую сложность, что затрудняет оценку точности получаемых результатов в случае сетей большой размерности. Для  $p=3$  сравнение результата на небольшой сети с 170 вершинами, полученного нашим алгоритмом и алгоритмом случайного поиска без адаптации (хаотический поиск, при котором на каждой итерации случайным образом выбираются  $p$  вершин) с известным точным результатом (горизонтальная асимптота), показано на рис. 2.6. Для построения графика, к шагу 6 Алгоритма 2.4 добавлены команды, которые через каждые 10 итераций фиксируют в специальном массиве достигнутое значение целевой функции и время с момента запуска алгоритма. Запуск алгоритмов осуществлен по 10 раз, на рис. 2.6 показан усредненный (для 10 запусков) результат хаотического поиска и худший (из 10 запусков) результат нашего алгоритма на базе метода изменяющихся вероятностей.

Для оценки эффективности метода произведено сравнение с результатами простейшего метода – локального поиска с мультистартом (Алгоритм 2.2) из случайным образом сгенерированных начальных решений, а также с генетическим алгоритмом (ГА) [204] с процедурой кроссинговера, заданной Алгоритмом 2.3 (специальная жадная

агломеративная эвристика) в качестве одного из наиболее эффективных существующих методов. В качестве тестовых наборов данных были использованы  $p$ -медианные задачи из библиотеки ORLIB [244]. Та же библиотека тестовых примеров была использована в работе [204]. Поскольку эта библиотека содержит задачи с числом узлов лишь до  $n = 900$ , Алгоритм 2.8 был использован для генерации больших задач.

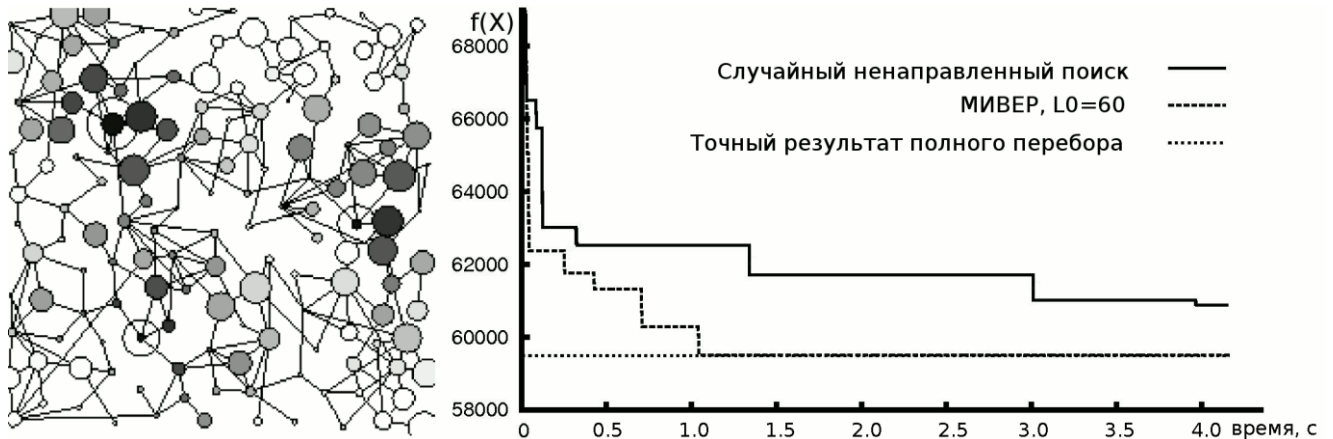


Рисунок 2.6 – Схема задачи с оптимальным решением и сравнение результата алгоритмов для  $p=3$  с результатом хаотического поиска и полного перебора

Результаты нашего алгоритма, основанного на методе изменяющихся вероятностей, использованного в качестве самостоятельного метода решения задачи, обнаруживают его медленную сходимость (рис. 2.7).

Задачи "rmed22" и "rmed39" включены в ORLIB, задача с  $n = 5000$  была сгенерирована Алгоритмом 2.8. На данном рисунке показаны усредненные результаты 10 запусков алгоритмов, а также наихудшие из результатов, достигнутые после 10 запусков к определенному моменту времени. Результаты комбинированного метода (МИВЕР+ГА, обозначено как "Probability changing+GA") разъяснены ниже. Для определения количества генерируемых на каждой итерации решений  $N_{POP}$  использована формула

$$N_{POP} = \left\lceil \frac{\sqrt{n} C \binom{n}{p}}{100 \lceil n/p \rceil} \lceil n/p \rceil \right\rceil. \quad (2.9)$$

ГА с жадной эвристикой использует формулу

$$N_{POP} = \left\lceil \frac{n C \binom{n}{p}}{100 \lceil n/p \rceil} \lceil n/p \rceil \right\rceil. \quad (2.10)$$

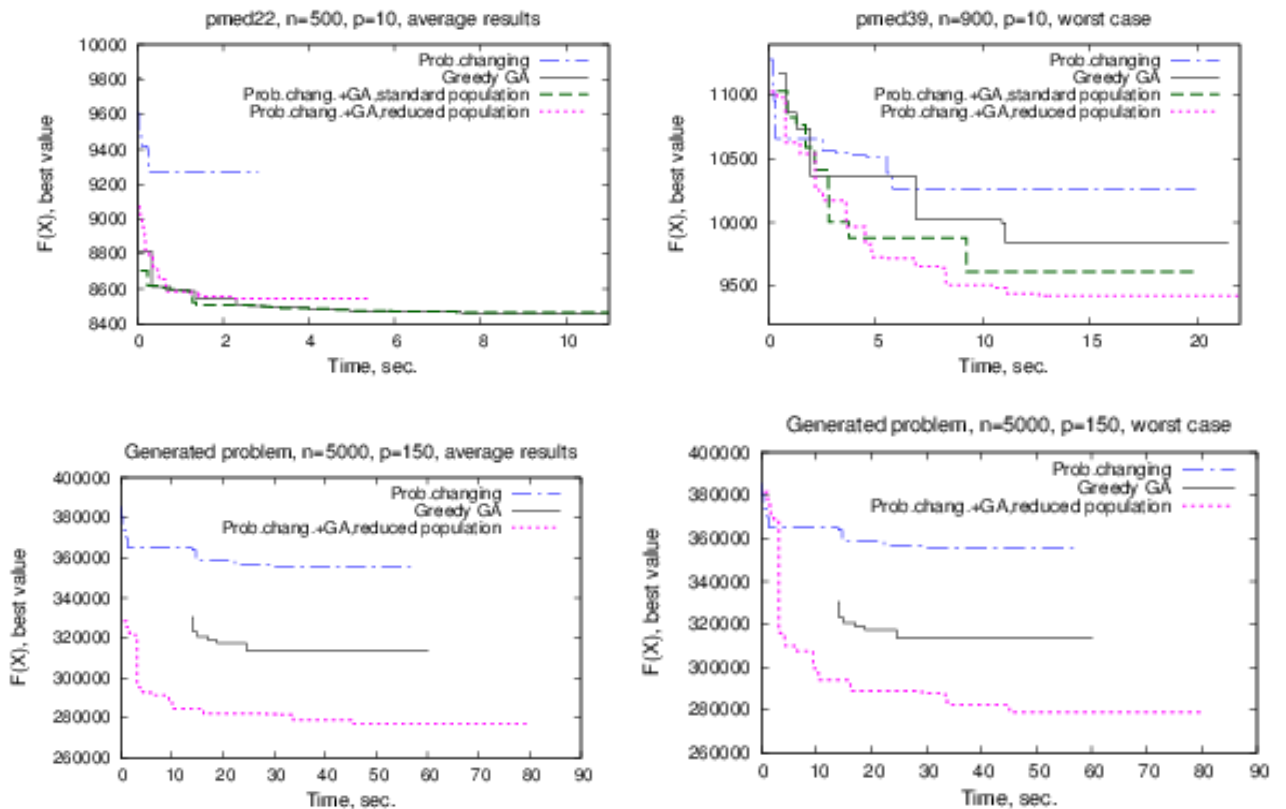


Рисунок 2.7 - Сравнение исходного ГА с жадной эвристикой (обозначен “Greedy GA”) и комбинированных методов: МИВЕР+ГА со стандартной популяцией (“Prob.chang.+GA, standard population”) и уменьшенной популяцией (“Prob.chang.+GA, reduced population”)

## 2.6 Результаты комбинированных методов

Генетический алгоритм с жадной эвристикой [204] использует детерминированный метод инициализации начальной популяции. В случае больших задач (например, pmed39, pmed32, а также сгенерированные Алгоритмом 2.8 задачи с  $n = 2000$ ,  $n = 5000$ ) эксперименты со случайным образом заполненной начальной популяцией в некоторых случаях показывают незначительное снижение точности результата.

Эксперименты с использованием последней популяции метода изменяющихся вероятностей в качестве начальной популяции ГА с жадной эвристикой показывают ускорение сходимости такого комбинированного алгоритма в сравнении с исходным ГА с жадной эвристикой. Были использованы два варианта определения размера популяции ГА: стандартная популяция (2.10) и уменьшенная популяция (2.9). Оба варианта

показывают существенное ускорение сходимости для большинства задач. Вариант с уменьшенной популяцией демонстрирует в некоторых случаях снижение точности, но ускоряет сходимость, позволяя за короткое время получить результат, достаточно близкий к оптимальному.

Вначале выполнялись 3-7 итераций Алгоритма 2.4 (на рисунках 2.7-2.9 показаны результаты при  $N_{STEPS} = 5$  на Шаге 6 Алгоритма 2.4) с адаптацией вероятностей (Алгоритм 2.6), затем популяция решений, полученная на последней итерации,  $\{X_{5,i} \mid i = \overline{1, N_{POP}}\}$  была использована в качестве начальной популяции ГА. «Хромосомы»  $M \in \{X_{5,i} \mid i = \overline{1, N_{POP}}\}$  были затем использованы в процедуре кроссинговера ГА (Алгоритм 2.3).

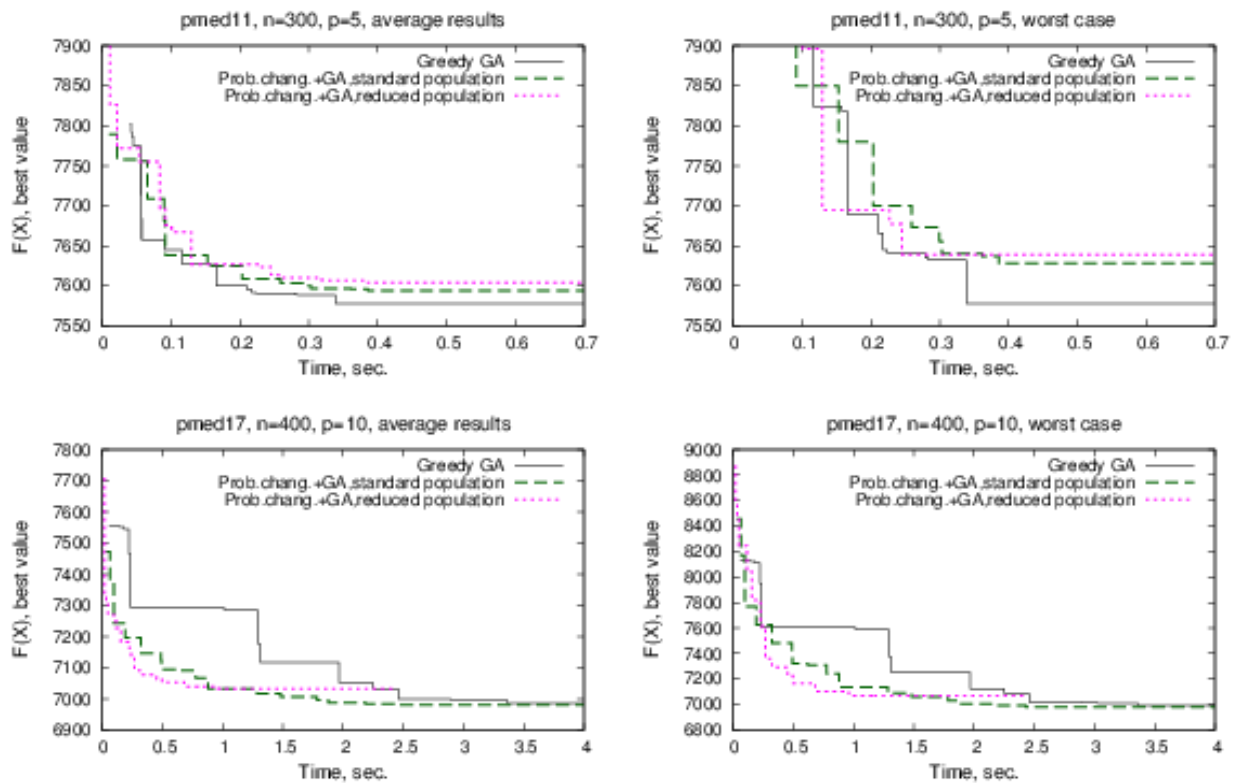


Рисунок 2.8 - Сравнение исходного ГА с жадной эвристикой (обозначен “Greedy GA”) и комбинированных методов: МИВЕР+ГА со стандартной популяцией (“Prob.chang.+GA, standard population”) и уменьшенной популяцией (“Prob.chang.+GA, reduced population”)

Результаты, показанные на рис. 2.7-2.9, были рассчитаны после 10 запусков каждого из представленных комбинированных алгоритмов. Зависимость усредненного и наихудшего из полученных значений целевой функции от затраченного времени показана для задач из библиотеки ORLIB со сравнительно небольшим значением отношения  $l_{avg}/L_{avg}$ , см. (2.4) и (2.8). Таковы, в том числе, задачи ”pmed11”, ”pmed12”,



”pmed14”, ”pmed16”, ”pmed18”, ”pmed19”, ”pmed21”, ”pmed23”, ”pmed35”, ”pmed31”. Была использована комбинация из трех условий останова: достижение наилучшего результата, анонсированного в ORLIB (если таковой известен), выполнение  $[\sqrt{n \cdot p}]$  итераций, не дающих улучшения достигнутого результата или превышение лимита времени.

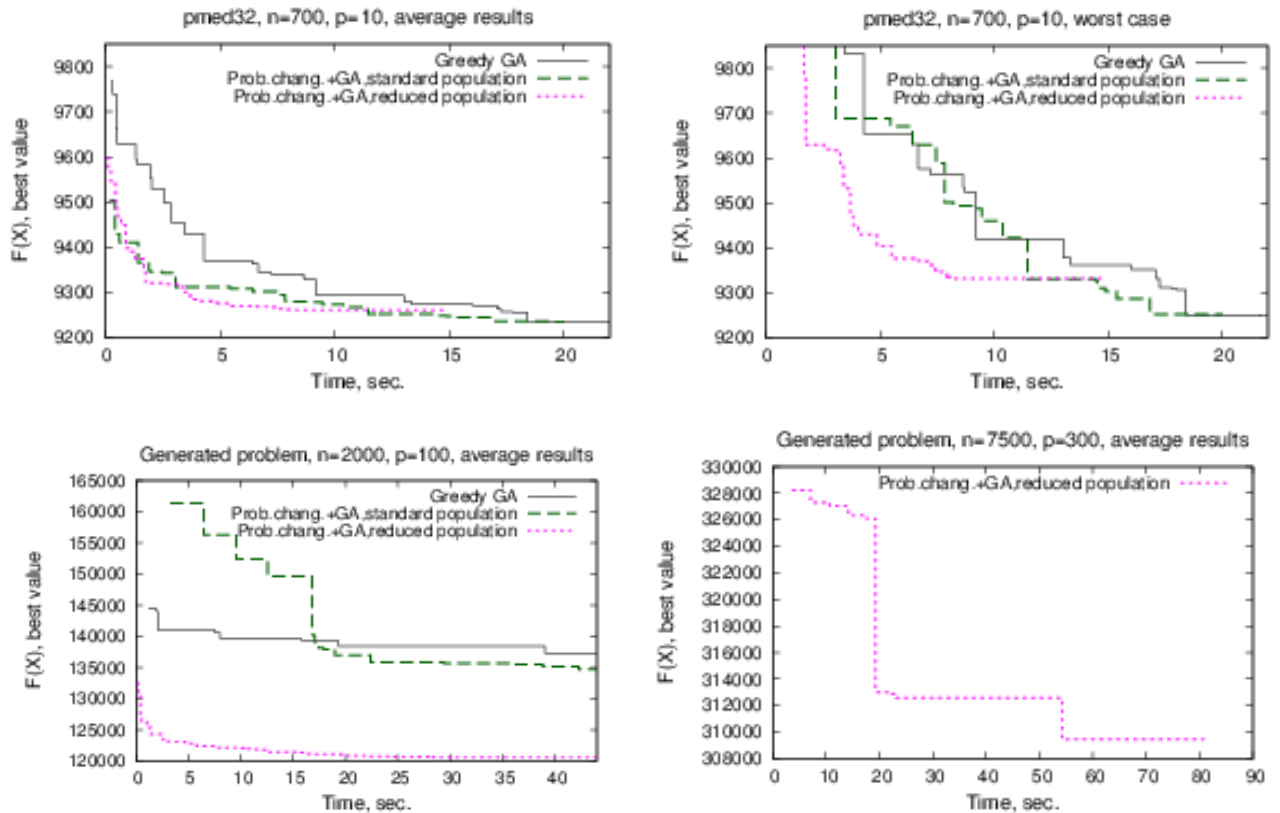


Рисунок 2.9 - Сравнение исходного ГА с жадной эвристикой (обозначен “Greedy GA”) и комбинированных методов: МИВЕР+ГА со стандартной популяцией (“Prob.chang.+GA, standard population”) и уменьшенной популяцией (“Prob.chang.+GA, reduced population”)

Для задачи с  $n = 7500$  показаны лишь результаты комбинированного алгоритма с уменьшенной популяцией (2.9), поскольку в случае стандартной популяции (2.10) не удалось избежать проблем, связанных с нехваткой оперативной памяти.

В случае использования метода изменяющихся вероятностей алгоритм быстрее начинает уменьшать разброс значений целевой функции в генерируемых решениях (рис. 2.10). В комбинированном алгоритме поиск продолжается с популяцией сравнительно «хороших» решений, причем скорость дальнейшего улучшения популяции также увеличивается.

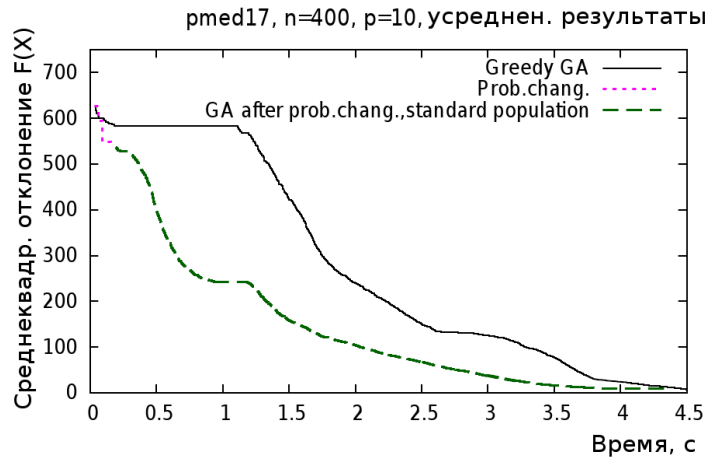


Рисунок 2.10 – Сравнение динамики изменения стандартного отклонения значений целевой функции в популяциях генетического алгоритма с жадной эвристикой (обозначен “Greedy GA”) и комбинированного алгоритма: метода изменяющихся вероятностей (“Prob.changing”) с продолжением расчета генетическим алгоритмом с жадной эвристикой (“GA after prob.chang.standard population”)

Кроме использования процедуры локального поиска (Алгоритм 2.2) из случайным образом сгенерированных начальных решений с мультистартом в качестве самостоятельного метода решения задачи, локальный поиск был применен как процедура в составе алгоритма на основе метода изменяющихся вероятностей. Шаг 2 Алгоритма 2.4 был модифицирован следующим образом:

2: В соответствии с распределением, заданным элементами вектора  $P$ , сгенерировать  $N_{POP}$  векторов  $X_{k,i}, i \in \overline{1, N_{POP}}$ ;

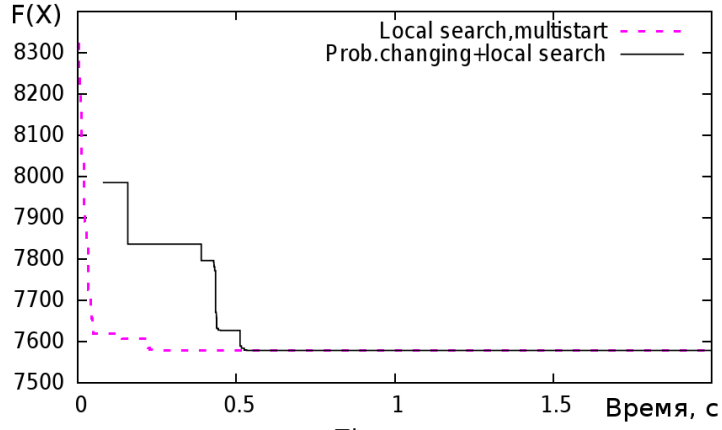
2.1: Если  $[k/5] = k/5$  и  $k > 0$ , то применить Алгоритм 2.2 к каждому из векторов  $X_{k,i}$ ;

Результаты показаны на рис. 2.11. Алгоритмы также запускались по 10 раз. Размер популяции алгоритма на базе метода изменяющихся вероятностей был рассчитан в соответствии с (2.9). Для задач с небольшим объемом входных данных (на малой сети) локальный поиск в обоих его вариантах оказался эффективнее, чем ГА.

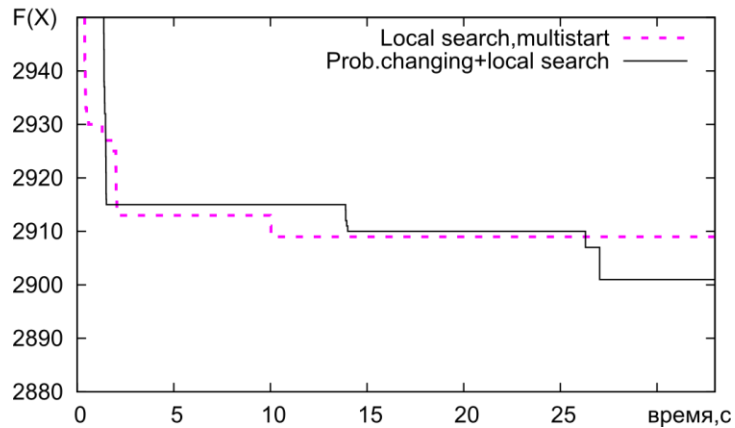
Для большинства задач, включенных в ORLIB, результаты комбинированного метода примерно соответствуют результатам локального поиска с мультистартом (часть “а” рис. 2.11), поскольку 2-100 запусков процедуры локального поиска оказываются достаточными для достижения точного решения задачи. Исключением являются задачи с высокой плотностью (плотностью называется отношение  $p/n$ ) на сравнительно большой

сети ("rmed19", "rmed24", "rmed25", "rmed29"). В этом случае (часть "б" рис. 2.11) комбинированный алгоритм позволяет достичь результата быстрее.

а) стандартная задача с низкой плотностью  
rmed11,  $n=300, p=5$ , усреднен. результаты



б) стандартная задача с высокой плотностью  
rmed24,  $n=500, p=100$ , усреднен. рез-ты



в) задача большой размерности

сгенерированная задача,  $n=7500, p=300$ , усреднен. результаты

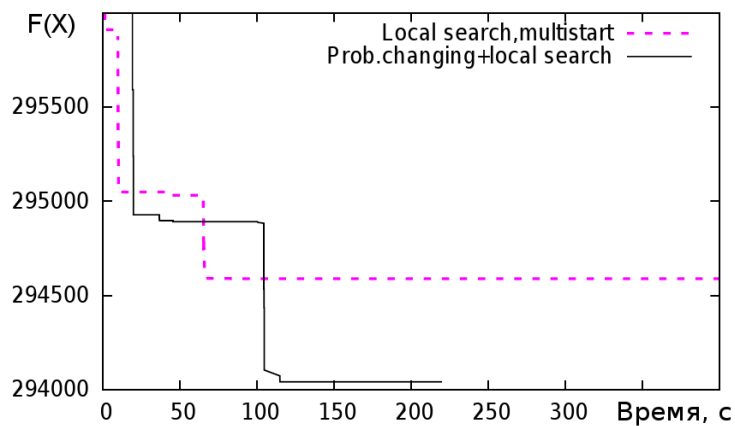


Рисунок 2.11 – Сравнение метода локального поиска (мультистарт) и метода изменяющихся вероятностей с встроенной процедурой локального поиска (обозначен «Prob.changing+local search»)

Для больших задач ( $n \geq 2000$ , часть "в" рис.2), сгенерированных Алгоритмом 2.8, комбинированный алгоритм дает стабильно лучшие результаты. В сравнении с исходным алгоритмом с жадной агломеративной эвристикой [204] новый метод, комбинирующий адаптированный МИВЕР с модифицированным генетическим алгоритмом с жадной эвристикой позволяя етв случае большой размерности задачи получать стабильно более точные результаты за одинаковое время, либо получать точные результаты быстрее в сравнении с известными эволюционными алгоритмами с жадной агломеративной эвристикой.

## 2.7 Параллельная версия алгоритма метода изменяющихся вероятностей

Оценки времени выполнения алгоритма (результаты профайлинга) и основной его процедуры — вычисления целевой функции — для различных задач стандартного тестового набора ORLIB [244] представлены в Таблице 2.1. Поскольку вопрос об условии останова и оптимальной численности популяции — дискуссионный [157], алгоритм запускался с различными формулами расчета данных параметров для различных задач. Для каждой задачи алгоритм был запущен 10 раз [157, 245]. В таблице представлены усредненные значения.

Таблица 2.1 – Результаты профайлинга последовательной версии алгоритма

Задача pmed	p	n	Общ время, мс	Время на выч-е $F(\chi)$ , всего, мс	Число шагов	Попу- ляция	Сред. время на 1 шаг	Сред. время на 1 выч-е $F(\chi)$
11	5	300	767	742	106	61	7,24	0,115
13	30	300	366	302	323	11	1,13	0,09
17	10	400	2538	2428	321	41	7,9	0,18
22	10	500	2333	2255	149	51	15,66	0,3
24	100	500	3831	2741	1111	12	3,44	0,21

На рисунке 2.12 представлена схема алгоритма 2.4 с процедурой генерации новых решений в виде Алгоритма 2.6 с адаптацией вероятностей согласно (2.12)-(2.14).

Здесь для целей тестирования также были использованы задачи ORLIB [244] и автоматически сгенерированные тестовые задачи размерности до  $n=7500$ ,  $p=200$ .

Использовался кластер с 4-ядерными узлами Xeon X5650 2.67 GHz, 12Gb RAM на каждом узле, HyperThreading отключен, компилятор GNU Fortran + OpenMPI, сеть Myrinet. Эксперименты были организованы следующим образом. На работу последовательной версии алгоритма отводилось время, равное 1/10 полного времени его работы (табл.2.1). Достигнутое значение целевой функции фиксировалось (контрольное значение). Затем выполнялось по 10 запусков последовательного алгоритма и параллельного алгоритма в различных конфигурациях кластера, при этом в качестве условия останова принималось достижение контрольного значения целевой функции.



Рисунок 2.12 – Блок-схема параллельной версии метода изменяющихся вероятностей для  $r$ -медианной задачи на сети

Параллельная работа алгоритма на 2.4 ядрах единственного узла для задач  $rmed22$ ,  $rmed24$ ,  $rmed32$ ,  $rmed39$  показывает параллельную эффективность 69-95%, которая растет с ростом размерности задач.

Эффективность работы в MPI-кластере с 2-8 узлами продемонстрирована в Таблице 2.2 ( $N_n$  – число узлов вычислительного кластера,  $N_p$  – число процессов на узел). Параллельная эффективность для 4 узлов с 1 запущенным процессом на узел, несмотря на необходимость пересылки значительных объемов информации, остается на том же уровне, что и при запуске на единственном узле, что можно объяснить отсутствием конкурентного использования каналов доступа к RAM и кэш-памяти.

Таблица 2.2 – Результаты параллельной версии алгоритма

Задача	p	n	Контр. знач. $F(\chi)$	Среднее время, 10 попыток, мс				Популяция
				$N_n=1$ $N_p=1$	$N_n=4$ $N_p=1$	$N_n=4$ $N_p=4$	$N_n=8$ $N_p=2$	
pmed11	5	300	7653	79,1	21,9	5,88	5,79	64
pmed13	30	300	4363	41	12,3	3,41	3,34	16
pmed17	10	400	7298	262	71,1	19,1	18,5	48
pmed22	10	500	8521	237	65,1	20,3	20	64
Pmed24	100	500	2914	392	127	35,5	34,9	16
Сгенерированная 1	100	2000	13971	2903	798	204	202	64
Сгенерированная 2	300	7500	32469	9306	2449	632	625	64

Таким образом, предложенный эвристический алгоритм на базе метода изменяющихся вероятностей для решения  $p$ -медианной задачи на сети большой размерности может быть эффективно распараллелен как для систем с общей памятью, так и с распределенной, без принципиального изменения общей схемы алгоритма.

В целом, алгоритм на основе метода изменяющихся вероятностей – эффективное средство инициализации генетического алгоритма при решении  $p$ -медианной задачи на сети при условии, что длины ребер сети много меньше математического ожидания длины пути из случайным образом выбранного узла до ближайшего узла, входящего в решение задачи. При этом очень медленная сходимость алгоритма не позволяет эффективно применять его в качестве самостоятельного метода решения задачи, но множество промежуточных решений, полученных после нескольких итераций данного алгоритма, является хорошей стартовой популяцией для генетического алгоритма с жадной эвристикой (по сравнению со случайным образом заполненной популяцией и с детерминированной процедурой инициализации [204]).

## 2.8 Жадная агломеративная эвристическая процедура для задач псевдобулевой оптимизации и некоторые свойства задач размещения

В работах [204, 157, 199], а также в предыдущих параграфах настоящей главы, предлагаются различные модификации генетического алгоритма с жадной эвристикой для  $p$ -медианной задачи на сети. Отметим, что некоторые модификации, применимые к задачам на сети, описаны в следующей главе.

Взаимосвязь задач размещения с задачей минимизации псевдобулевой функции известна [246] Задача о  $p$ -медиане на сети ( $p$ -медианная задача) может быть сформулирована в виде задачи минимизации псевдобулевой функции [247]:

$$F(x_1, \dots, x_N) = \sum_{i=1}^N w_i \min_{j \in \{1, \dots, N\}; x_j=1} L(i, j),$$

$$\sum_{i=1}^N x_i \leq p.$$

Здесь  $N$  – число булевых переменных, равное числу узлов сети ( $N=n$ ).

Задачу  $k$ -медоид можно сформулировать следующим образом. Даны  $N$  векторов данных  $A_1, \dots, A_N$  в  $d$ -мерном пространстве,  $A_i = (a_{i,1}, \dots, a_{i,d})$ ,  $A_i \in \mathbb{R}^d$ . Данная задача и методы ее решения более подробно рассмотрены в следующей главе. Здесь отметим лишь, что, несмотря на то, что формулировка данной задачи предполагает рассмотрение исходных данных как точек (векторов) в некотором непрерывном  $d$ -мерном пространстве, фактически задача является дискретной, имеет свойства, близкие к свойствам  $p$ -медианной задачи на сети и, так же, как и  $p$ -медианная задача на сети, может быть эффективно решена с применением алгоритмов, включающих в себя жадную агломеративную эвристику, в том числе – генетическим алгоритмом с жадной эвристикой в специальных модификациях. Целевую функцию также можно представить в виде функции с булевыми переменными:

$$F(x_1, \dots, x_N) = \sum_{i=1}^N w_i \min_{j \in \{1, \dots, N\}; x_j=1} \|A_i - A_j\|.$$

Здесь  $\|\bullet\|$  – расстояние в некоторой метрике.

Аналогичным образом в виде функции с булевыми переменными может быть представлена и непрерывная  $p$ -медианная задача [248, 249, 250]:

$$\arg \min_{X_1, \dots, X_N \in \mathbb{R}^d} F(X_1, \dots, X_N) = \arg \min_{X_1, \dots, X_N \in \mathbb{R}^d} \sum_{i=1}^N w_i \min_{Y \in \{X_1, \dots, X_N\}} \|A_i - Y\|.$$

В этом случае целевая функция должна определяться алгоритмически [250]:

Алгоритм 2.9. Определение значения целевой функции непрерывной  $p$ -медианной задачи с применением локального поиска.

Дано: булевы переменные  $x_1, \dots, x_N$ , векторы данных  $A_1, \dots, A_N$  в  $d$ -мерном пространстве.

1. Сформировать множество  $\chi = \{A_i / i = \overline{1, N}, x_i = 1\}$ .
2. Запустить ALA-процедуру (процедуру чередующегося размещения-распределения) или аналогичную процедуру локального поиска с начальным решением  $\chi$ . Получить результат – множество точек  $\chi^*$  в  $d$ -мерном пространстве.
3. Возвратить  $F(\chi^*)$ .

Жадный эвристический алгоритм (жадная эвристика) [251, 67] является одним из способов рекомбинации в генетическом алгоритме [252, 197]. Для решения всех рассмотренных выше задач предлагаются различные модификации специальной итеративной жадной эвристической процедуры, которую в общем виде можно описать следующим образом.

Алгоритм 2.10. Жадная эвристика для  $p$ -медианной задачи.

Дано: подмножество индексов вершин  $\chi, |\chi| > p$ .

1. Если  $|\chi| \leq p$ , то ОСТАНОВ, возврат  $\chi$ .
2. Попытаться улучшить решение  $\chi$  каким-либо методом локального поиска.
3. Выбрать  $j = \arg \min_{i \in \chi} F^*(\chi \setminus \{i\})$ . Здесь  $F^*(\cdot)$  – модифицированная целевая функция, которая в некоторых версиях алгоритма эквивалентна исходной целевой функции  $F(\cdot)$ , а в других версиях соответствует значению целевой функции  $F(\cdot)$  после улучшения ее множества-аргумента локальным поиском.
4. Удалить  $j$  из множества  $\chi$ .
5. Повторять с шага 1.

В качестве метода локального поиска применяются, в зависимости от особенностей конкретной задачи, либо поиск по смежным вершинам [157], либо поиск в SWAP-окрестности [248, 253, 254], ALA-процедура [255, 250, 256], РАМ-процедура [253, 248] или иные методы.



Целевая функция любой из рассматриваемых задач, будучи сформулированной с булевыми переменными, является монотонной [234, 237]. Данный факт легко доказать. Например, для задачи р-медоид сформулируем следующую лемму [257].

Лемма 2.3. Целевая функция задачи р-медоид – монотонно убывающая.

Доказательство. Векторы данных  $A_j, j = \overline{1, N}$ , такие, что  $x_j = 1$ , будем называть центрами кластеров. Для каждого вектора данных  $A_i, i = \overline{1, N}$  имеется ближайший центр  $A_{i'}, i' = \overline{1, N} : x_{i'} = 1, \exists m \neq i' : x_m = 1, \|A_i - A_m\| < \|A_i - A_{i'}\|$ .

Увеличим значение произвольно выбранной переменной. Для этого выберем произвольно индекс  $n$ , такой, что  $x_n = 0$ .

Обозначим

$$X = (x_1, \dots, x_N), \quad X' = (x_1, \dots, x_{n-1}, 1, x_{n+1}, \dots, x_N).$$

Тогда

$$\begin{aligned} F(X') &= \sum_{i=1}^N w_i \min \left\{ \min_{j \in \{\overline{1, N}\} \setminus \{n\} : x_j = 1} \|A_i - A_j\|; \|A_i - A_n\| \right\} \leq \\ &\leq \sum_{i=1}^N w_i \min_{j \in \{\overline{1, N}\} \setminus \{n\} : x_j = 1} \|A_i - A_j\| = F(X). \end{aligned}$$

Лемма доказана.

Таким образом, решение р-медоидной задачи, из которого удален один элемент (один центр), всегда хуже (точнее – не лучше) решения, в котором этот элемент присутствует.

Это же свойство аналогичным образом доказывается и для задач на сети.

Функцию  $F(X)$  можно превратить в монотонно возрастающую, взяв со знаком «-».

Очевидно, что левая часть ограничения  $\sum_{i=1}^N x_i \leq p$  является монотонно возрастающей функцией.

Отметим, что алгоритмы с жадной эвристикой являются наиболее эффективными при достаточно больших  $N$  и  $p \ll N$ . При малых значениях  $p$  (скажем,  $p < 10$ ) могут применяться точные алгоритмы вплоть до полного перебора. Так, при решении задач на сети [204, 157, 247] алгоритмы с жадной эвристикой эффективны при  $N > 400, p > 10$ . При этом данными алгоритмами решены достаточно большие задачи [249], со значениями до  $N = 560000, p = 100$ .

Выдвинем гипотезу о том, что, если и целевая функция  $F(X)$ , и единственная

функция-ограничение  $G(X) \leq B$ , где  $B = const$ , являются монотонно возрастающими, то алгоритм с жадной эвристикой может быть эффективно применен при выполнении дополнительного условия  $\sum_{i=1}^N x_i^* \ll N$ . Здесь  $x_i^*$  – оптимальное решение.

Отметим, что в работах [236, 247], а также в следующих параграфах настоящей диссертации рассмотрена задача об оптимальной загрузке производственных мощностей литейно-прокатного производства с целевой функцией  $F(X) = \sum_{i=1}^N x_i$ , очевидно являющейся монотонной, и нелинейными, но монотонными [236] ограничениями. Для решения задачи был применен эволюционный алгоритм с жадной эвристикой [247], основанный на алгоритме для р-медианной задачи, что подтверждает гипотезу применимости этого алгоритма для данного случая.

Предложим следующую общую схему алгоритма.

Алгоритм 2.11. Жадная эвристика для задач псевдодобулевой оптимизации.

Дано: монотонно возрастающая функция  $F(X)$ , ограничение  $G(X) \leq B$ , начальное решение  $X^* = (x_1, \dots, x_N) \in B^N$ ,  $G(X^*) > B$ .

1. Методом локального поиска попытаться «улучшить» решение  $X^*$  таким образом, чтобы значение  $F(X^*)$  возросло, а значение  $G(X)$  не возросло.
2. Если на шаге 1 улучшенное решение найдено, то повторить шаг 1.
3. Иначе выбрать  $k \in \overline{1, N}$ :  $x_k = 1$ , присвоить  $x_k = 0$ . Выбор осуществляется либо случайным образом, либо выбирается такое  $k$ , при котором обнуление значения соответствующей переменной дает наилучшее значение целевой функции.
4. Если  $G(X^*) \leq B$ , то ОСТАНОВ. Иначе перейти к шагу 1.

Рассмотрим, например, классическую задачу о рюкзаке [258]:

$$F(X) = \sum_{i=1}^N a_i x_i;$$

$$G(X) = \sum_{i=1}^N b_i x_i \leq B.$$

Подобная простая модель полезна при решении широкого класса практически важных задач: планирование ассортимента торгового предприятия [259], мультиверсионное программирование [260, 261, 262] и прочие задачи выбора оптимальной конфигурации технических систем [263].

В данном случае в качестве вспомогательного метода локального поиска применим следующий алгоритм.

Алгоритм 2.12. Локальный поиск в SWAP-окрестности для задачи псевдодулевой оптимизации «рюкзачного» типа.

Дано: значения булевых переменных  $x_1, \dots, x_N$ .

1. Выбрать случайным образом  $m \in \{\overline{1, N}\} : x_m = 1$ .
2. Выделить подмножество  $\xi \in \{i = \overline{1, N} / x_i = 0, a_i > a_m, b_i \leq b_m\}$ .
3. Выбрать случайным образом  $n \in \xi$ .
4. Присвоить  $x_m = 0; x_n = 1$ . Повторить с шага 1.

На Шаге 2 Алгоритма 2.11 выбор значения  $k$  будем осуществлять случайным образом.

Рассмотрим подробнее задачу распределения инвестиций в социальную сферу на примере Красноярского края. Имеются  $N$  объектов социальной сферы (в нашем случае  $N=452$ ). Модернизация  $j$ -го объекта требует денежных вложений в объеме  $b_j$  рублей (зависит от степени износа основных фондов и других факторов), при этом такая модернизация дает суммарный эффект  $a_j$ . В данном случае  $a_j$  – безразмерная величина, экономический смысл которой зависит от типа объекта (больницы, школы и т.д.), но в целом большие значения этой величины соответствуют большей степени удовлетворенности потребителей и муниципальных органов объемом и качеством соответствующих услуг, оказываемых  $j$ -м объектом. Сравнительные результаты решения данной задачи представлены в Таблице 2.3. При этом задача решалась генетическим алгоритмом [252] с жадной эвристикой (Алгоритм 2.11) и генетическим алгоритмом с «классической» рекомбинацией (однородный оператор скрещивания) [191]. В случае генетического алгоритма с жадной эвристикой получение «дочернего» решения из двух «родительских» (процедура кроссинговера - скрещивания) осуществляется следующим образом.

Алгоритм 2.13. Кроссинговер в ГА с жадной эвристикой для задач псевдодулевой оптимизации.

Дано: два набора «родительских» булевых переменных  $x_1, \dots, x_N$  и  $y_1, \dots, y_N$ .

1. Сформировать промежуточное решение  $Z=(z_1, \dots, z_N)$ :  $z_i = x_i \vee y_i$ .
2. Запустить Алгоритм 2.11 с начальным решением  $Z$ .

В случае генетического алгоритма с «классическим» кроссинговером в случае несовпадения значений «родительских» переменных  $x_i$  и  $y_i$  значение соответствующей «дочерней» переменной выбиралось случайным образом. При этом применялся механизм динамических штрафов [67, 264, 198, 265].

Отметим, что наша жадная эвристика для задач размещения (Алгоритм 2.13) – итеративная процедура, отличающаяся от двойственной жадной эвристики для задач о рюкзаке (см., например, работы Дюбина и Корбута [266, 267]) включением процедур локального поиска и процедурой оценки последствий исключения элемента из решения (т.е. обнуления соответствующей переменной).

Кроме того, мы сравнивали результаты двух алгоритмов для задач со случайным образом сгенерированными коэффициентами. Результаты для одной из таких задач ( $N=10000$ ) также приведены в Таблице 2.3.

Дополнительно была исследована задача следующего вида:

$$F(X) = \sum_{i=1}^{N/2} ((a_{(2i)}x_{(2i)} + 1) \cdot (a_{(2i+1)}x_{(2i+1)} + 1) - 1); \quad (2.11)$$

$$G(X) = \sum_{i=1}^N b_i x_i \leq B.$$

При неотрицательных коэффициентах целевая функция (2.11) данной задачи также остается монотонной. Следует также отметить, что подобные функции линеаризуются [268, 269] с вводом дополнительных ограничений.

Результаты, приведенные в Таблице 2.3, показывают эффективность нового алгоритма при небольших значениях правой части ограничения  $B$ , что соответствует дополнительному условию в выдвинутой гипотезе.

Решение задачи о распределении социальных инвестиций методом целочисленного линейного программирования с применением программного пакета GLPK показывает, что новый алгоритм дает для этой задачи во всех случаях оптимальное значение.

Жадная эвристика в составе генетического алгоритма показала свою эффективность в решении больших задач «рюкзачного» типа, при этом наилучшие результаты достигаются при меньших значениях правой части ограничения. Таким образом расширен класс задач, для которых может применяться жадная эвристика,

изначально разработанная для решения р-медианных задач размещения на сети.

Таблица 2.3 – Сравнительные результаты работы алгоритмов для различных задач

Задача, время	Алгоритм	Среднее значение	Лучшее значение	Среднеквадр. отклонение
Нелин. Ц.ф., N=100000 $B = 0,1\sum_{i=1}^N b_i x_i$ , 100 с.	Классич. ГА	9309557	9951901	254391,6
	ГА с ЖЭ*	11012998,9	11204749	106175,586
Нелин. Ц.ф., N=100000 $B = 0,5\sum_{i=1}^N b_i x_i$ , 100 с.	Классич. ГА*	113296909	115206924	717499,1
	ГА с ЖЭ	92107888	93431637	436778,62
Нелин. Ц.ф., N=100000 $B = 0,1\sum_{i=1}^N b_i x_i$ , 1000 с.	Классич. ГА	9309557	9951901	254391,6
	ГА с ЖЭ*	11012998,9	11204749	106175,586
Нелин. Ц.ф., N=100000 $B = 0,5\sum_{i=1}^N b_i x_i$ , 1000 с.	Классич. ГА*	113351101	114822282	827922,63
	ГА с ЖЭ	113318733	114822282	724457,01
Распределение социальн. инвестиций V=50000, 10 с.	Классич. ГА	6407	6441	6,76267
	ГА с ЖЭ*	6441	6441	0
Распределение социальн. инвестиций V=50000, 10 с.	Классический ГА	6389,7	6441	8,17658
	ГА с ЖЭ*	6441	6441	0
Распределение социальн. инвестиций V=100000, 1 с.	Классический ГА	6931,9	6952	32,12299
	ГА с ЖЭ*	6952	6952	0
Распределение социальн. инвестиций V=100000, 10 с.	Классический ГА	6935,933	6952	23,00964
	ГА с ЖЭ*	6952	6952	0

Примечания: ЖЭ – жадная эвристика. Знаком «\*» отмечен алгоритм, дающий лучший результат.

## 2.9 Постановка задачи составления расписаний загрузки производственных мощностей литейно-прокатного и химического производства

Базисом для обеспечения высокоэффективной работы любого промышленного предприятия является процедура календарного планирования производства, включая расчет производственного расписания (Production Scheduling, PS) [270].

В настоящем параграфе рассматривается проблема оперативно-календарного планирования [271] литейных и других аналогичных производств с ограниченной номенклатурой выпускаемой продукции и непрерывным характером производственного цикла. В таких производствах сложно определить ритм (такт) выпуска продукции, каждая производственная линия может выпускать одновременно только один вид продукции, а перенастройки с выпуска одного вида продукции на другой требуют значительных затрат. В этой связи одной из основных целей оперативного планирования становится составление производственного графика согласно заданным

объемам и срокам выпуска каждой номенклатурной позиции с минимальным числом перенастроек оборудования. Результаты опубликованы в [247, 272].

Алгоритмы планирования требовательны к вычислительным ресурсам. С другой стороны, имеются жесткие ограничения на время решения задач [273].

Рассмотрим следующую задачу [236]. Ограниченная номенклатура продукции производится на имеющихся производственных линиях, каждая из которых может производить некоторое подмножество общей номенклатуры продукции. Поток производственных заказов задает объемы и сроки выпуска каждого вида продукции.

В работе А. Н. Антамошкина и И. С. Масича [236, 274] рассматриваемая задача решается как задача псевдоболевой оптимизации (задача с булевыми переменными). Данная модель с небольшими изменениями используется и в настоящей работе.

Пусть имеется  $K$  производственных линий для выпуска  $L$  видов продукции. Производительность всех линий одинакова, для заданного  $l$ -го вида продукции линия может произвести  $V_l$  единиц продукции за смену при трех сменах в сутки. Требуется построить график с указанием вида продукции, производимой каждой производственной линией. Производственные линии могут производить лишь ограниченный ассортимент продукции. Вводится матрица  $Z$  булевых констант  $z_{k,l}$ ,  $k = \overline{1, K}$ ,  $l = \overline{1, L}$ , равных 1, если  $k$ -я линия может производить  $l$ -й вид продукции и 0 — в противном случае. Для каждого вида продукции установлен производственный план в объеме  $W_l$  единиц ( $l = \overline{1, L}$ ), который должен быть выполнен за  $T_l$  суток. Кроме того, установлена минимальная суммарная загруженность производственного комплекса в сутки в объеме  $W_{min}$  единиц.

Виды продукции объединены в  $M$  классов  $C_c$ ,  $c = \overline{1, M}$ . При смене класса продукции, выпускаемой производственной линией, требуются некоторые технологические операции продолжительностью в одну смену, в ходе которых выпуск продукции невозможен. Пусть возможность безостановочной смены продукции с вида  $l$  на вид  $r$  описывается симметричной матрицей булевых констант  $C_{l,r}$  размерности  $L \times L$ : значение  $C_{l,r} = 1$  означает необходимость останова производственной линии при смене продукции с  $l$ -го вида на  $r$ -й вид, нулевое значение — возможность безостановочного переключения производства (естественно,  $C_{l,l} = 0 \forall l \in \{\overline{1, L}\}$ ).

Вводятся булевы переменные  $y_{i,k,l}$ , которые принимают значение 1, если график предусматривает выпуск  $l$ -го вида продукции  $k$ -й производственной линией на  $i$ -е сутки.

График на  $I$  суток требуется составить так, чтобы при условии выполнения плана выпуска по видам продукции и срокам, с учетом требования минимальной загруженности, требовалось минимальное число изменений видов продукции.

Таким образом, в булевых переменных задача формулируется следующим образом:

$$\min \sum_{i=1}^I \sum_{k=1}^K \sum_{l=1}^L y_{i,k,l} (1 - y_{(i-1),k,l}); \quad (2.12)$$

$$V_l \sum_{i=1}^{T_l} \sum_{k=1}^K (3 \cdot y_{i,k,l} - y_{i,k,l} \cdot (1 - y_{(i-1),k,l}) \cdot \sum_{r=1}^L y_{(i-1),k,r} C_{r,l}) \geq W_l \forall l = \overline{1, L}; \quad (2.13)$$

$$\sum_{k=1}^K \sum_{l=1}^L V_l (3 \cdot y_{i,k,l} - y_{i,k,l} \cdot (1 - y_{(i-1),k,l}) \cdot \sum_{r=1}^L y_{(i-1),k,r} C_{r,l}) \geq W_{min} \forall i = \overline{1, I}; \quad (2.14)$$

$$\sum_{l=1}^L y_{i,k,l} \leq 1 \forall i = \overline{1, I}, k = \overline{1, K}; \quad (2.15)$$

$$y_{i,k,l} \leq z_{k,l} \forall i = \overline{1, I}, k = \overline{1, K}, l = \overline{1, L}; \quad (2.16)$$

$$y_{i,k,l} \in \{0, 1\} \forall i = \overline{1, I}, k = \overline{1, K}, l = \overline{1, L}. \quad (2.17)$$

Отметим, что приведенная в работе [236] модель не учитывает возможность безостановочной смены видов продукции. Для учета возможности замены выпускаемой продукции без остановки производства в выражения (2.13) и (2.14) добавлен дополнительный множитель  $\sum_{r=1}^L y_{(i-1),k,r} C_{r,l}$ .

В [236] рассмотрен лишь частный случай  $C_{r,l} = 1 \forall r, l \in \{\overline{1, L}\}, r \neq l$ . Описанная в [236] модель предполагает, что в начальный момент производство на всех линиях остановлено ( $y_{0,k,l} = 0 \forall k = \overline{1, K}, l = \overline{1, L}$ ). В реальных задачах в условиях регулярного планирования непрерывного производства такая ситуация практически невозможна, любая производственная линия в любой момент времени (за исключением случаев аварий, ремонтов и прочих ситуаций, когда данная линия исключается из производственного графика) настроена на выпуск какой-либо продукции.

В дополнение к переменным  $y_{i,k,l}$  авторы работы [236] вводят дополнительные булевы переменные  $x_{i,k,l}$ :

$$x_{i,k,l} = y_{i,k,l} (1 - y_{(i-1),k,l}) \forall i = \overline{1, I}, k = \overline{1, K}, l = \overline{1, L}.$$

Значения переменных  $y_{i,k,l}$  могут быть определены исходя из значений  $x_{i,k,l}$  с использованием специального алгоритма, реализующего формулу:

$$y_{i,k,l} = \begin{cases} x_{i,k,l}, & \sum_{r=1}^L x_{i,k,r} = 0, \\ y_{(i-1),k,l}, & \sum_{r=1}^L x_{i,k,r} > 0, \end{cases} \quad \forall i = \overline{1, I}, k = \overline{1, K}, l = \overline{1, L}.$$

Напомним математическую постановку  $p$ - медианной задачи [217]. На некоторой сети (связном графе)  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , где  $\mathcal{V}$ -множество узлов (вершин), а  $\mathcal{E}$  – множество попарно соединяющих их ребер  $E_{i,j}, i, j \in \mathcal{V}$ , каждому из которых поставлено в соответствие число – его длина  $\mathcal{L}_{i,j}$ , определена метрика расстояния  $D(i, j)$  между любой парой узлов  $i$  и  $j$  как минимальная длина пути между этими узлами (длина пути – сумма длин входящих в него ребер). Цель состоит в выборе множества узлов сети (вершин графа)  $\mathcal{S}$  заданной мощности  $p$ :

$$\arg \min_{\mathcal{S} \subset \mathcal{V}, |\mathcal{S}|=p} f_{\mathcal{G}}(\mathcal{S}) = \arg \min_{\mathcal{S} \subset \mathcal{V}, |\mathcal{S}|=p} \sum_{i \in \mathcal{V}} \min_{j \in \mathcal{S}} D(i, j). \quad (2.18)$$

Представим производственный график в виде трехмерной решетки в дискретной системе координат. По оси  $i$  расположим сутки планирования, по оси  $k$  – производственные линии, по оси  $l$  – виды продукции. Каждый узел такой сети-решетки будем описывать тройкой его координат  $(i, k, l)$ . Матрице наших булевых переменных  $[x_{i,k,l}, i = \overline{1, I}, k = \overline{1, K}, l = \overline{1, L}]$  поставим в соответствие множество  $\chi$  узлов сети-решетки, описывающей график:

$$x_{i,k,l} = \begin{cases} 1, & (i, k, l) \in \chi, \\ 0, & (i, k, l) \notin \chi. \end{cases}$$

В нашем случае задача состоит в выборе минимального по мощности множества  $\chi$  узлов сети (вершин графа), удовлетворяющего ограничениям (2.15) – (2.17). Выбранному узлу соответствует значение булевой переменной  $x_{i,k,l} = 1$ . Иными словами, наша задача состоит в выборе минимального по мощности множества точек переключения выпуска продукции на сетке графика производственного планирования.

Решение задач составления расписаний методами целочисленного линейного программирования исследовано во многих работах, например, [275, 276, 277]. Подобный подход можно считать классическим. Наша задача так же может быть линеаризована [278, 279, 269, 268].

Введем  $I \cdot K \cdot L$  новых переменных:

$$y_{i,k,l} = y_{(i-1),k,l} \cdot x_{i,k,l}, \quad i = \overline{1, I}, k = \overline{1, K}, l = \overline{1, L}. \quad (2.19)$$



Тогда, введя  $3 \cdot I \cdot K \cdot L$  дополнительных ограничений [268, 269]:

$$\gamma_{i,k,l} \leq y_{i,k,l}, \quad (2.20)$$

$$\gamma_{i,k,l} \leq y_{(i-1),k,l}, \quad (2.21)$$

$$\gamma_{i,k,l} \geq y_{(i-1),k,l} + y_{i,k,l} - 1, \quad (2.22)$$

и ограничения

$$\gamma_{i,k,l} \in \{0,1\}, \quad i = \overline{1, I}, k = \overline{1, K}, l = \overline{1, L}, \quad (2.23)$$

имеем эквивалентную задачу:

$$\min \sum_{i=1}^I \sum_{k=1}^K \sum_{l=1}^L (y_{i,k,l} - \gamma_{i,k,l}); \quad (2.24)$$

с ограничениями (2.15)-(2.17), (2.20)-(2.23), а также

$$V_l \sum_{i=1}^{T_l} \sum_{k=1}^K (3 \cdot y_{i,k,l} - (y_{i,k,l} - \gamma_{(i-1),k,l}) \cdot \sum_{r=1}^L y_{(i-1),k,r} C_{r,l}) \geq W_l \forall l = \overline{1, L}; \quad (2.25)$$

$$\sum_{k=1}^K \sum_{l=1}^L V_l (3 \cdot y_{i,k,l} - (y_{i,k,l} \cdot -\gamma_{i,k,l}) \cdot \sum_{r=1}^L y_{(i-1),k,r} C_{r,l}) \geq W_{min} \forall i = \overline{1, I}; \quad (2.26)$$

Аналогичное преобразование произведений  $(y_{i,k,l} \cdot -\gamma_{i,k,l}) \cdot y_{(i-1),k,r} \forall i = \overline{1, I}, k = \overline{1, K}, l = \overline{1, L}, r = \overline{1, L}$  к линейному виду привело бы к необходимости ввода  $2 \cdot I \cdot K \cdot L \cdot (L-1)$  дополнительных переменных (фактически, увеличение числа переменных в  $L-1$  раз). Вместо этого мы вначале введем  $I \cdot K$  дополнительных переменных, равных 1, если  $k$ -я линия на  $i$ -е сутки не производит продукцию, и  $I \cdot K \cdot M$  дополнительных переменных, равных 1, если на  $i$ -е сутки на  $k$ -й линии производится продукция  $c$ -го класса:

$$y_{i,k,0} = 1 - \sum_{r=1}^L y_{i,k,r} \quad \forall i = \overline{1, I}, k = \overline{1, K}. \quad (2.27)$$

Обозначим

$$\alpha_{i,k,c} = \sum_{r \in C_c} y_{i,k,r} \quad \forall i = \overline{0, I}, k = \overline{1, K}, c = \overline{1, M}.$$

Тогда условие (2.15) принимает вид

$$\sum_{l=0}^L y_{i,k,l} = 1 \quad \forall i = \overline{1, I}, k = \overline{1, K}. \quad (2.28)$$

Тогда выпуск  $l$ -го вида продукции  $k$ -й линией на  $i$ -е сутки без смены класса продукции может быть выражен следующим образом:

$$\begin{aligned} & 1 - (y_{i,k,l} \cdot -\gamma_{i,k,l}) \cdot \sum_{r=1}^L y_{(i-1),k,r} C_{r,l} = y_{i,k,l} (\alpha_{(i-1),k,C(l)} + y_{(i-1),k,0}) = \\ & = \sum_{r \in C_C(l)} y_{i,k,l} \cdot y_{(i-1),k,r} + y_{i,k,l} \cdot y_{(i-1),k,0}. \end{aligned}$$

Здесь  $C(l)$  – номер класса продукции, которому принадлежит  $i$ -й вид продукции.

Наконец, введем  $I \cdot K \cdot (L + \sum_{c=1}^M (|C_c|^2 - |C_c|))$  дополнительных переменных:

$$\omega_{i,k,l,r} = y_{i,k,l} \cdot y_{(i-1),k,r} \quad \forall i = \overline{1, I}, k = \overline{1, K}, l = \overline{1, L}, r \in C_{C(l)} \setminus \{l\} \cup \{0\}. \quad (2.29)$$

При этом добавим  $3 \cdot I \cdot K \cdot (L + \sum_{c=1}^M (|C_c|^2 - |C_c|))$  ограничений:

$$\omega_{i,k,l,r} \leq y_{i,k,l}, \quad (2.30)$$

$$\omega_{i,k,l,r} \leq y_{(i-1),k,r}, \quad (2.31)$$

$$\omega_{i,k,l,r} \geq y_{(i-1),k,r} + y_{i,k,l} - 1, \quad (2.32)$$

а также ограничения

$$\omega_{i,k,l,r} \in \{0, 1\} \quad \forall i = \overline{1, I}, k = \overline{1, K}, l = \overline{1, L}, r \in \{C_{C(l)} \setminus \{l\}\} \cup \{0\}. \quad (2.33)$$

Ограничения (2.25) и (2.26) преобразуем к виду:

$$V_l \sum_{i=1}^{T_l} \sum_{k=1}^K \left( 2 \cdot y_{i,k,l} + \sum_{r \in C_{C(l)} \setminus \{l\} \cup \{0\}} \omega_{i,k,l,r} \right) \geq W_l \quad \forall l = \overline{1, L}, c = C(l); \quad (2.34)$$

$$\sum_{k=1}^K \sum_{l=1}^L V_l \left( 2 \cdot y_{i,k,l} - \sum_{r \in C_{C(l)} \setminus \{l\} \cup \{0\}} \omega_{i,k,l,r} \right) \geq W_{min} \quad \forall i = \overline{1, I}, c = C(l). \quad (2.35)$$

Таким образом, имеем задачу целочисленного линейного программирования (2.19)-(2.24), (2.27)-(2.35), (2.16)-(2.17). Число переменных такой задачи равно  $I \cdot K + 2 \cdot I \cdot K \cdot L + I \cdot K \cdot (L + \sum_{c=1}^M (|C_c|^2 - |C_c|))$ . Так, для реальной задачи при  $L=26$ ,  $M=13$ ,  $I=31$ ,  $K=46$ , имеем модель целочисленного линейного программирования с 481988 булевыми переменными и более миллиона ограничений.

При построении алгоритма, учитывая ограничения (2.15), в целях экономии оперативной памяти, сокращения времени доступа к ней и для упрощения некоторых расчетов целесообразно перейти к целочисленным переменным.

Введем переменные  $y'_{i,k}$ , принимающие значения от 0 до  $L$ . Значение  $y'_{i,k} = l$  будет означать выпуск  $l$ -го вида продукции на  $k$ -й линии в  $i$ -е сутки (значение  $y'_{i,k} = 0$  – отсутствие выпуска). Пусть  $y'_{0,k}, k = \overline{1, K}$  – целочисленные константы, имеющие значения от 0 до  $L$ , показывающие, на какой вид продукции настроена каждая из  $K$  линий в начальный момент времени. Аналогично введем дополнительные переменные  $x'_{i,k}$ :

$$x'_{i,k} = \begin{cases} y'_{i,k}, & y'_{i,k} \neq y'_{(i-1),k}, \\ 0, & y'_{i,k} = y'_{(i-1),k}, \end{cases} \quad \forall i = \overline{1, I}, k = \overline{1, K}. \quad (2.36)$$

Значения переменных  $y'_{i,k}$  могут быть получены из  $x'_{i,k}$ :

$$y'_{i,k} = \begin{cases} y'_{(i-1),k}, & x'_{i,k} = 0, \\ x'_{i,k}, & x'_{i,k} \neq 0 \end{cases} \quad \forall i = \overline{1, I}, k = \overline{1, K}. \quad (2.37)$$

Значения исходных булевых переменных могут быть получены следующим образом:

$$x_{i,k,l} = \begin{cases} 1, & x'_{i,k} = l, \\ 0, & x'_{i,k} \neq l, \end{cases} \quad \forall i = \overline{1, I}, k = \overline{1, K}, l = \overline{1, L}, \quad (2.38)$$

$$y_{i,k,l} = \begin{cases} 1, & y'_{i,k} = l, \\ 0, & y'_{i,k} \neq l, \end{cases} \quad \forall i = \overline{1, I}, k = \overline{1, K}, l = \overline{1, L}. \quad (2.39)$$

В такой нотации ограничение (2.15) становится лишним, а ограничения (2.13) и (2.14) можно записать в более удобной форме:

$$V_l \sum_{i=1}^{T_l} \sum_{k=1}^K y_{i,k,l} (3 - C'_{y'_{(i-1),k}, y'_{i,k}}) \geq W_l \forall l = \overline{1, L}, \quad (2.40)$$

$$\sum_{k=1}^K \sum_{l=1}^L V'_{y'_l} (3 - C'_{y'_{(i-1),k}, y'_{i,k}}) \geq W_{min} \forall i = \overline{1, I}. \quad (2.41)$$

Здесь  $C'_{l,r}$  – матрица булевых значений  $C_{l,r}$ , дополненная строкой с нулевым индексом и столбцом с нулевым индексом ( $l, r \in \{\overline{0, L}\}$ ):

$$C'_{l,r} = \begin{cases} C_{l,r}, & l > 0, r > 0, \\ 0, & r = 0, \\ 1, & l = 0, r > 0, \end{cases}$$

$V'$  – вектор норм производства по видам продукции за смену  $V$ , дополненный нулевым элементом:

$$V'_l = \begin{cases} V_l, & l > 0 \\ 0, & l = 0. \end{cases}$$

Предлагаемый в настоящей работе генетический алгоритм не сохраняет значения булевых переменных  $x_{i,k,l}$  и  $y_{i,k,l}$ , они определяются согласно (2.38) и (2.39) “на лету”.

В соответствии с нотацией, принятой для  $p$ -медианных задач, промежуточные решения описываются множествами  $\chi$  выбранных узлов (медиан) сети. Эти множества также удобно представлять матрицами целочисленных переменных:

$$x'_{i,k} = \begin{cases} l, & (i, k, l) \in \chi, \\ 0, & (i, k, l) \notin \chi. \end{cases} \quad (2.42)$$

Операция включения узла  $(i, k, l)$  в множество  $\chi$ , таким образом, сводится к присвоению элементу матрицы  $x'_{i,k}$  значения  $l$ , исключения – к присвоению нулевого значения.

## 2.10 Алгоритм с жадной эвристической процедурой для задачи о составлении расписания

Напомним основные идеи генетического алгоритма с жадной эвристикой (ГА с процедурой кроссинговера в виде Алгоритма 2.3) для  $p$ -медианной задачи на сети [157]. Имеется некоторый массив (“популяция”) решений задачи, каждое из которых представляет собой множество из  $p$  выбранных узлов сети. Случайным образом выбираются два решения (родительские “особи”). В качестве промежуточного решения принимается множество узлов, являющееся объединением выбранных “родительских” множеств. Далее из этого промежуточного решения по одному исключаются те узлы, удаление которых дает наименьший прирост целевой функции (2.18), до тех пор, пока не останется  $p$  узлов.

Такой генетический алгоритм в целом выглядит следующим образом:

Алгоритм 2.14. Генетический алгоритм с жадной эвристикой для  $p$ -медианной задачи (ГА с процедурой кроссинговера, реализованной Алгоритмом 2.3 и встроенной процедурой локального поиска, реализованной Алгоритмом 2.2).

Дано: количество искомых узлов  $p$ , множество всех узлов сети  $\mathcal{V}$ , целевая функция (метрика расстояния)  $f_G(\cdot)$ .

1. Сгенерировать начальный массив множеств узлов  $A = \{\chi_j\} = \{\mathcal{V}_{m_1}, \dots, \mathcal{V}_{m_p}\}$ ,  $j = \overline{1, N_{POP}}$  по  $p$  вершин в каждом множестве. Здесь  $N_{POP}$  – число “особей” (размер популяции) генетического алгоритма.

2. Выбрать случайным образом два индекса элементов массива  $A$  (двух “родительских” особей)  $j_1, j_2 = \overline{1, N_{POP}}$ ,  $j_1 \neq j_2$ . Выбрать случайным образом  $j_3 \in w$ . Здесь  $w$  – некоторое множество индексов особей (элементов массива  $A$ ), оцениваемых как “плохие”. В работе [157] всегда выбирается особь с худшим (максимальным) значением целевой функции:  $\chi_w = \{\operatorname{argmin}_{\mathcal{S} \in A} f_G(\mathcal{S})\}$ .

3. Присвоить  $\chi_{j_3} = \chi_{j_1} \cup \chi_{j_2}$ ;

4. Пока  $|\chi_{j_3}| > p$  выполнять:

4.1. Присвоить  $f_{best} = +\infty$ ,  $FOUND = 0$ .

4.2. Для каждого узла  $\mathcal{V} \in \chi_{j_3}$  выполнять:

4.2.1.  $\xi = \chi_{j_3} \setminus \{\mathcal{V}\}$ .

4.2.2. Если  $f_G(\xi) < f_{best}$ , то присвоить  $f_{best} = f_G(\xi)$ ,  $\chi_{best} = \xi$ ,  $FOUND = 1$ .

4.2.3. Следующая итерация цикла 4.2.

4.3. Присвоить  $\chi_{j_3} = \chi_{best}$ . Следующая итерация цикла 4.

5. Проверить условие останова и, если оно не выполняется, перейти к Шагу 2.

При определенной схожести с  $p$ - медианной задачей на сети, наша задача имеет и ряд существенных отличий, в том числе:

Свойство 1. В  $p$ - медианной задаче целевая функция – монотонно убывающая, в нашей задаче функция (2.12) – монотонно возрастающая (как относительно переменных  $x_{i,k,l}$ , так и относительно переменных  $y_{i,k,l}$ ).

Свойство 2. При решении  $p$ -медианной задачи [157] мы имеем дело с безусловной оптимизацией. В настоящей же работе рассматривается задача условной оптимизации. Левые части ограничений (2.13) и (2.14) – монотонно возрастающие функции переменных  $y_{i,k,l}$ , относительно же переменных  $x_{i,k,l}$  они в общем случае монотонными не являются: возможны такие изменения значений переменных  $x_{i,k,l}$  с 1 на 0 (такое изменение соответствует операции исключения узлов из сети), при которых значения левых частей некоторых неравенств (2.13) и (2.14) уменьшаются, других – увеличиваются.

Свойство 3. В  $p$ -медианной задаче число вершин  $p$  известно, в нашей задаче целью является минимизация числа переключений видов продукции.

Благодаря Свойству 1, при исключении любой вершины из множества вершин (Шаг 4.2.1 Алгоритма 2.14) значение целевой функции гарантированно уменьшается на единицу, проверка на Шаге 4.2.2 становится бессмысленной. В то же время, проверка соответствия полученного промежуточного решения ограничениям является актуальной благодаря Свойству 2. Благодаря Свойству 3 условие цикла 4 Алгоритма 2.14 также теряет значение. Исключение вершин из промежуточного решения следует повторять до тех пор, пока эта операция дает некоторые улучшения с точки зрения соответствия промежуточного решения ограничениям.

Ограничения (2.15) и (2.16) могут быть учтены на этапе генерации начальных и промежуточных решений. Ограничения (2.13) и (2.14) должны быть преобразованы в штрафную функцию.

В предлагаемом в настоящей работе алгоритме предлагается замена всех ограничений (2.13), которым соответствуют ограничения (2.40) в нотации с целочисленными переменными, на ограничение следующего вида:

$$f_2(\chi) = -\sum_{l=1}^L \max\{0, W_l - V_l \sum_{i=1}^{T_l} \sum_{k=1}^K y_{i,k,l} (3 - C'_{y'_{(i-1),k} y'_{i,k}})\} = 0, \quad (2.43)$$

А ограничение (3), которому соответствует ограничение (12) – на ограничение

$$f_3(\chi) = -\sum_{i=1}^I \max\{0, W_{min} - \sum_{k=1}^K \sum_{l=1}^L V'_{y'_l} (3 - C'_{y'_{(i-1),k} y'_{i,k}})\} = 0. \quad (2.44)$$

Функции  $f_2(\cdot)$  и  $f_3(\cdot)$  здесь определены как функции множества  $\chi$ . Значения целочисленных переменных  $y'_{i,l}$  в булевых переменных  $y_{i,k,l}$  определяются по формулам (2.42), (2.36)-(2.37), (2.39).

Определим также целевую функцию

$$f(\chi) = |\chi| = \sum_{i=1}^I \sum_{k=1}^K \sum_{l=1}^L x_{i,k,l},$$

значения булевых переменных  $x_{i,k,l}$  определяются по формулам (2.42), (2.37), (2.38).

Экономический смысл функции  $f_2(\chi)$  – общее количество продукции, выпускаемой с отставанием от плана, функции  $f_3(\chi)$  – суммарное невыполнение суточного минимума выпускаемой продукции.

Функции  $f_2(\chi)$  и  $f_3(\chi)$  имеют схожий экономический смысл, одинаковую размерность (число единиц продукции), поэтому возможно использование единой штрафной функции – их аддитивной свертки и соответствующего единого ограничения:

$$f_4(\chi) = f_2(\chi) + f_3\chi = 0. \quad (2.45)$$

Задачу поиска решений, удовлетворяющих условию (2.45) и, соответственно, условиям (2.13) и (2.14) исходной задачи, можно рассматривать как задачу минимизации функции  $f_4(\chi)$ .

Характер генетического алгоритма не требует наличия единственного критерия оценки решений. Возможна оценка решений одновременно по двум критериям: значение целевой функции  $f(\chi)$  и значение штрафной функции  $f_4(\chi)$ . Забегая вперед, Отметим, что использование свертки (2.45) не дает ощутимых преимуществ перед использованием трех отдельных критериев  $f(\cdot)$ ,  $f_2(\cdot)$  и  $f_3(\cdot)$ , недостатков этого подхода также не выявлено. В то же время, подход к оценке и сравнению решений в популяции генетического алгоритма (см. ниже) по трем отдельным критериям может быть применим и для тех случаев, когда задача имеет штрафные функции  $f_2(\cdot)$  и  $f_3(\cdot)$  различной размерности.

Последовательное исключение узлов сети, положенное в основу Алгоритма 2.14, соответствует минимизации целевой функции  $f(\cdot)$ . Таким образом, в ходе работы алгоритма нет необходимости сравнивать значения целевой функции при выполнении Шагов 4 – 4.3.

Общую схему алгоритма для задачи (2.12) – (2.17) можно описать следующим образом.

Алгоритм 2.15. Генетический алгоритм с жадной эвристикой для планирования непрерывного производства

1. Сгенерировать начальный массив множеств узлов нашей сети, представленных тройками индексов  $(i, k, l)$ :  $A = \{\chi_j\} = \{(i_1, k_1, l_1), \dots, (i_{p_j}, k_{p_j}, l_{p_j})\}$ ,  $j = \overline{1, N_{POP}}$ . Здесь  $N_{POP}$  – число ”особей” популяции генетического алгоритма. Число узлов  $p_j$  в каждом из элементов массива  $A$  может быть различным.

2. Выбрать случайным образом два индекса родительских ”особей”  $j_1, j_2 \in \overline{1, N_{POP}}$ ,  $j_1 \neq j_2$ . Выбрать случайным образом  $j_3 \in w$ . Здесь  $w$  – некоторое множество индексов ”особей” (элементов массива  $A$ ), оцениваемых как ”плохие”. Методика оценки приведена отдельно ниже.

3. Присвоить  $\chi_{j_3} = \chi_{j_1} \cup \chi_{j_2}$ .

4. Выполнить процедуру мутации множества  $\chi_{j_3}$  (опционально, описана ниже).

5. Для каждого узла  $\mathcal{V} = (i_1, k_1, l_1) \in \chi_{j_3}$  выполнять:

5.1. Если  $\exists \mathcal{V}_2 = (i_2, k_2, l_2) \in \chi_{j_3}: l_2 \neq l_1, i_2 = i_1, k_2 = k_1$ , то выбрать случайным образом с равной вероятностью индекс  $l \in \{l_1, l_2\}$ , присвоить  $\chi_{j_3} = \chi_{j_3} \setminus \{(i_1, k_1, l)\}$ .

5.2. Следующая итерация цикла 5.

6. Вычислить соответствующие множеству  $\chi_{j_3}$  матрицы булевых и целочисленных переменных  $[x_{i,k,l}]$ ,  $[y_{i,k,l}]$ ,  $[x'_{i,k}]$  и  $[y'_{i,k}]$  по формулам (2.42), (2.37) – (2.39). Отметим, что, поскольку множества  $\chi_j$  в реализации алгоритма представлены матрицами целочисленных переменных, данный шаг и аналогичные шаги неже сводится к вычислению  $[y'_{i,k}]$  согласно выражению (2.37).

7. Присвоить  $FOUND = 0$ .

8. Разместив узлы множества  $\chi_{j_3}$  в случайном порядке, для каждого узла  $\mathcal{V} = (i', k', l') \in \chi_{j_3}$  выполнять:

8.1. Присвоить  $\xi = \chi_{j_3} \setminus \{\mathcal{V}\}$ . Вычислить соответствующие множеству  $\xi$  матрицы булевых и целочисленных переменных  $[x_{i,k,l}^\xi], [y_{i,k,l}^\xi], [x'_{i,k} \in [y'_{i,k}^\xi]$ . Если  $f_4(\xi) < f_4(\chi_{j_3})$ , то присвоить  $\chi_{j_3} = \xi$ ,  $FOUND = 1$ , перерассчитать соответствующие матрицы булевых и целочисленных переменных  $[x_{i,k,l}], [y_{i,k,l}], [x'_{i,k}]$  и  $[y'_{i,k}]$ . Перейти к следующей итерации цикла 7. Отметим, что исключение вершины  $\mathcal{V} = (i', k', l')$  из множества и перерасчет соответствующих целочисленных переменных сводится к обнулению значения  $x'_{i',k'}$  и перерасчету значений  $y'_{i'',k'}$ , для  $i'' = \overline{i', T_l}$ .

8.2. Выполнить некоторые процедуры локального поиска в окрестности узла  $\mathcal{V}$ . Данные опциональные процедуры, описанные ниже, значительно повышают эффективность работы алгоритма.

8.3. Следующая итерация цикла 7.

9. Если  $FOUND = 1$ , то вернуться к Шагу 7 (начать цикл заново).

10. Проверить условия останова, если они не выполняются, то перейти к Шагу 2.

Аналогично Алгоритму 2.14, данный алгоритм последовательно исключает по одной вершине из множества вершин  $\chi_{j_3}$  (Шаг 4.2.1 Алгоритма 2.14 и Шаг 8.1 Алгоритма 2.15). При работе Алгоритма 2.14 заранее известно конечное число вершин  $p$  (Шаг 4), при работе Алгоритма 2.15 узлы сети-решетки исключаются до тех пор, пока это дает улучшение значений штрафной функции  $f_4(\cdot)$ , см. Шаги 7, 8.1, 9. Шаги 5 – 5.2 гарантируют выполнение условия 4. При представлении множеств  $\chi_{j_1}, \chi_{j_2}, \chi_{j_3}$  матрицами целочисленных переменных  $[x_{i,k}^{j_1}], [x_{i,k}^{j_2}], [x_{i,k}^{j_3}]$ , данные шаги сводятся к копированию:

$$x_{i,k}^{j_3} = \begin{cases} \max\{x_{i,k}^{j_1}, x_{i,k}^{j_2}\}, & \min\{x_{i,k}^{j_1}, x_{i,k}^{j_2}\} = 0, \\ x_{i,k}^{j_r}, & \min\{x_{i,k}^{j_1}, x_{i,k}^{j_2}\} > 0. \end{cases}$$

Здесь  $r \in \{1, 2\}$  выбирается случайным образом для каждой пары  $(i, k): \min\{x_{i,k}^{j_1}, x_{i,k}^{j_2}\} > 0$ .

Мы использовали следующий алгоритм генерации начальной популяции, гарантирующий соответствие каждого решения ограничениям (2.15)–(2.17).



Алгоритм 2.16. Создание особи для начальной популяции генетического алгоритма.

Дано: число узлов сети  $p$ .

Шаг 1. Присвоить  $x'_{i,k} = 0 \forall i = \overline{1, I}, k = \overline{1, K}$ .

Шаг 2. Для  $j = \overline{1, p}$  выполнять:

Шаг 2.1. Выбрать случайное  $k \in \overline{1, K}$  (здесь и ниже при случайном выборе целого числа из указанного множества предполагаются равные математические ожидания выбора каждого из чисел). Рассчитать  $Z_k = \sum_{l=1}^L z_{k,l}$  – число видов продукции, которую можно производить на  $k$ -й линии. Выбрать случайное  $m \in \overline{1, Z_k}$ . Найти индекс  $m$ -го по счету ненулевого элемента  $k$ -го столбца матрицы  $Z = [z_{k,l}]$ , сохранить этот индекс в переменную  $l$ . Выбрать случайное  $i \in \overline{1, T_l}$ . Если  $x'_{i,k} = 0$ , то присвоить  $x'_{i,k} = l$ . Иначе повторить Шаг 2.1 с начала.

Шаг 2.2. Следующая итерация цикла 2.

Шаг 3. Возврат матрицы целочисленных переменных  $[x_{i,k}]$ , представляющей множество узлов сети  $\chi$ .

Параметром данного алгоритма является число узлов сети  $p$ . Практика показывает, что наилучшие результаты достигаются, если  $p$  в различных экземплярах исходной популяции варьируется в широких пределах – в диапазоне  $\overline{\{L, K \cdot I/2\}}$ . Возможен либо случайный выбор  $p$  из указанного множества для каждого генерируемого экземпляра начальной популяции (в вычислительных экспериментах, приведенных в настоящей статье, использован этот подход), либо определение его по формуле  $p = [L + j(K \cdot I/2 - L)/N_{POP}]$ , результаты при этом статистически равнозначны. Здесь  $j$  – номер генерируемого экземпляра,  $[\cdot]$  – целая часть числа.

Число “особей”  $N_{POP}$  популяции генетического алгоритма – важный параметр. Недостаточное количество снижает качество решений. Эмпирическим путем установлено, что значение  $N_{POP} = I + K + L$  является достаточным, дальнейшее увеличение не приводит к улучшению результатов, лишь увеличивая время их достижения (проиллюстрировано приведенными в настоящей статье результатами экспериментов).

В качестве вспомогательных процедур (Шаг 8.2 Алгоритма 2.15), ускоряющих поиск допустимых решений, используются процедуры локального поиска в окрестности узла  $\mathcal{V}$  (см. Шаг 4). Окрестностью в данном контексте будем называть множество узлов, отличающихся от  $\mathcal{V}$  значением координаты  $l$  (соответствует замене вида продукции) либо отличающихся от нее значением координаты  $i$  не более чем на 1 (перенос начала выпуска продукции на 1 день позже или раньше). Мы использовали три такие процедуры:

Процедура 1. (Шаг 8.2 Алгоритма 2.15). Выполняется для вершины  $\mathcal{V} = (i, k, l)$  при условии  $V_l \sum_{i=1}^{T_l} \sum_{k=1}^K y_{i,k,l} (3 - C'_{y'_{(i-1),k}, y'_{i,k}}) > W_l$  (условие перевыполнения плана по  $l$ -му виду продукции).

1. Составить множество индексов видов продукции  $S_L$ , удовлетворяющих условию  $V_l \sum_{i=1}^{T_l} \sum_{k=1}^K y_{i,k,l} (3 - C'_{y'_{(i-1),k}, y'_{i,k}}) < W_l$  (условие перевыполнения плана).

2. Выбрать случайным образом индекс  $l_2$  из этого множества.

3. Вычислить  $\xi = (\chi_{j_3} \setminus \{\mathcal{V}\}) \cup \{(i, k, l_2)\}$ . Если  $f_4(\xi) < f_4(\chi_{j_3})$ , то присвоить  $\chi_{j_3} = \xi$ ,  $FOUND = 1$ .

Процедура 2. (Шаг 8.2 Алгоритма 2.15). Выполняется для вершины  $\mathcal{V} = (i, k, l)$  при условии  $i > 1$  и  $(i - 1, k, l) \notin \chi_{j_3}$ .

Вычислить  $\xi = (\chi_{j_3} \setminus \{\mathcal{V}\}) \cup \{(i - 1, k, l)\}$ . Если  $f_4(\xi) < f_4(\chi_{j_3})$ , то присвоить  $\chi_{j_3} = \xi$ ,  $FOUND = 1$ .

Процедура 3. (Шаг 8.2 Алгоритма 2.15). Выполняется для вершины  $\mathcal{V} = (i, k, l)$  при условии  $i < T_l(i + 1, k, l) \notin \chi_{j_3}$ .

Вычислить  $\xi = (\chi_{j_3} \setminus \{\mathcal{V}\}) \cup \{(i + 1, k, l)\}$ . Если  $f_4(\xi) < f_4(\chi_{j_3})$ , то присвоить  $\chi_{j_3} = \xi$ ,  $FOUND = 1$ .

Следует отметить, что включение данных процедур в алгоритм существенно сокращает время поиска допустимых решений. Данные процедуры реализуют локальный поиск в окрестности каждого из узлов нашей сети-графика, заменяя узел  $\mathcal{V}$  на узел из его окрестности, если такая замена дает улучшение значений штрафной функции  $f_4(\cdot)$ . В данном случае процедура 1 для узла  $\mathcal{V} = (i, k, l)$  реализует поиск в

множестве узлов  $\{(i, k, l') | l' = \overline{1, L}\}$ , а процедуры 2 и 3 – в множестве узлов  $\{(i', k, l) | i' = i \pm 1, 1 \leq i' \leq T_l\}$ .

Применение локального поиска приводит к снижению разнообразия популяции, что может быть частично компенсировано вводом процедуры мутации, отсутствующей в Алгоритме 2.14.

Процедура 4. (Шаг 4 Алгоритма 2.15). Мутация множества  $\chi_{j_3}$ . Выполняется с некоторой заданной вероятностью  $p_m$ .

Сгенерировать множество  $\xi$  с помощью Алгоритма 2.16 с параметром  $p = |\chi_{j_3}|$ . Присвоить  $\chi_{j_3} = \chi_{j_3} \cup \xi$ .

Экспериментально установлены оптимальные значения  $p_m \in [0,005,0,03]$  при наличии в Алгоритме 2.15 процедур 2 и 3 и  $p_m \in [0,01,0,05]$  при наличии всех процедур локального поиска (Процедуры 1, 2, 3).

Алгоритм 2.15 порождает новые "особи" (промежуточные решения – множества узлов)  $\chi_{j_3}$ , которые заменяют собой существующие "особи" с индексом  $j_3$ , выбираемым случайным образом из множества "плохих" решений  $w$ . При этом определение этого множества должно производиться с учетом значения целевой функции  $f(\cdot)$  и штрафной функции  $f_4(\cdot)$ , вычисляемой в соответствии с выражениями (2.43) – (2.44). В предлагаемом алгоритме используется следующая процедура.

Процедура 5. Определение множества  $w$ .

1. Присвоить  $S_n = 0 \forall n = \overline{1, N_{POP}}$ .
2. Присвоить  $F_n = |\chi_n|$ .
3. Отсортировать массив  $F$  в порядке возрастания.
4. Найти медианное значение  $f' = F_{[N_{POP}/2]}$ .
5. Для  $n \in \{\overline{1, N_{POP}}\}$ :  $|\chi_n| \leq f'$  присвоить  $S_n = S_n + 1$ .
6. Присвоить  $F_n = f_4(\chi_n) \forall n = \overline{1, N_{POP}}$ .
7. Отсортировать массив  $F$  в порядке возрастания.
8. Найти медианное значение  $f'_4 = F_{[N_{POP}/2]}$ .
9. Для  $n \in \{\overline{1, N_{POP}}\}$ :  $f_4(\chi_n) \leq f'_4$  присвоить  $S_n = S_n + 2$ .
10. Для  $n \in \{\overline{1, N_{POP}}\}$ :  $f_2(\chi_n) = 0$  присвоить  $S_n = S_n + 1$ .

11. Для  $n \in \overline{\{1, N_{POP}\}}$ :  $f_3(\chi_n) = 0$  присвоить  $S_n = S_n + 1$ .

12. Найти наименьший индекс  $n \in \overline{\{1, N_{POP}\}}$ :  $f_4(\chi_n) = 0$  и  $|\chi_n| = \min_{q \in \{1, N\}} |\chi_q|$ . Если

индекс, удовлетворяющий данным условиям, существует, то присвоить  $S_n = S_n + 10$ .

13. Скопировать массив  $S$  в массив  $S'$ . Отсортировать массив  $S'$  в порядке возрастания.

14. Найти медианное значение  $s' = S'_{[N_{POP}/2]}$ .

15. Возвратить результат – множество индексов  $w = \{n \in \overline{\{1, N_{POP}\}} | S_n \geq s'\}$ .

Процедура реализует принцип скоринга (подсчета очков): решение получает 1 очко за то, что целевая функция имеет значение лучше, чем медианное по популяции, 2 очка – за значение штрафной функции  $f_4(\cdot)$ , лучшее, чем медианное по популяции, по 1 очку – за нулевое значение штрафных функций  $f_2(\cdot)$  и  $f_3(\cdot)$  и дополнительные 10 очков – за лучшее решение, удовлетворяющее ограничениям. ”Плохими” считаются решения, набравшие число очков не больше медианного значения по популяции.

Данная процедура требует достаточно больших вычислительных затрат, даже если значения целевой функции  $f(\cdot)$  и штрафной функций  $f_4(\cdot)$  для каждой ”особи”, заданной множеством  $\chi_n$ , не вычислять каждый раз заново, а хранить в массивах. Поэтому в предлагаемом Алгоритме 2.15 данная процедура выполняется на Шаге 2 после генерации начальной популяции, а также через каждые  $[N_{POP}/4]$  шагов (“особи” – кандидаты на выбывание определяются после смены четверти популяции). При включении в популяцию новой  $j_3$ -й особи полагаем, что  $j_3 \notin w$  (только что появившиеся “особи” кандидатами на выбывание не являются).

## 2.11 Результаты вычислительных экспериментов для задачи о составлении расписания

Алгоритм 2.15 реализован на языке Fortran 90 (компилятор ifort с опцией оптимизации кода и распаралеливания вычислений -O3). Для экспериментов использовалась вычислительная система Depo X8Sti (6-ядерное ЦПУ Xeon X5650 2.67 ГГц, 12 Гб ОЗУ), технология hyperthreading отключена. Также эксперименты

проводились на системе малой мощности с 2-ядерным ЦПУ Atom N270 1.6 ГГц, 1 Гб ОЗУ (те же результаты, время выполнения увеличивается в 16 – 21 раз).

На рис. 13 – 16 приведены результаты для задачи с параметрами задачи с параметрами, аналогичными приведенным в работе [236] ( $L = 34, K = 28, I = 31$ ), но более жестким ограничениями (2.13) и (2.14):  $T_l \in \{\overline{10,31}\}$ ,  $W_{min} = 1925$ ,  $V_l \in [40,50]$ ,  $W_l \in [20,25000]$ .

Всего использовано 10 наборов исходных данных, для каждого набора данных производилось по 10 запусков Алгоритма 2.15 в различных его модификациях.

На рисунках приведена зависимость наилучшего по значению целевой функции  $f(\cdot)$  допустимого ( $f_4(\chi) = 0$ ) решения от затраченного времени. Приведены усредненные значения целевой функции  $f(\chi) = |\chi|$  для 10 запусков алгоритма.

На рис. 2.13 и рис. 2.14 представлены сравнительные результаты Алгоритма 2.15 с различными параметрами. Процедуры локального поиска и мутации (Процедуры 1 – 4) очевидно улучшают результат для всех 10 наборов исходных данных.

Рис. 2.15 иллюстрирует зависимость динамики изменения достигнутых результатов от размера популяции ГА. Экспериментальным путем выведено приемлемое для большинства случаев значение  $N_{POP} = I + K + L$ . Также график включает результаты алгоритма локального поиска, составленного только из Процедур 1, 2, 3.

Наконец, рис. 2.16 демонстрирует результаты работы алгоритма для задачи с параметрами, в точности повторяющими параметры задачи, приведенной в работе [236] ( $L = 36, K = 28, I = 31, T_l = 31 \forall l = \overline{1, I}, \alpha = 0,5, W_{min} = \alpha \cdot \sum_{l=1}^L W_l / I = 484, V_l \in [40,50], W_l \in [20,25000], y_{0,k,l} = 0 \forall k, l$ ). С учетом, того, что время работы алгоритма, предложенного в работе [236], составляет  $8 \cdot 10^5$  секунд, результаты, достигнутые в настоящей работе, можно считать убедительными.

Попытки решения данной задачи с помощью пакетов программ линейного программирования, таких как GLPK, COIN-OR, CPLEX, не приводят к точному решению. Тем не менее, эвристики GLPK и CPLEX позволяют получить субоптимальные решения, которые, однако, значительно уступают решениям, полученным с помощью алгоритма, предложенного в настоящей работе. Результаты вычислительных алгоритмов приведены в Таблице 2.4.

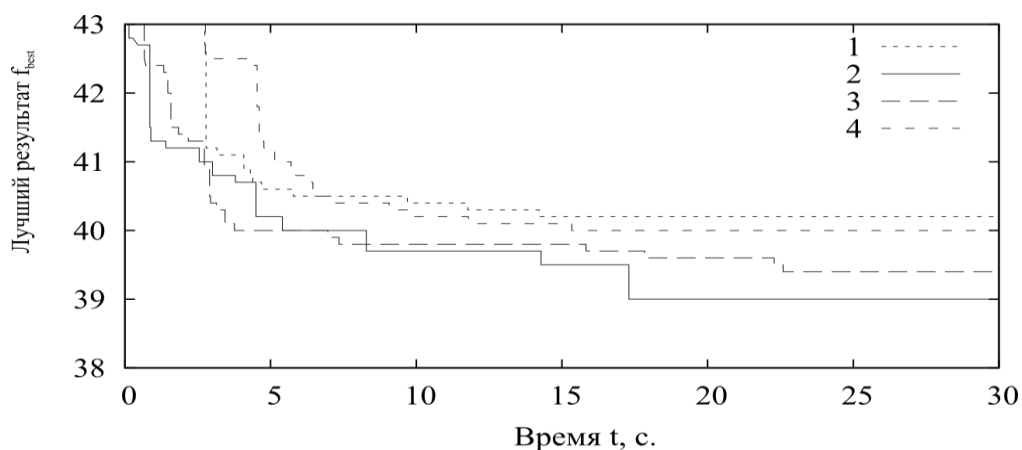


Рисунок 2.13 – влияние процедур локального поиска на результат.

1 – Алгоритм 2.15 без Процедур 1 – 3, 2 – с Процедурами 1 – 3,  
3 – только с Процедурой 1, 4 – с Процедурами 2 – 3.

Параметры задачи:  $L = 34, I = 31, P = 28, W_{min} = 1925$ .

Параметры алгоритма:  $N_{POP} = 89$ , вероятность мутации  $p_m = 0,01$ )

Решение задачи с меньшим числом видов продукции ( $L=10, M=6$ ) поддается решению методами целочисленного программирования. Решение упрощенной задачи с  $L=M$  (каждый вид продукции образует отдельный класс, задание матрицы  $C_{l,r}$  при этом не требуется, следовательно ввод переменных  $\omega_{i,k,l,r}$  также не является необходимым) с  $L=M=26, I=31, K=46$  также поддается решению пакетами COIN-OR, CPLEX, GLPK. При этом CPLEX дает более точное решение, чем предлагаемый в настоящей работе алгоритм, при этом требуется продолжительное время (более 7 часов счета).

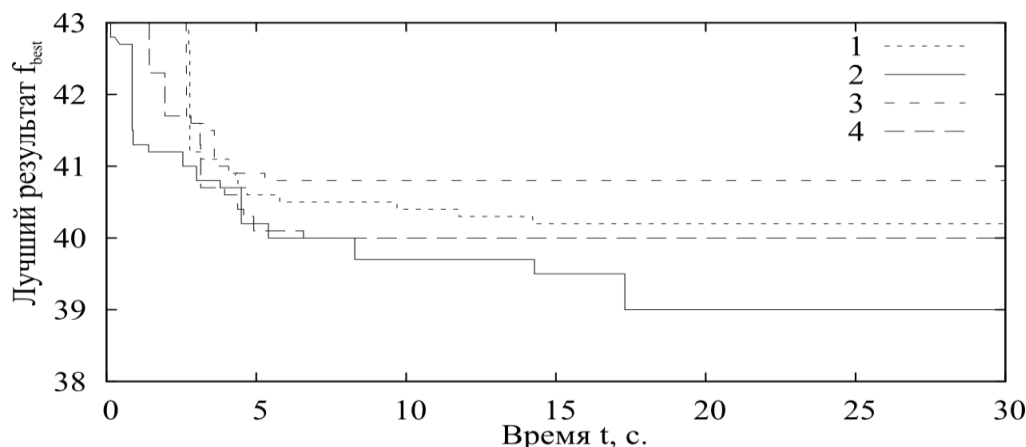


Рисунок 2.14 – влияние процедуры мутации на результат.

1 – Алгоритм 2.15 без Процедур 1 – 4, вероятность мутации  $p_m = 0,01$ ,  
2 – с Процедурами 1 – 3,  $p_m = 0,01$ , 3 – без Процедур 1 – 3,  $p_m = 0$ ,  
4 – с Процедурами 1 – 3,  $p_m = 0$ .

Параметры задачи:  $L = 34, I = 31, P = 28, W_{min} = 1925$ .

Параметры алгоритма:  $N_{POP} = 89$ )

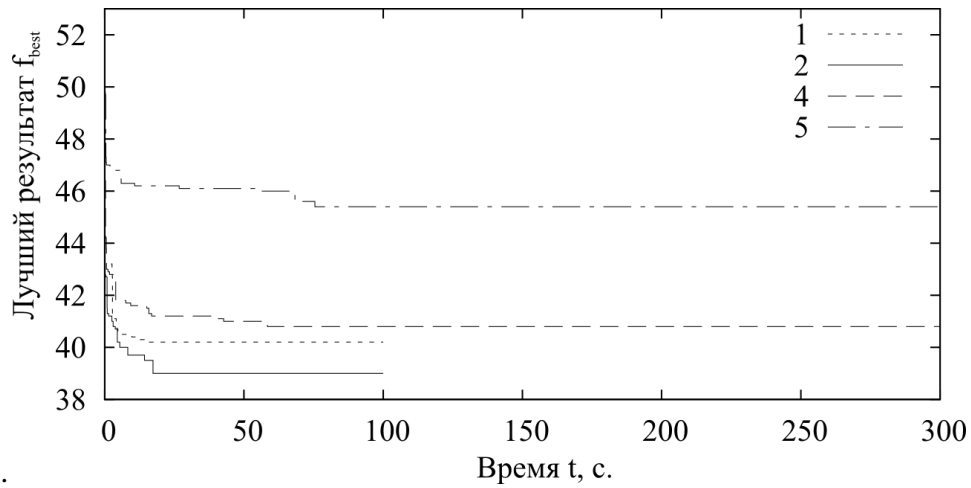


Рисунок 2.15 – влияние размера популяции на результат.

1 – Алгоритм 2.15 без Процедур 1 – 3,  $N = 89$ , 2 – с Процедурами 1 – 3,  $N_{POP} = 89$ ,  
 3 – без Процедур 1 – 3,  $N_{POP} = 500$ , 4 – с Процедурами 1 – 3,  $N_{POP} = 500$ ,  
 5 – алгоритм локального поиска Процедурами 1 – 3 с мультистартом.

Параметры задачи:  $L = 34$ ,  $I = 31$ ,  $P = 28$ ,  $W_{min} = 1925$ .

Параметры алгоритма:  $N_{POP} = 89$ , вероятность мутации  $p_m = 0,01$

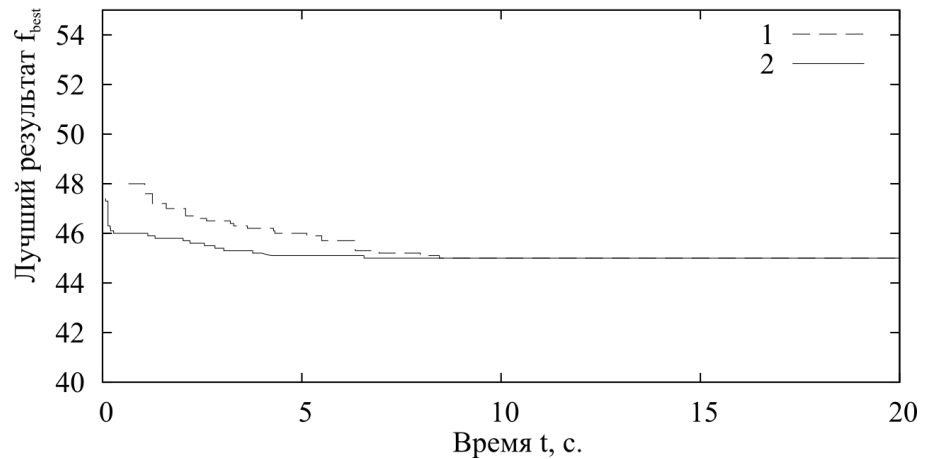


Рисунок 2.16 – Результаты для задачи с параметрами, приведенными в работе [236].

1 – Алгоритм 2.15 без Процедур 1 – 3, 2 – Алгоритм 2.15 с Процедурами 1 – 3.

Для подготовки задачи линейного программирования, решаемой пакетами CPLEX, GLPK, COIN-OR, использован язык программирования Python 2.7 и модуль PULP. В таблице 2.4 указано время решения задачи без учета времени подготовки задачи линейного программирования с использованием Python+PULP.

Таблица 2.4 – Результаты вычислительных экспериментов с различными алгоритмами для задачи о составлении расписания

Параметры задачи	Время для решения	Программный продукт для решения	Результат – значение целевой функции (число переключений)
$L=26, M=13, I=31, K=46$	1 час	GLPK	-
	12 часов	COIN-OR CPLEX	- -
	1 мин.	GLPK COIN-OR CPLEX Новый алгоритм	- - (сбой) 519 45
$L=M=26, I=31, K=46$	1 мин.	GLPK	-
	1 час	COIN-OR CPLEX	- -
	12 часов	GLPK COIN-OR CPLEX	- - 46
$L=10, M=6, I=31, K=46$	1 мин.	GLPK COIN-OR CPLEX	- - 15
	1 час	GLPK COIN-OR CPLEX	- 19 12
	12 часов	GLPK COIN-OR CPLEX	- 12 12
$L=10, M=6, I=31, K=46$	1 мин.	Новый алгоритм	12

Таким образом, задача планирования загрузки производственных мощностей литейно-прокатного или химического производства может рассматриваться как дискретная задача размещения. Предложенный алгоритм на базе генетического алгоритма с жадной эвристикой для  $p$ -медианных задач решает поставленные задачи, при этом затрачиваемое на решение время на несколько порядков меньше, чем в случае применения существующего алгоритма.

Алгоритм реализован в виде программного приложения [280] и внедрен в эксплуатацию на производстве изделий из полипропилена ООО «ИСТ» (г.Красноярск).



## Результаты и выводы к Главе 2.

Предложенный в настоящей работе комбинированный алгоритм решения задач на сети, сочетающий в себе метод изменяющихся вероятностей и генетический алгоритм со специальной жадной эвристикой, позволяет решать задачи большой размерности быстрее и точнее. Экспериментально доказана эффективность алгоритма для задач псевдодобулевой оптимизации с монотонными целевой функцией и левой частью ограничения. Примененный подход является основой новых алгоритмов для решения непрерывных задач размещения, приведенных в следующей Главе.

Проведенный анализ методов случайного поиска показал, что алгоритм, основанный на идеях метода изменяющихся вероятностей, может быть применен для решения задач автоматической группировки на сети. Обладая достаточно медленной сходимостью, алгоритм в комбинации с генетическим алгоритмом с жадной эвристикой повышает как быстродействие, так и точность последнего, что продемонстрировано на различных примерах задач большой размерности. Предложенный метод, состоящий в применении комбинации МИВЕРА и ГА с жадной эвристикой, успешно применяется при решении задач автоматической группировки и размещения на сетях, в частности – при решении задачи оптимального размещения пунктов экстренного реагирования в чрезвычайных ситуациях на сети автодорог Красноярского края, что подтверждается прилагаемым актом внедрения ФГБУ ВНИИ ГОЧС (ФЦ) МЧС России (Приложение Г).

На основе проведенного анализа существующих эволюционных алгоритмов впервые предложены модификации жадной эвристики в составе генетического алгоритма для задачи автоматической группировки на сети, позволяющие в случае задач с большим объемом входных данных получать стабильно более точные результаты за фиксированное время, либо получать точные результаты быстрее в сравнении с известными эволюционными алгоритмами с жадной агломеративной эвристической процедурой. Отметим, что дальнейшие модификации жадной агломеративной эвристической процедуры, приведенные в следующей Главе и применимые как к непрерывным, так и к дискретным задачам автоматической группировки, позволяют еще существеннее (до 55 раз) сократить время счета при неизменной точности результата по сравнению с другими методами.

Аналогичные модификации той же эвристической процедуры могут эффективно применяться для широкого круга задач псевдодбулевой оптимизации с монотонной целевой функцией и монотонными правыми частями ограничений, что показано на примере задачи оптимальной загрузки производственных мощностей литейно-прокатных и химических производств, для которой впервые предложен алгоритм, позволяющий получать решения в интерактивном режиме. Данный алгоритм внедрен в эксплуатацию на производстве изделий из полипропилена ООО «ИСТ» (г.Красноярск, см. Приложение Г).

Таким образом, в результате проведенных исследований, область применения генетических алгоритмов с жадной агломеративной эвристической процедурой для р-медианной задачи на сети была расширена как в направлении увеличения объема входных данных решаемых задач, так и в направлении охвата новых классов задач монотонной псевдодбулевой оптимизации.

### **ГЛАВА 3. МЕТОД ЖАДНЫХ ЭВРИСТИК ДЛЯ НЕПРЕРЫВНЫХ И ДИСКРЕТНЫХ ЗАДАЧ АВТОМАТИЧЕСКОЙ ГРУППИРОВКИ ОБЪЕКТОВ**

В настоящей главе рассмотрено применение различных вариантов алгоритмов с жадной эвристической процедурой для задач автоматической группировки объектов в метрическом пространстве в составе различных схем глобального поиска. Разработанные модификации алгоритмов были применены к задачам автоматической группировки как в непрерывной, так и в дискретной постановке. По результатам решения задач на тестовых примерах и реальных данных составлена схема совместимости различных стратегий глобального поиска, вспомогательных алгоритмов локального поиска, способов инициализации поисковых алгоритмов с различными модификациями жадных эвристических алгоритмов при различных постановках задач автоматической группировки, размещения или задач монотонной псевдобулевой оптимизации. Приведен способ оценки применимости жадных агломеративных эвристик и выбора модификаций алгоритмов на основе жадных агломеративных эвристик при построении автоматизированных систем решения задач автоматической группировки, размещения или монотонной псевдобулевой оптимизации в зависимости от условий и параметров решаемых задач. Эффективное сочетание перечисленных компонентов представляет собой метод жадных эвристик для потокового решения задач автоматической группировки и размещения.

#### **3.1 Общая постановка непрерывных задач автоматической группировки объектов**

Задачи автоматической группировки объектов в некотором метрическом пространстве или ином непрерывном пространстве, на котором определена мера (функция) расстояния (или мера сходства) между двумя точками данного пространства, являются наиболее широко применяемыми моделями. Например, к подобным задачам относится наиболее популярная в кластерном анализе модель – задача  $k$ -средних, в которой требуется разбить  $N$  объектов на  $k$  групп таким образом, чтобы сумма квадратов расстояний от объектов до центра каждой из групп достигало минимума. Центры (иногда называемые центроидами) – точки в том же пространстве, положение которых

требуется определить. Другая популярная модель автоматической группировки –  $p$ -медианная задача – имеет аналогичную постановку, но вместо суммы квадратов расстояний используются евклидовы расстояния или расстояния в других метриках. Таким образом, в задаче  $k$ -средних мерой расстояния является квадратичное евклидово расстояние (заметим, что мера расстояния, равная квадрату евклидова расстояния, метрикой не является: не выполняется неравенство треугольников), в задаче о  $p$ -медиане (иначе – о  $k$ -медиане) в качестве меры расстояния используется метрика. Общность непрерывной  $p$ -медианной задачи и задачи  $k$ -средних была подчеркнута многими исследователями [183, 187, 114, 281-283].

Как уже отмечалось в Главе 1,  $p$ -медианная задача является одной из классических моделей теории размещения. Общая цель непрерывной задачи размещения [113] состоит в нахождении местоположения одной или нескольких точек (центров, центроидов, медоидов и т.д. – в зависимости от конкретной постановки задачи) в непрерывном пространстве (рассматривается континуум возможных местоположений искомым точек). Забегая вперед, отметим, что существует промежуточный класс задач, фактически дискретных (число возможных местоположений является конечным), но при этом оперирующих понятиями, характерными для непрерывных задач. К таковым, в частности, относится и рассматриваемая в настоящей главе задача  $p$ -медоид [284, 71] (в литературе называемая также дискретной  $p$ -медианной задачей [285] или задачей  $k$ -медоид). Основными параметрами таких задач являются координаты объектов и расстояния между ними [225, 226, 114]. Примеры непрерывных задач автоматической группировки и задач размещения включают определение местоположения складов [114] таким образом, чтобы минимизировать расстояния от потребителей до ближайшего склада (иная постановка задачи – разбить множество потребителей на группы так, чтобы минимизировать расстояния от потребителей до центра каждой из групп), размещение узлов компьютерных и коммуникационных сетей (иная постановка – разбить множество элементов компьютерной инфраструктуры на группы так, чтобы минимизировать расстояния до центров групп), базовых станций беспроводных сетей [238, 286]. Задачи автоматической группировки данных в непрерывном пространстве возникают в статистике (например, задачи оценивания), статистической обработке данных [287], обработке сигналов и изображений и других инженерных приложений. Многие задачи

автоматической группировки [288-290] можно рассматривать как задачи размещения [291, 162] с квадратичными евклидовыми расстояниями [288, 130], евклидовыми [291, 73] или другими метриками и мерами расстояния [69] и наоборот.

Цель непрерывной  $p$ -медианной задачи [225] состоит в нахождении  $p$  точек (центров, центроидов, медиан), таких, чтобы сумма взвешенных (взятых с весовыми коэффициентами) расстояний от  $N$  известных точек, называемых точками требования, потребителями или векторами данных в зависимости от постановки и предметной области задачи, до ближайшего из  $p$  центров достигала минимума.

Непрерывные задачи размещения с евклидовой, манхэттенской (прямоугольной), чебышёвской метриками хорошо изучены (все эти метрики являются частными случаями метрик, основанных на  $l_p$ -нормах Минковского [150]), предложено множество алгоритмов для решения задачи Вебера для этих метрик. В частности, известная процедура Вайсфелда [144] была обобщена для метрик, основанных на нормах Минковского.

В традиционном понимании, в случае евклидовой метрики  $L(X_j, A_i) = \sqrt{\sum_{k=1}^d (x_{j,k} - a_{i,k})^2}$  мы имеем собственно  $p$ -медианную задачу. Здесь  $X_j = (x_{j,1}, \dots, x_{j,p}) \quad \forall j = \overline{1, p}$ ,  $A_i = (a_{i,1}, \dots, a_{i,k}) \quad \forall i = \overline{1, N}$ . В случае квадратичной евклидовой метрики  $L(X_j, A_i) = \sum_{k=1}^d (x_{j,k} - a_{i,k})^2$  при  $w_i = 1 \forall i = \overline{1, N}$  мы имеем задачу  $k$ -средних.

Допустив использование в  $p$ -медианной задаче произвольной меры расстояния (не обязательно метрики), будем считать задачу  $k$ -средних частным случаем  $p$ -медианной задачи с квадратичными евклидовыми расстояниями.

Простейший случай непрерывной задачи (случай с  $p=1$ ) – задача Вебера [292, 225, 114] – задача поиска такой точки, чтобы сумма взвешенных евклидовых расстояний от этой точки до заданных точек (существующих объектов, которые также называются «точки требования» или «векторы данных» в случае задачи автоматической группировки) достигала минимума:

$$\operatorname{argmin}_{X \in \mathbb{R}^2} F(X) = \sum_{i=1}^N w_i L(X, A_i).$$

Здесь  $L()$  – функция расстояния (норма, метрика или иная мера – произвольная функция, для которой, возможно, справедливы условия симметрии и тождества:  $L(X, Y) = L(Y, X)$ ,  $L(X, X) = 0$ ), евклидова в случае классической задачи Вебера.

Для решения этой задачи (поиска центра множества точек) мы можем использовать процедуру Вайсфелда [144] или ее улучшенные модификации [293, 148]. Аналогичные задачи с манхэттенской метрикой и метрикой Чебышева хорошо изучены [155, 243, 294]. Сходимость этих алгоритмов доказана для различных метрик [154]. Разработано множество алгоритмов для решения задачи Вебера с иными метриками и мерами расстояния [295, 296, 158]. Таким задачам посвящена Глава 5. Как будет показано ниже, существование эффективного алгоритма решения задачи Вебера – условие применимости предлагаемого в настоящей работе подхода к решению непрерывных задач автоматической группировки ( $p$ -медианных,  $k$ -средних) с соответствующей метрикой или мерой расстояния.

Собственно  $p$ -медианная задача является одним из возможных обобщений [114, 228] задачи Вебера:

$$\arg \min F(X_1, \dots, X_p) = \sum_{i=1}^N w_i \min_{j \in \{1, p\}} L(X_j, A_i). \quad (3.1)$$

Здесь  $\{A_i | i = \overline{1, N}\}$  – набор выбранных точек (векторов данных, в случае «геометрических» задач размещения используется термин точки-потребители),  $\{X_j | j = \overline{1, p}\}$  это набор новых размещаемых объектов,  $w_i$  – весовой коэффициент  $i$ -й заданной точки,  $L(\cdot)$  – функция расстояния в дискретном или непрерывном пространстве [217, 114]. В простейшем случае,  $L(\cdot)$  вычисляется как евклидово расстояние. В этом случае процедура Вайсфелда осуществляется до  $p$  раз на каждой итерации метода итеративного чередующегося расположения-распределения (ALA – alternating location-allocation – чередующееся размещение-распределение) [129, 297].

Простейшей и популярнейшей моделью автоматической группировки [298, 299, 289] является задача  $k$ -средних [48, 289]. Данная задача может быть сформулирована как  $p$ -медианная задача (с допущением, о котором говорилось выше), где  $w_i = 1 \forall i = \overline{1, N}$  и  $L(\cdot)$  – квадратичное евклидово расстояние  $l_2^2$ :

$$L(X, Y) = \sum_{i=1}^d (x_i - y_i)^2,$$

где  $X = (x_1, \dots, x_d) \in \mathbb{R}^d$ ,  $Y = (y_1, \dots, y_d) \in \mathbb{R}^d$ .

Если мера расстояния – квадратичная евклидова ( $l_2^2$ ), то решением задачи Вебера является точка (центроид) [114]:

$$x_j = \sum_{i=1}^N w_i a_i / \sum_{i=1}^N w_i. \quad (3.2)$$

В данном случае мы считаем, что  $X = (x_1, \dots, x_d)$ ,  $A_i = (a_{i,1}, \dots, a_{i,d}) \forall i = \overline{1, N}$ .

Поиск центроида для задачи с квадратичной евклидовой мерой расстояния занимает меньше вычислительных ресурсов, чем итеративный поиск центра в случае задачи Вебера и ALA алгоритм работает быстрее в этом случае.

Подобные  $p$ -медианные задачи с евклидовой ( $l_2$ ) метрикой, квадратичными евклидовыми расстояниями ( $l_2^2$ ) или другими  $l_p$  метриками [150] являются задачами общей оптимизации: целевая функция не является выпуклой [297]. ALA-процедура и ее аналоги (вариант – стандартная процедура  $k$ -средних [48]) могут найти один локальный минимум целевой функции, их результат зависит от заданного исходного решения. Более того, такие задачи глобальной оптимизации являются NP-трудными [300, 301, 151] как для непрерывного пространства, так и для дискретного множества возможных решений [219, 220, 184], что делает невозможным использование метода частичного или полного перебора для больших задач. В теории вычислительной сложности NP трудными называются задачи, которые «как минимум также сложны, как труднейшая из проблем класса NP». Более точно, задача является NP-трудной, если другая задача, принадлежащая классу NP, может быть сведена к ней за полиномиальное время [302, 303]. NP-класс можно определить как класс задач, которые можно решить на недетерминированной машине Тьюринга [304, 305, 306] за время, не превосходящее полинома от размера данных.

Наиболее популярная ALA-процедура для задачи  $k$ -средних основана на алгоритме, предложенном Ллойдом [48]. Алгоритм также известен как стандартная процедура  $k$ -средних [289] или алгоритм Ллойда. Тем не менее, многие авторы предложили более быстрые методы, основанные на этой стандартной процедуре [103, 307, 308] для наборов данных и непрерывных потоков поступающих данных. ALA-процедура (а также процедура  $k$ -средних) являются алгоритмами, последовательно улучшающими известное решение, позволяя найти локальный минимум. При этом они не являются в строгом смысле алгоритмами локального поиска, поскольку поиск нового решения ведется не обязательно в  $\varepsilon$ -окрестности имеющегося решения.

Многие авторы предлагают подходы, основанные на уменьшении объема данных [309]: упрощение задачи путем случайного (или детерминированного) выбора части начального набора данных и использование этих результатов в качестве начального решения ALA-процедуры на полном наборе данных [71, 108, 111, 310, 311, 312]. Аналогичные подходы были предложены для дискретных  $p$ -медианных задач [223].

Современная литература по методам решения задач размещения предлагает множество эвристических методов [313] задания начальных центров ALA-процедуры, в основном являющихся разнообразными эволюционными методами и методами случайного поиска. Методом случайного поиска фактически является и популярная процедура k-means++ [314].

Популярной идеей при решении задач автоматической группировки в непрерывном пространстве является использование генетических алгоритмов (ГА) и иные эволюционные подходы для улучшения результатов локального поиска [315, 84, 194, 199]. Многие из таких эволюционных алгоритмов в ходе своей работы рекомбинируют начальное решение, полученное ALA-процедурой.

В случае применения генетических и, в более широком смысле, эволюционных алгоритмов, применяются различные способы кодирования допустимых решений, образующих «популяцию» эволюционного алгоритма. Собственно варианты решений, закодированные определенным способом, являются «особями» в этой популяции. Хосадж и Годчайлд [201] предложили первый генетический алгоритм для  $p$ -медианной задачи на сети. Весьма точный, хотя и очень медленный алгоритм на основе специальной жадной агломеративной эвристики в 2002г. предложили Бозкая, Чжан и Эркут [203]. Следует напомнить, что алгоритм, содержащий аналогичные идеи – агломеративную эвристику, был предложен в 1963 году Куном и Гамбургером [131].

Достаточно точный и быстрый алгоритм с особым способом рекомбинации, названным агломеративной эвристикой, для  $p$ -медианной задачи на сети предложен Альпом, Эркутом и Дрезнером в том же 2002 году. Тем не менее, как показано в настоящем исследовании (см. предыдущую Главу), алгоритм может быть существенно улучшен по быстродействию. Данный алгоритм был адаптирован для непрерывных задач в работе Неэма в 2011 году, было предложено использовать жадную агломеративную эвристику наряду с другими эвристиками рекомбинации для создания начальных решений для ALA-процедуры [199]. Следует отметить, что для непрерывных задач алгоритм генерирует начальные решения для ALA-процедуры или другой процедуры локального поиска, многократно выполняемой на каждой итерации, что делает алгоритм чрезвычайно медленным при росте значения  $p$ . Иной способ рекомбинации предлагают, например, Шен и Лиу (2006) [82]. Эти алгоритмы работают быстрее, но менее точны. Также Лим и Сю в 2003 году предложили генетический



алгоритм, основанный на рекомбинации подмножеств центров фиксированной мощности [83].

Неоднократно предпринимались попытки применения и других эвристических алгоритмов. Например, в работах [157, 316, 317] используется алгоритм, основанный на методах изменяющихся вероятностей, который, однако, показал свою эффективность лишь как вспомогательный метод для ускорения сходимости генетического алгоритма на начальных итерациях. В работе Ливановой и Лореш [318] предложен алгоритм муравьиных колоний и симуляции обжига, который показал свою ограниченную эффективность.

Отметим, что для задачи  $k$ -средних существуют алгоритмы глобального поиска, гарантирующие получение точного результата, сводящиеся к перебору возможных вариантов начального решения ALA-процедуры, в качестве которого принимается подмножество векторов данных [77]. Такой подход требует неприемлемо больших вычислительных затрат даже на относительно небольших объемах данных. Для  $p$ -медианной задачи с прямоугольной метрикой несложно выделить конечное множество потенциальных точек минимума [319, 294], что также в принципе позволяет найти глобальный минимум путем перебора, не избавляя от необходимости применять эвристические методы при больших объемах данных.

Идея жадной агломеративной эвристики, примененной в главе 2 к задачам на сети, заключается в следующем. После выбора двух «родительских» решений («особей» генетического алгоритма), новое промежуточное решение, в общем случае – недопустимое (решение-кандидат), является «объединением» родительских особей. Из нового решения по одной исключаются элементы решения (центры, центроиды, медоиды), пока решение не станет допустимым, т.е. пока число элементов решения (центров, центроидов, медоидов и соответственно – число групп, на которые разбивается множество векторов данных) не снизится до  $p$ . На каждом шаге алгоритма исключается элемент, дающий наименьший прирост целевой функции. Если алгоритм используется для непрерывной задачи, то он создает начальное решение для ALA-процедуры [199], которая должна быть выполнена на каждом этапе для оценки результата устранения каждого объекта из промежуточного решения.

Генетический алгоритм с жадной эвристикой [204] не предусматривает использования мутации, характерной для многих генетических алгоритмов [320, 321].

В настоящей работе, кроме нескольких модификаций к ГА с жадной агломеративной эвристикой для р-медианной задачи, предлагается новый ГА, использующий вещественный алфавит (в реализации на ЭВМ – алфавит с плавающей запятой), основанный на идеях алгоритма, предложенного Альпом и др. [204]. В оригинале алгоритм Альпа использует целочисленный алфавит (номера вершин сети) в «хромосомах» (промежуточных решений) генетического алгоритма. Эта версия для задач размещения на плоскости [199] использует целочисленный алфавит для кодирования номеров векторов данных, используемых в качестве исходных решений ALA-процедуры. В нашем алгоритме мы используем кодирование элементов решения с помощью вещественных чисел. Элементы «хромосом» нашего ГА – это координаты центров или центроидов промежуточных решений, которые были изменены на шаге ALA-процедуры, которые устраняются, пока не будет получено допустимое решение. Такое сочетание жадной агломеративной эвристики и ALA-процедуры позволяет алгоритму получить более точные результаты.

При решении непрерывных задач с помощью ГА с жадной эвристикой выполняется собственно процедура жадной эвристики. В настоящей работе предлагается новая процедура, которая позволяет удалять из промежуточного решения целые наборы центров или центроидов за одну итерацию, снижая при этом временные затраты.

### 3.2 Известные методы

Как говорилось выше, в случае непрерывных задач одной из популярнейших и при этом простых и эффективных процедур локального поиска является ALA-процедура (ALA-алгоритм, стандартная процедура k-средних). Основная идея ALA-процедуры – изменяющееся размещение центров (пересчет центров групп вершин сети или векторов данных - кластеров) и распределение вершин векторов данных между центрами (отнесение к группе - кластеру, центр которого является ближайшим).

Алгоритм 3.1. ALA-процедура (чередующееся размещение-распределение) [315].

Дано: Множество  $V = \{A_1, \dots, A_N\}$  с  $N$  векторами данных в  $d$ -мерном пространстве,  $A_1 = \{a_{1,1}, \dots, a_{1,d}\}, \dots, A_N = \{a_{N,1}, \dots, a_{N,d}\}$ , исходные данные: множество

центров  $p$  кластеров  $X_1 = \{x_{1,1}, \dots, x_{1,d}\}, \dots, X_p = \{x_{p,1}, \dots, x_{p,d}\}$ .

1: Для каждого вектора найти ближайший центр:

$$C_i = \operatorname{argmin}_{j=1,p} L(A_i, X_j) \forall i = \overline{1, N}.$$

2: Для каждого кластера  $C_j^{clust} = \{i \in \{1, N\} | C_i = j\}$ , пересчитать его центр  $X_j$ . В случае использования евклидовой ( $l_2$ ) метрики может быть применена процедура Вайсфелда либо его модификация. В случае квадратичной Евклидовой ( $l_2^2$ ) метрики, решение (3.2) используется для получения очередного местоположения центра (в данном случае называемого центроидом).

3: Повторять с шага 1, если состав хотя бы одного кластера изменился.

4: В противном случае ОСТАНОВ.  $X_1, \dots, X_p$  являются локальными минимумами.

В случае квадратичных евклидовых расстояний и векторов данных с равными весовыми коэффициентами поиск нового центра кластера  $X_j = (x_{j,1}, \dots, x_{j,d})$  на Шаге 1 ALA-процедуры представляет собой простейшую задачу и осуществляется итеративно:

$$x_{j,k} = \sum_{i \in C_j^{clust}} a_{i,k} / |C_j^{clust}| \forall k = \overline{1, d}.$$

Связанные с этим низкие вычислительные затраты во многом обуславливают популярность модели  $k$ -средних. В случае евклидовой метрики новый центр является решением «классической» задачи Вебера [292], которая может быть решена градиентным спуском – процедурой Вайсфелда или более совершенными ее модификациями [144, 148].

Для снижения вычислительных затрат на Шаге 2 ALA-процедуры алгоритм не пересчитывает центры кластеров, состав которых не был изменен на Шаге 1. Для евклидовой метрики  $l_2$  это позволяет избежать применения процедуры Вайсфелда или аналогичной процедуры градиентного спуска в каждом кластере при каждой итерации. В случае задачи  $k$ -средних в пространстве небольшой размерности данное замечание можно не учитывать.

ALA-процедура – это процедура локального поиска, соответственно, результаты ее зависят от начальных данных. В простейшем случае векторы данных  $p$  могут быть случайным образом выбраны в качестве начальных центров. Процедура, называемая ее авторами  $k$ -means++ [314], гарантирует точность результата  $O(\log(p))$  при ее

использовании для выбора начального решения стандартной процедуры  $k$ -средних. Идея этого метода основана на изменении вероятности выбора векторов данных в качестве исходных центров в зависимости от расстояния до ближайших векторов данных, выбранных ранее в качестве элементов начального решения. Аналогичный метод, основанный на идеях метода изменяющихся вероятностей, для дискретных задач размещения [157] описан в предыдущей Главе. Алгоритм  $k$ -means++ состоит в следующем:

Алгоритм 3.2.  $k$ -means++ [314].

Дано: Множество  $V = \{A_1, \dots, A_N\} \in \mathbb{R}^d$ , с числом кластеров  $p$ .

1: Установить начальное значение вектора вероятности  $P = (p_1, \dots, p_N)$  с равными значениями (например, равными  $1/N$ ). Установить множество центров  $\chi = \emptyset$ .

2: Выбрать один вектор данных  $X$  из множества векторов данных  $V$  случайным образом с учетом вероятностей  $P$ : вычисляем  $S = \sum_{i=1}^N p_i$ , выбираем случайное значение  $r \in [0; S)$  с равномерным распределением и используем  $i_{min} = \operatorname{argmin}_{i \in \{1, \dots, N\}: \sum_{j=1}^i p_j < r} i$ .

Дополняем множество  $\chi = \chi \cup \{A_{i_{min}}\}$ .

3: Для каждого  $i \in \{1, \dots, N\}$  присвоить  $p_i = \min_{X \in \chi} L(X, A_i)$ . Здесь  $L(\cdot)$  - мера расстояния.

4: Если  $|\chi| < p$  тогда переходим к Шагу 2.

5: Иначе ОСТАНОВ.  $\chi$  – первоначальный набор центров.

Существует множество подходов к оптимизации работы ALA-алгоритма [308], например сэмплинг [299, 309] (решение задачи на случайным образом выбранной части данных и использование результата в качестве начального решения при решении полной задачи), различные потоковые алгоритмы для работы с большими объемами данных [308, 322, 323] и др. Одна из простейших реализаций метода  $k$ -средних с сэмплингом [309, 310] такова:

Алгоритм 3.3.  $k$ -средних с сэмплингом (простейший вариант).

Дано: Множество  $V = (A_1, \dots, A_N) \in \mathbb{R}^d$ , числом кластеров  $p$ , с параметрами  $s \in (0; 1)$ .

1: Случайным образом выбрать  $s$  векторов данных из  $V$  и сформировать новое множество  $V_s$ .

2: Выбрать множество начальных центров (центроидов)  $\chi$ . Традиционно начальные центры выбираются из множества векторов данных. Выбор может быть случайным, с равной вероятностью. Для улучшения результатов может быть выполнена процедура k-means++.

3: Запускается Алгоритм 3.1 с начальным значением данных множества  $\chi$  и множества векторов данных  $V_s$ . После этого мы получаем модифицированное множество центров  $\chi$ .

4: Запускаем Алгоритм 3.1 с начальным множеством  $\chi$  для всего набора векторов данных  $V$ .

5: ОСТАНОВ.

В [310] авторы предлагают метод выбора оптимального значения параметра  $s$ .

Такие методы, как k-средних с сэмплингом, процедура k-means++ для генерации начальных решений и другие методы улучшают результаты процедуры k-средних, однако они не устраняют его наиболее важный недостаток: все они выполняют процедуру локального поиска. Простейший подход, используемый для глобальной оптимизации – многократный старт (мультистартом) локального поиска [235]. В этом случае процедура локального поиска работает со случайным образом сгенерированными начальными данными. Для  $p$ -медианной задачи или задачи  $k$ -средних этот алгоритм состоит в следующем:

#### Алгоритм 3.4. Случайный мультистарт.

Дано: Множество  $V = (A_1, \dots, A_N) \in \mathbb{R}^d$ , числом групп (кластеров)  $p$ .

1: Множество  $F^* = +\infty$ .

2: Инициализировать множества индексов векторов данных  $\chi: \chi \subset \{\overline{1, N}\}, |\chi_k| = p$ . Может быть использован случайный выбор с равномерным распределением или процедура k-means++.

3: Выполнить ALA-процедуру с начальными решениями  $\chi$  и получить локальный минимум целевой функции  $F^*$  (3.1) и набор соответствующих центров  $\chi^*$ .

Вместо ALA-процедуры в «чистом» виде,  $k$ -средних с сэмпингом может быть использован в случае большого набора данных.

4: Если  $F^{**} > F^*$ , то множество  $F^{**} = F^*$ ;  $\chi^{**} = \chi^*$ .

5: Проверяем условия ОСТАНОВА. Если условия не достигнуты, переходим к Шагу 2.

6: Иначе, STOP. Возвратить решение  $\chi^{**}$ .

Схема алгоритма с жадной эвристикой предложена Альпом и др. для непрерывных задач размещения и представлена [204, 199].

### Алгоритм 3.5. ГА с жадной эвристикой.

Дано: Множество  $V = (A_1, \dots, A_N) \in \mathbb{R}^d$ , число кластеров  $p$ , размер популяции  $N_p$ .

1: Сформировать  $N_p$  множеств индексов векторов данных  $\chi_1, \dots, \chi_{N_p}: \chi_i \subset \overline{\{1, N\}}, |\chi_k| = p \forall k = \overline{1, N_p}$ . Для каждого  $k \in \overline{\{1, N_p\}}$  вычисляем функцию пригодности (целевую функцию). В случае непрерывной -медианной задачи для получения значения функции пригодности  $\chi_k$  применяется ALA-процедура с начальным значением  $\chi_k$  и вычисляется

$$\mathcal{F}_k = F(\chi_k^*) = \sum_{i=1}^N w_i \min_{X \in \chi_k^*} L(X, A_i). \quad (3.3)$$

Здесь  $\chi_k^*$  - результат выполнения ALA процедуры с исходным множеством векторов данных  $\{A_i | i \in \chi_k\}$ .

2: Если условия ОСТАНОВА достигнуты, переход к Шагу 7.

3: Выбираются случайным образом два «родительских» множества  $\chi_{k_1}$  и  $\chi_{k_2}$ ,  $k_1, k_2 \in \overline{\{1, N_p\}}, k_1 \neq k_2$ . Запускается специальная кроссинговер-процедура с жадной эвристикой, выбирается «дочернее» множество с индексами  $\chi_c$ . Вычисляется значение функции пригодности  $\mathcal{F}_c$  в соответствии с (3.3).

4: Если  $\exists k \in \overline{\{1, N_p\}}: \chi_k = \chi_c$ , то переходим к Шагу 2.

5: Выбираем индекс  $k_{worst} = \operatorname{argmax}_{k=\overline{1, N_p}} \mathcal{F}_k$ . Если  $\mathcal{F}_{worst} < \mathcal{F}_c$ , то переходим к Шагу 2.

6: Заменить  $\chi_{k_{worst}}$  на  $\chi_c$ , заменить соответствующее значение целевой функции

$\mathcal{F}_{k_{worst}} = \mathcal{F}_c$ . Перейти к Шагу 2.

7: СТОП. Результат представляет собой набор  $\chi_k^*$ ,  $k^* = \operatorname{argmin}_{k=\overline{1, N_p}} \mathcal{F}_k$ .

В приведенной версии алгоритма на Шагах 5 и 6 худшее решение  $\chi_{worst}$  заменяется новым. В нашем случае мы использовали другую процедуру на Шаге 5 (простая турнирная селекция): случайно выбираются 2 индекса  $k_1$  и  $k_2$ ,  $k_1 \neq k_2$ ; множество  $k_{worst} = \operatorname{argmax}_{k \in \{k_1, k_2\}} \mathcal{F}_k$ . Данная версия Шага 5 дает лучшие результаты на всех исследованных тестовых примерах с  $N > 199$ ,  $p > 5$ .

Отметим, что в следующих параграфах рассмотрены, кроме мультистарта и генетического алгоритма, и другие схемы глобального поиска с применением рассматриваемой жадной агломеративной эвристики.

В случае случайного мультистарта и генетических алгоритмов могут быть использованы различные условия останова. Мы использовали ограничение по времени. В отличие от большинства генетических алгоритмов, этот метод не использует процедуру мутации. Однако, процедура кроссинговера использует особую эвристику:

Алгоритм 3.6. Эвристическая процедура жадного скрещивания (жадная эвристика).

Дано: Множество  $V = (A_1, \dots, A_N) \in \mathbb{R}^d$ , количество  $p$  кластеров, два "родительских" набора центров или центроидов  $\chi_{k_1}$  и  $\chi_{k_2}$ , кодируемых индексами векторов данных.

1: Присвоить значение множества  $\chi_c = \chi_{k_1} \cup \chi_{k_2}$ . Отметим, что  $p \leq |\chi_c| \leq 2p$ , т.е., дочернее решение  $\chi_c$  не возможно.

2: Если  $|\chi_c| = p$ , то ОСТАНОВ и возврат к решению  $\chi_c$ .

3: Посчитать  $j^* = \operatorname{argmin}_{j \in \chi_c} F(\chi_c \setminus \{j\})$ .

4: Присвоить значение множества  $\chi_c = \chi_c \setminus \{j^*\}$ .

5: Перейти к шагу 2.

На каждой итерации один центр или центроид удаляется из промежуточного решения (Шаг 4). На шаге 3 Алгоритм 3.6 выбирает центр или центроид, при удалении которого целевая функция дает наименьший прирост. При этом для оценки значения

целевой функции должен быть выполнен запуск ALA-процедуры. Таким образом, Шаг 3 Алгоритма 3.6 требует значительных вычислительных затрат. В случае евклидовой метрики, кроме того, процедура Вайсфелда должна быть запущена  $|\chi_c|$  раз на каждой итерации ALA алгоритма.

Поэтому Алгоритм 3.6 является вычислительной процедурой, медленной для больших наборов данных в случае задачи  $k$ -средних, практически непригодной в случае больших непрерывных  $p$ -медианных задач с евклидовой метрикой. Идея этой эвристики соотносится с методом информационного бутылочного горлышка (IBC – Information Bottleneck Clustering – кластеризация информационным бутылочным горлышком) [73]. В начале работы IBC-алгоритмов все векторы данных образуют отдельный кластер. На каждом шаге один кластер удаляется и его составляющие присоединяются к другим кластерам. Чтобы выбрать кластер, алгоритм вычисляет "информационные потери". В случае «геометрической» кластеризации, основанной на расстояниях между объектами, эти потери могут быть оценены функцией расстояния. Вычислительная нагрузка в случае IB-кластеризации позволяет реализовать этот метод на небольших наборах данных ( $N < 1000$ ).

Целевые функции практических задач имеют, как правило, большое число локальных минимумов [303] (растет экспоненциально с ростом  $p$ ). Вследствие этого поиск универсального алгоритма, гарантирующего точное решение (кроме задач очень малой размерности) давно признан бесперспективным. Некий компромисс по точности и быстрдействию представляют собой генетические алгоритмы [113] с агломеративной эвристикой [204], изначально разработанные для решения  $p$ -медианной задачи на сети [217]. В версии, предложенной в работе [199], такой алгоритм может быть применен для решения непрерывных задач. Данные алгоритмы, тем не менее, обладают невысокой производительностью на больших объемах данных. В настоящей работе повышена производительность данных алгоритмов без существенного снижения точности.

Этот алгоритм может быть непосредственно использован для решения  $p$ -медианной задачи (3.1) с дополнительным условием:

$$X_j \in V \forall j \in \overline{1, p}, \quad (3.4)$$

Заметим, что данное условие фактически превращает  $p$ -медианную задачу или задачу  $k$ -средних в задачу  $k$ -медоид, о которой более подробно говорится в следующих параграфах.



Полученный результат может такого генетического алгоритма быть использован как начальное решение для ALA-процедуры.

Алгоритм 3.7. Процедура кроссинговера с жадной эвристикой для создания начального решения ALA-процедуры (Модификация 1 жадной эвристики).

Дано: Множество  $V = \{A_1, \dots, A_N\} \in \mathbb{R}^d$ , количество  $p$  кластеров, два «родительских» набора центров или центроидов  $\chi_{k_1}$  и  $\chi_{k_2}$ .

- 1: Присвоить значение множества  $\chi_c = \chi_{k_1} \cup \chi_{k_2}$ .
- 2: Если  $|\chi_c| = p$ , то ОСТАНОВ и вернуть решение  $\chi_c$ .
- 3: Вычислить  $j^* = \operatorname{argmin}_{k \in \chi_c} \sum_{i=1}^N (\min_{j \in (\chi_c \setminus \{k\})} w_i L(A_i, A_j))$ .
- 4: Присвоить  $\chi_c = \chi_c \setminus \{j^*\}$ .
- 5: Перейти к шагу 2.

В этом случае ALA процедура всегда начинает с локального минимума дискретной задачи (3.1) – задачи р-медоид – с дополнительным ограничением (3.4). Данная версия алгоритма намного быстрее и его результат лучше, чем мультистарт случайного поиска (Алгоритм 3.4) для наиболее популярных тестовых примеров (см. ниже). Однако такие результаты могут быть улучшены.

### 3.3 Модификация жадной эвристической процедуры – частичное объединенное решение

В настоящем параграфе представлена модификация жадной агломеративной эвристики, повышающая эффективность алгоритмов как для непрерывной р-медианной задачи и задачи k-средних, так и дискретных задач – р-медианной задачи на сетях и задачи р-медоид. Эффективность данных модификаций подтверждается решением тестовых задач и практическими примерами [324, 325].

Шаг 1 Алгоритмов 3.6 и 3.7 создает промежуточное решение  $\chi_c$ . В общем случае это решение не является допустимым: мощность его может достигать  $2p$ . На шагах 3–4 Алгоритм 3.7 исключает один элемент из  $\chi_c$  до тех пор, пока не выполняется  $|\chi_c| = p$ . Данные шаги требуют множественных стартов ALA-процедуры, которая в этом случае запускается  $|\chi_c|$  раз на каждой итерации. Таким образом, ALA-процедура запускается

до  $2p + 2p - 1 + 2p - 2 + \dots + p + 1$  раз. Кроме того, вычислительная сложность ALA-процедуры зависит от  $|\chi_c|$ .

Генетические алгоритмы, в том числе Алгоритм 3.5, могут использовать различные условия останова (Шаг 2) [204, 199, 157]. Алгоритм может быть остановлен после определенного числа выполненных итераций, не дающих улучшения целевой функции (после некоторого числа безрезультатных шагов).

Иная идея глобального поиска в случае  $p$ -медианной задачи включает улучшение результата локального поиска путем замены части центров (центроидов) на случайным образом выбранные точки [199]. Здесь предлагается комбинация двух идей. В нашей следующей модификации Алгоритмов 3.6 и 3.7 мы добавляем случайное число  $r$  центров одного «родительского» решения к другому «родительскому» решению («особи») генетического алгоритма. Распределение  $r$  должно быть таково, чтобы оно могло позволить добавлять  $1 \dots p$  элементов решения, при этом в случае больших задач прослеживалась бы тенденция добавления небольшого количества элементов решения. Таким образом, жадная эвристика при этом удаляет  $r = |\chi_c| - p$  «лишних» центров из промежуточного решения

Шаг 1 Алгоритмов 3.6 и 3.7 приобретает следующий вид.

Алгоритм 3.8. Частичное объединенное решение – модификация Шага 1 Алгоритмов 3.6 и 3.7 (Модификация 2 жадной эвристики).

1.1: Сгенерировать случайное значение  $r_{init} \in [0; 1)$  с равномерным распределением (стандартная функция Random).

1.2: Присвоить  $r = 1 + [(p - 1) \cdot r_{init}^2]$ .

1.3: Из множества  $\chi_{k_2}$  выбрать случайным образом подмножество  $\chi_{k_2}^*$  мощности  $r$ .

1.4: Сформировать промежуточное решение  $\chi_c = \chi_{k_1} \cup \chi_{k_2}^*$ .

Здесь мощность  $r \in \overline{\{1, p\}}$  множества  $\chi_{k_2}^*$  может быть выбрана в указанном интервале с равной вероятностью. Тем не менее, выражение на Шаге 1.2 дает лучшие результаты в случае больших задач.

Некоторые модификации Алгоритмов 3.6 и 3.7 были предложены автором диссертации в [249]. Модификация, предложенная выше, может быть использована

совместно с модификациями, рассмотренными в [249], либо применяться отдельно от них к генетическому алгоритму с жадной эвристикой, предложенному в [204, 199].

Результаты предложенных модификаций генетических алгоритмов с жадной эвристикой для  $p$ -медианных задач в непрерывном пространстве представлены ниже.

Для целей тестирования мы использовали данные из репозитория UCI [326], а также автоматически сгенерированные данные, представляющие собой случайным образом выбранные пары координат с равномерным распределением на плоскости в квадрате  $10 \times 10$ . Отметим, что такие задачи с равномерным распределением, то есть те задачи, в которых выраженная кластерная структура фактически отсутствует, являются наиболее трудными.

Среди тестовых задач максимальное число векторов данных для задачи  $k$ -средних было 169309 ( $N = 169309$ , набор данных "Europe" из репозитория UCI), максимальная размерность пространства для непрерывных задач  $d = 32$ . Кроме того, мы прорешали несколько  $p$ -медианных задач с евклидовой метрикой.

Важным параметром генетических алгоритмов, в том числе Алгоритма 3.5, является размер популяции  $N_{pop}$ . Неэма и др. [199] не предлагают какой-либо методики определения данного значения. В исходном генетическом алгоритме с жадной эвристикой для  $p$ -медианной задачи на сети [204] размер популяции зависит от числа вершин сети  $N$  и числа выбираемых в качестве центров вершин  $p$ . В предыдущей главе было показано, что этот алгоритм дает весьма точные результаты за меньшее время при значительном уменьшении размера популяции, по крайней мере, при использовании специальной процедуры инициализации (основанной на идеях метода изменяющихся вероятностей - МИВЕР). В отличие от  $p$ -медианной задачи на сети, в случае непрерывной задачи вычислительные эксперименты не выявили какой-либо взаимосвязи между размером задачи (параметрами  $N$  и  $p$ ) и размером популяции, при котором достигается наискорейшая сходимость. Как правило, оптимальные значения  $N_{pop}$  – выше для задач большого размера по сравнению с малыми задачами наподобие классической задачи Iris. Тем не менее, значения  $N_{pop}$ , при которых достигаются наилучшие результаты при фиксированном времени счета, находятся в пределах  $\overline{\{10,27\}}$  для всех исследованных задач ( $N \in \overline{\{150,169309\}}$ ,  $p \in \overline{\{3,100\}}$ ), как для исходного Алгоритма 3.5 в сочетании с Алгоритмом 3.6, так и для нашей модификации.

Использование популяций меньшего размера снижает точность: алгоритм вырождается, генерируя повторяющиеся множества центров, большие размеры популяции лишь увеличивают время достижения лучшего по точности решения. Во всех описанных ниже экспериментах мы использовали популяции из  $N_{pop} = 20$  решений-«особей». Результаты вычислительных экспериментов с различными размерами популяций на 2-мерном наборе данных, сгенерированном автоматически с параметрами  $N = 10000$ ,  $p = 100$ , показаны на рис. 3.1. Даны усредненные результаты 20 запусков.

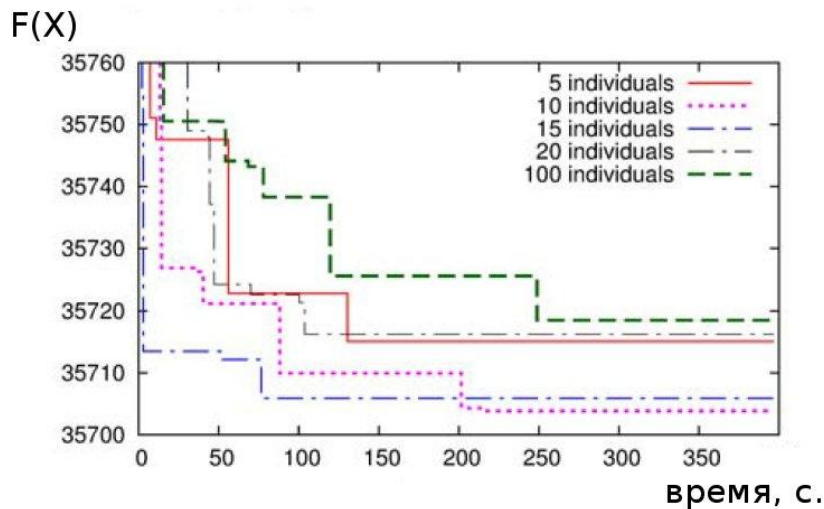


Рисунок 3.1 – Сравнение результатов модифицированного ГА с жадной эвристикой (частичное объединенное решение) для  $p$ -медианной задачи при различных размерах популяции ГА (указано различное число «особей» - “individuals”). Усредненные результаты 20 запусков. Параметры задачи:  $d=2$ ,  $N=10000$ ,  $p=100$

Эксперименты были проведены на вычислительной системе с ЦПУ Xeon 5650 2.76 GHz, 12 Gb ОЗУ, технология HyperThreading отключена. Использован компилятор GNU Fortran.

Вычислительные эксперименты были организованы следующим образом. Алгоритм 3.5 в сочетании с исходной версией Алгоритма 3.6 в качестве оператора кроссинговера запускался 20 раз с лимитом времени в качестве условия останова. Наилучшие значения целевой функции достигнутые при каждом запуске алгоритма, усреднялись для 20 запусков. После этого исходный алгоритм и его модифицированная версия запускалась еще по 20 раз, при этом в качестве условия останова задавалось достижение значения усредненного минимума целевой функции, достигнутого первоначальными двадцатью запусками исходного алгоритма. Кроме того, для сравнения выполнялся многократный запуск (мультистарт) процедуры k-means++ для

тех же задач.

Для всех задач с  $N > 1000$  и  $p > 30$  наша модификация Алгоритма 3.6 и 3.7 оказалась быстрее, чем исходный алгоритм с жадной эвристикой [204, 199] на начальных этапах. Таким образом, применение модификации оправдано при достаточно больших значениях  $p$ , если требуется получить быстрый результат. На рис.3.2 и 3.3 показаны усредненные результаты, достигнутые нашим алгоритмом и исходным алгоритмом, зафиксированные после каждой итерации (скрещивания). Следует, однако, отметить, что применение данной модификации во многих случаях снижает точность счета и повышает разброс результатов алгоритма при увеличении времени счета.

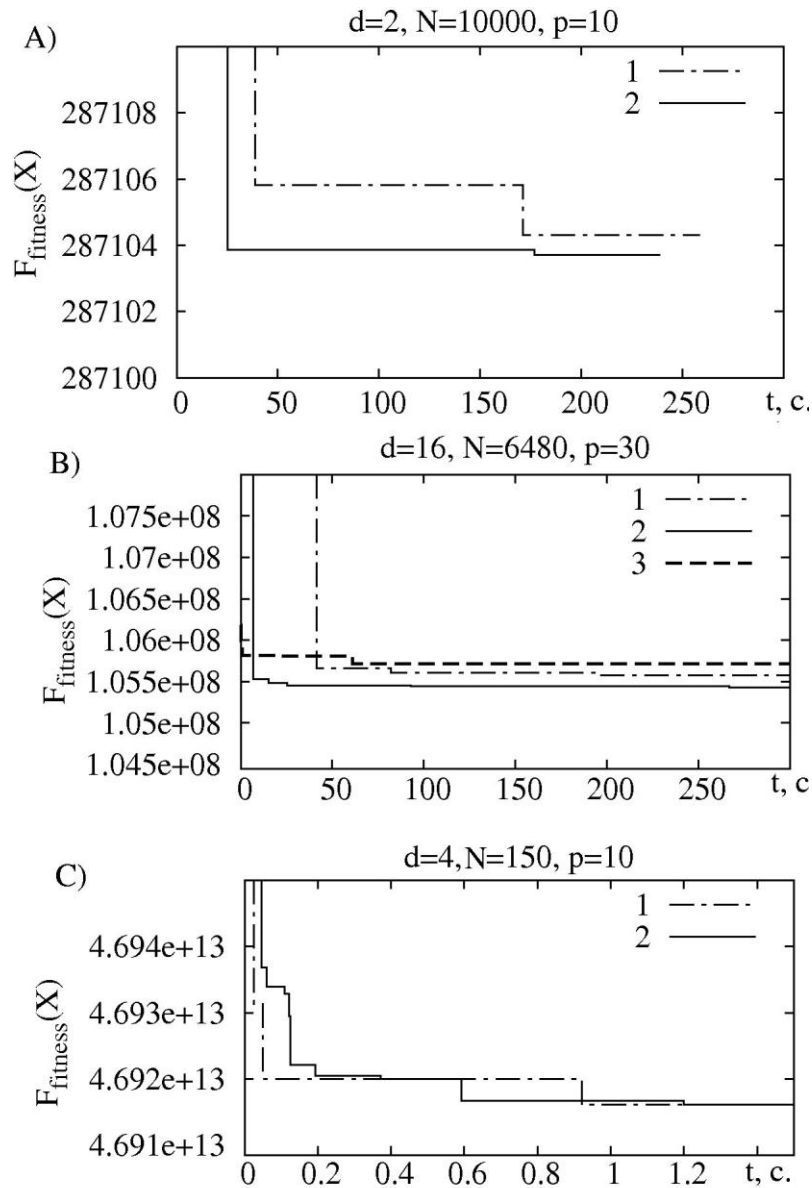


Рисунок 3.2 – Сравнение результатов исходного и модифицированного генетических алгоритмов с жадной эвристикой для задачи  $k$ -средних. 1 – исходный ГА с жадной эвристикой, 2 – модифицированный алгоритм (Модификация 2 жадной эвристики), 3 – процедура  $k$ -means++ с мультистартом

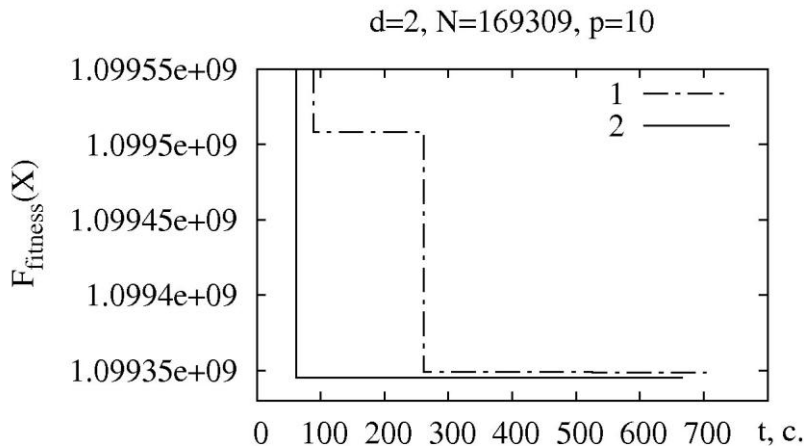


Рисунок 3.3 – Сравнение результатов исходного и модифицированного ГА с жадной эвристикой для непрерывной  $p$ -медианной задачи. Набор данных Europe. 1 – исходный генетический алгоритм с жадной эвристикой, 2 – модифицированный алгоритм (Модификация 2 жадной эвристики)

### 3.4 Дальнейшие модификации жадной эвристики

В настоящем исследовании предлагаются две дальнейшие модификации жадной эвристики. Одна из них уменьшает вычислительную сложность алгоритма, вторая увеличивает его точность. Их комбинация дает новый алгоритм, более быстрый и применимый для задач с большими наборами данных.

Рассмотрим шаги 3 и 4 Алгоритмов 3.6 и 3.7. На каждой итерации на Шаге 3 выбирается один из векторов данных и исключается из промежуточного решения. Будем считать, что на некоторой  $k$ -й итерации удаляется  $j^*$ -й индекс и на  $(k + 1)$ -ой итерации алгоритм удаляет  $j^{**}$ -й индекс. Первая модификация основана на предположении, что если  $A_{j^*}$  далека от  $A_{j^{**}}$  (т.е.  $L(A_{j^*}, A_{j^{**}}) > L_{min}$ , где  $L_{min}$  некая постоянная величина), то удаление или сохранение  $j^{**}$ -го индекса «почти» не зависит от удаления или сохранения  $j^*$ -го индекса на предыдущей итерации.

Если факты выбора индексов двух удаляемых векторов данных на шаге 3 в двух последовательных итерациях являются независимыми друг от друга, то решения об их удалении из решения (сохранении в решении) могут быть сделаны одновременно, в ходе одной итерации алгоритма. Предлагаем следующую модификацию Шагов 3 и 4.

Алгоритм 3.9. Быстрый кроссинговер с жадной эвристикой: модифицированные шаги 3-4 процедуры жадной эвристики – Алгоритма 3.6 (Модификация 3 жадной эвристики).

3: Для каждого  $j \in \chi_c$  вычисляется  $\delta_j = F(\chi_c \setminus \{k\})$ .

4.1: Оценить  $\delta_i$  и выбрать подмножество  $\chi_{elim} = \{e_1, \dots, e_{n_\delta}\} \subset \chi_c$  из  $n_\delta$  индексов с минимальными значениями  $\delta_i$ . Значение  $n_\delta \in \overline{\{1, |\chi_c| - p\}}$  должно вычисляться на каждой итерации. Максимальное количество элементов множества  $\chi_c$  должно быть удалено на первых итерациях и только один – на последних (последние итерации совпадают с Алгоритмами 3.6 или 3.7):

$$n_\delta = \max\{(|\chi_c| - p) * \sigma_e, 1\}. \quad (3.5)$$

Предлагается использовать Алгоритм 3.9 со значением  $\sigma_e = 0.2$ . При наименьшие значения ( $\sigma_e = 0.0$ ) новый алгоритм вырождается в исходный Алгоритм 3.6. Большие значения ( $\sigma_e > 0.3$ ) меняют порядок удаления кластеров и снижают точность алгоритма.

4.2: Из  $\chi_{elim}$ , исключаются близкие друг к другу векторы данных. Для каждой  $j \in \overline{\{2, |\chi_{elim}|\}}$ , если  $\exists k \in \overline{\{1, j-1\}}: L(A_{e_j}, A_{e_k}) < L_{min}$  тогда удаляется  $e_j$  из  $\chi_{elim}$ .

4.3: Удаляем  $\chi_c = \chi_c \setminus \chi_{elim}$ .

Алгоритм 3.6 требует выполнения до  $p$  итераций. Для реальных больших наборов данных, вычислительные эксперименты показывают, что в большинстве случаев при  $N \gg p$  выполняются  $p$  или  $(p-1)$  итерации (векторы данных «родительских» решений на Шаге 3 Алгоритма 3.5 не совпадают). На каждой итерации запускается ALA-процедура  $|\chi_c|$  раз. Таким образом, ALA алгоритм запускается до  $2p + (2p-1) + \dots + 1 = 2p^2 - p + 1$  раз.

Популярные тестовые наборы данных BIRCH 1–3 были разработаны для тестирования алгоритмов при числе кластеров, равном 100. Таким образом, ALA-процедура должна быть запущена до 19901 раз.

В зависимости от параметра  $L_{min}$ , на каждой итерации алгоритм 3.9 удаляет до  $\sigma_e \cdot p$  элементов из  $\chi_c$ . Если  $L_{min}$  большое и  $\sigma_e = 0.2$ , на первой итерации ALA запускается  $2p$  раз, на второй итерации  $[1.8p]$  раз, затем  $[1.64p]$ ,  $[1.512p]$  раз и т.д. В случае 100 кластеров и большого  $L_{min}$ , ALA процедура запускается только  $200 + 180 + 164 + 152 + 142 + 134 + 128 + 123 + 118 + 116 + 113 + 111 + 109 + 108 + 107 + 106 + 105 + 104 + 103 + 102 + 101 = 2626$  раз. Учитывая вычислительную сложность или ALA процедуру такую как стандартный алгоритм -средних, который оценивается

как  $O(N^{34}p^34\log^4(N)/\sigma^6)$  в случае задачи на плоскости с независимо друг от друга сгенерированными векторами данных с нормальным распределением с дисперсией  $\sigma^2$  [327], сокращение количества запусков процедуры локального поиска является решающим при решении больших задач.

Шаг 3 Алгоритма 3.7 может быть модифицирован следующим образом:

Алгоритм 3.10. Быстрый кроссинговер с жадной эвристикой для создания начального решения ALA-процедуры: модифицированные шаги процедуры жадной эвристики – Алгоритма 3.7 (Модификация 3 жадной эвристики в комбинации с Модификацией 1).

3: Для каждого  $j \in \chi_c$ , вычисляется  $\delta_j = \sum_{i=1}^N (\min_{j \in (\chi_c \setminus \{k\})} w_i L(A_i, A_j))$ .

4.1: Сортируем  $\delta_i$  и выбираем подмножество  $\chi_{elim} = \{e_1, \dots, e_{n_\delta}\} \subset \chi_c$  индексов  $n_\delta$  с минимальным значением  $\delta_i$ .

4.2: Для каждого  $j \in \{2, |\chi_{elim}|\}$ , если  $\exists k \in \{1, j-1\}: L(A_{e_j}, A_{e_k}) < L_{min}$ , то удаляем  $e_j$  из  $\chi_{elim}$ .

4.3: Множество  $\chi_c = \chi_c \setminus \chi_{elim}$ .

Цель Шага 4.2 Алгоритма 3.9 – упорядочение удаления кластеров, представленные Алгоритмами 3.6 или 3.7. На рис.3.4 показаны два примера запуска Алгоритма 3.9. Будем считать, что  $p = 4$  и расстояние между центрами кластеров 1 и 3, 3 и 4, 1 и 4, 6 и 7 меньше чем  $L_{min}$ .

Будем считать, что параметр  $\sigma_e$  позволяет удалять до 3-х кластеров на первой итерации. После Шага 3 Алгоритма 3.9 и сортировки  $\delta_i$  мы имеем набор кластеров 4,3,6,... . Если Шаг 4.2 включен в Алгоритм 3.9, то только один из кластеров 1, 3 и 4 может быть удален на первой итерации (Случай А). Таким образом, только два кластера (4 и 7) удалены на первой итерации. Если мы уберем Шаг 4.2 из алгоритма или присвоим большое значение  $L_{min}$ , тогда одновременное устранение кластеров 3 и 4 допускается (Случай В), что дает худшие значения квадратов суммы расстояний. Если классический Алгоритм 3.7 запускается, он сначала удаляет кластер 4, после кластер 6. На третьей итерации Алгоритм 3.7 удаляет первый кластер, и мы имеем набор кластеров, показанный на рис. 3.4, случай А после двух итераций совпадает с



алгоритмом 3.9.

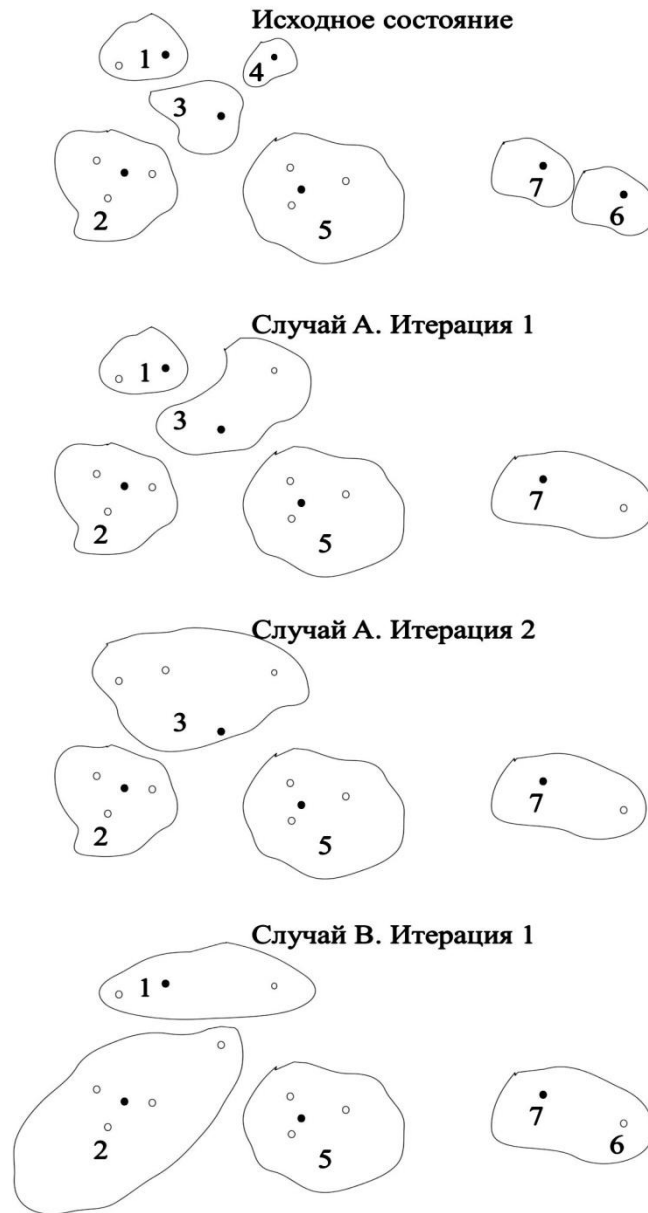


Рисунок 3.4 – Последовательное и одновременное удаление кластеров

Алгоритм 3.6 запускает ALA-процедуру множество раз, это достаточно точный, но медленный метод. Включив Алгоритм 3.9 в Алгоритм 3.6, мы уменьшаем количество запусков ALA-процедуры, однако, как описано выше, по меньшей мере, 2626 запусков алгоритма локального поиска в каждой итерации генетического алгоритма требуется в случае 100 кластеров, и применение этого метода невозможно при очень большом наборе данных, особенно для евклидовой метрики и других метрик и мер расстояния, требующих для определения центра группы векторов данных (кластера) значительных

вычислительных затрат.

Алгоритм 3.7 оптимизирует целевую функцию, вычисленную для начальных данных ALA-процедуры. Данный подход является очень быстрым, но оптимальное значение целевой функции для начальных данных ALA-процедуры не гарантирует ее оптимального значения в итоге ALA процедуры.

Здесь предлагается компромиссный вариант двух алгоритмов, который реализует один шаг ALA-процедуры после каждого удаления кластеров. Поскольку результат ALA-процедуры в общем случае не является подмножеством множества векторов данных  $\{A_i\}$ , использовать целые числа как алфавит ГА (т.е. для кодирования решений, образующих популяцию ГА) невозможно, и мы переходим к использованию векторов в пространстве  $R^d$  (координат элементов промежуточных решений ALA-процедуры) для кодирования решений в популяции ГА. Предлагается следующий алгоритм.

Алгоритм 3.11. ГА с жадной эвристикой и вещественным алфавитом.

Дано: Множество  $V = (A_1, \dots, A_N) \in \mathbb{R}^d$ , число кластеров  $p$ , размер популяции  $N_p$ .

1: Создать  $N_p$  множества с координатами  $\chi_1, \dots, \chi_{N_p}$ :  $\chi_i \subset \mathbb{R}^d$ ,  $|\chi_k| = p \forall k = \overline{1, N_p}$  с решениями ALA алгоритмом полученными процедурой  $k$ -means++ (Алгоритм 3.2). Таким образом, каждый  $\chi_i$  является локальным минимумом (3.1). Сохранить соответствующие значения функции 2 в массиве  $\mathcal{F}_1, \dots, \mathcal{F}_{N_p}$ .

2: Если условия остановки достигаются, тогда переходим к Шагу 7.

3: Случайно выбираем два «родительских» множества  $\chi_{k_1}$  и  $\chi_{k_2}$ ,  $k_1, k_2 \in \{\overline{1, N_p}\}$ ,  $k_1 \neq k_2$ . Запуская Алгоритм 3.12, получаем "дочерние" множества координат  $\chi_c$ , которые являются локальным минимумом (3.1). Сохраняем значение (3.1) в  $\mathcal{F}_c$ .

4: Если  $\exists k \in \{\overline{1, N_p}\}$ :  $\chi_k = \chi_c$  то перейти к шагу 2.

5: Выбрать индекс  $k_{worst} = \operatorname{argmax}_{k=\overline{1, N_p}} \mathcal{F}_k$ . Если  $\mathcal{F}_{worst} < \mathcal{F}_c$  то перейти к Шагу 2.

6: Случайным образом выбрать два индекса  $k_1$  и  $k_2$ ,  $k_1 \neq k_2$ ; массив  $k_{worst} = \operatorname{argmax}_{k \in \{k_1, k_2\}} \mathcal{F}_k$ .

6: Поменять местами  $\chi_{k_{worst}}$  с  $\chi_c$ , множество  $\mathcal{F}_{k_{worst}} = \mathcal{F}_c$  и перейти к Шагу 2.

7: ОСТАНОВ. Результатом является множество  $\chi_k^*$ ,  $k^* = \operatorname{argmin}_{k=1, \overline{Np}} \mathcal{F}_k$ .

Модификация жадной эвристики для этого генетического алгоритма описывается следующим образом.

Алгоритм 3.12. Эвристика жадного кроссинговера с вещественным алфавитом (Модификация 4 жадной агломеративной эвристики).

Дано: Множество  $V = (A_1, \dots, A_N) \in \mathbb{R}^d$ , количество  $p$  кластеров, два "родительских" набора центров или центроид  $\chi_{k_1}$  и  $\chi_{k_2}$ , характеристики  $\sigma_e$  и  $L_{min}$ .

1: Присвоить  $\chi_c = \chi_{k_1} \cup \chi_{k_2}$ . Выполнить ALA-процедуру для  $|\chi_c|$  кластеров, начиная с  $\chi_c$ . Сохранить результат в  $\chi_c$ .

2: Если  $|\chi_c| = p$ , то использовать ALA процедуру с начальным решением  $\chi_c$ , затем ОСТАНОВ и позврат результата.

2.1: Подсчитать расстояния от каждого вектора данных до ближайшего элемента  $\chi_c$ .

$$d_i = \min_{X \in \chi_c} L(X, A_i) \forall i = \overline{1, N}.$$

Определить каждый вектор данных к соответствующему центру кластера из элемента  $\chi_c$ .

$$C_i = \operatorname{argmin}_{X \in \chi_c} L(X, A_i) \forall i = \overline{1, N}.$$

Посчитать расстояния от каждого вектора данных до второго ближайшего элемента  $\chi_c$ .

$$D_i = \min_{Y \in (\chi_c \setminus \{C_i\})} L(Y, A_i).$$

3: Для каждого  $X \in \chi_c$  вычислить  $\delta_X = F(\chi_c \setminus \{X\}) = \sum_{i: C_i | X} (D_i - d_i)$ .

4.1: Вычислить  $n_\delta$  в соответствии с (3.5). Отсортировать  $\delta_X$  и выбрать подмножество  $\chi_{elim} = \{X_1, \dots, X_{n_\delta}\} \subset \chi_c$  из  $n_\delta$ , с координатами минимальных значений  $\delta_X$ .

4.2: Для каждого  $j \in \overline{2, |\chi_{elim}|}$ , если  $\exists k \in \overline{1, j-1}: L(X_j, X_k) < L_{min}$  то переместить  $X_j$  из  $\chi_{elim}$ .

4.3: Присвоить  $\chi_c = \chi_c \setminus \chi_{elim}$ .

4.4: Перераспределить векторы данных к ближайшим центрам или центроидам.

$$C_i^* = \operatorname{argmin}_{X \in \chi_c} L(X, A_i) \forall i = \overline{1, N}.$$

4.5. Для каждого  $X \in \chi_c$ , если  $\exists i \in \overline{1, N}: C_i = X$  and  $C_i^* \neq X$  то пересчитать центр или центроид  $X^*$  кластера  $C_X^{clust} = \{A_i | C_i^* = X, i = \overline{1, N}\}$ . Множество  $\chi_c = (\chi_c \setminus \{X^*\}) \cup \{X\}$ .

5: Перейти к Шагу 2.

Важным параметром Алгоритмов 3.9 и 3.12 является значение  $L_{min}$ . После проведения серии экспериментов на различных данных предложен следующий метод для его определения каждой пары из центров или центроидов  $X_j$  и  $X_k$  (см. Шаг 4.2 Алгоритма 3.12):

$$L_{min} = \min_{X \in \chi_c} \{\max\{L(X, X_j), L(X, X_k)\}\}.$$

Отметим, что данная модификация имеет следующие отличительные особенности. Во-первых, используется кодирование вещественным алфавитом, что позволяет в качестве промежуточных решений использовать конечный либо промежуточный результат АЛА-процедуры или иной процедуры, позволяющей получить локальный минимум в качестве результата. Во-вторых, АЛА-процедура на каждой итерации алгоритма (в отличие от Модификации 1) не выполняется целиком – вместо этого выполняются Шаги 2.1 и 4.4, которые фактически реализуют одну итерацию АЛА-процедуры – распределение векторов данных по кластерам и перерасчет центра каждого кластера – соответственно. В-третьих, для оценки возможности исключения элемента решения (центра, центроида) из решения используется критерий (см. Шаг 2.1), аналогичный критерию силуэта [328]. За исключением первой особенности (вещественный алфавит), остальные две могут эффективно применяться и для дискретных задач. В следующей главе продемонстрирована эффективность алгоритмов с использованием предлагаемых эвристик при решении практических задач автоматической группировки электрорадиоизделий как в дискретной, так и в непрерывной постановке.

Эффективность предложенной модификации генетического алгоритма с жадной эвристикой для непрерывных задач доказывается экспериментально запуском на задачах с большим объемом входных данных.

### 3.5 Вычислительные эксперименты с модификациями жадных эвристик в составе генетического алгоритма

Для тестирования были использованы реальные данные и сгенерированные наборы данных, собранные отделом обработки речи и изображений вычислительной Школы Университета Восточной Финляндии и репозиторий машинного обучения UCI.

Многие исследователи используют подобную задачу в своих экспериментах [103, 308, 312]. Число векторов данных  $N$  варьируются от 150 (классическая задача Iris) до 581013 (набор данных Cover type), число измерений  $d$  варьируется от 2 до 54, число кластеров – от 3 до 1000. Кроме того, мы использовали специально сгенерированный набор данных для  $k$ -медианной задачи (равномерно распределенные векторы данных в  $\mathbb{R}^2$ , каждая координата расположена в интервале  $[0; 10)$  с равномерно распределенными весовыми коэффициентами в интервале  $[0; 10)$ ).

Вычислительные эксперименты были применены для задач с евклидовыми ( $l_2$ ) и квадратичными евклидовыми ( $l_2^2$ ) расстояниями.

В нашем эксперименте мы использовали компьютер Depo X8Sti (6-ядерный CPU Xeon X5650 2.67 GHz, 12Gb RAM), технология hyperthreading отключена и компилятор ifort с полной оптимизацией и неявным параллелизмом (опция -O3).

С целью сравнения алгоритмов мы запускали каждый алгоритм с каждым набором данных 30 раз.

Запуская Алгоритм 3.11 для сгенерированных наборов данных ( $d = 2$ ,  $N = 1000$  либо  $N = 10000$ ,  $p = 10$  либо  $p = 100$ ) и реальных (MissAmerica1 with  $p = 100$ ), видим, что при большой популяции ( $N_p > 50$ ) сходимость замедляется без видимого увеличения точности, а при очень малой популяции ( $N_p < 10$ ), снижется точность.

Для всех наборов данных мы использовали генетический алгоритм с жадной эвристикой (Алгоритм 3.5) с различными эвристиками процедуры кроссинговера (Шаг 3 Алгоритма 3.5): его оригинальная версия, предложенная Альпом и др. [204, 199] (Алгоритм 3.6), его версия только для задания начальных центров кластеров для ALA-процедуры (алгоритм 3.7), наши модификации, позволяющие устранить многие центры кластеров в один шаг (алгоритм 3.9) и новый генетический алгоритм с вещественным алфавитом (алгоритм 3.12).

Результаты для каждого набора данных представлены в Таблицах 3.1 и 3.2.

Таблица 3.1 – Результаты работы нового Алгоритма 3.12 и генетического алгоритма с процедурой кроссинговера в виде жадной эвристики.

Наборы данных и их параметры, время	Расстояние	Метод	Средний результат	Среднее время, необходимое для достижения результата оригинальным методом, сек	Худший результат	Среднее увеличение быстродействия (новый метод в сравнении с исходным)
Iris ( $n = 150, d = 4, p = 3$ ), sec.	$l_2^2$	Original	$1.40026039044 \cdot 10^{14}$	0.0096	$1.40026039044 \cdot 10^{14}$	
		ALA mult.	$1.40026039044 \cdot 10^{14}$	0.0103	$1.40026039044 \cdot 10^{14}$	
		New	$1.400262 \cdot 10^{14}$	-	$1.4002858 \cdot 10^{14}$	-
Iris ( $n = 150, d = 4, p=10$ ), sec.	$l_2^2$	Original	46916083985700	2.4	46916083985700	
		ALA mult.	46917584582154	-	46935815209300	
		New	46916083985700	2.5	46916083985700	-
MissAmerica1 ( $n = 6480, d=16, p = 30$ ), sec.	$l_2^2$	Original	105571815.061	603	105663081.95	
		ALA mult.	105714622.427	-	106178506.965	
		New	105440299.602	13.8	105440299.601	43.69
Europe ( $n = 169309, d = 2, p=10$ ), sec.	$l_2^2$	Original	1099348562.46	1050.8	1099355026.03	
		ALA mult.	1099345009.09	15.6	1099345033.08	
		New	1099345067.99	123.8	1099345210.55	8.48

Примечание: "Original" - Алгоритм 3.5 с жадной эвристикой (Алгоритм 3.6),  
 "ALA mult." – алгоритм мультистарт ALA процедуры (Алгоритм 3.4),  
 "New"- генетический алгоритм с вещественным алфавитом (Алгоритм 3.11 с Алгоритмом 3.12 процедурой жадного кроссинговера).

Мы использовали процедуру сэмпинга (Алгоритм 3.3) для наборов данных с  $N \geq 10000$  на Шаг 1 Алгоритмов 3.5, 3.11 и 3.12. Для небольших наборов данных применяются все алгоритмы без сэмпинга. Использование алгоритмов без сэмпинга для больших наборов данных в равной степени замедляет генетический алгоритм со всеми рассмотренными модификациями жадных эвристик в качестве процедур кроссинговера. Каждый алгоритм запускался 30 раз. Ограничение по времени, показанное в первом столбце таблиц, используется в качестве условия останова. Значение этого максимального времени работы было выбрано таким образом, чтобы двукратное увеличение этого времени не позволяло получить лучшие результаты в случае использования первоначальной жадной эвристики (Алгоритма 3.7) в качестве процедуры кроссинговера для инициализации ALA-процедуры по меньшей мере в 27 попытках из 30. Кроме того, для решения задач, представленных в таблице 3.1, мы зафиксировали среднее время, необходимое для достижения усредненного результата

работы оригинального генетического алгоритма с жадной эвристикой в качестве процедуры кроссинговера (алгоритм 3.5 + Алгоритм 3.6, см [204, 199]).

Таблица 3.2 – Результаты работы Алгоритма 3.12 и генетического алгоритма с жадной эвристикой процедурой, использованных для задания начальных значений ALA-процедуры

Наборы данных и их параметры, время	Расстояние	Метод	Средний результат	Среднее время, необходимое для достижения результата оригинальным методом, сек	Худший результат	Среднее увеличение быстродействия (новый метод в сравнении с исходным)
Europe ( $n = 169309$ , $d = 2$ , $p = 100$ ), sec.	$l_2$	Orig. init.	400370576	397.6	400904292	
		ALA mult.	400755480	-	401007437	-
		New	400345100	193.6	400595350	2.05
Generic ( $N = 10000$ , $d = 2$ , $p = 100$ ), sec.	$l_2$	Orig. init.	85318.44	47.65	85482.67	
		ALA mult.	85374.62	-	86114.83	-
		New	85167.01	0.895	85179.72	53.24
Europe ( $N = 169309$ , $d = 2$ , $p = 100$ ), sec.	$l_2^2$	Orig. init.	$2.2767 \cdot 10^{12}$	557.6	$2.2933 \cdot 10^{12}$	
		ALA mult.	$2.3039 \cdot 10^{12}$	-	$2.3111 \cdot 10^{12}$	-
		New	$2.2752 \cdot 10^{12}$	143.1	$2.2862 \cdot 10^{12}$	3.89
BIRCH1 ( $N = 100000$ , $d = 2$ , $p = 100$ ), sec.	$l_2^2$	Orig. init.	$9.277283 \cdot 10^{13}$	46.08	$9.277287 \cdot 10^{13}$	
		ALA mult.	$9.746921 \cdot 10^{13}$	-	$9.276386 \cdot 10^{13}$	-
		New	$9.277282 \cdot 10^{13}$	31.56	$9.274231 \cdot 10^{13}$	1.46
BIRCH3 ( $N = 100000$ , $d = 2$ , $p = 1000$ ), sec.	$l_2^2$	Orig. init.	$4.08652 \cdot 10^{12}$	802.8	$4.105231 \cdot 10^{12}$	
		ALA Mult.	$4.16053 \cdot 10^{12}$	-	$4.162683 \cdot 10^{12}$	-
		New	$4.02040 \cdot 10^{12}$	8.81	$4.022190 \cdot 10^{12}$	91.12
MissAmerica2 ( $N = 6480$ , $d = 16$ , $p = 100$ ), sec.	$l_2^2$	Orig. init.	80264286	85.8	81039148	
		ALA Mult.	81869326	-	82316364	-
		New	79837119	0.752	80147971	114.1
CoverType ( $N = 581013$ , $d = 54$ , $p = 100$ ), sec.	$l_2^2$	Orig. init.	3122934.7	905.4	3146107.4	
		ALA Mult.	3163271.9	-	3182076.8	-
		New	3069213.6	53.1	3072299.5	17.05

Примечание: "Orig. init."- Алгоритм 3.5 с жадной эвристикой процедурой кроссинговера (Алгоритм 3.7),

"ALA mult." – мультистарт ALA процедуры (Алгоритм 3.4),

"New" генетический алгоритм с вещественным алфавитом (Алгоритм 3.11 с Алгоритмом 3.12 процедурой жадного кроссинговера).

Для более сложных задач, представленных в Таблице 3.1, где оригинальная процедура жадного кроссингвера применяется для решения задачи  $k$ -медоид с дальнейшим уточнением ALA-процедурой на завершающем шаге из-за огромной вычислительной сложности, мы зафиксировали среднее время, необходимое для достижения среднего результата работы оригинального генетического алгоритма с жадной эвристикой в качестве процедуры кроссингвера, применяемого для оптимизации значения целевой функции начального набора данных ALA процедуры (Алгоритм 3.5 + Алгоритм 3.7). Алгоритм 3.5 с оригинальной жадной эвристикой (см. Алгоритм 3.6), предложенный Альпом и др. [204, 199] показывает отличные результаты для сравнительно небольших наборов данных (см. Таблицу 3.1). Для менее сложных задач (набора данных "Iris"), алгоритм, предложенный в настоящем параграфе (Алгоритм 3.11, задача 1 и 3 в Таблице 3.1), снижает точность решения по сравнению с оригинальным алгоритмом Альпа и др. [204, 199] (Алгоритмы 3.5 и 3.6). Для больших наборов данных новый алгоритм (Алгоритм 3.11 + Алгоритм 3.12) работает быстрее и точнее.

Кроме того, использование оригинальной жадной эвристики в качестве оператора кроссингвера невозможно для больших наборов данных (для всех больших наборов данных с  $p > 30$ ,  $N \geq 10000$ ) из-за громоздких вычислений на каждой итерации. В случае таких наборов данных для сравнения мы использовали алгоритм Альпа и др., применяемый для оптимизации значения целевой функции начального набора данных ALA-процедуры (Алгоритмы 3.5 и 3.7).

В этом случае для всех решаемых задач большой размерности с евклидовой ( $l_2$ ,  $p$ -медианная задача на плоскости) и квадратичной евклидовой ( $l_2^2$ , задача -средних) мерами, наш Алгоритм 3.11 с вещественным алфавитом и модифицированной эвристикой жадного скрещивания (Алгоритм 3.12) работает быстрее и дает более точные результаты, чем Алгоритм 3.5 с жадной эвристикой, реализованный только для выбора подмножества векторов данных в качестве начального решения ALA-процедуры (Алгоритм 3.7, см. [204, 199]).

Для иллюстрации динамики процесса решения мы представляем временные диаграммы, которые показывают средние результаты 30 запусков каждого алгоритма для различных наборов данных на рис. 3.5 и 3.6. Диаграммы показывают, что новый алгоритм с вещественным алфавитом позволяет повысить точность на ранних стадиях



вычислительного процесса по сравнению с другими известными методами, что позволяет использовать его для получения быстрых приближенных решений. Кроме того, результаты быстрой жадной эвристики (Алгоритм 3.9) показаны на диаграммах. Использование этой эвристики без других модификаций генетического алгоритма может уменьшить точность результатов.

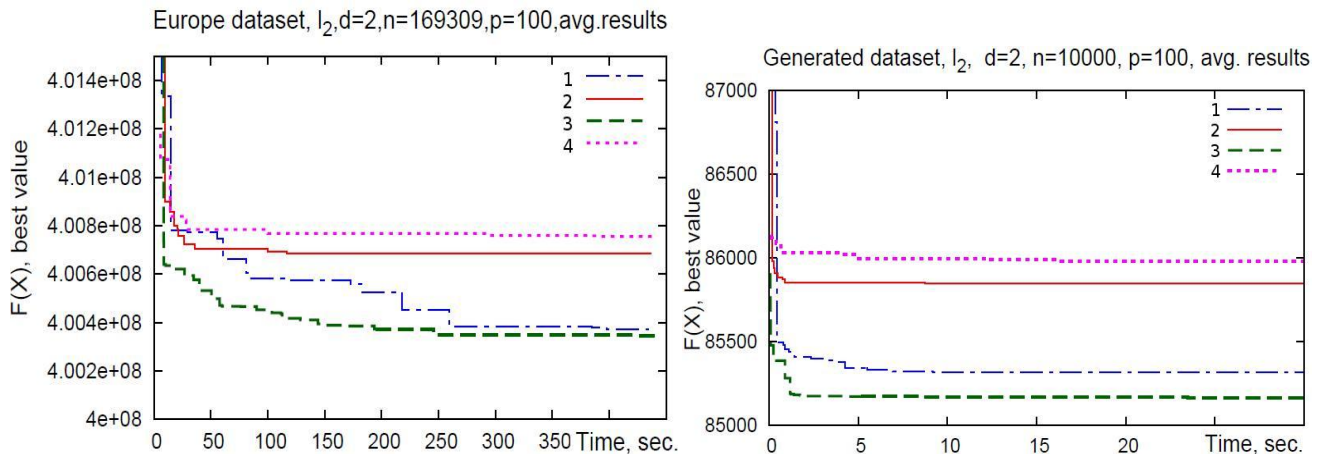


Рисунок 3.5 – Усредненный результат нового ГА в сравнении с другими алгоритмами для р-медианной задачи.

- 1 – использование ГА для генерации начальных значений ALA-процедуры (Алг.3.5+Алг.3.7),  
 2 – модиф. ГА для генерации начальных значений ALA-процедуры (Алг.3.5+Алг.3.10),  
 3 – новый ГА с вещественным алфавитом (Алгоритм 3.11+Алгоритм 3.12),  
 4 – мультистарт стандартной процедуры k-средних (ALA)

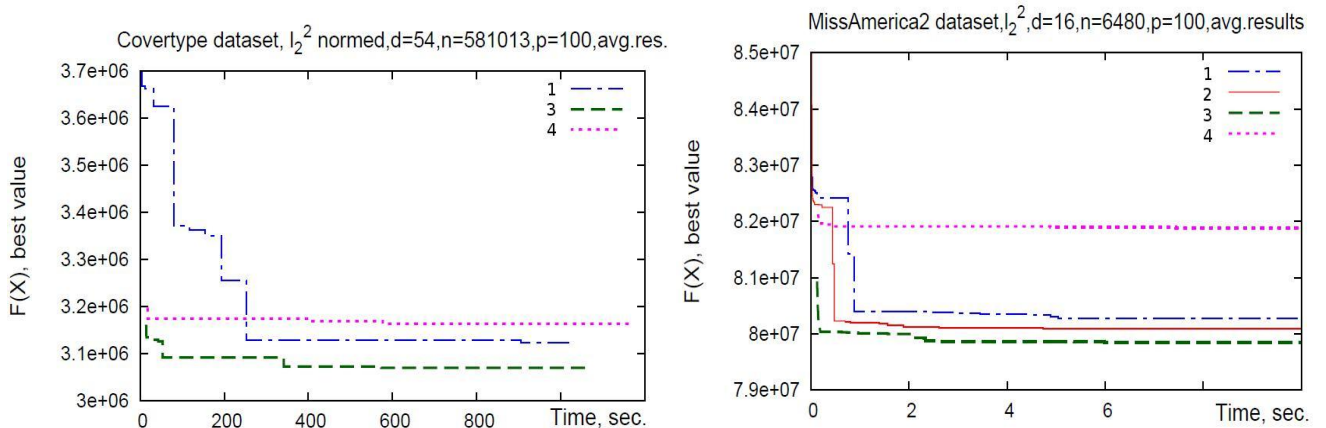


Рисунок 3.6 – Усредненный результат нового ГА в сравнении с другими алгоритмами для задачи k-средних. 1 – использование ГА для генерации начальных значений ALA-процедуры (Алг.3.5+Алг.3.7), 2 – модиф. ГА для генерации начальных значений ALA-процедуры (Алг.3.5+Алг.3.10), 3 – новый ГА с вещественным алфавитом (Алгоритм 3.11+Алгоритм 3.12), 4 – мультистарт стандартной процедуры k-средних (ALA)

Новый генетический алгоритм, основанный на идеях генетического алгоритма с жадной эвристикой для k-медианной задачи на сети с предложенными выше модификациями могут быть использованы для быстрого и достаточно точного решения

$p$ -медианных задач большой размерности и задачи  $k$ -средних. Для небольших задач ( $p < 10$ ,  $N < 500$ ) результаты нашего метода менее точны, чем результаты генетического алгоритма с жадной эвристикой, предложенной Альпом и др., при одинаковых временных затратах. Однако новый алгоритм, как правило, обеспечивает более точные результаты за приемлемое время для больших задач ( $p > 30$ ,  $N > 1000$ ).

### 3.6 Применение новых модификаций жадных эвристических процедур к дискретным задачам размещения

В задачах автоматической группировки данных недостатком метода  $k$ -средних и, соответственно, решения задачи  $k$ -средних, которую мы считаем вариантом обобщенной  $p$ -медианной задачи с квадратичной евклидовой мерой расстояния  $L(X, A_i) = \sum_{k=1}^d (x_{j,k} - a_{i,k})^2$ , является то, что координаты центроидов могут принимать произвольные значения в пространстве решений, благодаря чему точность, с которой получены значения центров, превышает точность исходных данных, которая может быть ограничена точностью измерительных приборов либо фактически дискретным характером измеряемых величин. Таким образом, полученное решение – трудно интерпретируемое. Теми же недостатками обладает и решение «традиционной» непрерывной  $p$ -медианной задачи с евклидовой метрикой (к этому добавляется еще и повышенная вычислительная сложность) – координаты центров (медиан) в общем случае не совпадают с координатами векторов данных.

Данная проблема частично устраняется использованием прямоугольной метрики,  $L(X, A_i) = \sum_{k=1}^d |x_{j,k} - a_{i,k}|$ . Решением данной задачи является множество центров, таких, что значения каждой координаты каждого из этих центров совпадает с медианным значением данной координаты [243], следовательно, совпадает со значением данной координаты одного из векторов исходных данных. В этом случае, однако, не гарантируется, что все координаты центра совпадут со значениями координат какого-либо одного вектора данных. Кроме того, кластеры в этой метрике, как правило, менее компактны, чем в классической для задачи  $k$ -средних квадратичной евклидовой метрике.

Другим подходом к повышению интерпретируемости результатов является переход к задаче  $k$ -медоид [82] со следующей постановкой:

$$\arg \min_{x_1, \dots, x_p \in \{A_1, \dots, A_N\}} \sum_{i=1}^N w_i \min_{j \in \{1, p\}} L(X_j, A_i).$$

Здесь в качестве  $L()$  может быть использована любая метрика или мера расстояния.

Иными словами, данная задача эквивалентна непрерывной  $p$ -медианной задаче (3.1) с дополнительным условием:  $x_1, \dots, x_p \in \{A_1, \dots, A_N\}$ .

Отметим, что данное условие соответствует условию (3.4), при выполнении которого работает Алгоритм 3.7 – жадная эвристика для поиска начального решения ALA-процедуры. Таким образом, один из вариантов применения жадной эвристики к задаче  $k$ -медоид нами уже был рассмотрен выше.

Собственно медоидами мы будем называть искомые точки  $X_1, \dots, X_p$ . Иногда [329, 329] такую задачу называют дискретной  $p$ -медианной задачей, в таком случае искомые точки называют медианами или центроидами.

Отметим, что задача  $k$ -медоид является дискретной задачей размещения: число возможных местоположений искомым точек является конечным.

В качестве средства локального поиска для задачи  $k$ -медоид может также быть использована ALA-процедура [297] либо более быстрые процедуры [248, 330, 82], не гарантирующие точного нахождения центра каждого из формируемых кластеров, например – аналогичный PAM-алгоритм [82]. Оба алгоритма (ALA- и PAM-процедуры) являются процедурами локального поиска. Оба начинаются с указания некоторого начального решения — начального множества центров, в качестве которого используется подмножество из  $p$  векторов данных. Оба состоят в поочередном нахождении подмножеств векторов данных для каждого из центров, для которых данный центр является ближайшим, и нахождении нового центра для каждого из этих множеств.

Предположить применимость и эффективность алгоритмов с жадной эвристикой к задаче  $k$ -медоид можно, как минимум, по трем причинам [331]:

а) задача  $k$ -медоид решается Алгоритмом 3.5 (ГА) с Алгоритмом 3.7 в качестве процедуры кроссинговера, полученное решение (множество медоидов) используется как начальное решение для ALA-процедуры при решении  $p$ -медианной задачи и задачи  $k$ -средних, и такой подход достаточно эффективен в случае больших задач;

б) задача  $k$ -медоид является дискретной задачей с постановкой, аналогичной  $p$ -

медианной задаче, а дискретная  $p$ -медианная задача на сетях эффективно решается генетическим алгоритмом с жадной эвристикой;

в) постановка  $k$ -медоидной задачи в целом аналогична  $p$ -медианной задаче, для решения обеих задач могут использоваться одни и те же либо аналогичные процедуры локального поиска, в обоих случаях начальными решениями процедур локального поиска могут являться подмножества векторов данных мощности  $p$ , следовательно, одни и те же способы рекомбинации этих подмножеств могут быть применены, например, с использованием генетического Алгоритма 3.5.

Поскольку в случае  $k$ -медоидной задачи элементы решения всегда составляют подмножество векторов данных, решения всегда могут быть закодированы в виде множеств индексов векторов данных. Соответственно, переход к генетическому вещественному алфавиту в генетическом алгоритме (см. Алгоритмы 3.11 и 3.12) является бессмысленным, при этом другие особенности Модификации 4 жадной эвристики (Алгоритм 3.12), а именно – использование отдельных шагов процедуры локального поиска и оценка удаляемых элементов решения специальным критерием – так же, как и в случае непрерывных задач, ускоряет поиск решений, близких к оптимальному.

С учетом модификаций, рассмотренных выше (за исключением генетического алгоритма с вещественным алфавитом – Алгоритмов 3.11 и 3.12), составим общую схему генетического алгоритма (Алгоритм 3.5) с жадной эвристикой (Алгоритмы 3.6 или 3.7 – в зависимости от способа включения локального поиска в схему алгоритма) с учетом возможных модификаций, таких как частичное объединение «родительских» решений-множеств генетического алгоритма (Алгоритм 3.8) и быстрого исключения элементов из промежуточного решения (Алгоритмы 3.9 и 3.10).

Алгоритм 3.13. Генетический алгоритм с жадной эвристикой для  $p$ -медианной и  $k$ -медоидной задачи [204, 317] (в редакции, предложенной в работах [199, 249, 324]).

Дано: Размер популяции  $N_{POP}$ .

1. Сформировать (случайным образом или с использованием процедуры  $k$ -means++)  $N_{POP}$  различных начальных решений  $\chi_1, \dots, \chi_{N_{POP}} \subset \overline{\{1, N\}}$ ,  $|\chi_i| = p \forall i = \overline{1, N_{POP}}$  – множеств индексов векторов данных мощности  $p$ , используемых в качестве исходных решений ALA-алгоритма. Для каждого из начальных решений оценить значение

целевой функции  $F_{fitness}(\chi)$  с использованием Алгоритма 3.14, сохранить значения данной функции в переменных  $f_1, \dots, f_{N_{POP}}$ .

2. Если достигнуты условия останова, то ОСТАНОВ. Решением является начальное решение  $\chi_{i^*}$ , которому соответствует наименьшее значение  $f_i$ . Для нахождения окончательного решения запускается ALA-процедура (Алгоритм 3.1) или РАМ-процедура.

3. Выбрать случайным образом два индекса  $k_1, k_2 \in \overline{\{1, N\}}, k_1 \neq k_2$ .

4. Получить промежуточное решение  $\chi_c = \chi_{k_1} \cup \chi_{k_2}$ , либо запустить Алгоритм 3.8 для получения частичного объединенного решения.

5. Если  $|\chi_c| > p$ , то перейти к Шагу 7:

6. Вычислить  $j^* = \arg \min_{j \in \chi_c} F_{fitness}(\chi_c \setminus \{j\})$ .

Исключить  $j^*$  из  $\chi_c$ :  $\chi_c = \chi_c \setminus \{j^*\}$ . Перейти к Шагу 5.

7. Если  $\exists i \in \overline{\{1, N_{POP}\}}: \chi_i = \chi_c$ , то перейти к Шагу 2.

8. Выбрать индекс  $k_3 \in \overline{\{1, N_{POP}\}}$ . В настоящей работе использована простейшая турнирная селекция: выбираются случайным образом два индекса  $k_4, k_5 \in \overline{\{1, N_{POP}\}}$ , если  $f_{k_4} > f_{k_5}$ , то  $k_3 = k_4$ , иначе  $k_3 = k_5$ . Использование других способов селекции не влияет существенным образом на результат.

9. Заменить  $\chi_{k_3}$  и соответствующее значение целевой функции:  $\chi_{k_3} = \chi_c$ ,  $f_{k_3} = F_{fitness}(\chi_c)$ . Перейти к Шагу 2.

Определение значения целевой функции во всех случаях осуществляется при помощи следующего алгоритма в различных его модификациях:

Алгоритм 3.14. Вычисление значения целевой функции  $F_{fitness}(\chi)$ .

Дано: начальное решение  $\chi = \{X_1, \dots, X_p\}$ .

1. Если локальный поиск встроен в жадную эвристику, то запустить Алгоритм 3.1 или РАМ-процедуру с начальным множеством центров  $\{A_i / i \in \chi\}$ , получить множество центров  $\{X_1, \dots, X_p\}$ .

2. Возвратить значение  $F_{fitness}(\chi) = \sum_{i=1}^N w_i \min_{j \in \{1, p\}} L(X_j, A_i)$ .

Шаг 4 Алгоритма 3.13 порождает промежуточное решение — множество мощности до  $2p$ , из которого последовательно исключаются (Шаг 6) элементы до достижения мощности  $p$ . При этом на каждой такой итерации требуется количество вычислений функции  $F_{fitness}$ , соответствующее текущей мощности множества — промежуточного решения. Таким образом, запуск ALA-алгоритма на Шагах 2-9 Алгоритма 3.13 выполняется до  $p^2$  раз. Вычислительная сложность ALA-алгоритма также зависит от количества центров. Так, вычислительная сложность Шага 2 Алгоритма 3.1 линейно зависит от  $N$  при использовании евклидовой метрики в задаче  $k$ -средних: производятся вычисления (3.1) каждого из кластеров, сложность вычислений зависит от числа точек в кластере  $|C_j^{clust}|$ , итого проводятся  $N$  вычислений (3.1). В случае задачи  $k$ -медоид вычислительная сложность данного шага значительно возрастает. В этом случае требуется определить

$$x_{j,k} = \min_{i \in C_j^{clust}} \sum_{j \in C_j^{clust}} \sum_{k=1}^d \sqrt{(a_i - a_j)^2}.$$

Фактически в случае  $k$ -медоид требуется оценка расстояния между каждой парой векторов данных, входящих в кластер, что повышает вычислительную сложность шага до  $N^2$ . Таким образом, вычислительная сложность всей ALA-процедуры существенно повышается, и сокращение числа ее запусков весьма желательно. Быстрее работает RAM-процедура. Тем не менее, и ее вычислительная сложность зависит от  $p$  и  $N$ .

Алгоритм 3.15. Модификация генетического алгоритма с жадной эвристикой для задачи  $k$ -медоид (шаг 6 Алгоритма 3.13) – Модификация 3 + Модификация 1 жадной эвристики для задачи  $k$ -медоид.

6.1. Для каждого  $j \in \chi_c$  выполнять:

6.1.1. Сформировав множество  $\chi^- = \chi_c \setminus \{j\}$ , для каждого вектора данных  $A_i \in A_1, \dots, A_N$  найти ближайший центр  $C_i^- = \arg \min_{j=1, |\chi^-|} L(A_i, X_j)$ . Сформировать  $|\chi^-|$

множеств векторов данных, для которых каждый из центров является ближайшим:

$C_j^{clust} = \{i \in \{1, N\} / C_i^- = j\}$ ; для каждого кластера  $C_j^{clust}$ ,  $j=1, |\chi_c|$  рассчитать его центр  $X_j^-$ ,

после чего рассчитать  $f_j = \sum_{i=1}^N w_i \min_{k \in \chi^-} L(X_{\bar{k}}, A_i)$ .

6.1.2. Следующая итерация цикла 6.1.

6.2. Упорядочить множество пар  $(j, f_j = \sum_{i=1}^N w_i \min_{k \in \chi^-} L(X_{\bar{k}}, A_i))$  по возрастанию значения  $f_j$ , сформировать подмножество  $E_{elim}$  первых  $N^{elim}$  индексов векторов данных из этого упорядоченного множества. Удалить из множества  $E_{elim}$  такие индексы  $i$ , что  $\exists j \in E_{elim} : \|A_i - A_j\| < L_{min}, j < i$ .

Здесь  $N_{elim} = [0.2 (|\chi_c| - p)] + 1$ . Значение параметра  $L_{min}$  принимается равным 0,1-0,5 значения математического ожидания расстояния между двумя случайно выбранными векторами данных. Мы использовали значение  $L_{min} = 0,2 \mu(L(A_i, A_j))$ .

6.3. Исключить  $E_{elim}$  из  $\chi_c : \chi_c = \chi_c \setminus \{j^*\}$ . Перейти к Шагу 5.

Для целей тестирования мы использовали задачи ORLIB [244] и наборы данных из библиотеки UCI [326], а также автоматически сгенерированные тестовые наборы данных, представляющие собой набор равномерно распределенных случайных точек в квадрате 10x10.

Размерность задач — до 169309 векторов данных ( $N=169309$ , набор данных «Euclidean» из библиотеки UCI), размерности до  $d=32$ , число кластеров – до 1000. Для решения  $p$ -медианной задачи с евклидовой метрикой использовались исключительно наборы данных с  $d=2$ , поскольку такие задачи имеют географическую интерпретацию. Для остальных наборов данных использовались квадратичные евклидовы расстояния.

В ходе вычислительных экспериментов были выявлены следующие особенности. В отличие от задачи  $k$ -средних и непрерывной  $p$ -медианной задачи, при решении которых генетическим алгоритмом с жадной эвристикой [324] требуется очень небольшая популяция «особей»  $N_{POP}$ , порядка 10-25 штук, при решении задачи  $p$ -медоид требуется популяция порядка 50-150 «особей» (в тестах мы использовали 75). Меньший размер популяции снижает точность результата, больший размер увеличивает время получения результата, не увеличивая точность. Тем не менее, и в этом случае, в отличие от  $p$ -медианной задачи на сети [204, 157], при решении генетическим алгоритмом непрерывных задач не прослеживается явная зависимость оптимального числа «особей»  $N_{POP}$  в популяции ни от числа векторов данных  $N$ , ни от количества центров  $p$ .

Использовалась вычислительная система с процессором Xeon X5650 2.67 GHz, 12Gb RAM, HyperThreading отключен, компилятор GNU Fortran.

В результате экспериментов предлагаемая в настоящей работе версия алгоритма позволяет получить результат более высокой точности, что и оригинальный генетический алгоритм с жадной эвристикой [249], причем значительно быстрее. В сравнении с другим широко применяемым генетическим алгоритмом, основанным на рекомбинации смешанных подмножеств [199] (в дальнейшем будем называть данный алгоритм MIX-GA), новый алгоритм дает более точные результаты за меньшее время.

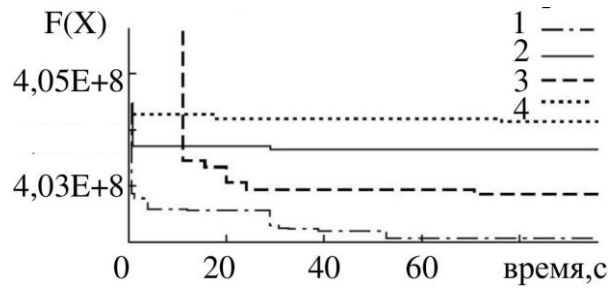
Эксперименты организованы следующим образом. Запускался какой-либо алгоритм, при этом задавалось максимальное время работы алгоритма в качестве условия останова. Среднее значение целевой функции, достигнутое в результате десяти запусков, фиксировалось. В результате экспериментов модифицированный алгоритм показал преимущество перед исходным алгоритмом с жадной эвристикой [249] и алгоритмом MIX-GA на всех тестовых примерах при  $N > 1000$  и  $p > 5$ . На рис.3.7 представлены графики зависимости наилучшего достигнутого результата от затраченного времени. Представлены также результаты, достигнутые многократным запуском процедуры k-means++.

Отметим, что применение Модификации 2 – частичного объединенного решения (см.Алгоритм 3.8) к задачам k-медоид, дает результаты, аналогичные результатам для непрерывных задач: для некоторых классов задач при достаточно больших значениях  $p$  ( $p > 30$ ) наблюдается ускорение сходимости алгоритма по сравнению с ГА с жадной эвристикой алгоритмом без данной модификации. В то же время, идентифицировать такие классы задач по их параметрам ( $N$ ,  $p$ ,  $d$ ) пока не удается – другие задачи с аналогичными параметрами могут решаться с применением данной модификации значительно хуже. Таким образом, необходимость применения Модификации 2 следует определять экспериментальным путем при построении автоматизированной системы группировки путем решения серии практических задач с применением и без применения Модификации 2, либо применяя коэволюционные подходы [197].

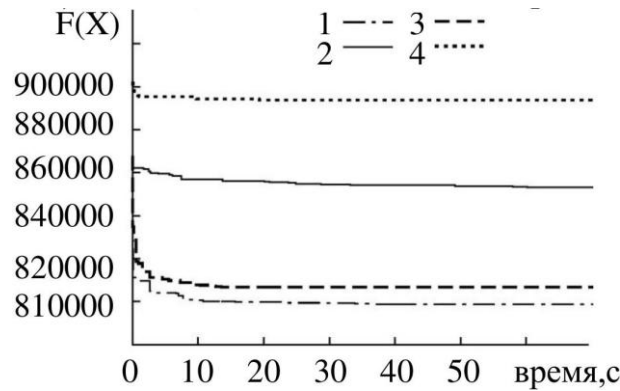
Для  $p$ -медианных задач на сети может быть предложена следующая модификация жадной эвристики (см. Алгоритм 2.3 из предыдущей главы) процедуры рекомбинации «особей» генетического алгоритма, аналогичная Модификации 2 для непрерывных задач.



А)  $N=169309, p=100, d=2$ , мера расстояния  $l_2^2$



В)  $N=6480, p=100, d=16$ , мера расстояния  $l_2^2$



В)  $N=980, p=30, d=120$ , метрика  $l_1$

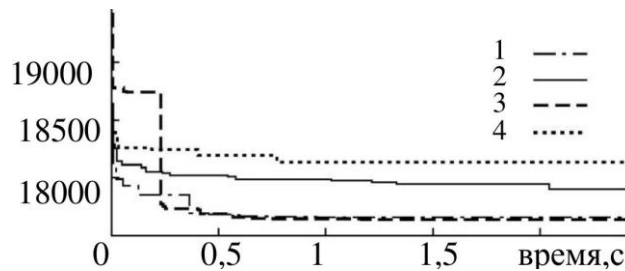


Рисунок 3.7 – Зависимость достигнутого значения целевой функции от затраченного времени для задачи k-средних. Наборы данных: А — Europe, В — MissAmerica1, С — набор данных неразрушающих тестов микросхемы. Алгоритмы: 1 — новый Алгоритм 3.13, 2 — MIX-GA, 3 — генетический с жадной эвристикой [249], 4 — мультистарт процедуры k-means++

Алгоритм 3.16. Процедура кроссинговера для ГА с жадной эвристикой на сетях

Дано: два множества индексов вершин  $M_1 = \{m_{1,1}, \dots, m_{1,p}\}$ ,  $M_2 = \{m_{2,1}, \dots, m_{2,p}\}$ .

1.1: Сгенерировать случайное  $r_{init} \in [0; 1)$  с равномерным распределением;

1.2: Присвоить  $r = 1 + [(p - 1) \cdot r_{init}]$ ;

1.3: Из «родительского» множества вершин  $M_2$  выбрать случайным образом подмножество  $M_2^*$  мощности  $r$ ;

1.4: Сформировать промежуточное решение  $M = M_1 \cup M_2^*$ .

2: Пока  $|M| > p$ , выполнять

- 2.1: присвоить  $f^* = +\infty$ ;
- 2.2: для каждой вершины  $m^*$  из множества  $M$  выполнять
- 2.2.1:  $M^* = M \setminus \{m^*\}$ ;
- 2.2.2:  $f^{**} = f(M^*)$ ;
- 2.2.2: Если  $f^{**} < f^*$ , то  $m^{**} = m^*$ ;
- 2.1.3: следующая итерация 2.2;
- 2.3: Присвоить  $M = M \setminus \{m^{**}\}$ ; следующая итерация 2;
- 3: вернуть новое решение («хромосому»)  $M$ .

Результаты модифицированного генетического алгоритма с жадной эвристикой для р-медианной задачи на сетях показаны в Таблице 3.3 и на рис. 3.8. Ускорение модифицированного алгоритма по сравнению с исходным алгоритмом в случае задач на сетях не столь значительно, как в случае непрерывных задач.

Таблица 3.3 – Сравнение результатов для задач на сетях

Задача	Алгоритм	Среднее время, с.	Средний результат
Сгенерированная (N=2000, p=100)	ЛП с мультистартом	27.5	129440.66
	Исходный ГА	73.76	120031.2
	Исходный ГА + ЛП	4.4	119885.61
	Модиф. ГА	53.52	120258.03
	Модиф. ГА+ЛП	3.4	119865.35
rmed11 (n=300, p=5)	ЛП с мультистартом	0.1	7578
	Исходный ГА	0.7	7578,89
	Исходный ГА + ЛП	0.05	7578
	Модиф. ГА	0.42	7602.11
	Модиф. ГА+ЛП	0.19	7578
rmed13 (N=300, p=30)	ЛП с мультистартом	1.15	4311
	Исходный ГА	13.25	4331
	Исходный ГА + ЛП	0,28	4314.25
	Модиф. ГА	0.52	4386.13
	Модиф. ГА+ЛП	0.24	4311
rmed17 (N=400, p=10)	ЛП с мультистартом	2.41	6980
	Исходный ГА	6.22	6982
	Исходный ГА + ЛП	0.95	6980
	Модиф. ГА	1	7044.6
	Модиф. ГА+ЛП	1.08	6981.2
rmed22 (n=500, p=10)	ЛП с мультистартом	7.09	8464
	Исходный ГА	15.48	8464
	Исходный ГА + ЛП	1.89	8464
	Модиф. ГА	2.69	8500,71
	Модиф. ГА+ЛП	0.98	8464

Примечание: здесь ГА – генетический алгоритм с жадной эвристикой, ЛП – локальный поиск.

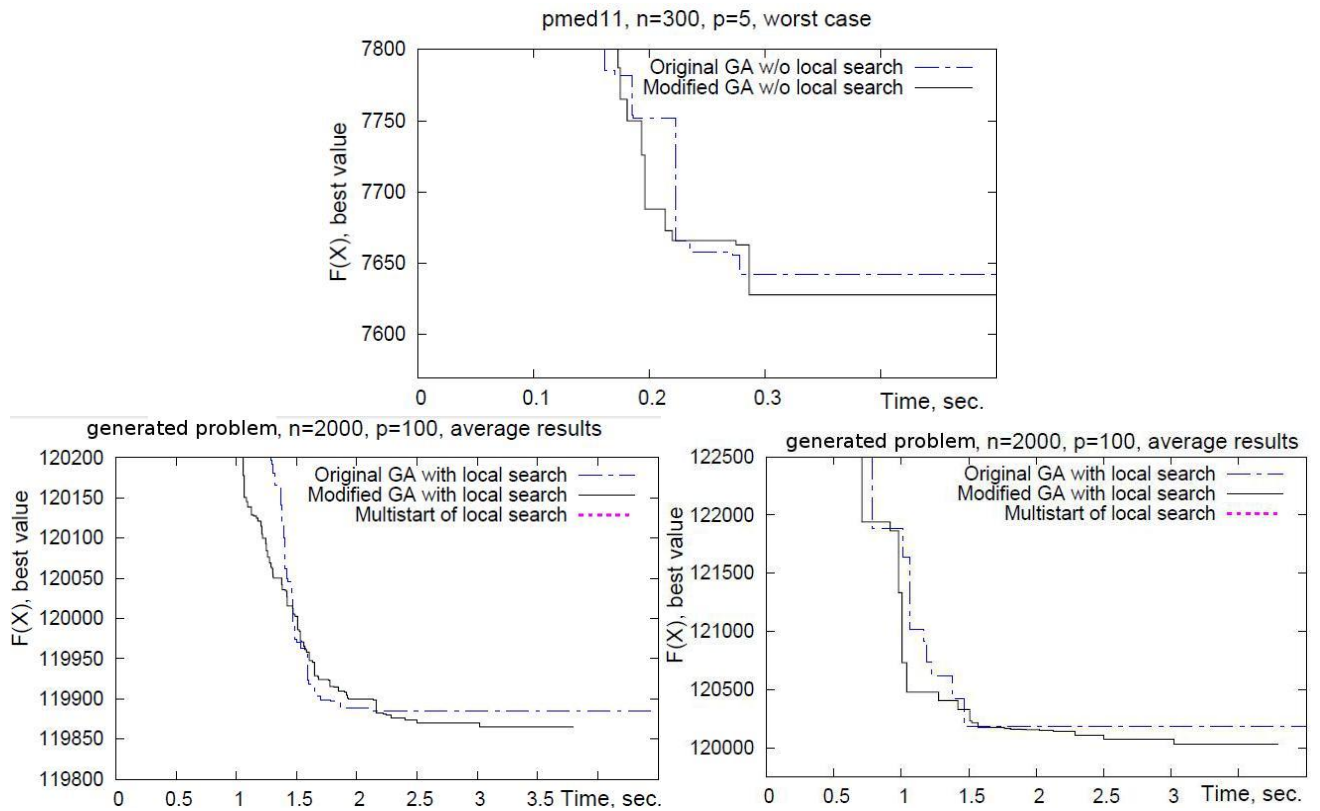


Рисунок 3.8 – Сравнение результатов исходного и модифицированного (Модификация 2) генетических алгоритмов с жадной эвристикой для  $p$ -медианных задач на сетях. Результаты локального поиска с мультисъстартом оказались за рамками графиков (выше).

Для целей сравнения мы использовали метод многократного старта (мультистарта) процедуры локального поиска (Алгоритма 2.2) из случайным образом сгенерированных начальных решений как пример простейшего, но достаточно эффективного алгоритма решения задачи, а также генетический алгоритм [204] с процедурой кроссинговера в виде исходной (не модифицированной) жадной эвристики (Алгоритм 3.16). В качестве тестовых примеров использованы задачи из репозитория OR Library [244]. Данная библиотека тестовых примеров содержит задачи с числом вершин сети до  $N = 900$ . Для создания тестовых задач большего размера был использован специальный алгоритм [157].

На рис. 3.8 показаны усредненные значения 10 запусков, а также худший результат 10 попыток. Для расчета размера популяции  $N_{pop}$  генетического алгоритма использовалось выражение

$$N_{pop} = \left\lceil \frac{\sqrt{N} C \binom{N}{p}}{100 \left\lceil \frac{N}{p} \right\rceil} \right\rceil \left\lceil \frac{N}{p} \right\rceil.$$

Таким образом, предложенные модификации жадных агломеративных эвристик

позволяют повысить точность решения задачи k-медоид и p-медианной задачи на сети при размерности задачи до нескольких сотен тысяч векторов данных по сравнению с известными ранее версиями алгоритмов с жадной эвристикой и по сравнению с другими известными методами или сократить время счета до 55 раз без потери точности. Алгоритм может быть использован при решении больших дискретных задач размещения, кластерного анализа и классификации с высокой ценой ошибки.

### 3.7 Детерминированный алгоритм с жадной эвристической процедурой

Рассмотренные выше алгоритмы позволяют существенно сократить ошибку получаемого ими субоптимального решения. Как показано ниже в сравнительных таблицах, предлагаемые модификации алгоритмов позволяют также сократить разброс получаемых ими оценок минимума целевой функции: результат становится более стабильным. Тем не менее, все предложенные выше модификации алгоритмов представляют собой вариации стохастических методов.

Во многих задачах важна стабильность результата. Таковой, например, является рассматриваемая в следующей Главе задача классификации электронной компонентной базы космического применения: в получаемом результате решения этой задачи заинтересованы две стороны – изготовитель электронных компонентов и потребитель, от имени которого выступает специализированный тестовый центр. Результаты должны быть не только точными, но и проверяемыми. То, что предлагаемые выше методы дают более точный и более стабильный результат, безусловно, упрощает проверку результатов, но идеальным был бы метод, который давал бы не только достаточно точный, но и абсолютно стабильный результат, т.е. был бы полностью детерминированным.

Как уже упоминалось выше, хорошие результаты могут быть получены для задач кластерного анализа и классификации с использованием методов Information Bottleneck Clustering (IBC) – метод «бутылочного горлышка» при кластеризации [253]. Работа данного алгоритма начинается с рассмотрения каждого из векторов данных как отдельного кластера. Затем из системы один за другим удаляются «лишние» кластеры, пока не останется требуемое количество. Каждый раз удаляется кластер, удаление которого дает наименьший прирост суммарного квадратичного расстояния от векторов

данных до ближайшего из центров кластеров. Такие алгоритмы чрезвычайно медленны [253]. Полная противоположность – процедуры локального поиска, такие как ALA для задачи  $k$ -средних или PAM для  $k$ -медоидной задачи – на практических задачах эти алгоритмы работают весьма быстро, и для большинства практических задач, в которых не предъявляются особые требования к точности получаемого результата, эти алгоритмы в сочетании со специальными методами инициализации, такими как  $k$ -means++ [314], являются весьма приемлемыми. В этом смысле генетические алгоритмы с жадной эвристикой представляют собой некий компромисс быстродействия и точности.

Отметим, что жадная агломеративная эвристика, аналогичная как методу ИВС была предложена в 1963 году [131] для задачи размещения складов. Как в [131], так и в методе ИВС в качестве начального решения для жадной эвристики выбирают множество центров (недопустимое решение), равное множеству векторов данных (точек требования в терминологии, характерной для задач размещения в реальном пространстве).

Неоспоримым достоинством метода ИВС является его детерминированность – точная воспроизводимость результатов. Метод чрезвычайно медленный (включает запуск Алгоритма 3.14 или аналогичного для оценки последствий удаления каждого элемента из промежуточного решения). Поэтому актуальным является разработка аналогичного более быстрого детерминированного метода.

В предыдущих параграфах были предложены несколько модификаций жадной эвристики, включая жадную эвристику для ГА с вещественным алфавитом (Модификация 4 – Алгоритм 3.12).

Важным настраиваемым параметром данного алгоритма является  $\sigma_e$  - доля одновременно исключаемых из решения «лишних» центров (центроидов, медоидов). Как было сказано выше, опытным путем установлены значения, близкие к оптимальным для ГА с процедурой кроссинговера в виде Алгоритма 3.12, в пределах  $\sigma_e \in [0.2; 0.25]$ . При близких к нулю значениях алгоритм удаляет из решения за одну итерацию лишь один центр (как в Алгоритме 3.5), что замедляет работу алгоритма практически без увеличения точности результата, при больших значениях точность резко падает. Далее в настоящем параграфе мы принимаем  $\sigma_e = 0.25$ .

Фактически жадная (Алгоритм 3.12) эвристика представляет собой комбинацию собственно жадной эвристики [204, 199, 332], в которой за одну итерацию удаляется до

20-25% «лишних» центров, пока не останется требуемое число центров  $p$ , с шагом модифицированной ALA-процедуры, выполняемым на каждой итерации. Если Алгоритм 3.5 требует выполнения  $p(k_0-k)$  запусков ALA-процедуры (здесь  $k_0$  – начальное число центров), то Алгоритм 3.12 снижает число итераций до  $O(\log(k_0-k))$ , кроме того, на каждой итерации выполняется не ALA-процедура целиком, но лишь отдельные ее оптимизированные шаги (размещение – распределение).

Нетрудно заметить, что при  $k_0=N$  работа Алгоритма 3.5 на начальной итерации аналогична методу ИВС: число центров, следовательно – и число кластеров совпадает с числом векторов данных. Более того, выбор начальных центров при  $k_0=N$  перестает быть случайным: в качестве таковых выбираются все векторы данных. Таким образом, можно построить следующий детерминированный алгоритм.

Алгоритм 3.17. Новый детерминированный алгоритм.

Дано: множество векторов данных  $V=(A_1, \dots, A_N) \in R^d$ , число кластеров.

1. Присвоить  $\chi_c = V$ .
2. Запустить Алгоритм 3.12 с шага 2.

Новый Алгоритм 3.17 может быть применен к задачам средней размерности, до  $N=6400$ , что продемонстрировано в Таблице 3.4. В качестве тестовых примеров были использованы данные тестовых испытаний электрорадиоизделий [332,333] и репозитория UCI (<http://www.cs.uci.edu/mllearn/mlrepository.html>). Используются различные метрики. Решены  $k$ -медоидные задачи [71, 254].

Достигнутые результаты для некоторых задач незначительно уступают по точности методу ИВС, вычислительные же затраты при этом требуются несопоставимо меньшие. В таблице приведены результаты для  $\sigma_e = 0,25$  и  $\sigma_e = 0,001$ . При  $\sigma_e = 0,001$  алгоритм фактически становится процедурой, при которой центры исключаются последовательно – один за другим. При этом способ включения элементов ALA-процедуры в жадную эвристику остается неизменным, и такой алгоритм, хотя и работает существенно медленнее, по скорости многократно превосходит ИВС, предполагающий использование Алгоритма 3.4 для оценки значения целевой функции на каждой итерации. В дополнение таблица содержит результаты «облегченной» версии ИВС, в которой выбор

исключаемых центров осуществляется без запуска процедур локального поиска: фактически решается  $k$ -медоидная задача, затем результат уточняется на последнем шаге ALA-процедурой.

Таблица 3.4 – Сравнительные результаты работы детерминированного алгоритма

Набор данных, число векторов данных, размерность пространства, тип данных	Число классов $k$ , метрика, тип задач	Алгоритм	Результат	Время, с.	Средне-квадр. отклонение результата 10 запусков
Chess (UCI), $N=3197$ , $d=50$ , булевы	$50$ , $l_1$ , $k$ -средних	Лок.поиск (мультистарт)	10020,2	35	19,11
		ГА с фикс.подмнож.	9328,4	35	6,066
		ГА в веществ.алфавитом	9290	35	9,148
		ИВС	-	-	-
		ИВС без лок. поиска	9796	29,24	Детерм.
		ИВС без лок.поиска, $\sigma_e=0.25$	10057	0,696	Детерм.
		Нов.алгоритм, $\sigma_e=0.25$	9649	0,694	Детерм.
Нов.алгоритм, $\sigma_e=0.001$ .	9610	34,52	Детерм.		
Отбраковочные испытания диода, $N=701$ , $d=18$ , веществ.	$30$ , $l_1$ нормирован., $k$ -средних	Лок.поиск (мультистарт)	5451,79	35	4,158
		ГА с фикс.подмнож.	5427,32	35	6,31
		ГА в веществ.алфавитом	5381,55	35	1,46
		ИВС	5390,12	2251,1	Детерм.
		ИВС без лок. поиска	5494,05	0,686	Детерм.
		ИВС без лок.поиска, $\sigma_e=0.25$	5483,89	0,111	Детерм.
		Нов.алгоритм, $\sigma_e=0.25$	5420,67	0,0669	Детерм.
Нов.алгоритм, $\sigma_e=0.001$ .	5414,06	0,8962	Детерм.		
Отбраковочные испытания диода, $N=701$ , $d=18$ , веществ.	$30$ , $l_2^2$ нормирован., $k$ -медоид	Лок.поиск (мультистарт)	1950,52	7	5,362
		ГА с фикс.подмнож.	1887,29	7	4,11
		ГА в веществ.алфавитом	1885,79	7	1,457
		ИВС	1902,93	1062,1	Детерм.
		ИВС без лок. поиска	1917,14	0,637	Детерм.
		ИВС без лок.поиска, $\sigma_e=0.25$	1959,97	0,0324	Детерм.
		Нов.алгоритм, $\sigma_e=0.25$	1918,56	0,039	Детерм.
Нов.алгоритм, $\sigma_e=0.001$ .	1912,43	0,6696	Детерм.		
Отбраковочные испытания микросхемы 1526ЛЕ2, $N=620$ , $d=120$ , веществ.	$30$ , $l_2^2$ нормирован., $k$ -сред.	Лок.поиск (мультистарт)	12873,22	30	50,71
		ГА с фикс.подмнож.	12536,19	30	3,11
		ГА в веществ.алфавитом	12539,5	30	2,59
		ИВС	12556,04	4453,9	Детерм.
		ИВС без лок. поиска	12727,13	0,5484	Детерм.
		ИВС без лок.поиска, $\sigma_e=0.25$	12788,47	0,0088	Детерм.
		Нов.алгоритм, $\sigma_e=0.25$	12834,9	0,1258	Детерм.
Нов.алгоритм, $\sigma_e=0.001$ .	12682,32	0,7528	Детерм.		
MissAmerica1 (UCI), $N=6480$ , $d=16$ , целочисленные	$100$ , $l_2^2$ , $k$ -сред.	Лок.поиск (мультистарт)	717488,7	60	1107,24
		ГА с фикс.подмнож.	698055,6	60	460,62
		ГА в веществ.алфавитом	698054,5	60	406,42
		ИВС	-	-	-
		ИВС без лок. поиска	715762,6	159,8	Детерм.
		ИВС без лок.поиска, $\sigma_e=0.25$	716150,6	1,85	Детерм.
		Нов.алгоритм, $\sigma_e=0.25$	707529,5	2,02	Детерм.
Нов.алгоритм, $\sigma_e=0.001$ .	703786,4	199,5	Детерм.		

Продолжение таблицы 3.4

Набор данных, число векторов данных, размерность пространства, тип данных	Число кластеров, метрика, тип задач	Алгоритм	Результат	Вре-мя, с.	Средне-квадр. отклоне-ние резуль-тата 10 запусков
PimaIndians Diabetes (UCI), $N=768$ , $d=9$ , веществ.	30, $l_2^2$ , k-сред.	Лок.поиск (мультистарт)	2072,425	36	2,4422
		ГА с фикс.подмнож.	2045,196	36	1,7294
		ГА в веществ.алфавитом	2045,365	36	1,238
		ИВС	2043,481	1150,1	Детерм.
		ИВС без лок. поиска	2096,06	0,81	Детерм.
		ИВС без лок.поиска, $\sigma_e=0.25$	2144,771	0,031	Детерм.
		Нов.алгоритм, $\sigma_e=0.25$	2087,46	0,042	Детерм.
Нов.алгоритм, $\sigma_e=0.001$ .	2056,35	1,058	Детерм.		
BreastCancer (UCI), $N=699$ , $d=10$ , категории	20, метрика Жаккара, k-медоид	Лок.поиск (мультистарт)	184,1	5	0,8725
		ГА с фикс.подмнож.	172,81	5	0,365
		ГА в веществ.алфавитом	172,62	5	0,0787
		ИВС	177,5	370,16	Детерм.
		ИВС без лок. поиска	174,4	0,8898	Детерм.
		ИВС без лок.поиска, $\sigma_e=0.25$	174,6	0,8072	Детерм.
		Нов.алгоритм, $\sigma_e=0.25$	175,2	0,437	Детерм.
Нов.алгоритм, $\sigma_e=0.001$	175,7	0,7352	Детерм.		
Zoo (UCI), $N=101$ , $d=10$ , категории	10, метрика Жаккара, k-медоид	Лок.поиск (мультистарт)	6,4285	1	0,0287
		ГА с фикс.подмнож.	6,4118	1	0
		ГА в веществ.алфавитом	6,4118	1	0
		ИВС	6,4706	4,533	Детерм.
		ИВС без лок. поиска	6,4706	0,015	Детерм.
		ИВС без лок.поиска, $\sigma_e=0.25$	6,4706	0,010	Детерм.
		Нов.алгоритм, $\sigma_e=0.25$	6,4706	0,009	Детерм.
Нов.алгоритм, $\sigma_e=0.001$	6,4706	0,012	Детерм.		
Ionosphere (UCI), $N=351$ , $d=35$ , веществ.	10, $L_1$ , k-сред.	Лок.поиск (мультистарт)	2530,40	4	2,117
		ГА с фикс.подмнож.	2526,77	4	0,0257
		ГА в веществ.алфавитом	2527,00	4	0,0082
		ИВС	2536,06	76,13	Детерм.
		ИВС без лок. поиска	2546,54	0,1124	Детерм.
		ИВС без лок.поиска, $\sigma_e=0.25$	2572,40	0,0198	Детерм.
		Нов.алгоритм, $\sigma_e=0.25$	2531,03	0,025	Детерм.
Нов.алгоритм, $\sigma_e=0.001$	2533,31	0,151	Детерм.		

Для всех задач, за исключением задач с метрикой Жаккара, новый алгоритм показал лучшие результаты среди приведенных детерминированных алгоритмов по производительности, лишь слегка уступая по точности методу ИВС и превосходя по скорости на несколько порядков. Результаты, получаемые новым детерминированным алгоритмом, уступают по точности эволюционным алгоритмам. За исключением задач с метрикой Жаккара, данное отставание по точности не является значительным (не превышает 2,5% для задач с вещественными координатами векторов данных и 4% для



задачи с булевыми данными).

Решение задачи автоматической группировки в некоторых случаях [332] сводится к решению серии задач  $k$ -средних при  $k = \overline{k_{min}, k_{max}}$ . Алгоритм 3.6 может быть применен для задачи с  $k=k_{max}$ , а затем Алгоритм 3.5 начиная с шага 2 может снова быть запущен для получения последующих результатов вплоть до  $k=k_{min}$ . Таким образом возможно получение результатов для всех значений  $k = \overline{k_{min}, k_{max}}$  одним цикле.

Предлагаемый в настоящей работе детерминированный алгоритм способен за приемлемое время решать задачи автоматической группировки  $k$ -средних и  $k$ -медоид. Результаты по точности незначительно уступают результатам эволюционных алгоритмов, но являются при этом детерминированными, что облегчает контроль и интерпретацию результатов всеми заинтересованными сторонами.

### 3.8 Адаптивный метод жадных эвристик

Детерминированный Алгоритм 3.17 является частным случаем Алгоритма 3.12 при начальном значении  $|\chi_c| = N$ . Работа процедуры кроссингвера генетического алгоритма с вещественным алфавитом (Алгоритм 3.12) начинается с объединенного промежуточного решения  $\chi_c = \chi_{k_1} \cup \chi_{k_2}$ . При  $p \ll N$  в большинстве случаев (по крайней мере на начальных итерациях генетического алгоритма) элементы «родительских» множеств  $\chi_{k_1}$  и  $\chi_{k_2}$ , как правило, не повторяются, и на начальной итерации Алгоритма 3.5  $|\chi_c| = 2p$ . В предыдущих параграфах предлагаются эффективные модификации жадной эвристики для генетического алгоритма с частичным объединением «родительских» множеств. В этом случае  $|\chi_c| \in \{(\overline{p+1}, 2p)\}$ , т.е. мощность множества, являющегося начальным решением, в среднем уменьшается. Наконец, генетический алгоритм с рекомбинацией подмножеств фиксированной длины [82] оперирует исключительно подмножествами мощности  $p$ . Все перечисленные алгоритмы являются наиболее эффективными применительно к отдельным задачам. Таким образом, в зависимости от задачи оптимальным является различное значение мощности начального множества  $|\chi_c|$ . Результаты параграфа опубликованы в [335].

Обозначим количество центров в начальном решении  $p_{init} = |\chi_c|$ , а отношение

количества «лишних» центров в начальном решении к требуемому числу центров

$$\beta = (p_{init} - p) / p.$$

Жадная эвристика в исходном виде (шаги 2-4 Алгоритмов 3.6 и 3.7) и в модифицированном, с вещественным алфавитом (Алгоритм 3.12) может быть применена в качестве самостоятельного метода решения задач. Алгоритм может быть построен следующим образом.

Алгоритм 3.18. Метод жадных эвристик с мультистартом.

Дано: множество векторов данных  $V=(A_1, \dots, A_N) \in \mathbb{R}^d$ , число  $p$  искомым центров, параметры  $\beta$  и  $\sigma_e$ .

1. Присвоить  $p_{init} = p + \lceil \beta p \rceil$ . Выбрать случайным образом подмножество  $\chi_c \in V$  мощности  $p_{init}$ .
2. Запустить Алгоритм 3.12 с шага 2.
3. Если не достигнуты условия останова, то повторить с шага 1.

Такой подход требует указания параметра  $\beta$ . Данный параметр важен, что показывают различные результаты Алгоритма 3.18 при различных значениях данного параметра, собранные в Таблице 3.5.

Рассмотрим известный метод решения задач псевдобоулевой оптимизации – метод изменяющихся вероятностей (МИВЕР) [237, 230, 229]. Идея метода заключается в том, что элементы из какого-либо множества выбираются случайным образом, но вероятность выбора или не выбора того или иного элемента множества зависит от результатов, достигнутых на предыдущих итерациях. Алгоритм на базе метода изменяющихся вероятностей предложен и для  $p$ -медианной задачи на сети [157]. Однако алгоритм обладает весьма медленной сходимостью и используется лишь как вспомогательный метод для генетического алгоритма с жадной эвристикой [157]. Подобный же алгоритм может использоваться для приближенного решения  $p$ -медианной задачи с произвольной функцией расстояния [316]. Встраивание жадной эвристики с вещественным алфавитом в схему метода изменяющихся вероятностей позволяет создать новый эффективный алгоритм:

Алгоритм 3.19. Адаптивный метод жадных эвристик для непрерывных задач размещения.

1. Установить значения элементов вектора вероятностей  $p_i = 1/N$  для всех  $i = \overline{1, N}$ .  
Установить  $\beta = 0,5$ .

2. Для каждого  $j = \overline{1, N_p}$  выполнять:

2.1. Скопировать элементы вектора вероятностей  $q_i = p_i$  для всех  $i = \overline{1, N}$ . Присвоить  $\chi_c = \emptyset$ . Сгенерировать случайное  $r \in [0; 2)$  с равномерным распределением. Рассчитать  $p_{init} = p + \lceil \beta \cdot r \cdot p \rceil$ .

2.2. Пока  $|\chi_c| < p_{init}$  выполнять:

2.2.1. Сгенерировать случайное  $Q \in [0; \sum_{i=1}^N q_i)$  с равномерным распределением.

Присвоить  $S=0; l=1$ .

2.2.2. Пока  $S+q_j < Q$  выполнять:  $S=S+q_j; l=l+1$ ; повторять 2.2.2.

2.2.3. Пополнить множество  $\chi_c = \chi_c \cup \{A_l\}$ ; присвоить  $q_l=0$ .

2.2.4. Следующая итерация 2.2.

2.3. Скопировать полученное начальное множество  $\xi_c = \chi_c$ .

2.4. Запустить Алгоритм 3.12 с шага 2. Сохранить полученное значение  $f_j = F(\chi_c)$

; присвоить  $\chi_j = \chi_c$ ;  $\beta_j = (p_{init} - p) / p$ .

2.5. Следующая итерация 2.

3. Рассчитать  $\beta = \beta \sum_{j=1}^{N_{pop}} \frac{\beta_j (N_p - n_j)^2}{\sum_{l=1}^{N_p} (N_p - n_l)^2 \sum_{l=1}^{N_p} \beta_l}$ . Здесь  $n_j$  – порядковый номер

значения  $f_j$  в упорядоченной по возрастанию последовательности этих значений.

Если  $4\beta \cdot p > N$ , то присвоить  $\beta = N / 4p$ .

4. Выбрать индекс  $b$ , которому соответствует наименьшее значение  $f_b$  и индекс  $w$ , которому соответствует наибольшее значение  $f_w$ .

5. Для каждого  $i = \overline{1, N}$  выполнять:

5.1. Если  $A_i \in \chi_b$  и  $A_i \notin \chi_w$ , то  $p_i = \gamma p_i$ . Иначе если  $A_i \in \chi_w$  и  $A_i \notin \chi_b$ , то  $p_i = p_i / \gamma$ .

5.2. Следующая итерация 5.

## 6. Проверить условия останова, повтор с шага 2.

Данный алгоритм требует указания значений параметров  $N_p$  – размер популяции и  $\gamma$  – шаг (коэффициент) изменения вероятностей. Для работы алгоритма достаточны небольшие популяции. Мы использовали значения  $N_{pop} = 9$  и  $\gamma = 1,1$ . Параметр  $\beta$  автоматически настраивается специальной метаэвристикой (шаг 4).

Заметим, что новые алгоритмы, как и детерминированный алгоритм с жадной эвристикой [250, 334] обладают следующим важным свойством. Решение задачи автоматической группировки в некоторых случаях [332] сводится к решению серии задач  $k$ -средних при  $p = \overline{p_{min}, p_{max}}$ , где  $p_{min} = 1$  (предполагается однородная партия, неделимая на кластеры), а  $p_{max}$  принимается равным некоторому максимально возможному с технологической точки зрения значению. Алгоритм 3.19 может быть применен для задачи с  $p = p_{max}$ , а затем Алгоритм 3.12, начиная с шага 2, может снова быть запущен для получения последующих результатов вплоть до  $p = p_{min}$ . Таким образом возможно получение результатов для всех значений  $p = \overline{p_{min}, p_{max}}$  одним циклом.

Новые методы были опробованы на задачах из репозитория UCI [326], а также на реальных данных результатов отбраковочных тестов электрорадиоизделий высокой степени надежности [332] (отбраковочным испытаниям посвящена следующая Глава). Некоторые результаты приведены в Таблице 3.5. Результаты показаны в сравнении с методом Information Bottleneck Clustering [253, 250], с детерминированным алгоритмом с жадной эвристикой [250], а также с известным эволюционным алгоритмом для  $p$ -медианной и  $p$ -медоидной задач, основанном на рекомбинации подмножеств фиксированной длины [253], с генетическим алгоритмом с классическим способом рекомбинации, а также с генетическим алгоритмом с вещественным алфавитом для  $p$ -медианной задачи [249]. Полученные результаты показывают высокую эффективность нового адаптивного алгоритма. В некоторых случаях он показывает наилучшие результаты.

Для выполнения единственного старта алгоритма с жадной эвристикой может потребоваться время, большее, чем отведенный лимит времени. Алгоритм в этом смысле чувствителен к выбору параметра  $\beta$ . В этих случаях в таблице проставлены прочерки. Также для некоторых «больших» задач не приведены результаты

детерминированных алгоритмов из-за чрезвычайно большого времени их работы.

Таблица 3.5 – Сравнительные результаты вычислительных экспериментов

Набор данных, число векторов данных, размерность пространства, тип данных	Число кластеров k, метрика, тип задач	Алгоритм	Вре-мя, с.	Результат		Средне-квадр. отклоне-ние (30 запусков)	Средне-квадр. отклоне-ние (30 запусков)
				$\sigma_e = 0,25$	$\sigma_e = 0,001$		
Отбраков. испытания диода, N=701, d=18, веществ.	30, $l_2^2$ нор-миро-ван., к-сред-них	ALA	15		4896,913		9,41069
		ГАФП	15		4867,103		2,13825
		ГАЗЭ	15		4810,655		1,37712
		ГА классич.	15		4882,609		8,51498
		ИВС	989,3		4887,930		Детерм.
		Детерм. ЖЭ	0,02/0,9	4890,675	4888,21	Детерм.	Детерм.
		ЖЭ, $\beta = 0.5$	15	4855,235	4855,534	12,9294	7,6843
		ЖЭ, $\beta = 1$	15	4835,241	4825,410	4,0288	6,4184
		ЖЭ, $\beta = 3$	15	4834,817	4821,546	7,6421	5,2087
		ЖЛ $\beta = 0.5$	15	4854,183	4853,432	10,848	8,8929
		ЖЛ, $\beta = 1$	15	4839,104	4829,781	2,8491	3,1062
		ЖЛ, $\beta = 3$	15	4836,167	4820,190	4,1217	3,1191
		ЖЛ, $\beta = 3$	15	4827,753	4832,583	1,3607	1,6819
		ЖЭ адапт.					
BreastCancer, N=699, d=10, категории	20, мет-рика Жак-кара, к-медо-ид	ALA	5		184,1		0,8725
		ГАФП	5		172,81		0,3650
		ГАЗЭ	5		172,62		0,0787
		ГА классич.	5		182,98		1,0415
		ИВС	370,16		Детерм.		Детерм.
		Детерм. ЖЭ	0,4/0,7	175,2	175,7	Детерм.	Детерм.
		ЖЭ, $\beta = 0.5$	5	181,95	181,62	0,816	0,453
		ЖЭ, $\beta = 1$	5	177,10	177,77	1,108	0,513
		ЖЭ, $\beta = 3$	5	175,00	175,63	0,155	0,755
		ЖЛ $\beta = 0.5$	5	183,93	184,43	0,859	2,312
		ЖЛ, $\beta = 1$	5	181,43	181,60	3,163	1,012
		ЖЛ, $\beta = 3$	5	-	-	-	-
		ЖЛ, $\beta = 3$	5	175,39	175,67	1,741	0,468
		ЖЭ адапт.					
Ionosphere (UCI), N=351, d=10, веществ.	20, $l_1$ , к-сред-них	ALA	4		2530,40		2,117
		ГАФП	4		2526,77		0,0257
		ГАЗЭ	4		2527,00		0,0082
		ГА классич.	4		2526,93		0,1302
		ИВС	370,16		Детерм.		Детерм.
		Детерм. ЖЭ	0,03/0,2	2531,03	2533,31	Детерм.	Детерм.
		ЖЭ, $\beta = 0.5$	4	2526,81	2526,79	0,0286	0,0222
		ЖЭ, $\beta = 1$	4	2526,85	2526,81	0,0816	0,0222
		ЖЭ, $\beta = 3$	4	2526,86	2526,92	0,0821	0,1100
		ЖЛ $\beta = 0.5$	4	2528,02	2527,83	0,4523	0,5249
		ЖЛ, $\beta = 1$	4	2528,33	2527,92	0,9843	0,7072
		ЖЛ, $\beta = 3$	4	2528,99	2529,99	1,0165	0,8595
		ЖЭ адапт.	4	2526,79	2526,79	0,0231	0,0205

Продолжение Таблицы 3.5

Набор данных, число векторов данных, размерность пространст-ва, тип данных	Число клас- теров k, мет- рика, тип задач	Алгоритм	Вре-мя, с.	Результат		Средне- квадр. отклоне- ние (30 запусков)	Средне- квадр. отклоне- ние (30 запусков)
				$\sigma_e = 0,25$	$\sigma_e = 0,001$	$\sigma_e = 0,25$	$\sigma_e = 0,001$
BIRCH3 (UCI), N=100000, d=2, веществ.	100, $l_2^2$ ,  к-сред- них	ALA	60		$3,825 \cdot 10^{13}$		$3,060 \cdot 10^{11}$
		ГАФП	60		$3,886 \cdot 10^{13}$		$5,155 \cdot 10^{11}$
		ГАЗЭ	60		$3,751 \cdot 10^{13}$		$0,878 \cdot 10^{11}$
		ГА классич.	-		-		-
		IBC	-		-		-
		Детерм. ЖЭ	-		-		-
		ЖЭ, $\beta = 0.5$	60	$3,832 \cdot 10^{13}$	$3,735 \cdot 10^{13}$	$5,359 \cdot 10^{11}$	$0,759 \cdot 10^{11}$
		ЖЭ, $\beta = 1$	60	$3,827 \cdot 10^{13}$	-	$4,017 \cdot 10^{11}$	-
		ЖЭ, $\beta = 3$	-	-	-	-	-
		ЖЛ $\beta = 0.5$	-	-	-	-	-
		ЖЛ, $\beta = 1$	-	-	-	-	-
ЖЛ, $\beta = 3$	-	-	-	-	-		
ЖЭ адапт.	60	$3,772 \cdot 10^{13}$	$3,722 \cdot 10^{13}$	$3,802 \cdot 10^{11}$	$0,216 \cdot 10^{11}$		
MissAmerica1 (UCI), N=6400, d=16, веществ.	100, $l_2^2$ ,  к-сред- них	ALA	60		717488,7		1107,24
		ГАФП	60		698055,6		460,62
		ГАЗЭ	60		698054,5		406,42
		ГА классич.	60		714946,3		685,23
		IBC	-		-		-
		Детерм. ЖЭ	93/9977	703786,4	707529,5	Детерм.	Детерм.
		ЖЭ, $\beta = 0.5$	60	714287,9	713244,8	499,037	2263,9
		ЖЭ, $\beta = 1$	60	705838,9	-	334,656	-
		ЖЭ, $\beta = 3$	60	709468,4	-	195,671	-
		ЖЛ $\beta = 0.5$	60	-	-	-	-
		ЖЛ, $\beta = 1$	60	-	-	-	-
ЖЛ, $\beta = 3$	60	-	-	-	-		
ЖЭ адапт.	60	702860,8	708239,0	576,207	781,084		
Сгенерированн ая задача, N=500, d=2, веществ.	20, мет- рика с угло- выми  расс- тояни- ями, к- меди- анная	ALA	4		4461,24		25,9199
		ГАФП	4		4509,49		14,4368
		ГАЗЭ	4		4326,72		4,7587
		ГА классич.	4		4392,95		79,6913
		IBC	3141		4381,34		Детерм.
		Детерм. ЖЭ	-	4377,12	4376,43	Детерм.	Детерм.
		ЖЭ, $\beta = 0.5$	4	4350,36	4378,30	15,1717	15,4234
		ЖЭ, $\beta = 1$	4	4333,02	4371,50	5,8614	38,5620
		ЖЭ, $\beta = 3$	4	4389,32	-	22,5053	-
		ЖЛ $\beta = 0.5$	4	4428,97	4453,65	59,4167	34,3130
		ЖЛ, $\beta = 1$	4	4379,42	4359,41	46,1732	22,5052
ЖЛ, $\beta = 3$	4	4429,37	-	78,2398	-		
ЖЭ адапт.	4	4313,67	4309,43	3,7077	1,4427		

Продолжение таблицы 3.5

Набор данных, число векторов данных, размерность пространства, тип данных	Число кластеров k, метрика, тип задач	Алгоритм	Время, с.	Результат		Средне-квадр. отклонение (30 запусков)	Средне-квадр. отклонение (30 запусков)
				$\sigma_e=0,25$	$\sigma_e=0.001$	$\sigma_e=0,25$	$\sigma_e=0.001$
Сгенерированная задача, N=1300, d=2, веществ.	5, метрика Москвы-Карлсруэ, k-медианная	ALA	10		22661,347		0
		ГАФП	10		22661,347		0
		ГАЗЭ	10		22668,407		18,680
		ГА классич.	10		22661,347		0
		IBC	-		-		-
		Детерм. ЖЭ	9,8/1,2	23521,44	23221,94	Детерм.	Детерм.
		ЖЭ, $\beta = 0.5$	10	22661,347	22679,593	0	48,277
		ЖЭ, $\beta = 1$	10	22661,347	22661,347	0	0
		ЖЭ, $\beta = 3$	10	2675,467	22694,032	24,116	56,349
		ЖЛ $\beta = 0.5$	10	22679,591	22682,843	48,277	39,628
		ЖЛ, $\beta = 1$	10	22661,347	22661,347	0	0
		ЖЛ, $\beta = 3$	10	22693,666	22702,762	85,511	109,574
ЖЭ адапт.	10	22701,227	22661,347	68,703	0		
Отбраков. испытания изделия H5503XM1, d=229	15, косинусная мера k-медианная	ALA	5		8,841802		0,039958
		ГАФП	5		8,753999		0,044288
		ГАЗЭ	5		8,645197		0,000310
		ГА классич.	5		8,876550		0,060246
		IBC	306,3		9,022582		Детерм.
		Детерм. ЖЭ	0,98/4,54	8,800345	8,666864	Детерм.	Детерм.
		ЖЭ, $\beta = 0.5$	5	8,811074	8,814372	0,062142	0,069825
		ЖЭ, $\beta = 1$	5	8,835857	8,771366	0,022888	0,037968
		ЖЭ, $\beta = 3$	5	8,812487	8,790734	0,011947	0,026473
		ЖЛ $\beta = 0.5$	5	8,754117	8,776604	0,022399	0,035394
		ЖЛ, $\beta = 1$	5	8,721773	8,706887	0,025144	0,023189
		ЖЛ, $\beta = 3$	5	8,698656	8,699854	0,011932	0,019937
ЖЭ адапт.	5	8,706932	8,697230	0,019641	0,025038		

Обозначения алгоритмов: ALA – мультистарт ALA-процедуры, ГА – генетический с рекомбинацией подожеств фиксированной длины [82], ГАЗЭ – генетический с жадной эвристикой с вещественным алфавитом [249], ГА классич. – ГА с классическим способом рекомбинации, IBC – Information Bottleneck Clustering [253, 250], Детерм. ЖЭ – детерминированный с жадной эвристикой (Алгоритм 3.17) [250], ЖЭ – Алгоритм 3.18, ЖЭ – мультистарт исходной жадной эвристики (шаги 2-4 Алгоритма 3.6 при  $\sigma_e=0.001$  и Алгоритм 3.9 с быстрым удалением элементов из промежуточного решения аналогично шагам 4.1-4.3 Алгоритма 3.12 при  $\sigma_e = 0,25$ ), ЖЛ – мультистарт жадной эвристики с включенным локальным поиском (Алгоритмом 3.12), ЖЭ адапт. – адаптивная жадная эвристика (Алгоритм 3.19).

Открытым остается вопрос подбора оптимального значения параметра  $\sigma_e$ . Заметим, что при  $\sigma_e = 0.001$  центры исключаются из промежуточного решения последовательно, один за другим, как в исходной жадной эвристике. Если для детерминированного алгоритма с жадной эвристикой [250] значение  $\sigma_e = 0,25$

гарантирует быстрый результат (не всегда более точный) по сравнению с результатами, полученными при  $\sigma_e=0.001$ , то в случае стохастического алгоритма сложно предугадать, при каком значении он даст более точный результат за приемлемое время.

Как видно из таблицы 3.5, ни число векторов данных задачи (сравнить наборы данных MissAmerica1, BIRCH3 и данные отбраковочных испытаний), ни метрика не определяют однозначно оптимальное значение данного параметра. Перспективным направлением представляется разработка метаэвристики для подстройки данного параметра по ходу выполнения алгоритма аналогично тому, как подстраивается параметр  $\beta$ . Построение алгоритма, подстраивающего параметр  $\sigma_e$  при фиксированном значении  $\beta$  не представляется проблематичным. Подстройка же обоих параметров одновременно требует факторного анализа для оценки влияния каждой из величин на получаемые результаты. Поэтому более перспективной кажется разработка алгоритма с двумя конкурирующими популяциями, каждая из которых развивается при одном из фиксированных значений параметра  $\sigma_e$ . Аналогичный алгоритм с конкурирующими популяциями, реализующими различные бионические оптимизационные алгоритмы, рассмотрен, например, в [197]. Тем не менее, можно сделать однозначный вывод о том, что новый метод дает результаты лучшие, чем многократный локальный поиск из случайных начальных решений и лучшие, чем результаты многократного запуска жадной эвристики с фиксированным значением параметра  $\beta$ .

Таким образом, новый метод жадных эвристик может быть использован как самостоятельный метод решения  $p$ -медианных задач, задач  $k$ -средних и  $k$ -медоид наряду с использованием его в составе детерминированного алгоритма схемы Information Bottleneck Clustering [253, 250] и в качестве эвристики рекомбинации, применяемой в составе эволюционного алгоритма [249, 332, 324]. Новый адаптивный алгоритм, основанный на идеях метода изменяющихся вероятностей, позволяет существенно повысить точность получаемых результатов и в некоторых случаях достичь наилучших результатов за приемлемое время.

### **3.9 Комбинация жадных эвристических алгоритмов с альтернативными алгоритмами локального поиска.**

В настоящем исследовании для непрерывных  $p$ -медианных задач мы применяли в качестве процедуры локального поиска ALA-процедуру (стандартную процедуру  $k$ -



средних), для  $p$ -медоидных задач – как ALA-процедуру, так и более быструю PAM-процедуру [253, 248]. PAM-процедура – алгоритм поиска в окрестности каждого из медоидов, состоящую из решений, получаемых заменой медоида на один из  $p_{PAM}$  ближайших к нему векторов данных. В настоящем исследовании мы использовали значение  $p_{PAM}=3$ , то есть очень небольшую окрестность. В случае ALA-процедуры для задачи  $p$ -медоид окрестность образуется заменой медоида на любой вектор данных кластера, к которому принадлежит медоид. Эксперименты не выявили существенных преимуществ того или иного метода: PAM позволяет быстрее получить хороший результат при небольшом числе больших кластеров (большие значения отношения  $N/p$ ), в случае небольших кластеров более эффективна ALA-процедура.

Между тем, методы локального поиска для дискретных задач автоматической группировки, в частности – для задачи  $p$ -медоид, весьма разнообразны [181, 210, 67]. Давно отмечено [210], решения, весьма близкие к точному, дают алгоритмы, основанные на поиске в SWAP-окрестности, образуемой заменой медоида на любой вектор данных. Во многих случаях поиск в данной окрестности позволяет получить точные решения [67]. Вычислительная сложность одной итерации такой процедуры при этом лежит в пределах  $O(pN^2)$ . Традиционно данная процедура применяется для относительно небольших наборов данных ( $N < 5000$ ).

Еще большие окрестности, такие как K-SWAP (замена сразу  $K$  медоидов на другие векторы данных), безусловно, способны повысить ожидаемую точность результата, получаемого после единственного запуска такой процедуры, но затраты времени при этом растут настолько, что даже в окрестности 2-SWAP поиск возможен лишь на очень небольших наборах данных. Заметим, что при  $K=p$  поиск в данной окрестности, вырождаясь в полный перебор, дает точное решение.

Жадные алгоритмы, предложенные в настоящей главе (Алгоритмы 3.5, 3.6, 3.9, 3.12), используют ALA-процедуру (как отмечалось выше, в случае задачи  $p$ -медоид возможно использование PAM-процедуры). Дополним эту процедуру поиском в SWAP-окрестности. Алгоритм 3.20 может быть использован как вариант ALA-процедуры, дополненный поиском в окрестности SWAP, в Алгоритмах 3.5, 3.6, 3.9, 3.12.

Алгоритм 3.20. Чередование ALA-процедуры или PAM-процедуры с поиском в окрестности SWAP.

Дано: начальное решение  $\chi$  – множество медоидов или центроидов.

Шаг 1. Запустить ALA-процедуру (как вариант – PAM-процедуру) с начальным решением  $\chi$ .

Шаг 2. Если на Шаге 1 значение целевой функции не улучшилось, то ОСТАНОВ, возврат  $\chi$ .

Шаг 3. Сформировать массив  $I$  из чисел  $\{\overline{1, p}\}$ , расположить эти числа в массиве в случайном порядке (рандомизация несколько ускоряет процедуру).

Шаг 4. Для каждого  $i' \in I$  выполнять:

Шаг 4.1. Присвоить  $i = I_{i'}$ . Присвоить  $f' = +\infty$ .

Шаг 4.2. Присвоить  $j' = \arg \min_{j \in \{\overline{1, N}\}} F((\chi \setminus \{\chi_i\} \cup \{A_j\}))$ . Здесь  $\chi_i$  –  $i$ -й медоид/центроид в решении,  $A_j$  –  $j$ -й вектор данных.

Шаг 4.3. Если  $F((\chi \setminus \{\chi_i\} \cup \{A_j\})) < F(\chi)$ , то присвоить  $\chi = \chi \setminus \{\chi_i\} \cup \{A_j\}$ , перейти к 1.

Отметим, что для непрерывных задач ( $k$ -средних,  $p$ -медианная) существует процедура, аналогичная поиску в окрестности SWAP – процедура  $j$ -means [282], область применения которой также ограничена довольно небольшими задачами. Данная процедура сводится к замене центров/центроидов на один (лучший с точки зрения целевой функции) из векторов данных с последующим продолжением поиска стандартной ALA-процедурой. Нетрудно заметить, что именно этот принцип поиска реализован Алгоритмом 3.20.

Таким образом, Алгоритм 3.20 реализует чередование ALA- или PAM-процедуры с поиском в SWAP-окрестности для дискретных задач и процедуру  $j$ -means для непрерывных.

Применим Алгоритм 3.20 вместо Алгоритма 3.1 для решения дискретных и непрерывных задач автоматической группировки в составе генетических алгоритмов с жадной эвристикой.

На рис. 3.9 представлены некоторые результаты вычислительных экспериментов.

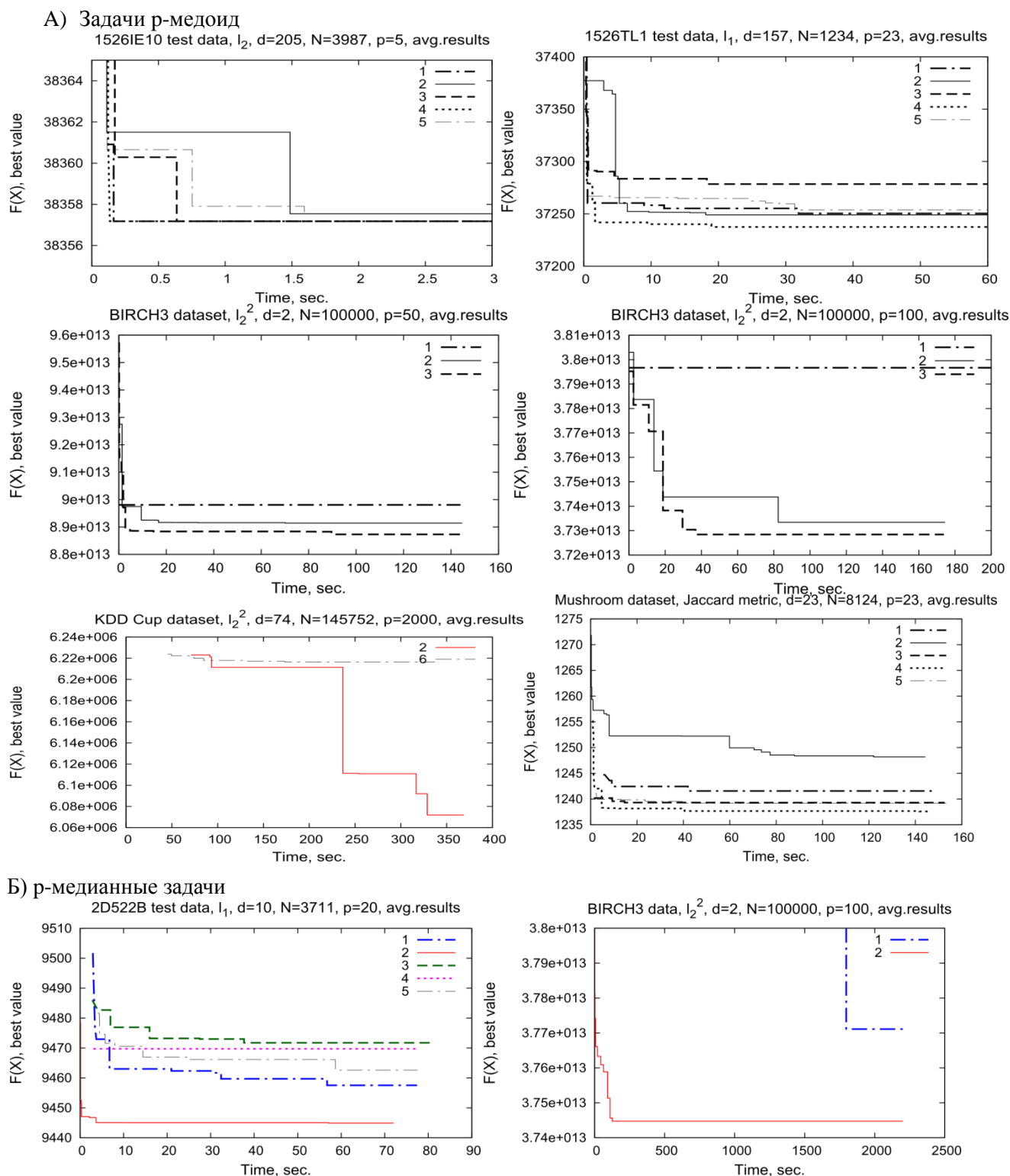


Рисунок 3.9 – Сравнительные результаты алгоритмов: 1 – мультистарт локального поиска в SWAP-окрестности (в случае p-медианной задачи – алгоритма j-means), 2 – ГА с жадной эвристикой (процедура PAM или k-средних в качестве лок.поиска), 3 – Алгоритм 3.13 (в комбинации с лок. поиском в SWAP-окрестности или j-means), 4 – Алгоритм 3.15 (в комбинации с лок. поиском в SWAP-окрестности или j-means), 5 – ГА с рекомбинацией подмножеств фикс. мощности (GA-MIX) в комбинации с локальным поиском в SWAP-окрестности или j-means, 6 – мультистарт обычного локального поиска

Для задач с числом векторов данных  $N > 10000$  применить Алгоритм 3.20 в составе генетического алгоритма с жадной эвристикой не удалось в виду того, что для единственного запуска Алгоритма 3.20 требовалось время, многократно превосходящее весь отводимый на решение задачи лимит времени.

Динамика изменения наилучшего полученного значения целевой функции показывает, что для небольших дискретных задач ( $N < 1000$ ) применение Алгоритма 3.20 имеет преимущество перед применением Алгоритма 3.1. Более того, для таких задач Применение Алгоритма 3.20 даже в качестве самостоятельного алгоритма, без использования генетического алгоритма или МИВЕРА в качестве стратегии глобального поиска имеет преимущество перед генетическим алгоритмом с жадной эвристикой в редакциях, изложенных в предыдущих разделах (т.е. в комбинации с Алгоритмом 3.1 в составе жадной эвристики).

«Средние» по объему входных данных ( $N < 10000$ ) задачи демонстрируют иную картину. В некоторых случаях генетический алгоритм с жадной эвристикой даже без применения поиска в окрестности SWAP показывает наилучшие результаты. В других случаях применение поиска в окрестности SWAP себя оправдывает, но применение этого вида локального поиска в составе генетического алгоритма ведет к дальнейшему улучшению результата. При этом применение Алгоритма 3.8 – частичного объединенного решения в качестве средства получения начального решения для жадной эвристики имеет преимущество перед полным решением. Более того, в некоторых случаях более простые способы рекомбинации в генетическом алгоритме, такие как генетический алгоритм с рекомбинацией подмножеств фиксированной мощности [82], дают лучшие результаты в сравнении с ГА с жадной эвристикой. Особенно это характерно для задач с булевыми и данными и данными о составе объектов. При этом получаемые результаты все же сопоставимы по точности, и, если априори неизвестен характер задачи, использование ГА с жадной эвристикой можно считать конкурентноспособным для задач с  $1000 < N < 10000$ .

В то же время, для непрерывных задач (за исключением самых небольших, с  $N < 1000$ , для которых вполне можно обойтись мультистартом процедуры j-means) применение алгоритмов с жадной эвристикой в комбинации с Алгоритмом 1 вполне оправдано в большинстве практических случаев.

Таким образом, применение поиска в окрестности SWAP (а возможно – и в еще более широких окрестностях) может быть оправданным при решении небольших задач, но не оправдано для задач с большим объемом входных данных ( $N > 10000$ ).

### 3.10 Модификации для решения серии задач

Обычные алгоритмы, такие как k-средних (ALA-процедура) или весьма эффективный генетический алгоритм с рекомбинацией множеств центров фиксированной мощности Шена и Лиу [82] требуют указания значения  $p$ . Другие, как, например, X-means [70], выбирают лучшее значение  $p$  по определенному критерию, адекватность которого не всегда очевидна. Ниже предложена простая модификация жадной эвристики: после достижения числа кластеров  $p$  процесс исключения элементов из решения продолжается, при этом фиксируются значения целевой функции для каждого значения числа центров. Таким образом можно получить решение сразу серии задач со значениями  $p = \overline{2, p_{max}}$ . Отметим, что некое представление о максимальном значении групп  $p_{max}$  иметь все же необходимо.

Специфика алгоритмов с жадной агломеративной эвристической процедурой состоит в последовательном сокращении мощности множества точек (центров, центроидов, едоидов), которым представлено промежуточное решение. Таким образом, на различных этапах работы алгоритма могут быть получены промежуточные решения, мощность которых лежит в диапазоне  $\{\overline{p, p_{init}}\}$ , где  $p_{init}$  – мощность промежуточного решения, с которой жадная эвристика начинает свою работу. Если принять  $p_{init} = p_{max}$ , а  $p = 2$ , можно получить все решения.

Алгоритм ниже иллюстрирует, как такую модифицированную жадную эвристику можно встроить в ГА, хотя тот же прием можно применять и для всех других алгоритмов с жадной агломеративной эвристикой, использующих иные стратегии глобального поиска, предложенных в настоящей Главе.

Алгоритм 3.21. ГА с жадной эвристикой для решения серии задач с  $p = \overline{2, p_{max}}$ .

Шаг 1. Инициализация популяции из  $N_{pop}$  особей. Каждая особь – множество из  $p_{max}$  вершин (обозначим его множеством  $X$ , а  $X_i$  –  $i$ -й элемент множества). Присвоить

$F_{new,j} = +\infty$  для  $j = \overline{1, N_{pop}}$ . Инициализация массива лучших значений целевой функции  $F_k^* = +\infty$  и лучших решений  $X_k^* = \{\}$  для  $k = \overline{2, p_{max}}$ .

Шаг 2. Выбрать случайным образом  $j_1, j_2 \in [1, N], j_1 \neq j_2$

Шаг 3.  $X_{new} = X_{j_1} \cup X_{j_2}$

Шаг 4. Пока  $|X_{new}| > p_{max}$ :

Шаг 4.1. Выбрать вершину  $j$ , исключение которой даёт наименьший прирост целевой функции:  $j = \arg \min_{j \in X_{new}} F(X_{new} \setminus \{j\})$

Шаг 4.2.  $X_{new} = X_{new} \setminus \{j\}$ . Продолжить цикл 4.

Шаг 5. Присвоить  $F_{new} = 0$ . Здесь мы вводим новую функцию полезности – сумму значений целевой функции для  $p = \overline{2, p_{max}}$ . Присвоить  $X^* = X_{new}$ .

Шаг 6. Пока  $|X_{new}| > 2$ ,

Шаг 6.1. Присвоить  $F_{new} = F_{new} + f(X_{new})$ ;  $k = |X_{new}|$ ;  $F_k = f(X_{new})$ ; если  $F_k < F_k^*$ , то присвоить  $F_k^* = F_k$ ;

Шаг 6.2. Выполнить действия шагов 4.1 и 4.2 для  $X_{new}$ . Продолжить цикл 6.

Шаг 7. Выбрать  $j_3$  турнирной селекцией по значению  $F_{new,j}$ . Присвоить  $F_{j_3} = F_{new}$ ;  $X_{j_3} = X^*$ ,  $F_{new,j_3} = F_{new}$ .

Шаг 7. Проверить условия останова, перейти к 2.

В данном алгоритме, в отличие от других ГА, рассмотренных в настоящей Главе, функция полезности отличается от целевой функции задачи. Отметим, что, поскольку решается сразу серия задач, то количество целевых функций соответствует количеству решаемых задач. Мы складываем целевые функции для каждого значения  $p$  и получаем новую функцию полезности.

Исследование применимости данного алгоритма к задачам кластерного анализа и сравнение его с другими находится на стадии проведения экспериментов. Некоторые результаты представлены в Таблице 3.6. Для непрерывных  $p$ -медианных задач в сочетании с разными способами локального поиска также получены хорошие результаты для задач с большим числом кластеров. При снижении этого числа метод начинает давать худшие результаты в сравнении с другими методами. Поэтому, вероятно, ГА с жадной эвристикой следует останавливать не при  $p=2$ , а раньше, примерно на значении  $p = p_{max}/3$ , а остальные решения получать, например, новым запуском алгоритма с меньшим значением  $p_{max}$ .

Таблица 3.6 – Сравнительные результаты вычислительных экспериментов для задачи  $k$ -средних и непрерывной  $p$ -медианной задачи.

Набор данных, его параметры	$p$ и мера расстоян	Алгоритм (см.прим.)	Время, с.	Следний результат	Среднеквадр. отклонение
Отбраков. испытания микросхемы 1526ТЛ1, N=1234, d=120, веществ.  (в ГА ЖЭС – $p \in \{2..20\}$ )	$p=14, l_2^2$	Мутьгистарт ALA	15	150,124869801	0,384203928
		Мультист. j-means	15	150,533299444	0,598587789
		ГА ФП+ALA	15	149,954679652	0,172789313
		ГА ФП+j-means	15	151,280175427	0,982922979
		ГА ЖЭС+ALA	15 (на все задачи)	149,78736565*	0,03157532*
		ГА ЖЭС+j-means	15 (на все задачи)	151,082443691	0,654212395
	$p=10, l_2^2$	Мутьгистарт ALA	15	198,375350991	0,018643710
		Мультист. j-means	15	198,426881563	0,044039446
		ГА ФП+ALA	15	198,377650812	0,024878118
		ГА ФП+j-means	15	198,450402498	0,032311263
		ГА ЖЭС+ALA	15 (на все задачи)	198,359747028	$2 \cdot 10^{-14}$ *
		ГА ЖЭС+j-means	15 (на все задачи)	198,35421865*	0,0070903
	$p=6, l_2^2$	Мутьгистарт ALA	15	362,70701636*	0*
		Мультист. j-means	15	362,70401636*	0*
		ГА ФП+ALA	15	362,70401636*	0*
		ГА ФП+j-means	15	362,704156850	0,000344112
		ГА ЖЭС+ALA	15 (на все задачи)	362,704051312	0*
		ГА ЖЭС+j-means	15 (на все задачи)	362,704051312	0*
UCI Mopsi Joensuu, N=6014, d=2, веществ.  (в ГА ЖЭС – $p \in \{2..20\}$ )	$p=10, l_2$	Мутьгистарт ALA	15	359,680203232	3,964320582
		Мультист. j-means	15	359,545287242	0,208756158
		ГА ФП+ALA	15	359,545250068	2,526439494
		ГА ФП+j-means	15	361,435624000	0,208770779
		ГА ЖЭС+ALA	15 (на все задачи)	359,410460803	0,177992934
		ГА ЖЭС+j-means	15 (на все задачи)	359,41036391*	0*
	$p=4, l_2$	Мутьгистарт ALA	15	596,825210394	0,000000442
		Мультист. j-means	15	596,825217410	0,000004148
		ГА ФП+ALA	15	596,82520843*	0,000000388
		ГА ФП+j-means	15	596,825208927	0,000000574
		ГА ЖЭС+ALA	15 (на все задачи)	596,825283111	0*
		ГА ЖЭС+j-means	15 (на все задачи)	596,825283111	0*
BIRCH-3, N=100000, d=2, веществ.  (в ГА ЖЭС – $p \in \{2..110\}$ )	$p=100, l_2^2$	Мутьгистарт ALA	30	$3,7513245 \cdot 10^{15}$	116786778766
		Мультист. j-means	3000	$3,7711179 \cdot 10^{13}$	158613580914
		ГА ЖЭС+ALA	30 (на все задачи)	$3,740432 \cdot 10^{13}$ *	21699776156*
		ГА ЖЭС+j-means	30 (на все задачи)	-	-
	$p=50, l_2^2$	ГА ФП+ALA	30	$9,0099578 \cdot 10^{13}$	9545892119
		ГА ФП+j-means	30	-	-
		ГА ЖЭС+ALA	30 (на все задачи)	$8,902789 \cdot 10^{13}$ *	0*
		ГА ЖЭС+j-means	30 (на все задачи)	-	-
	$p=20, l_2^2$	ГА ФП+ALA	30	$3,303278 \cdot 10^{14}$ *	0*
		ГА ФП+j-means	30	-	-
		ГА ЖЭС+ALA	30 (на все задачи)	$3,3049972 \cdot 10^{14}$	0*
		ГА ЖЭС+j-means	30 (на все задачи)	-	-

Примечание: Мутьгистарт ALA – мультистарт процедуры  $k$ -средних (ALA-процедуры), Мультист. J-means – мультистарт процедуры j-means, ГАФП+ALA – ГА с рекомбинацией подмножеств фиксированной длины (Шен, Лиу) в комбинации с ALA-процедурой в качестве средства локального поиска, ГАФП+ALA – тот же ГА в комбинации с процедурой j-means в качестве средства локального поиска, ГАЖЭС+ALA, ГАЖЭС+j-means – ГА с жадной эвристикой с вещественным алфавитом для решения серии задач в комбинации с ALA-процедурой и процедурой j-means соответственно. Знаком «\*» отмечен лучший результат.

### 3.11 Общая схема метода жадных эвристик

На рис.3.10-3.12 дана блок-схема оценки применимости метода жадных эвристик и процедуры выбора алгоритма/алгоритмов для реализации автоматизированной системы, предназначенной для потокового решения задач автоматизированной группировки данных, задач множественного размещения или решения задач псевдобулевой оптимизации с монотонной целевой функцией и ограничениями, в зависимости от объема и размерности данных, оценки количества получаемых групп, свойств задачи (например, дискретная/непрерывная), требований точности и стабильности результата и некоторых других параметров решаемых автоматизированной системой задач.

Схемы на рис. 3.10-3.13 обобщают практический опыт автора в области применения жадных агломеративных эвристик.

Начало блок-схемы дано на рис.3.10. Вначале определяется, является ли задача дискретной (в случае непрерывной задачи осуществляется переход к метке 1 на рис.3.11). Если дискретная задача не может быть сформулирована в виде дискретной  $p$ -медианной или аналогичной задачи (т. е. целевая функция не является суммой каких-либо расстояний либо имеются сложные условия), на при этом и целевая функция, и ограничения представлены в виде неравенств с монотонной левой частью и константой в правой части, предлагается использование генетического алгоритма с жадной эвристикой для задач монотонной псевдобулевой оптимизации. Для определения применимости алгоритма к таким задачам по аналогии с  $p$ -медианной оценивается значение  $p$ , которое в этом случае означает оценку количества булевых переменных, значение которых в оптимальном решении равно 1. Большинство практических задач позволяет априорно дать приблизительную оценку этого значения исходя из ограничений задачи или из практического опыта (жадные агломеративные эвристики эффективны при небольших значениях отношения  $p/N$ , где  $N$  – общее число переменных, в противном случае следует ориентироваться на другие методы).



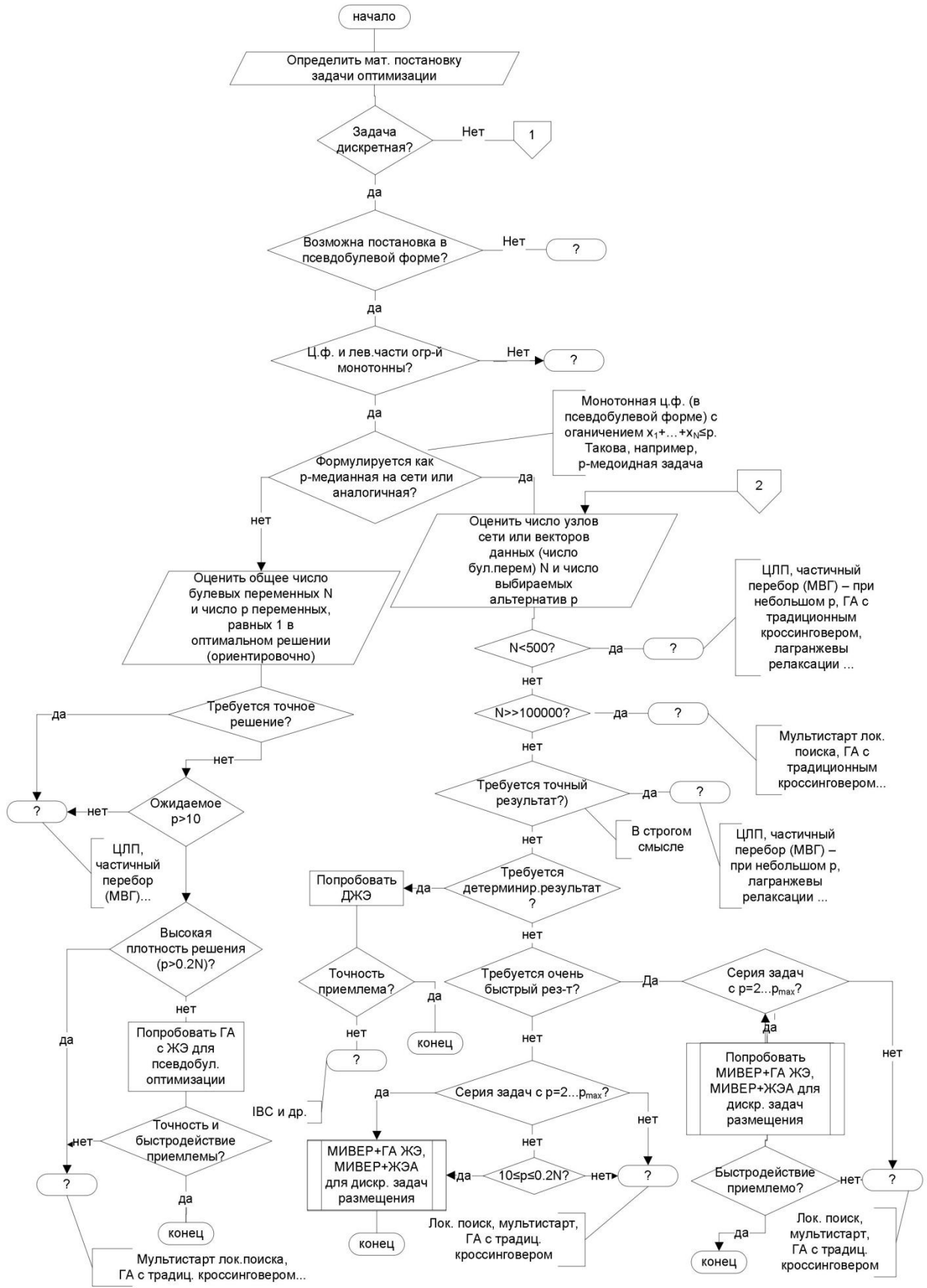


Рисунок 3.10 – Блок-схема определения применимости метода жадных эвристик и выбора его компонент для использования в автоматизированной системе, часть 1

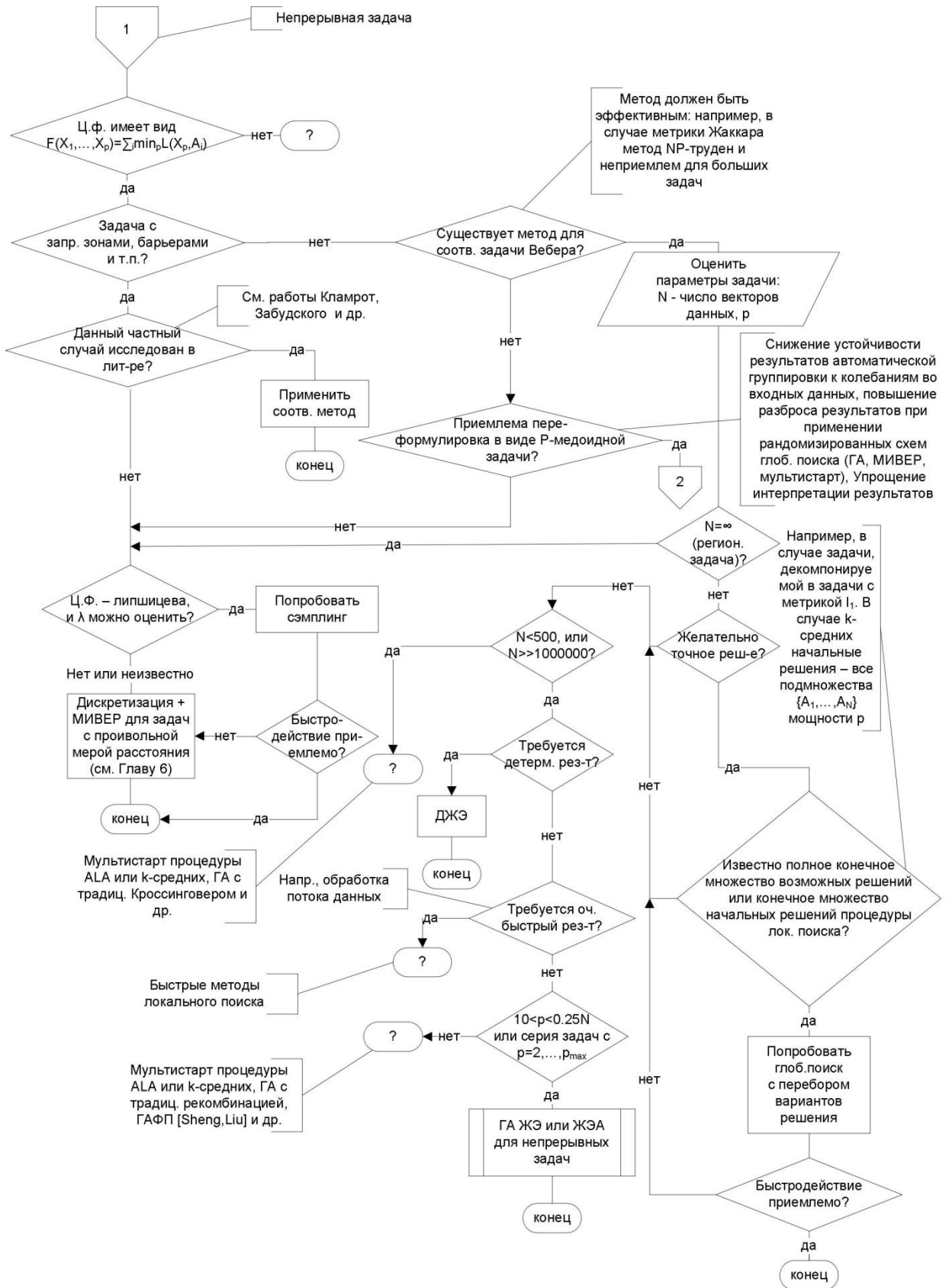


Рисунок 3.11 – Блок-схема определения применимости метода жадных эвристик и выбора его компонент для использования в автоматизированной системе, часть 2

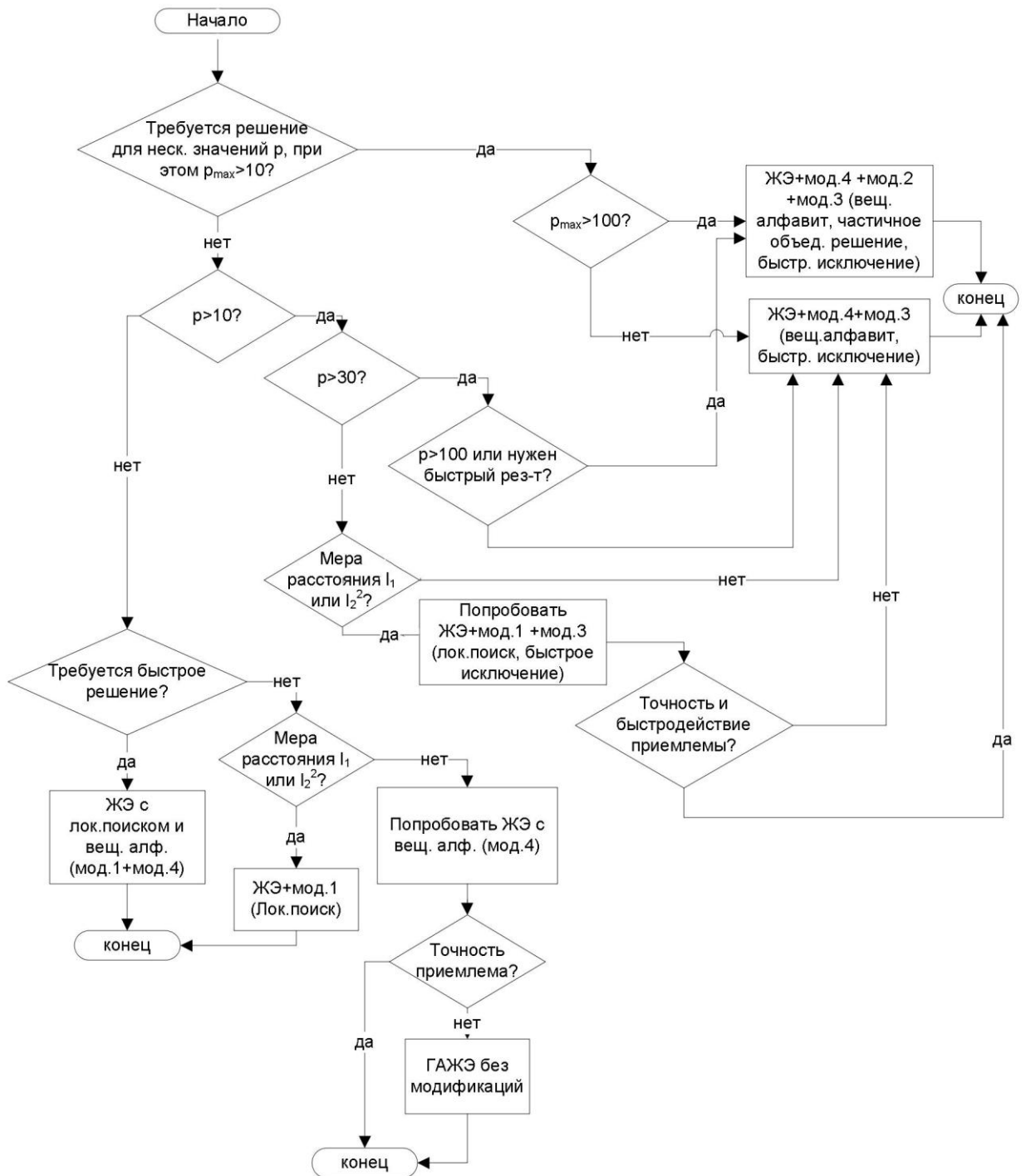


Рисунок 3.12 – Блок-схема выбора модификаций жадных эвристик для использования в автоматизированной системе в зависимости от условий и параметров задач

Отметим, что в случае, если требуется точное решение задачи, также следует использовать другие методы, например — сведение к задаче целочисленного линейного программирования, поэтому метод жадных эвристик следует использовать лишь при

невозможности получения такого решения в силу большого объема входных данных в решаемых задачах.

В случае, если дискретная задача допускает формулировку в виде  $p$ -медианной задачи размещения на сети, и объем входных данных достаточно велик ( $N > 500$ ), предлагается использовать метод жадных эвристик. При этом область его применения ограничена верхним пределом  $N \gg 100000$ . В зависимости от значения  $p$  и соотношения  $p/N$  предлагается выбор той или иной модификации алгоритма, основанного на применении жадных эвристик. Здесь ГАЖЭ — генетический алгоритм с жадной эвристикой, ЖЭА — жадная эвристика с адаптацией (адаптивный алгоритм с жадной эвристикой), МИВЕР — метод изменяющихся вероятностей для  $p$ -медианной задачи на сети, использующийся для инициализации генетического алгоритма (см. Главу 2).

Часть блок-схемы, изображенная на рис.3.11, показывает логику выбора алгоритма для использования в автоматизированной системе решения задач группировки или размещения в непрерывном пространстве. В зависимости от используемой метрики или иной меры сходства/различия, наличия ограничений (запрещенных зон, барьеров в задаче размещения), предлагается выбор различных вариантов комбинаций схем глобального поиска с жадными агломеративными эвристиками, при этом схема выбора конкретных модификаций жадных эвристик дана на рис.3.12. Для наиболее сложных случаев задач размещения с барьерами, запрещенными зонами и произвольной функцией расстояния (например, заданной алгоритмически), предлагается использовать алгоритм на основе метода изменяющихся вероятностей для дискретизированной задачи, описанный в Главе 6.

На рис. 3.13 дана схема, отражающая взаимную совместимость различных постановок задач (дискретная/непрерывная задача размещения или группировки, задача псевдоболевой монотонной оптимизации), различных стратегий глобального поиска (МИВЕР / ГА / мультистарт / детерминированные методы), различных вспомогательных методов локального поиска, эффективных при решении той или иной конкретной задачи, а также используемых мер расстояния при их применении в составе метода жадных эвристик.

Блок-схема алгоритма метода жадных эвристик в самом общем виде дана на рис. 3.14.

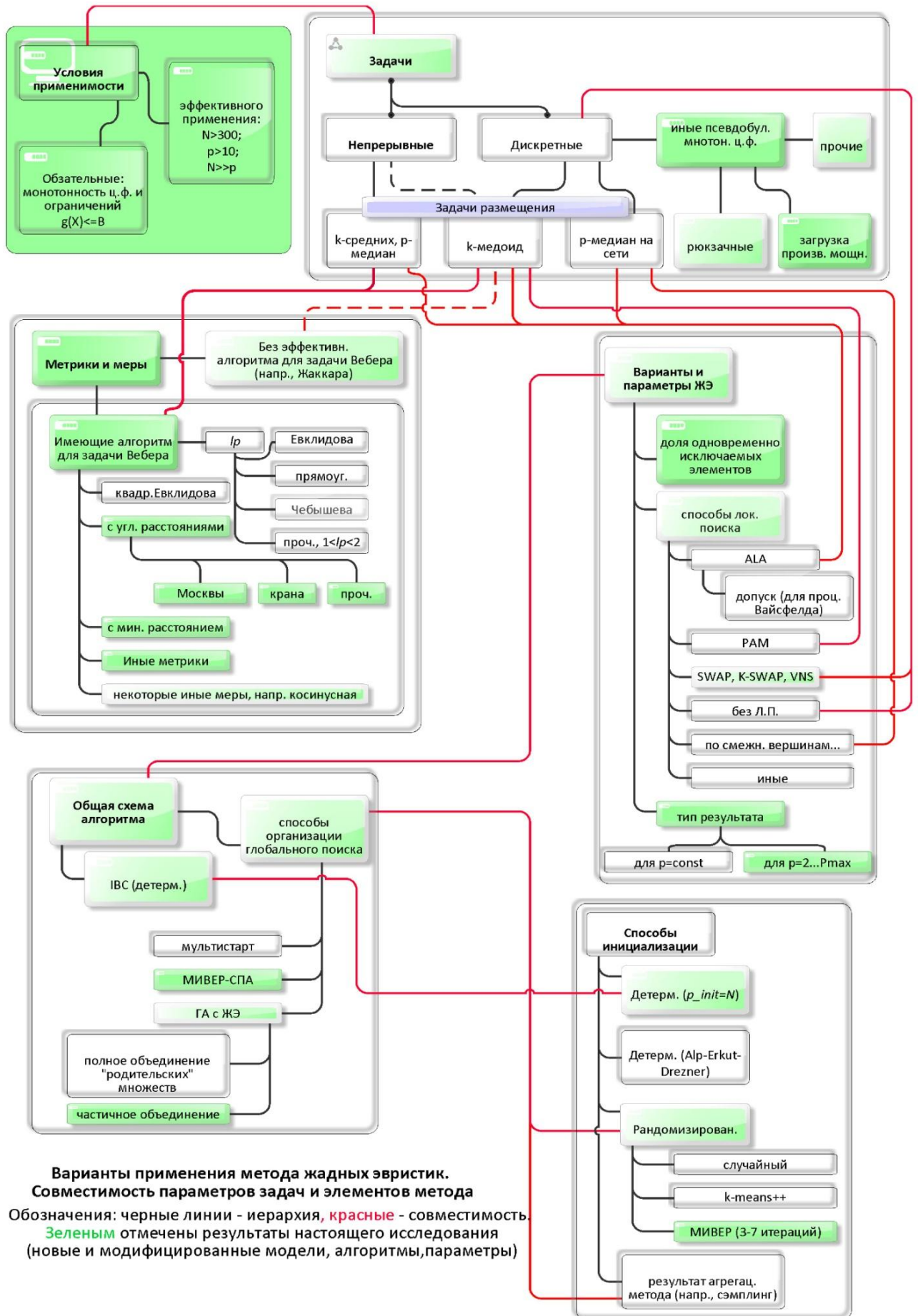


Рисунок 3.13 – Схема совместимости компонентов метода жадных эвристик



Рисунок 3.14 – Общая схема алгоритма метода жадных эвристик

На данной блок-схеме можно видеть три вложенных цикла. Первый из них – итерации стратегии глобального поиска (выполнение эволюционных операторов селекции, скрещивания – для ГА, генерация новой популяции решений для МИВЕРа и т.д.) Для детерминированного алгоритма предусмотрено выполнение всего одной итерации на этом уровне. Второй вложенный цикл – собственно выполнение жадной эвристической процедуры. Третий вложенный цикл в рамках этой процедуры предусматривает оценку последствий исключения элементов из промежуточного решения.

В случае, если жадная эвристическая процедура выполняется в модификации 1 (включение локального поиска) или же выполняется жадная эвристика (жадная эвристическая процедура) с вещественным алфавитом, то запускается один из существующих алгоритмов локального поиска для решаемой задачи. Так, для непрерывных задач это может быть ALA-процедура (стандартная процедура k-средних) или же другие двухшаговые процедуры чередующегося размещения-распределения, например, J-means или H-means. Для задачи составления расписания в Главе 2 были применены процедуры локального поиска в окрестностях, характерных для таких задач: окрестности, образуемые сдвигом начала выпуска продукции на одну смену или

же заменой вида выпускаемой продукции. Для других задач монотонной псевдодвулевой оптимизации должны применяться процедуры локального поиска, хорошо зарекомендовавшие себя для конкретного класса задач, при этом достаточно быстрые. Для дискретных задач автоматической группировки может применяться поиск в различных типах окрестностей. Так, для задачи  $p$ -медоид может применяться РАМ-процедура (окрестность из заданного количества соседних векторов данных), АЛА-процедура (более широкая окрестность – весь кластер), или же поиск в еще более широких SWAP или K-SWAP-окрестностях.

Как показывают результаты вычислительных экспериментов, включение в состав жадной эвристической процедуры алгоритмов поиска в очень широких окрестностях, таких как SWAP-рестность, не дает ощутимых преимуществ перед РАМ-процедурой или АЛА-процедурой. Как правило, получение приемлемого результата просто замедляется. Исключение составляют некоторые задачи с булевыми входными данными, а также задачи с метрикой Жаккара и метрикой Хемминга в качестве меры расстояния. Для дискретных задач с большим объемом входных данных ( $N > 10000$ ) поиск в SWAP-рестности или еще более широких окрестностях практически невозможен из-за очень высокой вычислительной сложности перебора такого числа вариантов для каждого из кластеров.

На рис. 3.15 дана структурная схема метода жадных эвристик (уровень расположения элементов по вертикали соответствует уровню вложенности данного элемента при выполнении алгоритма). Схема совмещена со схемой применимости стратегий глобального и локального поиска к различным классам задач.

### **Результаты и выводы к Главе 3.**

Результаты вычислительных экспериментов показывают, что предлагаемые жадные эвристические процедуры с целочисленным и вещественным алфавитом комбинируются с различными общими схемами организации глобального поиска (ГА, мультистарт, адаптивный – МИВЕР, детерминированный - ИВС). В зависимости от параметров задачи ( $N$ ,  $p$ , для непрерывных и  $p$ -медоидных задач –  $d$ , свойства ограничений) и требований, предъявляемых к времени решения и точности получаемых результатов, сформулированы условия применимости данных эвристических процедур в

составе различных схем глобального поиска, а также их модификаций (Модификаций 1,2 и 3), для решения задач автоматической группировки (р-медианной, р-медоидной, k-средних с различными метриками и мерами расстояния) и иных задач, обладающих свойствами монотонности целевой функции и левых частей ограничений в булевых переменных. Таким образом, алгоритмы метода жадных эвристик, включающий модификации жадной агломеративной эвристики в сочетании с условиями их применимости к тем или иным задачам в составе тех или иных схем глобального поиска, является универсальным эффективным методом решения широкого круга задач автоматической группировки объектов, задач размещения и задач монотонной псевдодобулевой оптимизации.

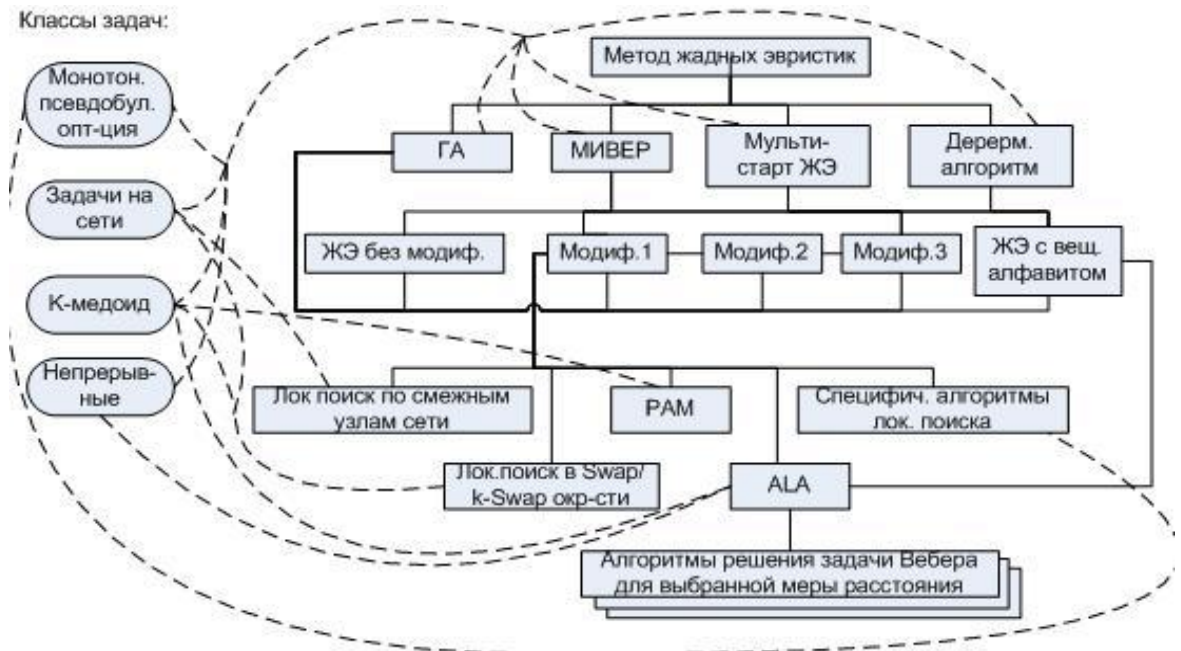


Рисунок 3.15 – Компоненты метода жадных эвристик, их взаимная совместимость (сплошные линии) и применимость к классам задач (курсивные линии)

Таким образом, метод жадных эвристик состоит в составлении эффективной комбинации данных компонентов при построении автоматизированной системы решения оптимизационных задач автоматической группировки, размещения или иных задач псевдодобулевой оптимизации в зависимости от условий и параметров решаемых задач. Алгоритм выбора этих компонентов, представленный на рис.3.10-3.12, позволяет строить автоматизированные системы, эффективно решающие поставленные задачи, что продемонстрировано практическими примерами, представленными в Главах 2 и 4.



Новая жадная эвристика с вещественным алфавитом для задачи автоматической группировки данных в непрерывном пространстве характеристик может быть эффективно использована как самостоятельный алгоритм, а также в составе эволюционного алгоритма и адаптивного алгоритма схемы МИВЕР для решения практических задач большой размерности с широким кругом метрик и иных мер расстояния. Эвристика комбинируется с различными методами локального поиска и методами агрегирования, что позволяет использовать передовые подходы для повышения быстродействия разрабатываемых алгоритмов. При этом новая эвристика, являясь компромиссным подходом по времени счета и точности результата, позволяет решать задачи группировки данных большой размерности (число векторов данных – от нескольких тысяч при числе измерений пространства характеристик от 2 до нескольких тысяч) в интерактивном режиме с точностью, не уступающей точности других известных методов, и при этом быстрее (в 1.4-220 раз) получать наиболее стабильный результат (наименьшее среднеквадратичное отклонение получаемого значения целевой функции), что продемонстрировано на примере задачи автоматической группировки электрорадиоизделий по производственным партиям.

В целом, предлагаемый метод жадных эвристик, основанный на применении новых жадных эвристик и их модификаций в составе различных схем глобального поиска (эволюционные алгоритмы, МИВЕР, детерминированный, мультистарт) в сочетании со сформулированными условиями применимости жадных эвристических процедур и их модификаций, схем глобального поиска в зависимости от параметров задачи, представляет собой эффективный метод решения задач автоматической группировки объектов и задач монотонной псевдобулевой оптимизации с большим объемом входных данных.

## **ГЛАВА 4. АЛГОРИТМ И СИСТЕМА АВТОМАТИЧЕСКОЙ ГРУППИРОВКИ ЭЛЕКТРОРАДИОИЗДЕЛИЙ КОСМИЧЕСКОГО ПРИМЕНЕНИЯ ПО ПРОИЗВОДСТВЕННЫМ ПАРТИЯМ**

В настоящей главе рассмотрены вопросы применения разработанных методов автоматической группировки к задаче классификации электрорадиоизделий космического применения в рамках проекта «Разработка алгоритмического обеспечения анализа однородности партий электрорадиоизделий для комплектации радиоэлектронной аппаратуры космических аппаратов». Приведенные результаты, являющиеся примером практического применения метода жадных эвристик, общая схема которого и модификации применяемых эвристик даны в Главе 3, внедрены в эксплуатацию в составе автоматизированной классификации электрорадиоизделий космического применения в ОАО ИТЦ – НПО ПМ (г.Железногорск, см. Приложение Г).

### **4.1 Общая постановка задачи**

При комплектации критически важных электронных узлов космических аппаратов компонентной базой (ЭКБ) соответствующего качества важно, чтобы однотипные элементы схемы имели близкие характеристики для обеспечения их согласованной работы, что наилучшим образом достигается в случае, если элементы изготовлены в рамках одной производственной партии из единой партии сырья. При комплектации критически важных узлов используются элементы, изготовленные отдельными специальными партиями, к которым предъявляются повышенные требования качества [336,337]. Главное отличие в требованиях к ЭКБ состоит в необходимости обеспечения длительной непрерывной работоспособности в течение 10-15 лет активного существования космического аппарата, а также в необходимости работы в агрессивной среде (температуры от -60 до +125 градусов цельсия и облучение, измеряемое в Мрад, тысячекратно превышающее смертельные для человека уровни).

Кроме того, каждая поступающая партия электрорадиоизделий перед использованием в космической отрасли проходит сложный комплекс как неразрушающих, так и разрушающих испытаний [336]. Разрушающие испытания проводятся выборочно – для нескольких экземпляров из каждой партии. При этом

распространять результаты именно разрушающих испытаний на всю поступившую партию можно лишь будучи уверенными в том, что данная партия состоит из экземпляров, изготовленных из одной партии сырья, по одной и той же технологии, с идентичной топологией (применительно к микросхемам), на одном и том же оборудовании, одновременно и подвергалась одним и тем же тестам на предприятии-изготовителе, хранилась и транспортировалась одинаковым образом. Все это невозможно гарантировать без обеспечения полного контроля всех перечисленных процессов. Для импортных компонентов такой контроль невозможен в принципе. Таким образом, партия должна быть предварительно проанализирована и разбита на однородные подмножества.

В России отсутствует специализированное производство электронной компонентной базы (ЭКБ) для космической отрасли, как это организовано в США и Западной Европе [338-340, 323]. Отечественные изготовители не выделяют изделия космического назначения в специальный класс. При этом требования к качеству и надежности компонентной базы, используемой в космической аппаратуре, выше, чем у стандартной продукции, выпускаемой российскими заводами [341, 342]. В этой связи сертификацию электрорадиоизделий с целью выделения экземпляров, потенциально ненадежных в условиях космического полета [343, 344], приходится производить в специализированных тестовых центрах [342, 341].

В процессе комплектации космического аппарата закупается компонентная база самого высокого уровня качества, существующего в России (ВП, ОС), затем проводится работа по отбраковке потенциально ненадежных компонентов. Как упоминалось выше, характеристики электронных приборов каждой партии контролируются с помощью разрушающих и не разрушающих тестов [336]. Для анализа однородности поступившей партии могут быть использованы данные неразрушающих тестов. Поставляемые партии ЭКБ могут быть неоднородными, собранными из нескольких производственных партий пластин. Поэтому результаты выборочных разрушающих испытаний на выборке ЭКБ распространять на всю поставленную партию компонентов можно лишь при условии, что партия изготовлена из одной партии пластин или, что разброс параметров различных кристалльных партий невелик. Это связано с тем, что относительно незначительные изменения в производственном процессе могут радикально повлиять на характеристики чувствительности, например, к радиационному воздействию.

Необходимо знать, из какого количества однородных групп собрана производственная партия компонентов.

Если поставленная совокупность компонентов состоит из нескольких различных групп, то для того, чтобы иметь обоснованное мнение о каждой из них, необходимо провести испытания для каждой группы, предварительно выявив их. Анализ партий ЭКБ на однородность становится актуальной задачей, особенно для ЭКБ с высоким уровнем интеграции, где флуктуации технологического процесса могут привести к ошибкам в оценке качества ЭКБ, идущей на комплектацию космического аппарата (КА).

Конечной целью исследования, которому посвящена настоящая глава, является предотвращение отказов электрорадиоизделий (ЭРИ) [345-347]. Отказ – скачкообразный (внезапный) или постепенный выход (дрейф) параметров изделия за установленные пределы. Дефектами [348, 349] называются неисправности, непосредственно в момент их обнаружения не приводящие к отказу. Для полупроводниковых приборов [345, 350] причинами отказов являются короткие замыкания, обрывы и изменения (дрейф) параметров. Нарушение технологии производства может приводить к дефектам, способным вызвать отказ любого типа. Тем не менее, даже в случае точного следования предусмотренной технологии случайные вкрапления в материалах, скрытые дефекты оборудования и множество иных причин могут приводить к дефектам, способным в будущем стать причиной отказа.

Скрытые дефекты снижают качество продукции с точки зрения риска отказов, поэтому многие исследователи отводят решающее значение неразрушающим испытаниям [351, 336] – дефектологии электрорадиоизделий, принципам и методам обнаружения дефектов [350, 352].

Первую группу неразрушающих испытаний составляют методы интегральной диагностики, основанные на измерении шумовых характеристик, в том числе электрических и акустических шумов. Вторую группу составляют методы локальной диагностики. К общим испытаниям относятся визуальный контроль, испытание давлением, акустическая и магнитная дефектоскопия, радиография и метод вихревых токов. К специфическим методам испытаний относятся рентгеновские, голографические, тепловые, оптические и электрические методы.

Взаимосвязь результатов различных видов испытаний с наличием дефектов в полупроводниковых приборах и других электрорадиоизделиях достаточно хорошо

изучена [345, 342, 353, 350, 352], разработаны регламентирующие документы [351, 354-357]. Наличие скрытого дефекта так или иначе отражается на результате одного или нескольких испытаний. В то же время, влияние дефекта на результат может быть различным и зависит как от типа дефекта, так и от типа изделия. Кроме того, параметры изделий, изготовленных в несколько отличающихся условиях и из различных партий сырья, могут довольно существенно отличаться. Выделение групп изделий, в которых с некоторой повышенной вероятностью содержатся изделия, имеющие дефекты, также выделение групп изделий, изготовленных одновременно из единой партии сырья, является основными задачами данной части настоящего исследования, для чего и применяются методы автоматической группировки. Данные, полученные с помощью не разрушающих тестов, могут быть использованы для анализа однородности партии.

#### 4.2 Типы испытаний электрорадиоизделий

Физической основой испытаний в форсированных режимах является различная стойкость потенциально ненадежных и качественных ЭРИ к электрическим, механическим, климатическим и радиационным нагрузкам [349, 343, 358-361]. Показатели безотказности являются функцией внутренних параметров ( $X$ ), формируемых в процессе производства, электрических нагрузок ( $Y$ ) и условий эксплуатации ( $Z$ ) [345]. Тогда время безотказной работы можно оценить как  $T_0=f(X,Y,Z)$ . Так как в процессе эксплуатации электрические, механические нагрузки и климатические факторы неизменны для всех элементов партии, то разброс наработки до отказа будет полностью определяться внутренними параметрами:  $T_0=f(X)$ .

Совокупность диагностических методов контроля основана на контроле информативных параметров. При этом предполагается, что отказы обуславливаются деградиационными физико-химическими процессами, влияющими на измеряемые параметры.

Отбраковочные испытания (ОИ) и диагностический неразрушающий контроль (ДНК) проводят с целью выявления потенциально-ненадежных ЭРИ и ЭРИ со скрытыми дефектами. Объем ОИ и ДНК устанавливается в ведомостях отбраковочных испытаний.

Отбраковочные испытания предназначены для исключения несоответствующих компонентов (компонентов со случайными дефектами [349]), которые могут привести к

раннему отказу, известному как «детская смертность») из приемлемой без этих компонентов партии, что позволяет повысить уверенность в надежности выбранных компонентов.

Квалификационные испытания состоят из механических, электрических и климатических испытаний, они предназначены для проверки того, что материалы, конструкция, характеристики и долгосрочная надежность компонентов соответствуют спецификации и предполагаемому применению, а также для подтверждения того, что производитель и его технологические процессы согласуются от партии к партии.

Снижение нагрузок представляет собой уменьшение электрических и тепловых нагрузок, приложенных к компоненту во время нормальной работы, чтобы уменьшить скорость деградации и продлить предполагаемый срок службы.

Применение отбраковочных испытаний основано на том, что форсированная нагрузка не должна приводить к появлению новых механизмов отказов, т.е. должно выполняться условие автономности механизмов отказов. Физический подход к выбору ускоряющих факторов предполагает знание закономерностей возникновения и развития отказов и определение влияния различных факторов на скорость развития деградационного процесса. Поэтому при определении условий форсированных испытаний необходимо выбирать номенклатуру и значение факторов, характерных для условий эксплуатации.

При изготовлении интегральных микросхем (ИМС) проводят стопроцентные отбраковочные испытания, в состав которых включено и испытание на воздействие изменения температуры среды. А при проведении приемосдаточных испытаний на заводе-изготовителе проводят испытания на воздействие изменения температуры среды на выборке, которая не подлежит поставке потребителю [362].

Испытание на воздействие изменения температуры среды проводят для выявления дефектов корпуса [363]. Учитывая, что данный вид испытаний относится к разрушающим, опасно не только повредить элемент и отбраковать его, но, что намного опаснее, - внести дефекты, которые могут проявиться при последующей эксплуатации.

На все электрические параметры компонентов, применяемых в космической отрасли, устанавливаются ужесточённые нормы. Определение ужесточённых норм осуществляется в следующей последовательности:

1. Измерение электрических параметров для выборки компонентов.

2. Статистическая обработка результатов измерений, в том числе:

- построение гистограммы распределения результатов измерений для каждого параметра;

- оценка типа распределения и расчёт параметров распределения.

3. Расчёт границ допустимых значений ужесточённых норм по следующей формуле:

$$X_{max} = \bar{x} \pm k \cdot \sigma,$$

где  $\bar{x}$  – среднее значение параметра,  $k$  – толерантный коэффициент,  $\sigma$  – среднеквадратичное отклонение.

Отклонения параметров изделий в производственной партии вызвано не обязательно нарушением технологического процесса. Причинами отклонений и разброса параметров является неизбежная неоднородность структуры качественно изготовленного материала сырья [364], неточность измерительных приборов и др. Поэтому дальнейшее ужесточение требований норм приводит к необоснованной отбраковке качественных изделий без повышения качества продукции.

Необходимо отметить, что для оценки достаточности принятых мероприятий на входном контроле очень важна информация о надёжности ЭРИ. Оперативность получения такой информации из сферы эксплуатации (обратная связь) чрезвычайно низка. Дополнительные испытания у потребителя ЭРИ [356, 342], повышают качество применяемых партий ЭРИ, идущих на комплектацию КА, снижают интенсивность отказов выборки ЭРИ за счет исключения ЭРИ со скрытыми дефектами и позволяют применять понижающий коэффициент  $K_{умц}$ , уточняющий значения базовой интенсивности отказов  $\lambda'_o$ , приводимую в справочниках по надёжности [365]. Причем наиболее эффективным испытанием является электротермотренировка (ЭТТ) с обязательной оценкой дрейфа параметров и классификацией по данному признаку [342].

Очевидно, что при прогнозировании работоспособности ЭРИ практически невозможно учесть реальный разброс времени наработки до отказа, связанный с разбросом электрофизических параметров ЭРИ и наличием в их структуре различных дефектов и неоднородностей. Поэтому даже при оптимистических результатах прогноза необходимым условием является научно-обоснованный выбор и введение дополнительных методов отбраковочных испытаний. И, в первую очередь, эффективных методов диагностирования до установки электрорадиоизделий в бортовую

аппаратуру. Физические и методологические основы применения методов диагностирования известны [351, 357].

В технологическом процессе изготовления ЭРИ предусмотрен производственный контроль качества, предназначенный для выявления явных дефектов, характеризующих процент выхода годных изделий, и скрытых дефектов, характеризующих надежность выпускаемых ЭРИ. Если рассматривать отказ как результат развития во времени дефекта, характеризуемого размером  $D_i$ , то суть сказанного можно проиллюстрировать графиком рис. 4.1 [353], на котором дана зависимость плотности дефектов ( $a$ ) от их величины ( $D$ ).

Дефекты величиной  $D_i \geq D_0$  (где  $D_0$  – критический размер дефекта, нарушающий работоспособность) приводят к браку или отказу и называются «явными». Дефекты величиной  $D_i < D_0$  обнаруживаются лишь в процессе эксплуатации и называются «скрытыми». Природа генерируемых технологическим процессом явных и скрытых дефектов одна и та же, а разброс начальных размеров дефектов  $D_i$  от 0 до  $\infty$  и определяет на практике уровень брака при изготовлении и разброс характеристик надежности однотипных изделий в одинаковых условиях эксплуатации. Пусть  $D_i$  – некоторая условная величина дефекта, при которой его развитие не достигает в течение времени  $t_i$  эксплуатации критического размера  $D_0$ . Тогда изделия с дефектами  $D_{ti} < D_i < D_0$  образуют группу потенциально ненадежных изделий с точки зрения заданных требований по надежности. На рис. 4.1 показана кривая интенсивности отказов интегральных схем (ИС)  $\lambda(D_i)$  в зависимости от величины скрытых дефектов. Нарастающий характер кривой  $\lambda(D_i)$  в диапазоне  $D_{ti} < D_i < D_0$  и определяет так называемую проблему «ранних» отказов в эксплуатации.

Необходимый уровень качества и надежности ЭРИ определяется на этапе оценки вероятности безотказной работы аппаратуры и может быть использован для разработки программы дополнительных отбраковочных испытаний.

Дополнительные отбраковочные испытания не повторяют испытания, проводящиеся на заводе-изготовителе. Минимально необходимый объем дополнительных отбраковочных испытаний должен включать [336] измерение электрических параметров, выборочный разрушающий физический анализ, испытания на наличие посторонних частиц в подкорпусном объеме, электротермотренировку.



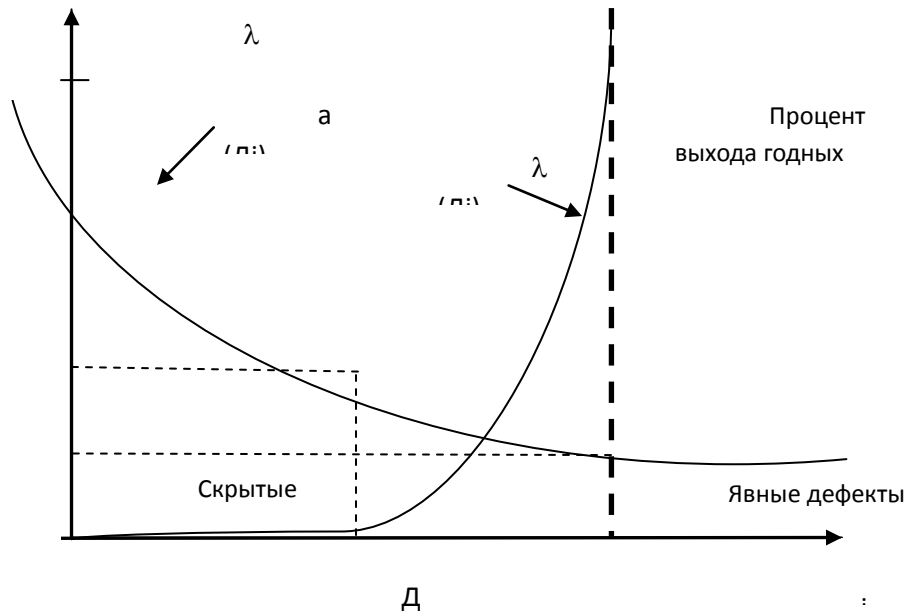


Рисунок 4.1 – Зависимость плотности дефектов от их величины

При выборе температурных режимов электротренировки принимаются во внимание коэффициенты ускорения деградационных явлений для различных энергий активации дефектов. К настоящему времени нет достаточно точных количественных оценок коэффициентов ускорения развития потенциальных дефектов, что связано с недостаточным знанием закономерностей дефектообразования в реальных элементах, а также сложностью и высокой стоимостью экспериментального получения таких оценок [366], что не снижает ценность получаемых при электротренировке данных для оценки однородности производственных партий.

### 4.3 Особенности комплектации космических аппаратов электрорадиоизделиями зарубежного производства

После того, как принято решение о применении в аппаратуре, особенно в космической технике, зарубежной ЭКБ, встаёт вопрос о закупке этой ЭКБ соответствующего качества. Наиболее высоким качеством характеризуются американские интегральные микросхемы (ИМС) классов S и B, а также европейские ИМС классов B и C [338, 323], надёжность которых выше надёжности коммерческих

ИМС примерно в 80-100 раз [367]. Приблизительно таково же и различие в стоимости между ИМС космического и коммерческого применения.

Уровни качества европейских и американских ИМС, ориентированные на космическое, военное, промышленное и коммерческого применения, обобщены и сведены в таблицу 4.1.

В последнее время в нашей стране и за рубежом наметилась тенденция к применению в радиоэлектронной аппаратуре (РЭА) космических систем элементов промышленного класса вместо дорогой и труднодоступной зарубежной ЭКБ космического класса после проведения широкой программы отбраковочных, сверхотбраковочных и диагностических испытаний на основе анализа методов неразрушающего контроля, а также конструкционного контроля и физического анализа представленных выборок ИМС от каждой партии.

Таблица 4.1 – Уровни качества зарубежных интегральных микросхем

Уровень качества ( $\pi_Q$ )	Европейские ИМС	Американские ИМС
Космический, $\pi_Q \cong 0,25$	Уровень В (приёмочные испытания LAT1, LAT2, LAT3)	JANS QPL38510
		CLASS V QML38535
Военный, $\pi_Q \cong 1,0$	Уровень С (приёмочные испытания LAT1, LAT2, LAT3)	JANS QPL 38510
		CLASS Q QML38535
Средний военный, $\pi_Q \cong 2,0$	Экспериментальный уровень (уровень С без приёмочные испытания LAT1, LAT2, без записи данных и визуального контроля перед герметизацией)	SMD (38536 ClassM). Производственная линия, соответствующая уровню качества В, испытания в соответствии с пар. С 1.2.1 MIL-STD-883
Низший военный, $\pi_Q \cong 5,0$	Объём испытаний сокращён по сравнению с классами В и С стандарта ESA/SCC 9000	Объём испытаний сокращён по сравнению с классами S и B MIL-STD-883
Промышленный, $\pi_Q \cong 10,0$	ИМС негерметичные с дополнительными температурными испытаниями: промышленный температурный диапазон от -25 до +100С	
Коммерческий, $\pi_Q \cong 20,0$	ИМС негерметичные без отбраковочных испытаний: коммерческий температурный диапазон от 0 до +80 С	

Интенсивная процедура отбора позволяет существенно повысить качество партий ИМС и в среднем в 10-30 раз снизить интенсивность отказов в партиях за счет исключения из них ИМС со скрытыми дефектами.

Реализация дополнительных отбраковочных испытаний при создании КА «SESAT» [368, 369], первого отечественного КА с гарантируемым (и фактическим на сегодня) сроком активного существования 10 лет, положила начало одного из основных направлений обеспечения надежности функционирования электронной аппаратуры и космических аппаратов в целом.

В российской и зарубежной литературе [338, 323, 350, 370, 371-373, 352, 364] подробно рассмотрены вопросы прогнозирования отказов, связанных с возникновением дефектов на этапе производства ЭРИ. в основном проводимые ими работы направлены на выявление и устранение дефектов в производственном процессе изготовления ЭРИ и сырья для них – кремниевых пластин [352, 372]. При этом предполагается, что тестовые испытания проводятся производителем ЭРИ, полностью контролирующим производственный процесс. В то же время, разработка методов эффективного контроля качества уже выпущенных партий на предмет соответствия заявленных и фактических условий производства по косвенным признакам, выявленным в результате тестовых испытаний партий ЭРИ и прогнозирование отказоустойчивости ЭРИ с использованием ретроспективного анализа данных о ранее выпущенных партиях представляется важной задачей, способной значительно увеличить время жизни летательных аппаратов в целом. Особую актуальность проблема приобретает при использовании компонентной базы зарубежного производства, когда непосредственный контроль на этапе выпуска продукции затруднен или невозможен.

В процессе испытаний и приемки ЭРИ предполагается двух заинтересованных лиц – производителя и потребителя (либо специального тестового центра). При этом тестовые испытания проводятся как производителем (выходной контроль), так и потребителем (при возможном участии посредника – тестового центра), который не располагает полной информацией о производственном процессе, что особенно актуально при использовании ЭРИ импортного производства.

#### 4.4 Проблема создания специальных партий ЭРИ

Основные характеристики качества и надежности определяются в процессе производства на заводе-изготовителе. Как отмечалось выше, в России отсутствует специализированное производство ЭКБ для космической отрасли, как это организовано в США и Западной Европе [374]. Большинство производителей электронных компонентов не стремятся к сотрудничеству с космической отраслью [347] вследствие высоких требований. В 2002 году 22ЦНИИ МО РФ выпустил проект документа, определяющего требования к организации производства и правилам приемки ЭКБ уровня качества «ОСД», но документ до сих пор не был реализован.

Поэтому при комплектации КА приходится закупать ЭКБ самого высокого доступного российским потребителям уровня качества, и затем проводить серьезную работу по отбраковке потенциально ненадежных компонентов. При этом, в случае использования импортных компонентов, предпочтение отдается той номенклатуре изделий, для которой имеются аналоги уровня качества Space. Однако опыт показал, что серийная продукция заводов-изготовителей не всегда удовлетворяет предъявляемым требованиям [347, 375], выполнение которых позволило бы повысить срок службы неремонтопригодных изделий в потенциально агрессивной космической среде [376, 377], вследствие чего организация работы с заводами-изготовителями по выпуску специальных партий ЭКБ для космической отрасли (так называемых «спецпартий») представляется актуальной.

Спецпартия является аналогом (прообразом) компонентов космического уровня качества. Характеристики отдельных изделий, входящих в нее, должны быть не ниже категории качества «ВП» или «ОС». Кроме того, характеристики всей совокупности изделий, входящих в спецпартию, разброс их эксплуатационных характеристик [333, 378], должны отличаться от параметров обычных партий.

Для того чтобы сформировать список дополнительных требований к спецпартиям ЭКБ, был проведен сравнительный анализ последовательности испытаний по американским стандартам MIL для интегральных схем (ИС) и полупроводниковых приборов (ППП) категории качества «SPACE» (для космического применения) и «MILITARY» (для военного применения) и требований Российских стандартов для категорий качества «ВП», «ОС» и «ОСД».

В результате проведенного сравнительного анализа последовательности испытаний по американским стандартам MIL для интегральных схем и полупроводниковых приборов категории качества «SPACE» (для космического применения) и «MILITARY» (для военного применения) [340] и требований Российских стандартов для категорий качества «ВП», «ОС» и «ОСД» выяснилось, что ЭКБ категорий качества SPACE и MILITARY имеют, кроме требования однородности партий сырья, лишь два отличия: оценка дрейфа параметров и контроль наличия посторонних частиц в подкорпусном пространстве (PIND), которые восполняются дополнительными отбраковочными испытаниями [379].

Для оценки дрейфа параметров необходимо присвоить каждому элементу (корпусу) индивидуальный номер, провести измерения электрических параметров (обычно – в объеме не менее приемосдаточных испытаний завода-изготовителя), провести электротермотренировку не менее 72 часов, повторные измерения электрических параметров после завершения ЭТТ в течение 6-24 часов, рассчитать дрейф параметров и оценить допустимые границы дрейфа.

Испытания проводятся в испытательном техническом центре, поскольку ни один завод-изготовитель в России не в состоянии откорректировать свой технологический процесс в соответствии с требованиями. В результате спецпартии являются совместным продуктом заводов-изготовителей и испытательно-тестовых центров (ИТЦ). От каждой партии ЭКБ берутся образцы, на которых проводится разрушающий физический анализ (РФА) для оценки технологических дефектов, которые обычно не выявляются на этапе электрических отбраковочных испытаний, а проявляются как временной фактор. В составе РФА входит исследование на растровом электронном микроскопе (РЭМ), контроль подкорпусной влаги, испытания на сдвиг кристалла и обрыв выводов, проверка паяемости и др.

В соответствии с нормативной действующей документацией [354] ЭКБ иностранного производства [358] в ракетно-космической технике необходимо сертифицировать с целью документального подтверждения соответствия требованиям потребителя. Такая сертификация должна подтвердить соответствие характеристик требованиям модели внешних воздействующих факторов.

Работа по формированию требований к спецпартиям проводилась в рамках международного проекта TACIS TELRUS 9503 (1996-1998 гг.), направленного на

создание в России двух Центров по сертификации ЭКБ космического назначения. В ходе проекта был проведен тщательный анализ и сравнение российского и западноевропейского подходов к квалификации, а также требований к испытаниям ЭКБ. Российские специалисты из ИСС им. Решетнева (в то время – НПО ПМ) прошли обучение в западноевропейских центрах, что позволило разработать схему соответствия российских и западноевропейских требований к ЭКБ. Немаловажную роль в определении понятия спецпартии сыграл опыт международного сотрудничества при создании спутника SESAT [368, 369] с 10-летним сроком активного существования и 12-летним техническим ресурсом, который успешно функционирует на орбите более 13 лет. На этом КА впервые применена технология комплектации ЭКБ с проведением индивидуальной отбраковки, РФА и выпуска документов, разрешающих использование конкретных типов ЭКБ в радиоэлектронной аппаратуре КА с указанием условий использования.

На начальном этапе в отечественной аппаратуре использовались зарубежные электрорадиоизделия исключительно индустриального класса, которые поставлялись без каких либо документов качества. При этом требовалось сложное контрольно-испытательное оборудование, программно-методическая документация, благодаря чему испытания могли затянуться на несколько месяцев, и суммарные затраты на порядки превышали стоимость самих изделий. В дальнейшем доля элементов Space, MIL, HI-REL в современной бортовой аппаратуре космических аппаратов значительно увеличилась и имела тенденцию к дальнейшему росту, чему объективно способствовали отказы ЭКБ ИП коммерческого (индустриального) уровня качества при испытаниях и эксплуатации бортовой аппаратуры. ЭКБ для космического применения изготавливаются в соответствии с общими и детальными спецификациями, утвержденными в установленном порядке соответствующих стран-производителей, в процессе изготовления подвергаются отбраковочным и квалификационным испытаниям, которые превосходят отечественные требования. В то же время отсутствие официальных межгосударственных актов по использованию ЭКБ на территории РФ и, соответственно, механизмов идентификации их подлинности, не позволяет применять без сертификации ЭКБ ИП даже, казалось бы, высших уровней качества Space и MIL.

Практика поставки ЭКБ уровней качества Space и MIL официальными

дистрибьюторами изготовителей ЭКБ показывает, что по запросу изготовители ЭКБ ИП поставляют сопроводительную документацию, содержащую отчеты по испытаниям, при наличии которых сертификация может полностью или частично проведена анализом этих отчетов на соответствие подтвержденных характеристик закупленных ЭКБ ИП требованиям модели внешних воздействующих факторов. При этом отчеты должны содержать те же данные, что и результаты испытаний, проводимых российскими ИТЦ.

При этом анализ сопроводительной документации выявил ряд проблемных вопросов, вызванных различиями в документации на иностранную и отечественную ЭКБ, в частности:

- испытания на воздействие синусоидальной вибрации для ЭКБ ИП проводятся в частотном диапазоне 20-2000 гц для интегральных микросхем (ИС) и 100-2000гц для полупроводников, а требования на аппаратуру КА – 5-2000гц;
- испытания на воздействие механического одиночного удара для ЭКБ ИП проводятся в режиме: амплитуда 1500g , длительностью 0,5 мс, требования на аппаратуру – частотный спектр в диапазоне 35-10000 гц, амплитуда до 5000g;
- требования стойкости к воздействию акустического шума, повышенной влажности, пониженного давления и сохраняемости, а, соответственно, и испытания по подтверждению в иностранной НД не предусмотрены;
- требования к сроку службы (ресурсу) ЭКБ ИП отсутствуют.

Главной причиной перехода к производству спецпартий – российского аналога класса качества Spase – является фактическая недоступность зарубежных изделий высших классов качества в современных условиях (необходимость импортозамещения и развития отечественных высокотехнологичных отраслей промышленности).

Проведенный анализ показал, что задача классификации ЭРИ по производственным партиям на основании результатов отбраковочных испытаний является недостаточно исследованной.

Современные исследования в данной области в основном сосредоточены на поиске причин возникновения производственного брака, в то время как задача отбора однородных по своим характеристикам качественных изделий в сборной поставляемой партии остается не исследованной.

Поскольку анализ результатов тестовых испытаний ЭРИ – задача, в результатах решения которой заинтересованы две стороны – изготовитель ЭРИ и изготовитель

электронной аппаратуры либо специальный тестовый центр – используемый метод анализа должен обеспечивать стабильность и высокую точность получаемых результатов за приемлемое время.

Современное состояние решаемой научно-технической проблемы наиболее полно отражено в работах В.И. Орлова и В.В. Федосова [336, 380, 381].

Целью разработки является выявление однородных партий в массивах поступающих от производителя электрорадиоизделий (ЭРИ) путем анализа данных, получаемых по результатам проведенных диагностических испытаний, что необходимо для проведения последующих испытаний ЭРИ (разрушающего физического анализа).

С учетом необходимости выполнения требований точности классификации и времени выполнения расчетов в качестве метода, вычисляющего начальное расположение центров групп, выбран генетический алгоритм, который в ходе разработки модернизирован для решения данной задачи. Таким образом, стандартный метод k-средних (ALA-алгоритм) в сочетании с применяемым генетическим алгоритмом представляют собой единую процедуру автоматической классификации ЭРИ.

Использование генетического алгоритма с жадной эвристикой для оценки однородности партий электронной компонентной базы позволяет проводить такую оценку в реальном времени, одновременно с проведением неразрушающих и разрушающих тестов. Результат работы алгоритма может быть легко визуализирован, что предоставляет дополнительный удобный инструментарий оценки качества партий.

Разработанные алгоритмы реализованы в виде программного приложения. Разработанные алгоритмы показали свою работоспособность при автоматической классификации на исходных данных и позволили удостовериться в правильности выбора направления разработки.

При работе над проектом создания автоматизированной системы группировки электрорадиоизделий космического применения решались, в том числе, следующие задачи:

- Исследование применимости различных методов машинного обучения и анализа данных для решения задачи автоматической группировки ЭРИ.
- Выбор наилучшего подхода, модификация методов (при необходимости) и настройка алгоритмов.



- Учет особенностей задачи и исходных данных, в том числе нестабильности испытаний (учет погрешности приборов измерений).

- Разработка процедур обоснования (подтверждения) результатов анализа данных: критерии определения числа групп ЭРИ (предполагаемых производственных партий), оценка достоверности отнесения объекта к группе.

- Реализация разработанных средств в программном приложении.

Решение поставленных задач осуществлялось методами машинного обучения и анализа данных совместно с применением методов поисковой оптимизации.

Методы интеллектуального анализа данных [49, 287] предназначены для выявления новых знаний из набора данных, в данном случае это знания о выявляемых группах изделий. Методы машинного обучения [382] направлены на создание классификатора, способного соотносить конкретные экземпляры изделий с выявленными группами изделий. Методы поисковой оптимизации, в свою очередь, являются инструментом, способствующем наилучшей реализации упомянутых средств, в частности, для оптимизации выделения групп и установления центров групп.

Исходными данными для анализа при решении задачи являются результаты тестовых воздействий на ЭРИ по контролю вольт-амперных характеристик входных и выходных цепей микросхем. Данные представляют собой таблицу, в строках которой приведены последствия различных электрических воздействий на элементы набора однотипных ЭРИ. Предполагается, что изделия с различающимися эксплуатационными характеристиками (фактически различные партии ЭРИ) будут иметь различия в полученных результатах испытаний. Такое предположение дает возможность применения методов анализа данных для осуществления необходимой группировки ЭРИ. Пример формата исходных данных приведен в Приложении А.

Подробные результаты сравнительных испытаний алгоритмов автоматической группировки приведены в Приложении Б.

В результате выполнения проекта реализовано программное приложение для решения задачи автоматической группировки ЭРИ. Программное приложение написано на языке объектно-ориентированного программирования python с использованием модулей на языке Fortran. Программа является частью комплексной автоматизированной системы, включающей средства организации пользовательского

интерфейса, базы данных результатов тестовых испытаний и средств интеграции с существующей системой.

#### **4.5 Методы машинного обучения и анализа данных для задачи автоматической группировки объектов электрорадиоизделий**

Как отмечалось в Главах 1 и 3, наиболее популярной моделью автоматической группировки (классификации без учителя) является задача  $k$ -средних. Данную задачу  $k$ -средних можно отнести к задачам непрерывной теории размещения [114]. Задача сводится к нахождению  $k$  точек (центров, центроидов, медоидов) в  $d$ -мерном пространстве характеристик, таких, при которых сумма расстояний от каждого из вектора данных (набор результатов измерений) до ближайшего к нему из  $k$  центров достигает минимума:

$$F(X_1, \dots, X_k) = \sum_{i=1}^n \min_{x \in \{x_1, \dots, x_k\}} \|A_i - X\|_2^2.$$

Требования качества, предъявляемые к электронным компонентам в космической отрасли, настолько высоки, что определять принадлежность изделий к одной либо различным производственным партиям приходится по расхождениям (расстояниям) совокупности измерений, едва превышающим точность измерений. Таким образом, результаты каждого из измерений фактически представляют собой дискретные значения, шаг изменения которых определяется точностью измерительного прибора.

Как отмечалось в предыдущей Главе, качестве меры расстояния в задачах автоматической группировки чаще всего используется квадратичные евклидовы расстояния [114]. В то же время выбор прямоугольной метрики позволяет получать результаты той же точности, что и исходные векторы данных (значение каждой координаты вектора-результата совпадает со значением соответствующей координаты одного из векторов данных) [71]. Кроме того, выбор прямоугольной метрики позволяет существенно снизить влияние выбросов, вызванных наличием в партиях единичных бракованных экземпляров изделий либо ошибками измерений.

Поскольку результаты измерений различных параметров имеют различный диапазон значений и разные единицы измерения, проводится нормировка данных. При этом для нормировки используются данные не только исследуемой партии изделий,

разброс показаний одного отдельно взятого параметра может быть очень незначительным, но и предыдущих исследованных партий электрорадиоизделий.

Результаты тестовых испытаний каждой партии ЭРИ представляют собой массивы многомерных данных: в состав испытаний каждого изделия могут входить измерения до нескольких тысяч параметров. При этом мы исходим из того, что все измерения информативны: как говорилось в предыдущем параграфе, в состав дополнительных отбраковочных испытаний включаются лишь испытания, позволяющие выявить тот или иной тип дефекта, при этом испытания не являются повторением отбраковочных испытаний, проводимых на этапе входного контроля. Таким образом, мы считаем, что уменьшить размерность данных без потери информации невозможно.

Альтернативным способом автоматической группировки (разбиение множества элементов на однородные подмножества) является решение задачи  $k$ -медоид [82]. Данная задача предполагает выбор в качестве центра кластера (подмножества элементов) одного из векторов данных – вектора характеристик в  $d$ -мерном пространстве характеристик. Применительно к задаче классификации ЭРИ решение задачи  $k$ -медоид сводится к выбору некоторых «центральных» экземпляров каждого из кластеров (групп ЭРИ в партии). Такое решение упрощает интерпретацию результатов разбиения на кластеры специалистом (выбирается не гипотетический, а реальный экземпляр изделия). В то же время, как показано в следующих параграфах, результат решения задачи  $k$ -медоид крайне неустойчив к малым изменениям исходных данных, что весьма актуально с учетом того, что колебания параметров соразмерны предельной точности измерительных приборов.

При решении задачи  $k$ -медоид для определения центра кластера (группы) в общем случае требуется перебор в качестве вариантов такого центра всех входящих в кластер векторов данных, что усложняет поиск решения. В то же время, существуют гораздо более быстрые аналогичные процедуры локального поиска [82], которые, хотя и не гарантируют точного решения задачи поиска центра кластера, являются более эффективными во многих практических задачах большой размерности.

Зависимость результатов ALA-алгоритма, как и любой процедуры локального поиска, от заданных начальных значений является серьезной проблемой также и с точки зрения воспроизводимости результатов работы алгоритма классификации: при разных запусках алгоритма, в зависимости от выбора начальных значений центров кластеров,

одни и те же векторы данных могут относиться к различным кластерам (в терминологии классификации элементной базы – относиться к различным партиям) либо к одному и тому же кластеру (к одной производственной партии). То же (возможно, в меньшей степени) относится и к другим рандомизированным методам, таким как генетический алгоритм. Таким образом, требуется разработка алгоритма автоматической группировки, дающего достаточно точный («почти» оптимальный) стабильный результат. Иными словами, требуется повышение точности используемого оптимизационного алгоритма.

Как показывают результаты вычислительных экспериментов приведенные в предыдущей Главе, детерминированные алгоритмы, такие как Information Bottleneck Clustering, будучи избавленными от проблемы нестабильности результата, часто дают далеко не самый лучший результат в сравнении с генетическим алгоритмом с жадной эвристикой и другими рандомизированными алгоритмами.

Важной задачей является задача выявления среди изделий, прошедших входной контроль, групп (партий) изделий, произведенных в одинаковых условиях из идентичного сырья. Эта задача решается как задача  $k$ -средних, которую можно считать частным случаем  $p$ -медианной задачи. В общем случае непрерывная  $p$ -медианная задача сформулирована в (3.1).

В качестве метода локального поиска используется ALA-процедура. Весовые коэффициенты мы здесь принимаем равными 1.

В случае использования в качестве модели  $p$ -медианной задачи с прямоугольной метрикой каждая из координат центра кластера определяется как медианное значение данной координаты векторов данных, входящих в кластер, что повышает интерпретируемость результатов.

Разработанный в настоящей диссертации (см. предыдущую Главу) детерминированный алгоритм с жадной эвристикой способен за приемлемое время решать задачи  $k$ -средних,  $k$ -медоид, а также иные  $k$ -медианные ( $p$ -медианные) задачи. При этом получаемые результаты по точности уступают результатам эволюционных алгоритмов, но являются при этом детерминированными, что облегчает контроль и интерпретацию результатов всеми заинтересованными сторонами. Точность и скорость работы программной реализации алгоритма вполне приемлемы для решения задачи автоматической группировки электрорадиоизделий по классам качества и

производственным партиям на основе данных тестовых испытаний. В то же время, эволюционный алгоритм с жадной эвристикой, разработанный в предыдущей главе дает все же более точные результаты при меньших временных затратах. Получаемые результаты при этом, хотя и не являются детерминированными, весьма стабильны – рандомизированные методы, основанные на модификациях жадных агломеративных эвристиках, предложенных в предыдущей Главе, дают наименьший разброс значения целевой функции, что показано на тестовых примерах и на реальных данных.

#### 4.6 Проблема сопоставимости точности входных данных и результата

Использование классического метода  $k$ -средних с квадратичной евклидовой мерой расстояния ( $l_2^2$ ) имеет неоспоримое преимущество: при использовании квадратичных евклидовых расстояний определение центра кластера представляет собой простейшую задачу, решаемую за один шаг – определяется среднее значение каждой координаты векторов данных, входящих в кластер, полученные значения являются координатами центра кластера [114].

Если центр  $i$ -го кластера  $X_i=(x_{i,1},\dots,x_{i,d})$  – вектор в  $d$ -мерном пространстве, и векторы данных  $A_j=(a_{j,1},\dots,a_{j,d})$ ,  $j=\overline{1,n}$  соответственно, также имеют  $d$  измерений, то новый центр определяется так [383]:

$$x'_{i,k} = \sum_{y \in C_i} y_k / |C_i|, k = \overline{1,d}.$$

Гораздо более сложной является задача с евклидовой метрикой ( $l_2$ ). В этом случае центр кластера представляет собой решение задачи Вебера [114, 144], определяемое с помощью итеративной процедуры Вайсфелда [144] или ее модификаций. Процедура представляет собой алгоритм градиентного спуска [385] и дает приближенное решение задачи. При этом существенно возрастает вычислительная сложность, что создает дополнительные проблемы использования алгоритма в интерактивном режиме.

Надо отметить, что использование вышперечисленных метрик имеет существенный недостаток с точки зрения интерпретируемости результатов. Задача классификации по производственным партиям ставится как с целью повышения качества электронных узлов путем их комплектации электронными компонентами, произведенными в рамках одной производственной партии и имеющими очень близкие эксплуатационные характеристики, так и с целью дальнейшего взаимодействия с

производителем или поставщиков электронных компонентов в направлении повышения качества поставляемой продукции. Если в поставляемой партии обнаружено фактическое количество производственных партий большее, чем заявленное поставщиком, потребитель электронных компонентов или специализированный испытательный центр должны обосновать отказ в приемке партии изделий.

Требования качества, предъявляемые к электронным компонентам, например, в космической отрасли, настолько высоки, что определять принадлежность изделий к одной либо различным производственным партиям приходится по расхождениям (расстояниям) совокупности измерений, едва превышающим точность измерений.

Таким образом, результаты каждого из измерений фактически представляют собой дискретные значения, шаг изменения которых определяется точностью измерительного прибора.

Например, приведем в табл.4.2 некоторые результаты испытаний микросхемы 1526ЛЕ2 по первым семи из 120 измеряемых параметров. В таблице 4.2 приведены результаты испытаний 14 экземпляров изделий, выбранных из сборной партии из 620 изделий, представляющей собой множество изделий, изготовленных в трех различных производственных партиях.

Как видно из таблицы 4.2, значение каждого из измерений колеблется в очень узких пределах.

Для исследуемой сборной партии,  $T_1 \in \{0,001001, 0,001002\}$ ,  $T_i \in \{0,0342, 0,0343, 0,0344, 0,0345, 0,0346, 0,0347, 0,0349\}, i = 2,7, |T_1|=2, |T_i|=7$ .

Остальные значения 117 параметров также колеблются в весьма узких пределах. В то же время, результат процедуры k-средних (центр кластера) может представлять собой, например, следующий набор значений:

$$T_1=0.001001458379, T_2=0.03456397208, T_3=0.035917650834, \dots$$

Но точность измерения параметра  $T_1$  составляет 6 знаков после запятой, параметров  $T_2 - T_7 - 4$  знака.

Центр кластера представляет собой реальное или смоделированное изделие с «идеальными» для данного кластера (т.е. партии изделий) параметрами. Но точность более 7 и 4 знаков после запятой невозможна ввиду отсутствия соответствующих измерительных приборов.

Таблица 4.2 – Результаты испытаний микросхемы 1526JE2

№ партии	T 1	T 2	T 3	T 4	T 5	T 6	T 7
1	0,001001	-0,0343	-0,0343	-0,0343	-0,0343	-0,0342	-0,0343
1	0,001001	-0,0343	-0,0343	-0,0343	-0,0342	-0,0343	-0,0343
1	0,001001	-0,0343	-0,0342	-0,0343	-0,0343	-0,0342	-0,0342
2	0,001001	-0,0345	-0,0345	-0,0345	-0,0345	-0,0345	-0,0345
2	0,001001	-0,0345	-0,0345	-0,0345	-0,0346	-0,0345	-0,0345
2	0,001001	-0,0345	-0,0345	-0,0345	-0,0345	-0,0345	-0,0345
2	0,001001	-0,0344	-0,0344	-0,0344	-0,0344	-0,0343	-0,0344
2	0,001001	-0,0346	-0,0346	-0,0346	-0,0346	-0,0346	-0,0346
3	0,001001	-0,0346	-0,0347	-0,0347	-0,0347	-0,0347	-0,0347
3	0,001001	-0,0347	-0,0347	-0,0347	-0,0348	-0,0347	-0,0347
3	0,001002	-0,0347	-0,0346	-0,0347	-0,0347	-0,0346	-0,0347
3	0,001002	-0,0347	-0,0347	-0,0347	-0,0347	-0,0347	-0,0347
3	0,001002	-0,0346	-0,0346	-0,0346	-0,0346	-0,0346	-0,0346
3	0,001002	-0,0349	-0,0349	-0,0349	-0,0349	-0,0349	-0,0349

Вследствие этого такой результат кластеризации затрудняет интерпретацию результатов специалистом: центр кластера – идеализированный объект в пространстве характеристик (результатов измерений) – предполагает гораздо более высокую точность измерений, чем позволяет имеющееся оборудование. Кроме того, отнесение изделия к тому или иному кластеру (партии) осуществляется по результатам расчета квадратичного евклидова расстояния в нормированном пространстве измерений, в то время как специалисту более понятен диапазон колебания значений параметра в пределах партии.

Выходом в данной ситуации является либо применение прямоугольной метрики (также именуется манхэттенской метрикой ( $l_1$ ) или использование метода  $k$ -медоид ( $k$ -medoids) [250] вместо  $k$ -средних.

При использовании метода  $k$ -медоид на шаге 2 ALA-процедуры вычисляется не центр кластера, а его медоид – вектор данных, принадлежащий кластеру, такой, что

суммарное расстояние от этого вектора до других векторов данных в кластере достигает минимума

$$X'_i = \arg \min_{x \in X_i} \sum_{y \in C_i} \|x - y\|.$$

Такой алгоритм определения минимума достаточно медленно работает, поскольку является комбинаторной процедурой и требует полного перебора всех векторов данных кластера в качестве кандидатур потенциального медоида. Существуют методы аппроксимации данного алгоритма [82, 250, 83], неприемлемые в случае, если точность решения особенно важна.

Другой важной в контексте настоящего раздела особенностью метода  $k$ -медоид является неустойчивость его результатов при небольших изменениях в исходных данных, благодаря чему, например, незначительные погрешности измерений могут приводить к сильно различающимся решениям задачи классификации. Продемонстрируем это свойство на простейшем примере [384]. Пусть требуется определить центр единственного кластера. В случае, изображенном в части А рис.4.2, в качестве медоида может быть выбран любой из векторов данных, значение суммарного расстояния до остальных векторов данных будет одинаковым. Причем это свойство данной задачи сохраняется при выборе любой из метрик и мер:  $l_1, l_2, l_2^2$ .

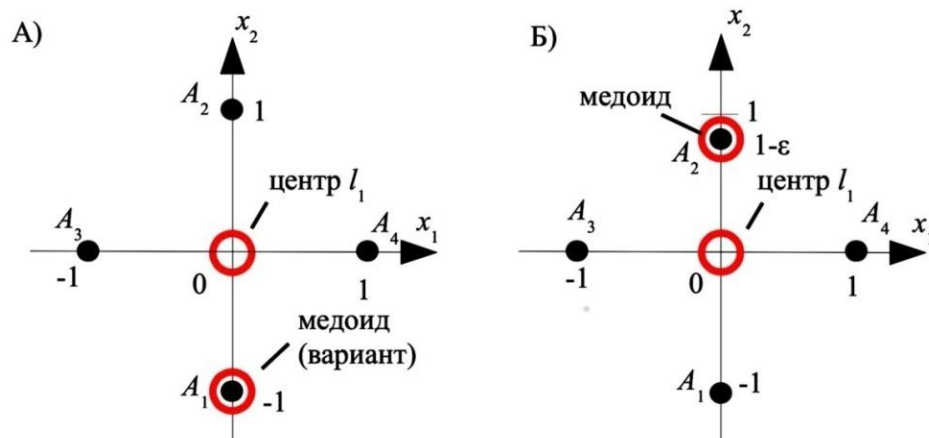


Рисунок 4.2 – Простейший пример выбора центра кластера с прямоугольной метрикой и медоида

В случае же задачи  $k$ -средних (также с любой из указанных метрик) центром будет являться точка  $(0,0)$ . Выбор конкретного вектора данных в качестве медоида зависит от конкретной версии алгоритма кластеризации и технических деталей его реализации. Предположим, что в качестве медоида выбран вектор  $A_1=(0,-1)$ . Перейдем к части Б рисунка 4.2. Здесь медоидом однозначно является вектор  $A_2$ , расположенный от



$A_1$  на весьма значительном расстоянии (в сравнении с расстоянием, на которое сдвинулся вектор  $A_2$ ). В случае же задачи  $k$ -средних центр кластера сместится незначительно, причем в случае использования прямоугольной метрики он останется в точке  $(0,0)$ .

Альтернативой является решение задачи  $k$ -средних в прямоугольной метрике. Целевая функция при этом выглядит следующим образом [114]:

$$F(X_1, \dots, X_k) = \sum_{i=1}^n \min_{x \in \{x_1, \dots, x_k\}} \|A_i - X\|_1.$$

При работе ALA-процедуры с прямоугольной метрикой каждая из координат центра кластера определяется независимо как медианное значение данной координаты векторов данных, входящих в кластер. Процедуру можно описать следующим образом.

Алгоритм 4.1. Определение центра  $i$ -го кластера (медианы) в метрике  $l_1$ .

1. Для каждого  $k = \overline{1, d}$  :

1.1. Отсортировать векторы  $A_i = (a_{j,1}, \dots, a_{j,d}) \in C_i$  по значению  $k$ -й координаты, получить последовательность значений  $a'_{1,k}, \dots, a'_{|C_i|,k}$ . Здесь  $|C_i|$  – мощность множества (кластера).

1.2. Рассчитать  $m = \left\lceil \frac{|C_i|}{2} \right\rceil$ . Присвоить  $x'_{i,k} = a'_{m,k}$ . Здесь квадратными скобками обозначена целая часть числа.

1.3. Повторять цикл 1.

2. Возврат  $X'_i = (x'_{i,1}, \dots, x'_{i,d})$

Данный алгоритм возвращает значение центра кластера, значение каждой из координат которого совпадает с одним из значений соответствующей координаты векторов данных.

Данный алгоритм возвращает значение центра кластера, значение каждой из координат которого совпадает с одним из значений соответствующей координаты векторов данных.

Например, результат кластеризации выборки на 3 кластера, выглядит в соответствии с таблицей 4.3.

Таблица 4.3 – Центры кластеров в прямоугольной метрике (пример)

№ кластера (партии)	T 1	T 2	T 3	T 4	T 5	T 6	T 7
1	0,001001	-0,0343	-0,0343	-0,0343	-0,0343	-0,0342	-0,0343
2	0,001001	-0,0345	-0,0345	-0,0345	-0,0345	-0,0345	-0,0345
3	0,001001	-0,0347	-0,0347	-0,0347	-0,0347	-0,0347	-0,0347

Состав тестовых испытаний с описанием получаемых результатов приведен в Приложении А, таблице А.1.

Следует отметить, что набор параметров (значения координат) центра кластера не обязательно полностью совпадают с параметрами одного из векторов данных, но значение каждого отдельного параметра обязательно совпадает со значением данного параметра одного из векторов данных (т.е. одного из изделий). Соответственно, сохраняется точность значения.

ALA-процедура является алгоритмом локального поиска, ее результаты зависят от выбора начальных центров кластеров. На практике используются следующие варианты запуска ALA-алгоритма:

- многократный запуск из случайно выбранных центров (в качестве центров выбираются векторы данных);
- многократный запуск из центров, выбранных процедурой k-means++ [314];
- генетические алгоритмы для рекомбинации множеств начальных центров.

Хорошие (по точности и стабильности) результаты показывает алгоритм с агломеративной эвристикой [204, 199] в версиях, предложенных в предыдущей Главе [324]. Данный алгоритм адаптирован к задачам кластеризации электронных компонентов. Алгоритм позволяет применять любой вариант ALA-алгоритма. Применительно к рассматриваемой задаче мы применим ALA-алгоритм с квадратичными евклидовыми расстояниями и прямоугольной метрикой.

#### **4.7 Влияние выбора меры расстояния на оценку состава производственных партий**

Сравним результаты разбиения партии микросхем 1526ЛЕ2 на 3 предполагаемые производственные партии с использованием стандартной процедуры *k*-средних с квадратичными евклидовыми расстояниями и процедуры с прямоугольной метрикой.

Результаты разбиения на 3 кластера представлены на рис. 4.3. Для визуализации кластеров на плоскости использована процедура MDS (Multi-dimensional Scaling–многомерное масштабирование) [386] и пакеты программ GNUPLOT и ELKI.

Изменение значения целевой функции в двух рассматриваемых метриках показано на рис. 4.4.

Как видно из рисунков, обе процедуры дают приблизительно одинаковое разбиение на кластеры. Значение целевой функции также меняются незначительно при переходе от разбиения на 3 кластера к разбиению на 4 кластера и более, что позволяет сделать вывод о наличии 3 производственных партий изделий в сборной партии [332, 387-389]. В то же время, результат процедуры с прямоугольной метрикой дает усредненные значения параметров в каждом кластере, совпадающие со значением соответствующего параметра одного из реальных изделий. На работу каждого алгоритма отводилась 1 минута.

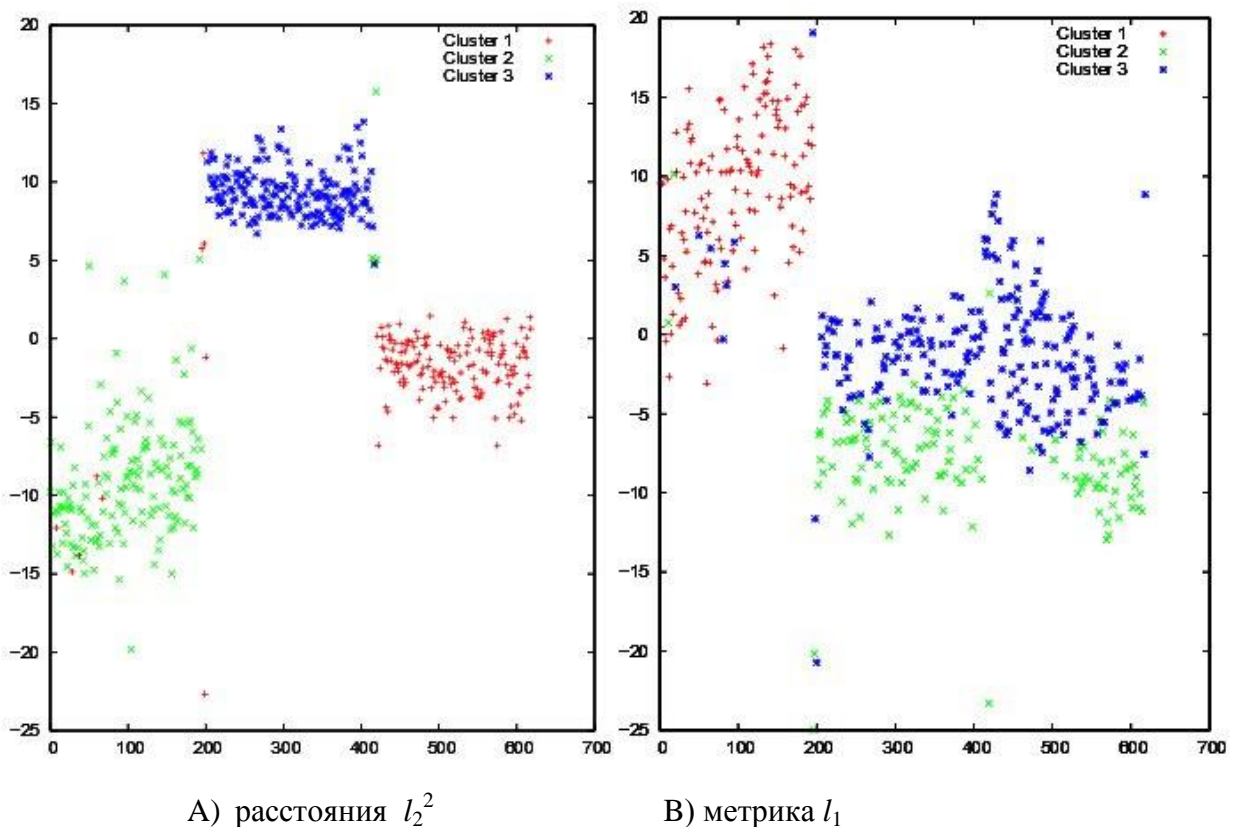


Рисунок 4.3– Визуализация кластеров в различных метриках

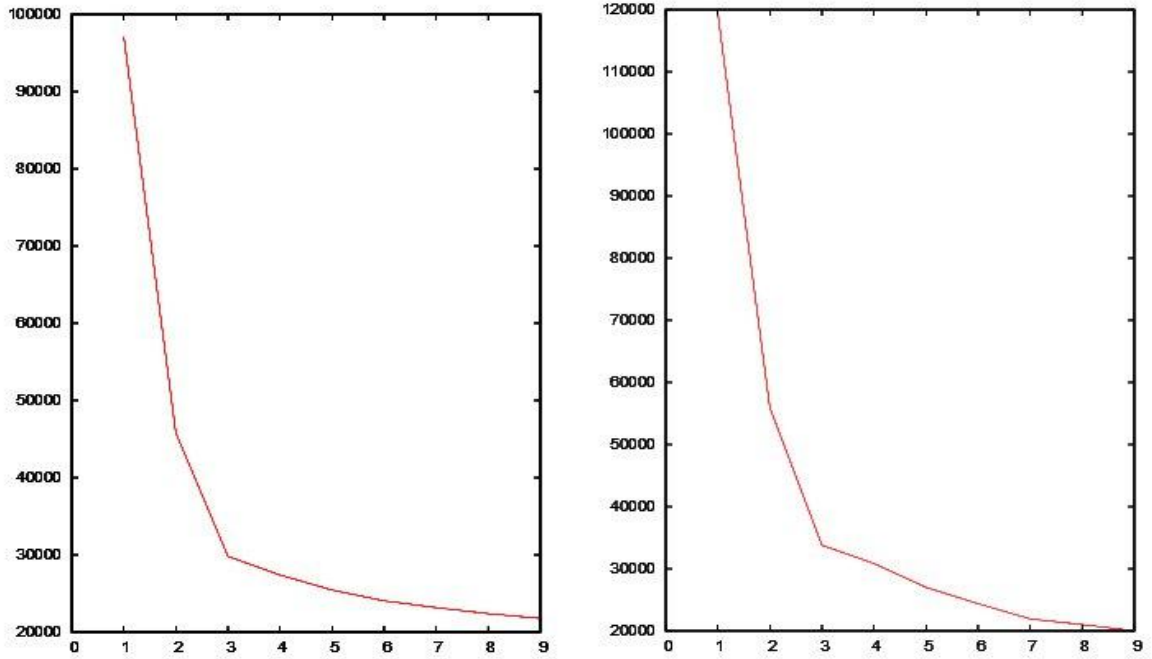
А) расстояния  $l_2^2$ В) метрика  $l_1$ 

Рисунок 4.4 – Зависимость целевой функции от числа кластеров

#### 4.8 Способы оценки числа групп и обоснования результатов группировки ЭРИ

Автоматическое определение числа групп является одной из труднейших задач в классификации данных. Большинство методов автоматического определения числа групп сводятся к проблеме выбора модели. Алгоритмы классификации запускаются при различном числе  $k$ , наилучшее значение  $k$  выбирается на основании определенного критерия. Перечислим основные способы оценки значения  $k$ .

А) Внутрикластерное расстояние – расстояние между объектами одного класса (среднее внутрикластерное расстояние), которое нужно минимизировать:

$$\Phi_1 = \frac{\sum_{i < j} [c(A_i) = c(A_j)] L(A_i, A_j)}{\sum_{i < j} [c(A_i) = c(A_j)]}$$

Для методов, в которых вычисляются центры групп, как и в используемом в данной работе подходе, подобный критерий можно вычислить более эффективно, используя расстояния от объектов до центров групп.

Так, в качестве целевой функции, по которой оценивается качество автоматической группировки ЭРИ и которая используется в разрабатываемой системе, используется суммарное расстояние от объектов выборки до ближайших центров групп:

$$F_1(\chi) = \sum_{i=1}^N \min_{j \in \{1, k\}} L(X_j, A_i).$$

Фактически, этот простейший критерий совпадает с целевой функцией. Чем меньше значение этой функции, тем ближе объекты выборки (экземпляры изделий) расположены к центрам групп, то есть лучше сгруппированы.

При увеличении числа групп значение этого критерия (по результатам работы алгоритма автоматической группировки) убывает (рис.4.5). При переборе возможных значений числа групп в порядке возрастания от 1 до  $K$  это убывания происходит не равномерно – вначале с большей скоростью, затем менее интенсивно.

Наличие точки резкого перегиба (излома) на графике может указывать на наиболее подходящее число групп.

Б) Межкластерное расстояние – расстояние между объектами различных групп:

$$\Phi_2 = \frac{\sum_{i < j} [c(A_i) \neq c(A_j)] L(A_i, A_j)}{\sum_{i < j} [c(A_i) \neq c(A_j)]}$$

Если известны центры групп, то можно использовать подобный критерий, вычисляемый как суммарное расстояние между центроидами:

$$F_2 = \sum_{\substack{i, j=1 \\ i < j}}^k L(X_i, X_j)$$

Чтобы учесть как внутрикластерные так и межкластерные расстояния, можно использовать составной критерий:

$$\frac{\Phi_1}{\Phi_2} \rightarrow \min \text{ или } \frac{F_1}{F_2} \rightarrow \min .$$

В) Индекс Дэвиса-Боулдина (DBI) [390, 391].

Пусть  $C_i$  – это некоторый кластер векторов. Пусть  $X_j$  – вектор признаков размерностью  $n$ , ассоциированный с кластером  $C_i$ .

$$S_i = \frac{1}{T_i} \sum_{j=1}^{T_i} \|X_j - A_i\|_p$$

Здесь  $A_i$  – центроид кластера  $C_i$ ;  $T_i$  – размер кластера  $i$ .  $S_i$  является мерой рассеяния внутри кластера. Обычно величина  $p=2$ , что делает эту функцию евклидовой функций

расстояния. Этот показатель расстояния должен совпадать с метрикой, используемой в модели автоматической группировки.

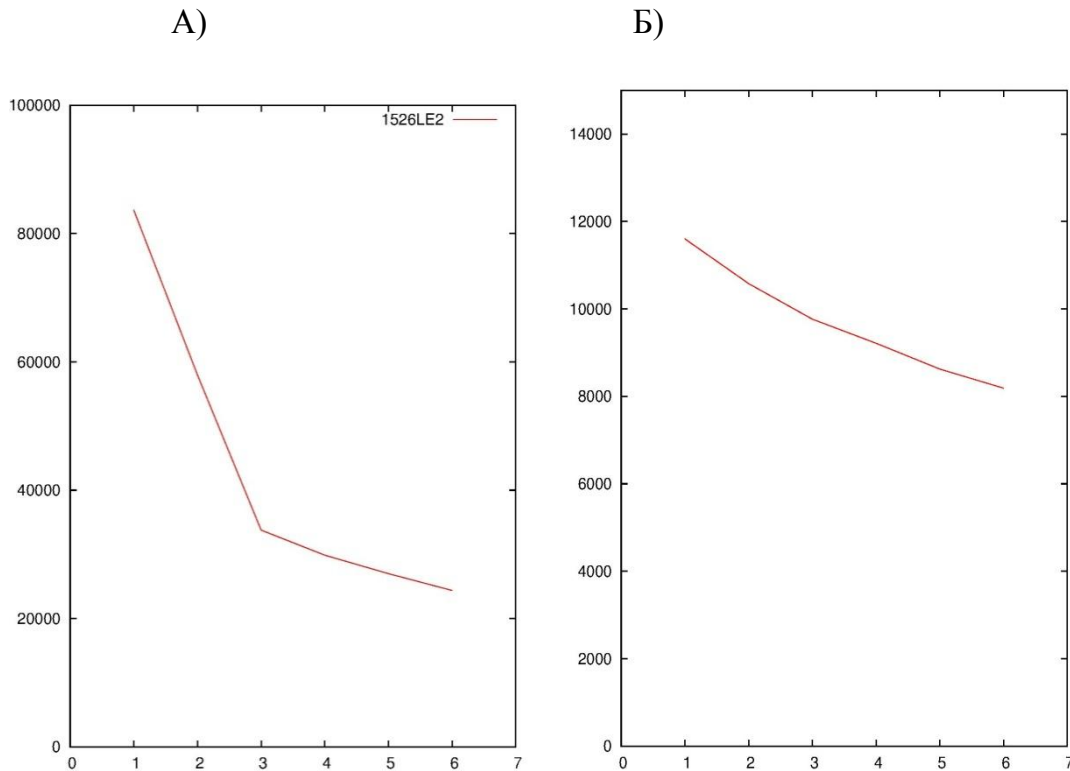


Рисунок 4.5 – Зависимость суммарного разброса параметров в партии от предполагаемого числа парий микросхемы 1526ЛЕ5. А) сборная партия (фактически 3 партии), Б) однородная партия (фактически 1 партия)

Тогда

$$DB \equiv \frac{1}{N} \sum_{i=1}^N D_i,$$

где  $N$  – число групп (кластеров),

$$D_i \equiv \max_{j \neq i} R_{i,j},$$

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}}.$$

$DB$  называется индексом Дэвиса-Боулдина, получаемое значение которого зависит как от алгоритма, так и от самих данных.

Число  $i$ , для которого значение критерия наименьшее, является показателем количества кластеров.

Г) Критерий Calinski-Harabasz. Количество кластеров выбирается как значение аргумента, максимизирующего функцию  $CH(K)$  [392]:

$$CH(k) = \frac{B(k)(k-1)}{W(k)(n-k)},$$

где  $B(k)$  и  $W(k)$ , соответственно, внешняя и внутренняя суммы квадратов элементов данных с  $k$  кластерами. Это один из самых первых предложенных методов. Он оказывается эффективным при данных небольших размерностей (рис.4.6).

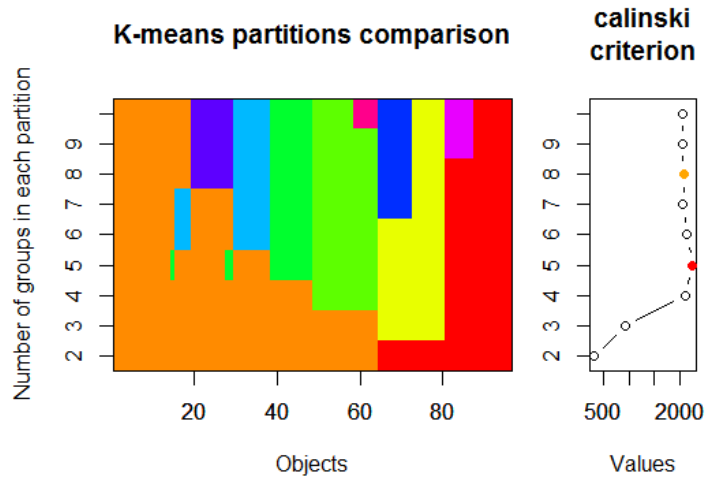


Рисунок 4.6 - Зависимость критерия Calinski-Harabasz от числа кластеров

Д) Критерий Krzanowski и Lai. Данный подход [393] максимизирует функцию  $KL(k)$ :

$$KL(K) = \left| \frac{DIFF(k)}{DIFF(k+1)} \right|$$

где  $DIFF(k) = (k-1)^{2/d} W_{k-1} - k^{2/d} W_k$ ,

$d$  — размерность пространства.

Основная идея заключается в измерении порядка изменчивости внутренних дисперсий.

Е) Критерий Хартигана [394]:

$$H_K = (W_K / W_{K+1} - 1)(N - K - 1),$$

в котором  $N$  — число объектов, а  $W_K$  и  $W_{K+1}$  — минимальные значения критерия суммарного разброса параметров при разбиениях на  $K$  и  $K+1$  кластеров. Согласно правилу Хартигана, надо последовательно разбивать множество на все большее число кластеров  $K$ , до тех пор, пока  $H_K$  не станет меньше 10 — именно это  $K$  и надо брать в качестве оценки числа кластеров.

Ж) Критерий «Силуэт» изначально был создан как средство интерпретации и подтверждения результатов автоматической группировки данных [71, 328].

Предположим, что данные были разгруппированы на  $k$  групп (кластеров). Для каждого экземпляра  $i$  определим  $a(i)$  – среднее расстояние от экземпляра  $i$  до других экземпляров того же кластера. Величина  $a(i)$  показывает, насколько правомерно объект  $i$  отнесен к данному кластеру (чем меньше значение, тем более обосновано отнесение этого объекта группе).

Будем определять среднее расстояние от объекта  $i$  до кластера  $c$  как среднее расстояние от  $i$  до объектов  $c$ . Определим  $b(i)$  как наименьшее среднее расстояние от объекта  $i$  до любого другого кластера, к которому этот объект не относится. Кластер с наименьшим средним расстоянием будем называть «соседним кластером» к  $i$ . Определим силуэт следующим образом:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}.$$

Иначе можно записать:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{если } a(i) < b(i), \\ 0, & \text{если } a(i) = b(i), \\ a(i)/b(i) - 1, & \text{если } a(i) > b(i). \end{cases}$$

Отсюда следует, что

$$-1 \leq s(i) \leq 1.$$

Для того, чтобы  $s(i)$  было близко к 1 требуется  $a(i) \ll b(i)$ . То есть  $s(i)$  близкое к единице означает, что объект правильно классифицирован. Если же значение  $s(i)$  близко к (-1), то это означает, что правильнее было бы отнести этот объект к соседнему кластеру. Если  $s(i)$  около нуля, то это означает, что объект находится на границе двух кластеров (рис.4.7).

Среднее  $s(i)$  по всем объектам кластера является мерой того, насколько плотной является соответствующая группа и того, насколько правильно проведена группировка в целом. Если при группировке было выбрано слишком малое или слишком большое число групп, то этот критерий будет принимать низкое значение. Таким образом, этот критерий можно использовать для выбора числа групп.

Недостатком оригинального критерия является то, что его использование требует значительных вычислительных ресурсов. Поэтому вместо него иногда имеет смысл



использовать ускоренный критерий, который вычисляется намного быстрее, но по качеству практически не уступает оригинальному критерию.

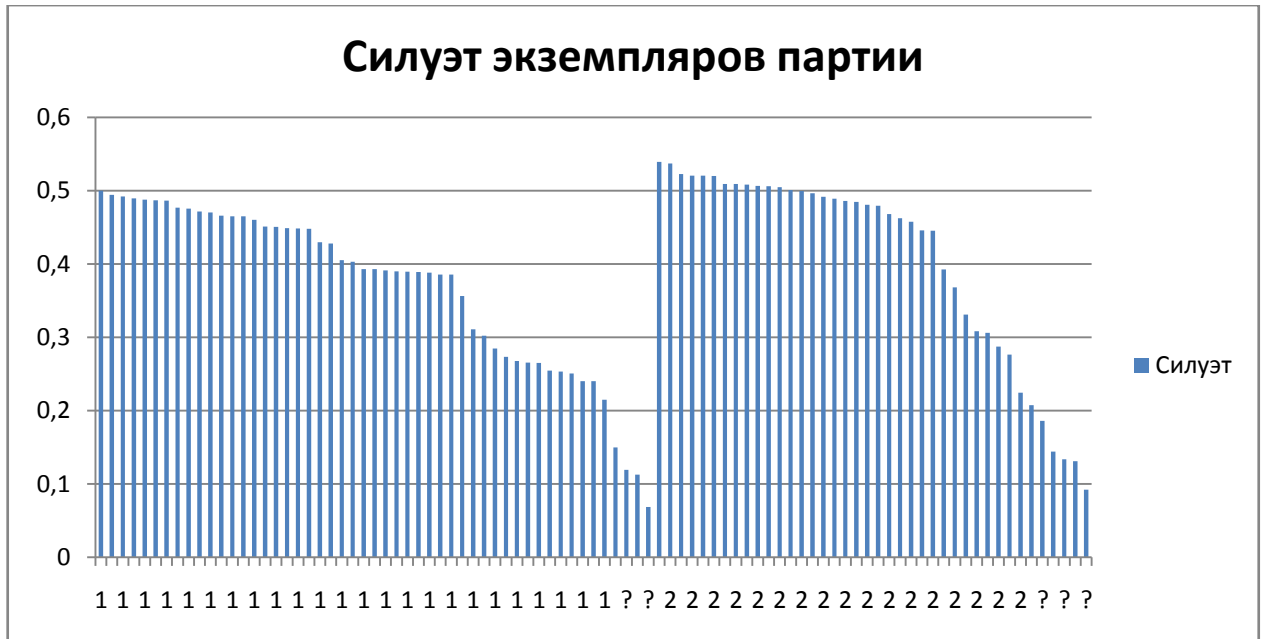


Рисунок 4.7 - Значения силуэта для элементов кластеров (разбиение на 2 кластера) на примере задачи разбиения множества интегральных схем по предполагаемым производственным партиям. Экземпляры со значением  $s(i) < 0,20$  помечены знаком «?» (нечетко классифицированные)

При вычислении ускоренного критерия определим  $a(i)$  как расстояние от объекта  $i$  до центра кластера, к которому он относится. В свою очередь,  $b(i)$  определяется как расстояние, наименьшее до центров групп, к которым данный объект не принадлежит, то есть расстояние до центра соседнего кластера.

3) GAP-критерий. Метод использует статистику расхождений [395], вычисляя критерий

$$GAP(k) = \frac{1}{B} \sum \log W_b^*(k) - \log W(k)$$

Метод использует  $B$  различных унифицированных множеств, каждое из которых состоит из того же количества элементов, что и оригинальное множество. Строится внутренняя сумма квадратов для различного количества кластеров. Требуется подобрать  $k$ , максимизирующее  $GAP(k)$ .

И) Байесовский информационный критерий.

Байесовский информационный критерий (Bayesian information criterion, BIC, иногда - Schwarz Criterion) - критерий выбора модели из класса параметризованных моделей, зависящих от разного числа параметров [396], используемый также в задачах

автоматической группировки данных. Для оценивания модели обычно используется метод нахождения максимума функции правдоподобия, значение которого можно увеличить добавлением дополнительных параметров. Однако, это может вызвать переобучение. Байесовский критерий устраняет проблему переобучения, штрафую увеличение числа параметров модели (в случае автоматической группировки – штрафую увеличение числа групп). Тесно связан с критерием Акаике, но в случае байесовского критерия увеличение параметров штрафуются строже.

С помощью критерия ВИС можно оценить наиболее подходящее число групп  $K$  при автоматической группировке объектов. Формула ВИС имеет следующий вид:

$$BIC(M_j) = \hat{l}_j(\delta) - \frac{p_j}{2} \log N$$

где  $\hat{l}_j(\delta)$  - логарифм максимального значения функции правдоподобия для данных  $\delta$  в соответствие с  $j$ -ой моделью,  $p_j$  - число параметров в модели  $M_j$ ,  $p_j = (D+1) \times K$ ,  $D$  обозначает размерность вектора входных данных. Следует выбирать модель с наибольшим значением этого критерия, и таким образом можно выбрать число кластеров.

Оценка максимального правдоподобия для разброса параметров может быть вычислена следующим образом:

$$\hat{\sigma}^2 = \frac{1}{N-K} \sum_{i=1}^N (A_i - X_{(i)})^2$$

Где  $(i)$  обозначает индекс центроида, который наиболее близок к входному вектору  $A_i$ ,  $N$  – число входных векторов и  $X_{(i)}$  – центроид, соответствующий входному вектору  $A_i$ .

Логарифм максимального значения функции правдоподобия может быть вычислен следующим образом:

$$\begin{aligned} \hat{l}_j(\delta) &= \log \prod_{i=1}^N \hat{P}(A_i) = \sum_{i=1}^N \left( \log \frac{1}{\sqrt{2\pi} \hat{\sigma}^D} - \frac{1}{2\hat{\sigma}^2} |A_i - X_{(i)}|^2 + \log \frac{N_i}{N} \right) = \\ &= -\frac{N}{2} \log 2\pi - \frac{ND}{2} \log \hat{\sigma}^2 - \frac{N-K}{2} + \sum_{i=1}^K N_i \log N_i - N \log N. \end{aligned}$$

Часть этого выражения зависит только от входных данных, которые остаются неизменными при изменении числа групп, поэтому выражение для критерия можно упростить:

$$BIC(M_j) = -ND \log \hat{\sigma}^2 + K + 2 \sum_{i=1}^K N_i \log N_i - p_j \log N = \sum_{i=1}^K N_i (2 \log N_i - D \log \hat{\sigma}^2) - p_j \log N + K.$$

Поскольку априори число групп (партий ЭРИ) неизвестно, решается сразу серия задач автоматической группировки. Далее производится оценка правдоподобия такой группировки с применением одного из критериев. Были проведены эксперименты по применению каждого из перечисленных критериев. В результате наиболее информативным и при этом не требующим подстройки значений каких-либо параметров оказался критерий силуэта. В программной реализации алгоритма автоматической группировки ЭРИ по производственным партиям применена оценка результатов по критериям внутрикластерного расстояния, силуэта и Байесовского информационного критерия. Фактически на производстве специальных партий ЭРИ задействован исключительно критерий силуэта, применение которого дает наименьшее число ошибок при определении числа производственных партий (в пределах 2% для партий от нескольких сотен до нескольких тысяч штук ЭРИ и фактическом количестве производственных партий от 2 до 5).

#### **4.9 Проблема нестабильности испытаний ЭРИ и проверка результатов автоматической группировки**

Исходными данными для алгоритмов автоматической группировки ЭРИ являются результаты диагностических испытаний, полученных на оборудовании испытательных стендов ИТЦ. На результаты испытаний могут накладываться погрешности измерений. Ниже приведен пример нестабильности по результатам диагностических испытаний микросхемы 1526ЛЕ5:

- тесты 11-26 - +- 2 нА (2E-6 мА);
- тесты 27-30 - +- 2 мВ (2E-3 В)
- тесты 31-38 - +- 10 мВ (1E-2 В)
- тесты 39-42 - +- 50 мкА (5E-2 мА)
- тесты 43-46 - +- 100 мкА (1E-1 мА)
- тесты 47-55 - +- 10 нА (1E-5 мА)

Так как существует погрешность измерений приборами при проведении тестовых испытаний, и соответственно, данные погрешности могут присутствовать в данных, по

результатам автоматической группировки предлагается осуществить проверку. Проверка заключается в попарном сравнении изделий на предмет схожести их характеристик. Если два изделия со схожими характеристиками находятся в одной группе, то это лишь подтверждает выполненную группировку и не требует дополнительного внимания. Если же два изделия со схожими характеристиками оказались в разных группах, то стоит с осторожностью относиться к таким изделиям с позиции оценки получаемых результатов. Проблема возникает в том, чтобы определить понятие «схожие характеристики», так как элементарные арифметические операции с величинами отклонений и погрешностей в многомерном пространстве признаков не могут дать необходимый результат (так, к примеру, известно, что диагональ единичного гиперкуба растет при увеличении размерности как корень значения размерности).

Для решения проблемы идентификации схожести характеристик изделий предлагается использовать элементы метода интервального кластер-анализа [397].

Многие методы кластер-анализа, в том числе рассматриваемые в Главе 3, основаны на использовании расстояний между объектами.

С ростом размерности пространства  $d$  евклидова пространства диагональ единичного куба растет как  $\sqrt{d}$ . Для определения погрешности, возникающей при решении задачи автоматической группировки, естественно, в первую очередь имеет значение погрешность при определении расстояний между объектами и их координат (значений характеристик). Рассмотрим два вектора  $X_0 = (x_1, x_2, \dots, x_p)$  и  $Y_0 = (y_1, y_2, \dots, y_p)$  размерности  $p$ . Пусть они известны с погрешностями  $\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_p)$  и  $\delta = (\delta_1, \delta_2, \dots, \delta_p)$ . То есть фактически приходится оперировать векторами  $X = X_0 + \varepsilon$ ,  $Y = Y_0 + \delta$ . Тогда

$$\rho^2(X, Y) = \rho^2(X_0, Y_0) + 2 \sum_{1 \leq i \leq p} (x_i - y_i)(\varepsilon_i - \delta_i) + \sum_{1 \leq i \leq p} (\varepsilon_i - \delta_i)^2. \quad (4.1)$$

Пусть все характеристики объектов измерены с некими предельными значениями разбросов

$$|\varepsilon_i| \leq \Delta, \quad |\delta_i| \leq \Delta, \quad i = 1, 2, \dots, n.$$

Если характеристики предварительно отнормированы такое предположение является вполне естественным.

Если  $p\Delta^2 \rightarrow 0$ , то последнее слагаемое в (4.1) не превышает значения  $4p\Delta^2$ , и им можно пренебречь, и норма евклидова расстояния имеет вид

$$N_{\rho^2}(X_0, Y_0) = 4 \sum_{1 \leq i \leq p} |x_i - y_i| \Delta$$

При одинаковых математических ожиданиях величин  $|x_i - y_i|$ , при большой размерности пространства  $p$  и малых  $\Delta$  существует константа  $C$ :

$$N_{\rho^2}(X_0, Y_0) = Cp\Delta$$

с точностью до бесконечно малых более высокого порядка при  $p\Delta^2 \rightarrow 0$ .

Из этого следует, что

$$\rho(X, Y) = \rho(X_0, Y_0) + \theta \frac{Cp\Delta}{2\rho(X_0, Y_0)} \quad (4.2)$$

при некотором  $-1 < \theta < 1$ .

Зная предельную погрешность отнормированных данных, можно определить минимально различимое расстояние при решении задачи автоматической группировки, уравнивая слагаемые в (4.2):

$$\rho_{\min} = \frac{Cp\Delta}{2\rho_{\min}}, \quad \rho_{\min} = \sqrt{\frac{Cp\Delta}{2}}. \quad (4.3)$$

То есть точки, лежащие друг от друга на меньшем расстоянии можно считать неразличимыми между собой. При этом важно знать значение  $C$ . Если точки (векторы данных)  $x_i$  и  $y_i$  независимы и распределены в пространстве с нормальным распределением вокруг центра координат (нулевое математическое ожидание) при дисперсии, равной 1, то  $\mu\{|x_i - y_i|\} = 2\sqrt{\pi}$ . Соответственно,  $C \approx 4,51$ . Следовательно, в этой модели  $\rho_{\min} = 1,5\sqrt{p\Delta}$ .

Таким образом, из (4.3) следует, что, несмотря на рост длины диагонали единичного куба, в пределах которого в основном сосредоточены отнормированные данные с ростом размерности пространства, таким же образом возрастает и величина естественного кванта расстояния – порога неразличимости  $\rho_{\min}$ . Следовательно, увеличение размерности пространства – добавление новых характеристик объектов – не улучшает возможности применения методов автоматической группировки [397].

В [397] даны выводы и для конкретных алгоритмов. Так, в дендрограммах [398] (например, результатах работы иерархических агломеративных алгоритмов ближнего соседа, дальнего соседа, средней связи) можно порекомендовать т.е. объединять уровни, отличающиеся менее чем на  $\rho_{\min}$ . Если все уровни таким образом будут объединены, то можно сделать вывод, что в данной математической постановке задачи данные

вообще не имеют кластерной структуры. В алгоритмах решения задачи  $k$ -средних типа FOREL [49] центр тяжести (центроид) текущего кластера определяется с точностью  $\pm \Delta$  по каждой координате, и, согласно (4.2), порог для включения вектора данных в группу (радиус шара  $R$ ) из-за погрешностей исходных данных может измениться на  $\pm \frac{2,25}{R} p\Delta$ .

Поэтому, кроме расчетов с  $R$ , для сравнения рекомендуется провести также расчеты с радиусами  $R_1$  и  $R_2$ :

$$R_1 = R \left( 1 - \frac{2,25}{R^2} p\Delta \right), \quad R_2 = R \left( 1 + \frac{2,25}{R^2} p\Delta \right)$$

Адекватными можно считать только выводы, общие для всех трех расчетов. Эти рекомендации развивают общую идею [399] о необходимости расчетов с различными значениями параметров алгоритмов для получения результатов, которые были бы инвариантны по отношению к выбору конкретного алгоритма.

Рассмотрим вычисление параметра  $\rho_{\min}$  на примере нестабильности по результатам диагностических испытаний микросхемы 1526JE5 (пример приведен в начале раздела). Результаты приведены в таблице 4.4.

В таблице указаны:

- разброс – разность между наибольшим и наименьшим значением признака для изделий выборки;
- абсолютная нестабильность – значение нестабильности для каждого тестера;
- среднеквадратическое отклонение для каждого признака изделий выборки;
- нормированная нестабильность – значение нестабильности, деленное на среднеквадратическое отклонение.

Из таблицы видно, что значение нестабильности очень велико по отношению к разбросу измеряемых параметров, а для некоторых тестов значение нестабильности превышает разброс измеряемого параметра (например, тесты 27-30). Это подтверждает сложность решаемой задачи и указывает на источник возможных ошибок при автоматической группировке ЭРИ.

Наибольшее нормированное значение нестабильности составляет 12,1, что само по себе чрезмерно велико. Значение  $\rho_{\min} = 3,47$ .

Таблица 4.4 – Показатели нестабильности для микросхемы 1526LE5

Номер признака	разброс (max-min)	абсолютная нестабильность	среднеквадратическое отклонение	нормированная нестабильность
Тест 11	0,00E+00	2,00E-06	0,00E+00	
Тест 12	2,00E-06	2,00E-06	2,36E-07	8,49E+00
Тест 13	3,00E-06	2,00E-06	3,31E-07	6,03E+00
Тест 14	3,00E-06	2,00E-06	3,47E-07	5,77E+00
Тест 15	9,00E-06	2,00E-06	4,91E-07	4,07E+00
Тест 16	3,00E-06	2,00E-06	4,66E-07	4,29E+00
Тест 17	5,00E-06	2,00E-06	3,69E-07	5,42E+00
Тест 18	3,00E-06	2,00E-06	3,27E-07	6,11E+00
Тест 19	3,00E-06	2,00E-06	6,75E-07	2,96E+00
Тест 20	4,00E-06	2,00E-06	8,39E-07	2,38E+00
Тест 21	3,00E-06	2,00E-06	5,35E-07	3,74E+00
Тест 22	2,00E-06	2,00E-06	5,50E-07	3,64E+00
Тест 23	4,00E-06	2,00E-06	5,69E-07	3,52E+00
Тест 24	5,00E-06	2,00E-06	7,87E-07	2,54E+00
Тест 25	1,10E-05	2,00E-06	1,12E-06	1,79E+00
Тест 26	3,00E-06	2,00E-06	5,51E-07	3,63E+00
Тест 27	1,00E-03	2,00E-03	3,81E-04	5,25E+00
Тест 28	1,00E-03	2,00E-03	3,59E-04	5,57E+00
Тест 29	1,00E-03	2,00E-03	3,67E-04	5,45E+00
Тест 30	1,00E-03	2,00E-03	3,39E-04	5,89E+00
Тест 31	1,00E-02	1,00E-02	4,69E-03	2,13E+00
Тест 32	1,00E-02	1,00E-02	4,74E-03	2,11E+00
Тест 33	1,00E-02	1,00E-02	4,72E-03	2,12E+00
Тест 34	1,00E-02	1,00E-02	4,69E-03	2,13E+00
Тест 35	1,00E-02	1,00E-02	4,70E-03	2,13E+00
Тест 36	1,00E-02	1,00E-02	4,70E-03	2,13E+00
Тест 37	1,00E-02	1,00E-02	4,70E-03	2,13E+00
Тест 38	1,00E-02	1,00E-02	4,70E-03	2,13E+00
Тест 39	1,47E+00	5,00E-02	3,03E-01	1,65E-01
Тест 40	1,47E+00	5,00E-02	2,97E-01	1,68E-01
Тест 41	1,51E+00	5,00E-02	2,98E-01	1,68E-01
Тест 42	1,50E+00	5,00E-02	2,99E-01	1,67E-01
Тест 43	3,27E+00	1,00E-01	6,93E-01	1,44E-01
Тест 44	3,01E+00	1,00E-01	6,69E-01	1,50E-01
Тест 45	3,28E+00	1,00E-01	6,69E-01	1,50E-01
Тест 46	3,03E+00	1,00E-01	6,63E-01	1,51E-01
Тест 47	8,00E-06	1,00E-05	9,18E-07	1,09E+01
Тест 48	6,00E-06	1,00E-05	8,23E-07	1,21E+01
Тест 49	1,00E-05	1,00E-05	9,15E-07	1,09E+01
Тест 50	1,10E-05	1,00E-05	1,02E-06	9,80E+00
Тест 51	1,40E-05	1,00E-05	1,23E-06	8,16E+00
Тест 52	2,20E-05	1,00E-05	2,23E-06	4,48E+00
Тест 53	2,00E-05	1,00E-05	2,24E-06	4,45E+00
Тест 54	3,00E-05	1,00E-05	2,70E-06	3,70E+00
Тест 55	4,00E-05	1,00E-05	4,30E-06	2,33E+00

#### 4.10 Экспериментальные исследования

Разработанные алгоритмы кластеризации были применены для классификации партий радиоэлектронных изделий по производственным партиям, различающимся условиями производства и, следовательно, характеристиками, измеряемыми при проведении неразрушающих испытаний. Кроме того, данные алгоритмы были применены и для определения количества партий в тестируемом множестве изделий.

Для визуализации результатов измерений и производимой с помощью разработанных алгоритмов классификации тестируемых радиоэлектронных изделий был применен метод MDS (Multi-Dimensional Scaling – многомерное масштабирование [386, 253]) и собственно средства визуализации ELKI[400] и GNUPlot.

Данные программные разработки распространяются по лицензии GNU GPL с открытым исходным кодом, что делает их удобными для интеграции в сложные прикладные разработки.

Пространство результатов измерений нормируется [383].

Были использованы 10 сборных партий различных ЭРИ (диоды, стабилитроны, полевые транзисторы, микросхемы), каждая из которых содержала от одной до семи производственных партий, различающихся условиями изготовления. Объемы сборных партий – от 60 до 1250 единиц продукции.

Алгоритмы классификации были запущены для каждой из партий 10 раз с различными значениями параметра  $k$  (предполагаемое число кластеров-партий ЭРИ), от 1 до 30.

Результаты разбиения сборной партии микросхемы 1526JE2 на предполагаемые производственные партии для  $k$  от 1 до 4 показаны на рис. 4.8. Результаты разбиения показаны в условном двумерном пространстве (результат процедуры MDS).

Визуально для микросхемы 1526JE5 можно различить 3 кластера, что соответствует фактически присутствовавшим в сборной партии (всего 620 единиц) экземплярам трех производственных партий.

Результатом работы алгоритмов кластеризации является собственно соответствие номеров единиц продукции в сборной партии и номеров предполагаемых производственных партий, а также суммарный разброс параметров результатов измерений в нормированном пространстве, являющийся целевой функцией алгоритмов.



Зависимость данного суммарного разброса от количества предполагаемых производственных партий (т. е. от количества кластеров  $k$ ) сведены в табл. 4.5.,

Таблица 4.5 – Зависимость значения целевой функции (суммарного расстояния до центров кластеров) от числа предполагаемых производственных партий  $k$  изделия 1526ЛЕ5

$k$	Партия 1 (сборная)		Партия 2 (однородная)	
	$F_{fitness}(\chi)$	В % от предыдущего значения	$F_{fitness}(\chi)$	В % от предыдущего значения
1	84359	-	32984	-
2	53968	63,9%	29355	89%
3	37985	70,4%	27110	92,3%
4	27329	71,9%	25320	93,4%
5	21098	77,2%	23927	94,5%
6	19865	94,2%	22754	95,1%
7	18952	95,4%	21821	95,9%
8	18091	95,5%	21035	96,4%
9	17357	95,9%	20425	97,1%
10	17021	98,1%	20078	98,3%

Из таблицы видно, что для изделия 1526ЛЕ5 значение целевой функции в первой партии изделий (620 единиц изделий) резко падает с ростом  $k$  до значения  $k=4$  (четыре производственных партии). Для второй партии (1132 изделия) такое поведение целевой функции не характерно вследствие того, что данная партия составлена из изделий лишь одной производственной партии. Все результаты получены за время, не превышающее 5 секунд (данное время было задано как условие останова алгоритма). Для сравнения: обработка данных той же первой партии изделия 1526ЛЕ5 из 620 единиц методом Information Bottleneck Clustering потребовала более 15 минут времени на машине с процессором Intel Xeon с тактовой частотой 2 ГГц при сравнимых по точности результатах. В табл. 4.6 приведены сравнительные результаты различных методов, примененных для классификации первой партии изделий. Новый алгоритм дает стабильные результаты (минимальный разброс значений) при высокой скорости выполнения расчета.

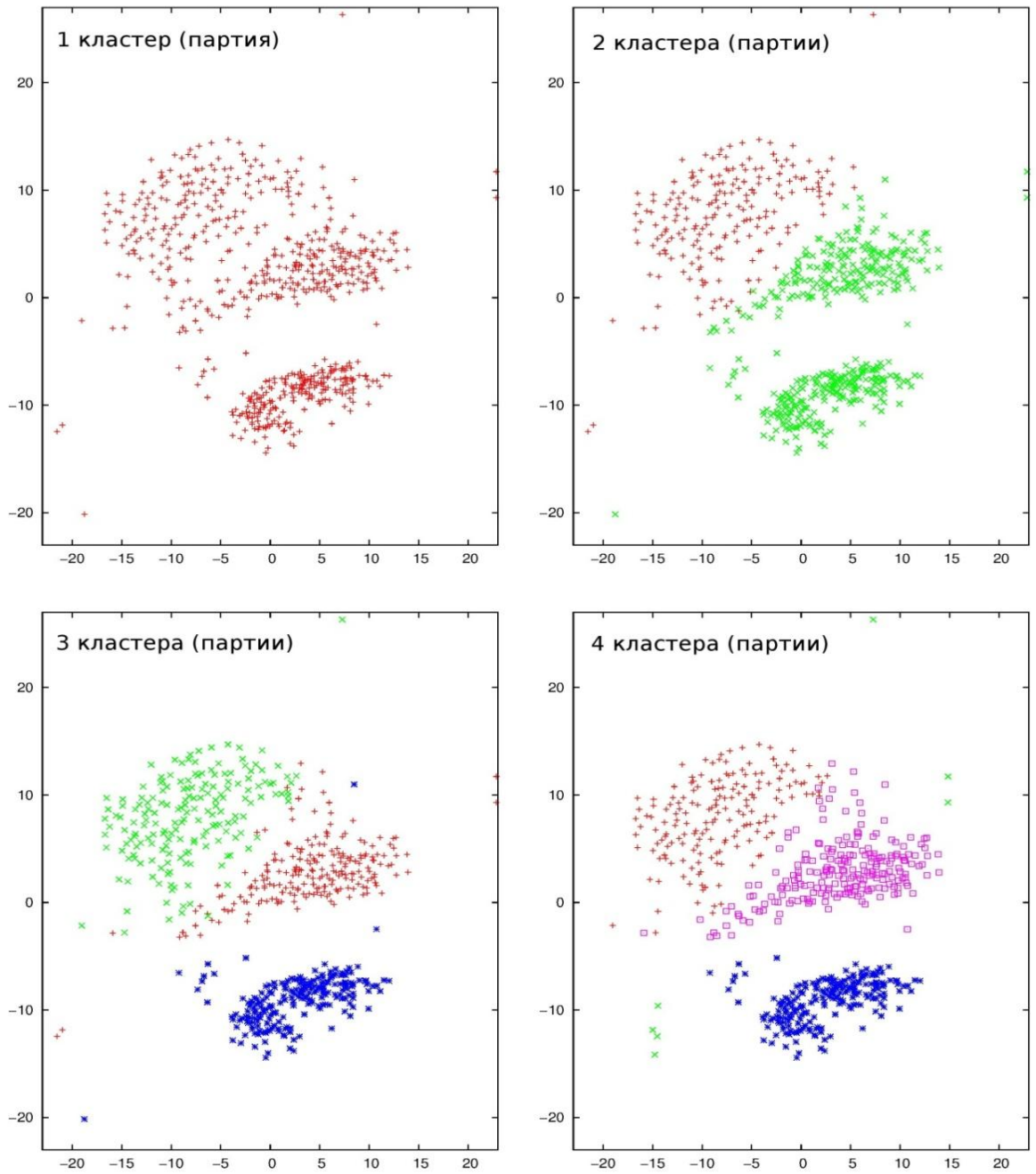


Рисунок 4.8 – Разбиение сборной партии микросхемы 1526ЛЕ5 на 1,2,3,4 кластера, представляющих предполагаемые производственные партии

На рисунках 4.9 и 4.10 представлены частотные гистограммы расстояний между векторами данных в нормированном пространстве для неоднородной и однородной партии ЭРИ соответственно.

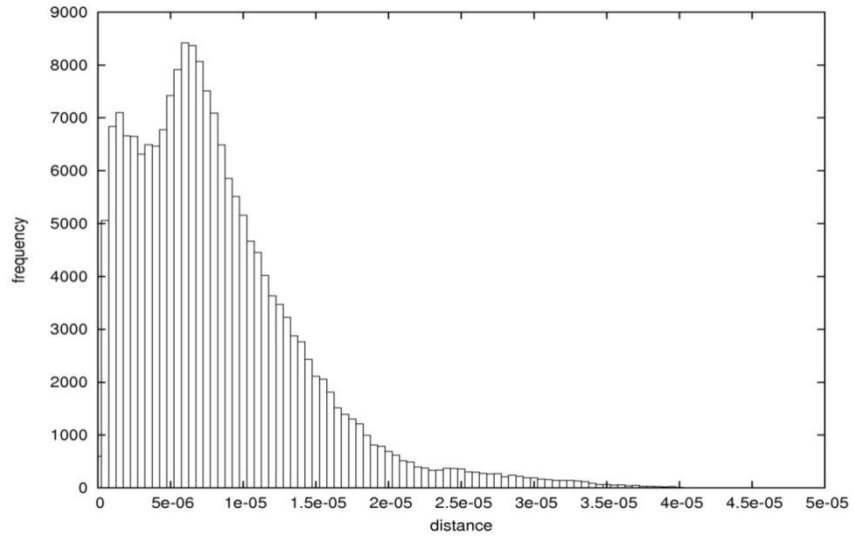


Рисунок 4.9 – Частотная гистограмма расстояний между векторами данных в нормированном пространстве для неоднородной партии ЭРИ

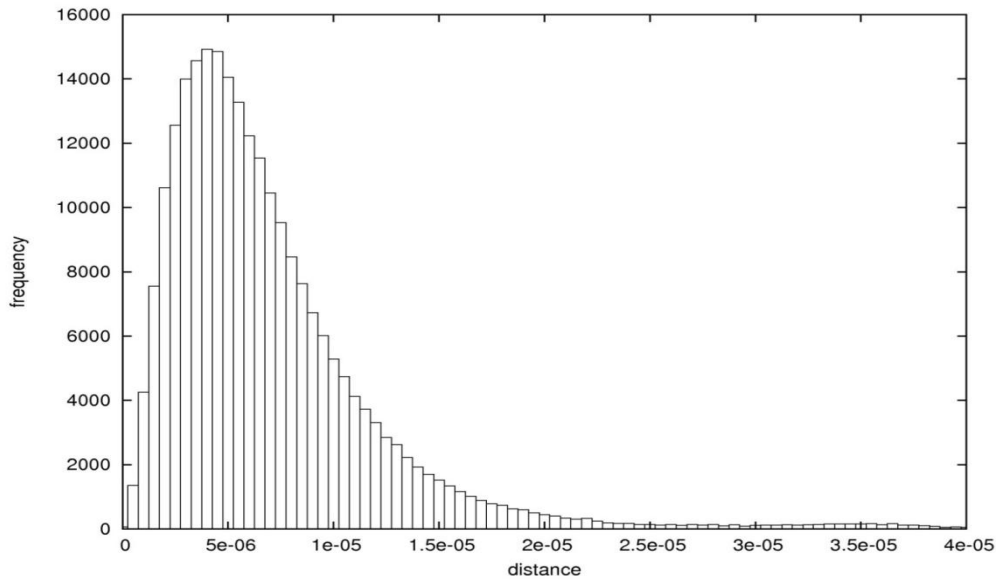


Рисунок 4.10 – Частотная гистограмма расстояний между векторами данных в нормированном пространстве для однородной партии ЭРИ

Таблица 4.6 – Сравнение результатов различных методов для первой партии изделия 1526ЛЕ5 при  $k=4$

Алгоритм	Число Запусков	Время на 1 запуск, с	Среднее значение $F_{fitness}(\chi)$	Точность в сравнении с наилучшим известным значением	Лучшее значение $F_{fitness}(\chi)$	Наихудшее значение $F_{fitness}(\chi)$	Разброс значений, %
Information Bottleneck	1	972	27318,1	99,996%	27317,1	27317,1	-
k-means++	30	5	27398,4	99,7%	27316,9	27416,9	0,37%
Новый алгоритм	30	5	27320,1	99,98%	27316,9	27321,1	0,015%

В качестве тестовых примеров для испытания генетического алгоритма с частичным объединением «родительских» решений были использованы данные тестовых испытаний ЭРИ, а также данные из библиотеки тестовых примеров UCI(<http://www.cs.uci.edu/mllearn/mlrepository.html>).

При решении задач были использованы различные метрики расстояния. Достигнутые результаты нового алгоритма незначительно уступают по точности методу Information Bottleneck Clustering (IBC) [253], при этом новый алгоритм требует несопоставимо меньших вычислительных затрат. Результаты вычислений сведены в таблицу 4.7. Кроме метода IBC в таблице приведено сравнение результатов с многократным запуском ALA-процедуры (локальный поиск) и с результатами популярного генетического алгоритма с рекомендацией подмножеств фиксированной длины [83].

Таблица 4.7 – Сравнительные результаты работы алгоритмов с различными метриками

Набор данных, число векторов данных, размерность пространства	Число классов, метрика, тип задач	Алгоритм	Результат	Время, с.	Средне-квадр. отклонение результата 10 запусков
Отбраковочные испытания диода, $N=701, d=18$	30, $l_1$ нормирован., k-средних	Лок.поиск (мультистарт)	5452,39	30	4,268
		ГА с фикс.подмнож.	5427,92	30	6,41
		ГА новый	5399,56	30	2,91
		IBC	5390,12	2151,2	Детерм.
Отбраковочные испытания диода, $N=701, d=18$	30, $l_2^2$ нормирован., k-медоид	Лок.поиск (мультистарт)	1951,62	10	5,217
		ГА с фикс.подмнож.	1888,31	10	4,02
		ГА новый	1887,01	10	1,894
		IBC	1902,93	1002,4	Детерм.
Отбраковочные испытания микросхемы 1526JE2, $N=620, d=120$	30, $l_2^2$ нормирован., k-средних	Лок.поиск (мультистарт)	12803,64	30	47,32
		ГА с фикс.подмнож.	12539,17	30	3,14
		ГА новый	12539,54	30	2,61
		IBC	12556,04	4953,3	Детерм.
Ionosphere (UCI), $N=351, d=35$	10, $L_1, k$ -средних	Лок.поиск (мультистарт)	2529,97	5	2,092
		ГА с фикс.подмнож.	2527,71	5	0,0249
		ГА новый	2526,99	5	0,0123
		IBC	2536,06	71,33	Детерм.

В таблице 4.7 также показано, что новый алгоритм может быть успешно применен для решения задачи  $k$ -медоид. Для всех задач новый алгоритм показал наименьший разброс значений целевой функции среди стохастических алгоритмов.

В Таблице 4.8 представлены результаты работы алгоритма автоматической группировки для различных ЭРИ, состав сборных партий которых заранее известен. Как видно из таблицы доля некорректно классифицированных предлагаемым алгоритмом изделий не превышает 2%. Таким образом, модель  $k$ -средних с прямоугольной метрикой позволяет эффективно выявлять однородные группы изделий несмотря на проблему неустойчивости, описанную в предыдущем параграфе.

Под «спорными» понимаются такие векторы данных (применительно к данной задаче – экземпляры ЭРИ), для которых расстояния до двух центров кластеров примерно одинаковы (в данном примере под таковыми понимаются расстояния, отличающиеся не более чем в 1,05 раз). «Спорные» экземпляры, доля которых не велика, могут быть удалены из специальных партий.

Таблица 4.8 – Результаты автоматической классификации реальных сборных партий ЭРИ

Вид изделия	№ партии	Объем сборной партии, шт.	Число измерений	Количество фактически произведенных партий, шт.	Число/% некорректно классифицированных экземпляров	Число/% «спорных» экземпляров
2Д522Б	000202	497	10	2	7/1,41	32/6,44
2Д522Б	000204	281	10	2	5/1,78	13/4,63
3ОТ122А	000357	279	14	2	4/1,43	11/3,94
1526ЛЕ5	000901	620	130	3	13/2,1	47/7,58
1256ЛЕ2	000139	826	130	4	18/1,94	79/9,56
ОР297GSZ	000017	700	18	4	13/1,86	49/7,0
537РУ9А	000682	1231	10	5	21/1,71	137/11,13
537РУ8А	000685	321	90	2	1/0,31	0/0,0

Задача автоматической группировки (разбиение на кластеры) требует задания числа групп (кластеров)  $k$ , значение которого априори неизвестно. Эффективной можно считать группировку, результатом которой являются компактные, удаленные друг от друга группы в нормированном пространстве. Это свойство (компактность и взаимная удаленность) в литературе по автоматической группировке и кластерному анализу [71] оценивается по различным критериям: критерий Хартигана [394], информационный критерий Байеса (BIC – Bayesian Information Criterion) [393],

информационный критерий Акаике (AIC), критерий силуэта [328]. В таблице 4.9 представлены значения данных критериев для партии изделий 1526JE5.

Таблица 4.9 – Значения целевой функции и критериев оценки эффективности разбиения на группы в зависимости от их числа для изделия 1526JE5

k	$F_{fitness}(\chi)$	Критерий Хартигана	Критерий силуэта ускоренный	Критерий силуэта	Критерий BIC	Критерий AIC
1	84359	220,4895	-	-	334844	5302219
2	53968	206,9374	0,53918	0,41129	316973	3189458
3	37985	54,3987	0,57784	0,45594	295186	1730708
4	27329	46,9809	0,53645	0,38642	291719	1549824
5	21098	35,9868	0,48232	0,34088	287171	1347527
6	19865	23,0933	0,44891	0,31559	282926	1182413
7	18952	22,6444	0,43588	0,29702	281246	1113269
8	18091	20,4564	0,42419	0,28621	279008	1032668
9	17357	15,0933	0,39457	0,25063	277946	988834
10	17021	13,4691	0,38525	0,23961	276951	968357

Результаты вычислительных экспериментов, проведенные с партиями изделий, состав которых априори известен, показали, что наиболее информативным является критерий силуэта и его ускоренная оценка. Остальные критерии фактически являются свертками двух критериев: собственно целевой функции  $F_{fitness}(\chi)$  и количества групп  $k$ . Определение количества групп (партий) изделий предлагается проводить в соответствии с максимальным значением критерия силуэта, в спорных случаях – полезной является MDS-визуализация.

#### 4.11 Общая схема метода принятия решений по комплектации электронных узлов КА

Данные проводимых тестовых испытаний аккумулируются в базе данных (БД) результатов тестов. В БД заносятся производители ЭРИ, наименования изделий (номенклатура тестируемых изделий), состав тестов для каждого изделия с указанием диапазона допустимых значений каждого измерения (рис. 4.11).

В базе данных регистрируется каждая партия того либо иного изделия с указанием предполагаемого (заявленного изготовителем) количества производственных партий (если партия – сборная). После проведения тестов их результаты также

фиксируются в базе данных, после чего автоматизированная система анализирует результаты, выдает их графическое представление (MDS-визуализация).

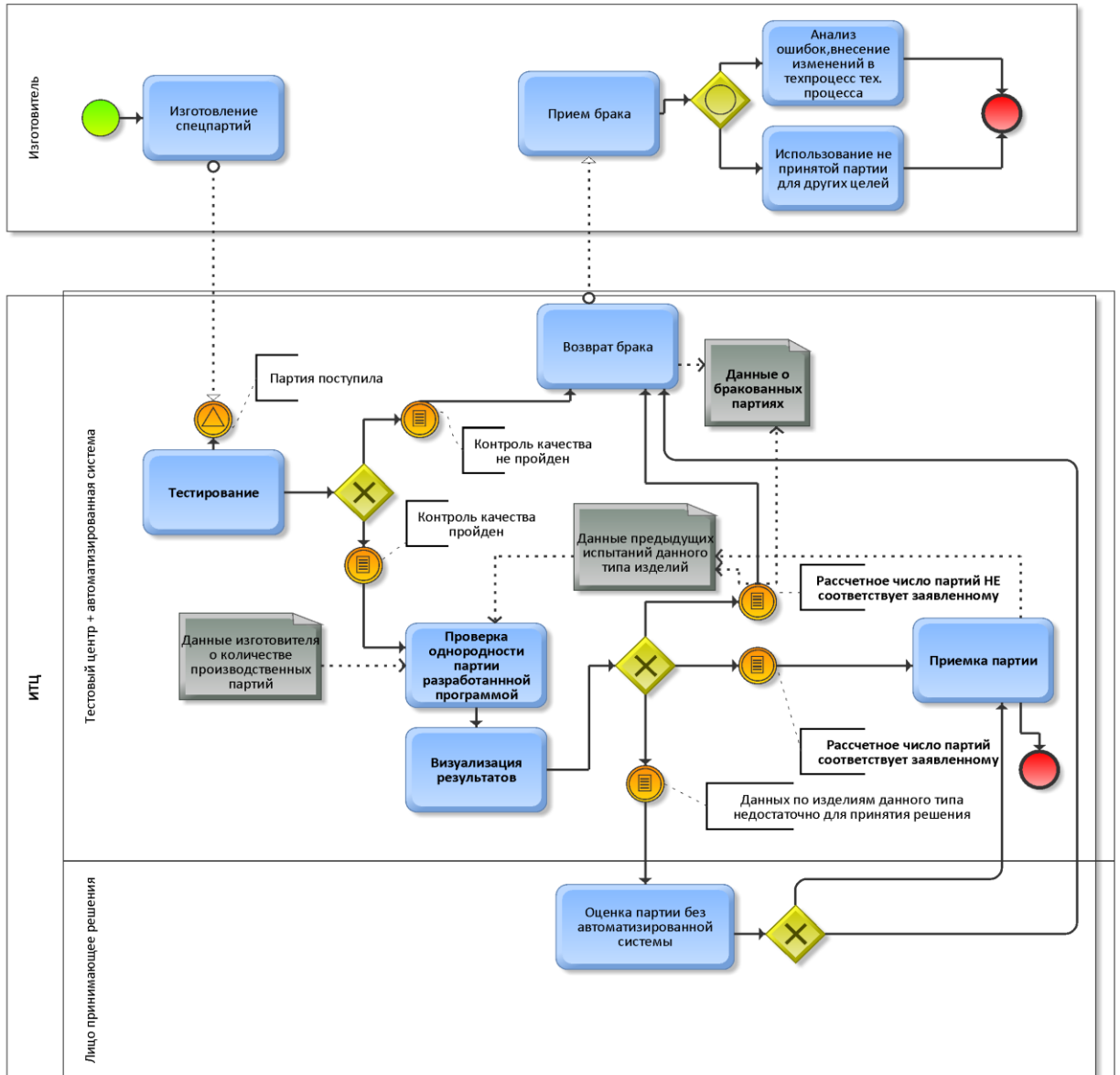


Рисунок 4.11 – BPMN-схема процесса формирования спецпартии ЭКБ

Далее специалист, проверив данные, подготовленные автоматизированной системой и результаты визуализации, принимает решение о приемке или отклонении партии. Данные о забракованных (с указанием причины) партиях также попадают в базу данных, благодаря чему становится возможным сбор и анализ статистической информации по изготовителям и видам изделий.

Учет результатов испытаний должен быть непрерывным и сплошным, иными словами, все данные о поставщиках и потенциальных поставщиках, номенклатуре изделий (ЭРИ), видах испытаний; программах испытаний для конкретных видов ЭРИ, партиях ЭРИ, результатах испытаний каждого экземпляра в партии, претензиях потребителей ЭРИ должно аккумулироваться в едином информационном пространстве. В соответствии с перечисленными выше сущностями, можно определить следующий состав таблиц информационной базы:

а) поставщики:

1. идентификатор поставщика,
2. наименование,
3. дополнительные реквизиты – адрес и т.п.;

б) классы ЭРИ:

1. идентификатор класса,
2. наименование;

в) номенклатура изделий (виды ЭРИ):

1. класс изделия,
  2. идентификатор,
  3. наименование, артикул,
4. дополнительная информация;

г) виды испытаний:

1. идентификатор вида испытаний,
  2. наименование,
  3. описание условий проведения испытаний,
4. задействованное оборудование (испытательные стенды);

д) программы испытаний:

1. вид изделия (в соответствии с таблицей «Номенклатура изделий»),
  2. порядковый номер испытания,
  3. вид испытания (в соответствии с таблицей «Виды испытаний»),
4. дополнительная информация;

е) партии ЭРИ:

1. поставщик или изготовитель (в соответствии с таблицей «Поставщики»),
2. идентификатор партии,



3. вид изделия (в соответствии с таблицей «Номенклатура изделий»),
  4. дата изготовления,
  5. дата поступления,
  6. количество элементов в партии,
  7. дата окончания испытаний,
  8. класс качества по результатам испытаний (для всей партии);
- ж) результаты испытаний:
1. идентификатор партии,
  2. вид изделия (в соответствии с таблицей «Номенклатура изделий»),
  3. порядковый номер испытания (в соответствии с таблицей «Программа испытаний»),
  4. номер изделия в партии,
  5. числовое значение параметра (результат испытания),
  6. дата испытания
7. номер группы (предполагаемой производственной партии)
8. класс качества (1– надежный. 2 – потенциально ненадежный, 3 – брак);
- з) претензии потребителей:
1. дата поступления претензии,
  2. идентификатор партии,
  3. вид изделия (в соответствии с таблицей «Номенклатура изделий»),
  4. класс проблемы (физические дефекты, несоответствие параметров изделия, спорная ситуация и т.п.),
  5. детальное описание проблемы.

Данные, поступающие с испытательных стендов, представлены текстовыми файлами, формат которых определен для каждого вида испытательного оборудования. Обработка текстовых данных осуществляется программным модулем, выполняющим функцию интеграции поступающих данных в разработанную единую информационную базу. База данных реализована с использованием СУБД PostGreSQL, при этом возможно использование практически любой другой СУБД, поддерживающей SQL.

От каждой партии ЭКБ берутся образцы, на которых проводится разрушающий физический анализ (РФА) для оценки технологического процесса изготовления и для

оценки технологических дефектов, которые обычно не выявляются на этапе отбраковочных испытаний, а проявляются со временем.

После проведения необходимых (взаимосогласованных) работ на заводе-изготовителе и в ОАО «ИТЦ-НПО ПМ» получается новый продукт, названный «Спецпартия».

В зависимости от типа ЭРИ и характерных для этого типа дефектов разработаны примерные перечни испытаний, служащие, в свою очередь, основанием для разработки программ испытаний конкретных наименований ЭРИ. Примерный перечень потенциальных дефектов ИС и выявляющие их методы отбраковочных испытаний приведены в табл. 4.10.

Таблица 4.10 – Отбраковочные испытания для выявления отказов, связанных с дефектами кристаллов ИС

Виды дефектов(отказов)	Методы отбраковочных испытаний				
	отжиг для стабилизации параметров	термоциклирование или тепловой	постоянное ускорение или механический удар	электротермо тренировка (ЭТТ) при обратном	ЭТТ в динамическом режиме
Загрязнение поверхности	+			+	+
Инверсионные каналы	+			+	
Несовершенство структуры кристалла	+	+			+
Трещина на кристалле	+	+	+		+
Точечные поры в окисле		+		+	+
Несовершенство структуры оксидной		+			+
Пробой в оксиде		+		+	+
Дефекты пассивирования		+			+
Диффузионные аномалии				+	+
Диффузионные включения				+	+
Дефекты фотолитографии					+
Открытые участки металлизации		+		+	+
Замыкания металлизации		+		+	+
Электромиграция					+
Уход электрических параметров	+			+	+

Ориентируясь на приведенную аргументацию, в качестве ДООИ у потребителя ЭРИ логично остановиться на следующих видах испытаний:

- измерение электрических параметров;

- выборочный разрушающий физический анализ;
- испытания на наличие посторонних частиц в подкорпусном объеме;
- электротермотренировка;
- термоциклирование;
- проверка герметичности.

В пользу такого выбора говорят и данные, приведенные в табл. 4.11.

Таблица 4.11– Отбраковочные испытания для выявления отказов, связанных со сборкой кристалла в корпусе

Виды дефектов (отказов)	Методы отбраковочных испытаний							
	отжиг для стабилизации параметров	Термоциклирование или тепловой удар	постоянное ускорение или механич. удар	Испытания на герметичность	ЭТТ при обратном смещении	ЭТТ в динамическом режиме	Рентгеноскопия	Визуальный контроль
Разрыв проволочного соединения		+			+	+		+
Отрыв проволочного соединения от кристалла		+			+	+		+
Некачественная сварка (передержка)		+				+		
Смещенное соединение сварки					+	+		+
Множественная сварка								+
Интерметаллические соединения	+	+	+			+		
Дефекты посадки кристалла		+	+			+	+	
Отрыв проволоки от вывода корпуса		+			+	+		+
Короткое замыкание					+	+		
Термические напряжения	+	+				+		
Инородные частицы							+	+
Негерметичный спай		+	+	+				
Дефект внешнего вида		+						+

Примеры дефектов, выявляемых при проведении РФА, помещены на рис. 4.12 – 4.14.

Комплекс испытаний разработан таким образом, что позволяет выявлять наиболее характерные дефекты. Например, закоротка или несплошная перемычка коммутационных шин (рис. 4.15 слева) выявляется при помощи ЭТТ при обратном смещении и ЭТТ в динамическом режиме (рис. 4.15 справа).

Однако, рассматривая приведенную информацию, надо помнить, что в табл. 4.10, 4.11 содержатся данные изготовления, а к потребителю поступают ЭРИ, испытанные по установленным правилам, и те испытания, которые проводят на этапе заводских отбраковочных испытаний на 100% ЭРИ, повторять на входном контроле у потребителя нецелесообразно. К таким видам испытаний относятся термоциклирование и проверка герметичности. При этом необходимо не забывать, что контроль герметичности в процессе изготовления проводят после операции герметизации до нанесения на корпус защитных покрытий [336]. Погружение в уайт-спирит при проверке на большие течи у потребителя ЭРИ и последующая промывка может оказать негативное воздействие на защитные покрытия корпуса и маркировку готовых изделий.

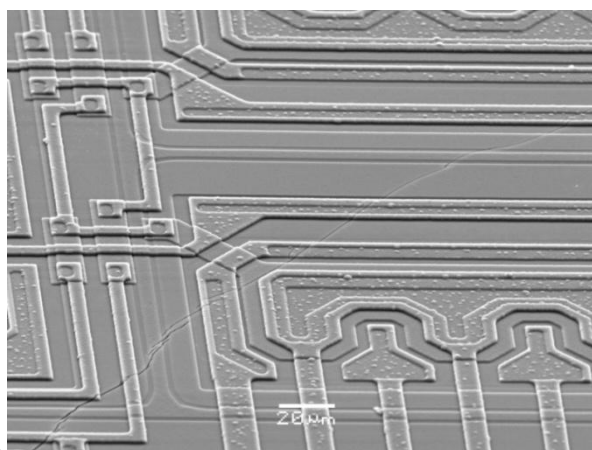


Рисунок 4.12 – Трещина на кристалле

Контроль герметичности на малые течи масс-спектрометрическим методом приводит к ложному забракованию ЭРИ из-за растворения гелия в защитном покрытии корпуса.

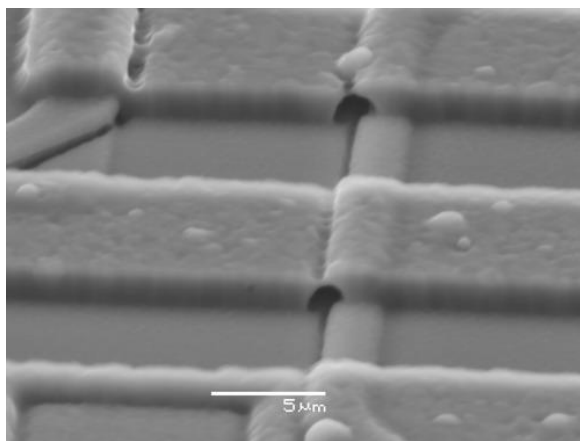


Рисунок 4.13 – Металлизация на ступеньке окисла меньше нормы

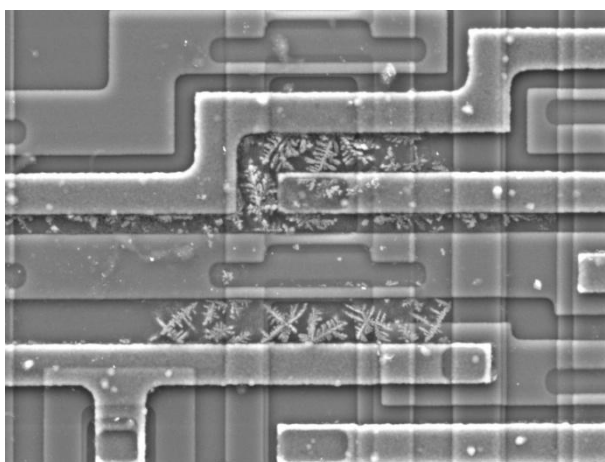


Рисунок 4.14 – Грязь на поверхности кристалла



Рисунок 4.15 – Выявление характерных дефектов

В табл. 4.12 приведен средний уровень отказов ИС для наиболее распространенных методов отбраковочных испытаний.

Таблица 4.12 – Средний уровень отказов ИС для наиболее распространенных методов отбраковочных испытаний

Методы отбраковочных испытаний	Средний процент отбракованных ИС, %
Измерение электрических параметров	0,32
Термоциклирование	1,08
Тепловой удар	0,67
Выдержка при высокой температуре	0,04
Электротермотренировка	1,19-10,5
Проверка герметичности	6,21

#### Заключение к Главе 4

В результате проведенных экспериментов исследована применимость различных методов машинного обучения и анализа данных для решения задачи автоматической группировки ЭРИ, осуществлен выбор наилучшего подхода, проведена модификация методов и настройка алгоритмов.

Задача классификации поступающих партий электрорадиоизделий по производственным партиям с различными условиями производства на основе данных неразрушающих тестов может быть сведена к задаче автоматической группировки. Применение предложенного в настоящей работе генетического алгоритма с особой эвристикой позволяет решать подобные задачи, получая стабильный результат.

Рассмотрены различные алгоритмы для вычисления расположения центров групп как результата задачи  $k$ -средних, а также разработаны их модификации. Из полученных результатов видно, что наиболее эффективным для решаемой задачи является генетический алгоритм с жадной эвристикой с вещественным алфавитом. Также хорошие результаты дает адаптивный алгоритм (метод изменяющихся вероятностей в сочетании с жадной эвристикой). Для ситуации, когда требуется строгая повторяемость получаемых результатов, специально разработан детерминированный алгоритм, который для многочисленных запусков на одной и той же выборке всегда выдает одинаковый результат. Результаты сравнительных испытаний приведены в Приложении Б.

Предложенный в настоящей работе алгоритм позволяет повысить точность решения непрерывной  $p$ -медианной задачи и задачи  $k$ -средних ( $p$ -средних) при размерности задачи до нескольких сотен тысяч векторов данных и количестве центров

(кластеров) до 100. Алгоритм может быть использован при решении задач автоматической группировки и классификации с высокой ценой ошибки. Главное достоинство предложенных алгоритмов – стабильность получаемых результатов при многократных запусках алгоритма – повышает воспроизводимость результата, позволяя использовать алгоритм как средство оценки однородности партий электрорадиоизделий или количества производственных партий в сборной партии изделий при разработке стандартов проведения тестовых испытаний электрорадиоизделий в специализированных тестовых центрах.

Применение прямоугольной метрики в задаче классификации электронных компонентов по производственным партиям позволяет получать результаты, аналогичные результатам классической процедуры  $k$ -средних с квадратичными евклидовыми расстояниями. В то же время, результаты процедуры кластеризации с применением прямоугольной метрикой легко интерпретируемы, поскольку полученный результат имеет ту же точность, что и исходные данные тестовых испытаний, используемые при классификации.

В результате экспериментов на исходных данных выбран наилучший способ автоматического определения числа производственных партий – критерий силуэта. Этот критерий служит также для подтверждения результатов группировки как по каждому изделию, так и по партиям.

Разработанные алгоритмы реализованы в виде программного приложения [401]. Эта версия была использована также для решения исследовательских задач: настройка внутренних параметров алгоритмов и анализ их эффективности, исследование адекватности критериев автоматического выбора числа предполагаемых производственных партий.

Использование ГА с жадной эвристикой для оценки однородности партий электронной компонентной базы позволяет проводить такую оценку в реальном времени, одновременно с проведением неразрушающих и разрушающих тестов. Результат работы алгоритма может быть легко визуализирован, что предоставляет дополнительный удобный инструментарий оценки качества партий.

Метод автоматической классификации электрорадиоизделий по производственным партиям лег в основу СППР автоматической классификации ОАО ИТЦ – НПО ПМ (г.Железногорск), что подтверждается прилагаемым актом внедрения.

## ГЛАВА 5. ЗАДАЧИ АВТОМАТИЧЕСКОЙ ГРУППИРОВКИ И РАЗМЕЩЕНИЯ СО СПЕЦИАЛЬНЫМИ МЕТРИКАМИ И МЕРАМИ РАССТОЯНИЯ

В главе рассматриваются задачи Вебера с нестандартными метриками и иными мерами расстояния, имеющие практическое значение. Наличие эффективного алгоритма решения задачи Вебера – условие применимости метода жадных эвристик (см. Главу 3). В этой связи результаты, изложенные в настоящей главе, расширяют область применения метода автоматической группировки, основанного на применении жадных агломеративных эвристик и иных методов, задействующих стандартную процедуру  $k$ -средних. Рассматриваемые в настоящей главе задачи возникают при работе автоматизированных технических систем, использующих специальные средства транспортировки, такие как, например, манипуляторы с вращающейся стрелой.

Непрерывные модели размещения являются старейшими аналитическими моделями теории размещения, оперирующими геометрическим представлением реальности. Классическим примером такой модели является задача Вебера. Расстояния в задаче Вебера часто принимаются евклидовыми, но могут быть задействованы функции расстояния практически любого типа. Решение особого вида задачи Вебера с квадратичными евклидовыми расстояниями – часть наиболее популярного алгоритма автоматической группировки [402, 403] – задачи  $k$ -средних и многих других задач размещения-распределения [297]. Задача  $k$ -средних также может рассматриваться как частный случай  $p$ -медианной задачи [217], одного из возможных обобщений задачи Вебера.

Классическая задача Вебера, сформулированная в 1909 году – задача поиска точки  $X^* \in \mathbb{R}^2$ , являющейся точкой минимума суммы взвешенных расстояний до  $N$  известных точек-потребителей  $A_1, \dots, A_N$  [225]:

$$X^* = \operatorname{argmin}_{X \in \mathbb{R}^2} f_w(X) = \operatorname{argmin}_{X \in \mathbb{R}^2} \sum_{i=1}^N w_i \|A_i - X\|_2. \quad (5.1)$$

Здесь  $w_i \in \mathbb{R}, w_i \geq 0$  – некоторые весовые коэффициенты,  $\|\cdot\|_2$  – евклидова норма на  $\mathbb{R}^2$ :  $\|A_i - X\| = \sqrt{(a_{i,1} - x_1)^2 + (a_{i,2} - x_2)^2}$ ,  $A_i = (a_{i,1}, a_{i,2})$  и  $X = (x_1, x_2)$ .

Рассмотрим следующую задачу группировки объектов. В некотором пространстве (на складе, на строительной площадке, в магазине и т.д.) имеется  $N$  объектов, которые



необходимо отнести к одной из  $p$  групп, каждая из которых будет складироваться отдельно от других. Если нет ограничений на выбор мест складирования объектов в нашем пространстве, нет ограничений на количество объектов в каждой из групп, и целью решения задачи ставится минимизация времени/стоимости/энергозатрат на перемещение объектов из их текущего местоположения к местам складирования, то мы имеем  $p$ -медианную задачу, рассмотренную в Главе 3. При этом в качестве меры расстояния могут выступать различные функции (не обязательно метрики). Подобные задачи возникают, например, при уборке улиц – если требуется  $p$  точек временного складирования мусора. Аналогично, на роботизированном складе может потребоваться собрать не несколько партий (поддонов) товара с различных ячеек и стеллажей таким образом, чтобы минимизировать затраты на их перемещение специальными механизмами к местам временного хранения, откуда партии товара могут транспортироваться за пределы склада иными способами (обычные складские погрузчики или самовывоз). В обоих случаях имеем задачу физической группировки объектов, которая в случае работы автоматизированных/роботизированных механизмов становится задачей автоматической группировки. Отметим также, что при  $p=1$  мы имеем задачу Вебера. В любом случае, задача сводится к нахождению оптимальных местоположений  $p$  центров каждой группы (кластера), т.е. к непрерывной  $p$ -медианой задаче.

В настоящей главе метрики и иные меры расстояния рассматриваются в контексте решения задачи Вебера, множественной задачи Вебера и в задачах приближительной оценки расстояния. Также дан анализ литературы и продемонстрирована эффективность таксономии в данной области. Впервые предложены алгоритмы решения задачи Вебера с некоторыми метриками и мерами расстояния, являющимися удобными моделями оценки затрат на перемещение объектов в автоматизированных и роботизированных системах, использующих специальные механизмы в качестве средств транспортировки. Сходимость и эффективность алгоритмов доказана аналитически. Даны некоторые вычислительные примеры.

### **5.1 Теория размещения и функции расстояний**

В настоящем и следующем параграфах дан обзор последних разработок в

моделировании расстояний, преодолеваемых при транспортировке объектов. Отмечен явный недостаток внимания в современной литературе к особым случаям метрик и иных мер расстояния, являющихся удобными моделями в системах с особыми средствами транспортировки, в связи с чем недостаточно развита и система классификации подобных метрик и мер [404].

В [405] показано, что функции расстояний играют важную роль во многих областях применения. Целью введения функции расстояния является дать инструмент точного измерения расстояния (длины пути/затрат на транспортировку) между любыми двумя точками в пространстве [406]. Между тем, в задачах физического размещения объектов, как правило, требуется найти кратчайший путь между точками с учетом транспортных сетей. Кроме задач размещения, функции расстояния применяются в различных областях от задач распределения, задачи о коммивояжере, до задачи о расчете оптимального пути транспортного средства [407]. Другие полезные области применения включают теорию аппроксимации [408], теорию оценивания [407-410], задачи обработки сигналов и изображений и другие технические применения [411], при этом используются различные меры сходства.

Задачи размещения можно охарактеризовать четырьмя компонентами:

- 1) точки-потребители (векторы данных), уже размещенные в пространстве;
- 2) искомые точки в пространстве;
- 3) пространство, в котором располагаются точки-потребители и искомые точки;
- 4) метрика или иная мера расстояния, которая указывает расстояние между точками-потребителями и искомыми точками [412].

Более подробное описание моделей теории размещения дано, например, в [140, 412, 225]. Отметим, что в качестве функции расстояния в подобных моделях используются не только нормы и метрики. Выбор функции расстояния играет важнейшую роль при оценке расстояний, преодолеваемых средством транспортировки в реальном окружении, в зависимости от режима транспортировки и типа задачи. Различные режимы транспортировки требуют различных способов оценки расстояния (времени транспортировки, стоимости транспортировки и т.п.). Следовательно, возрастающие усилия по построению моделей наиболее точной оценки расстояний определяются ростом числа разнообразных задач, возникающих в транспортном секторе [413-424, 159]. Конечно, данный параметр задачи (функция расстояния) широко изучен,

но его определение в специфических условиях может быть затруднено, и исследователи ищут пути оптимизации в задачах размещения с использованием концепции усредненных оценок расстояний, медианных расстояний и подходов, основанных на покрытии территорий [405]. Технические инструменты, используемые для задачи размещения, зависят от ее структуры. Следовательно, неявно подразумевают тесную взаимосвязь с функциями расстояния. Таким образом, вычислительные приемы (алгоритмы, эвристики и т.д.), возможность использования анализа выпуклых функций, непосредственно зависят от используемой метрики или меры расстояния. Структура и стабильность множества решений также зависят от сделанного выбора функции расстояния [425].

Имеется множество подходов к моделированию расстояний между двумя точками, от вычисления кратчайшего пути по точно заданной уличной сети до аналитических приближений, основанных на малом числе параметров [426]. Среди подходов, основанных на аналитических приближениях, предлагаемых для задач размещения с различными либо произвольными функциями расстояния, существуют подходы, преобразующие непрерывную задачу размещения в дискретную, требующие значительных вычислительных мощностей [218, 156], при этом не гарантирующие приемлемую точность результатов. Влияние функции расстояния на моделирование размещения, в частности, с точки зрения вычислительной сложности и качества получаемого решения исследовано в [425, 427]. Невыпуклые меры расстояния в моделях размещения с барьерами обсуждаются в [419, 420]. Применение в городской среде и сравнение функций расстояния и связанных с ним параметрах можно найти в [428-433]. Детальный обзор расстояний дан Бримбергом и Лавом в [434]. Непараметрический подход, использующий нейронные сети для оценки действительных расстояний, в реальной городской среде, на примере турецких городов дан в [407]. Один из последних примеров исследований функции расстояния в моделях размещений – [405], в которой дан наиболее общий обзор и систематизация применяемых функций расстояния.

Являясь основным компонентом моделей размещения, функции расстояния находят широкое применение в таких областях как, предупреждение чрезвычайных ситуаций [421], таких как пожары [435, 406], контроль за состоянием преступности [436], размещение телекоммуникационной инфраструктуры [437], автоматическая группировка [409] – тема настоящей диссертации.

Статистическое исследование литературы по таким ключевым словам как «функции расстояния» («distance functions»), «метрики» («metrics»), «расстояние при транспортировке» («travel distances»), «приблизительной оценки расстояний» («distance approximations») показывают следующее:

- найдено порядка 230 наименований;
- возрастающий интерес отмечается после 2000 года;
- около 90% источников – журнальные статьи и статьи конференций;
- статьи относятся к различным областям исследования, таким как, менеджмент, математика, региональное планирование, теория принятия решений, информатика, медицина, городское и промышленное планирование;
- только некоторые из результатов привлекли широкое внимание, например, [421, 438] и некоторые диссертации [439, 440, 406].

Наибольшее число статей по функциям расстояния опубликованы в следующих журналах: EJOR (9%), Operations Research (5%), Annals of Operations Research (3%), Facta Universitatis, Seires: Mathematics and Informatics (3%), Journal of the Operational Research Society (3%), SIAM Journals (2%), Advances in Operations Research (2%).

В настоящей работе представлены классы функции расстояния, ранее не освещенные в литературе, и соответствующие алгоритмы решения задач размещения. В частности представлены алгоритмы решения задачи Вебера, решение которой является частью предлагаемых в настоящей работе алгоритмов автоматической группировки.

Кроме того, в настоящем разделе ставится дополнительная цель – идентифицировать и классифицировать ключевые группы функций расстояния в задачах размещения.

Задача оптимального размещения объекта, обслуживающего заданное множество потребителей, известно довольно давно. Классическое исследование размещения промышленных объектов, впервые было представлено Вебером [292].

После работ Вебера интерес к анализу размещения стабильно возрастает. Предлагаются несколько расширений, модификаций и обобщений предложенной им модели.

Классическая постановка задачи Вебера – задача (5.1). Обобщенная математическая постановка – следующая:

$$\min F(X) = \sum_{i=1}^N w_i L(X, A_i),$$

где  $X=(x_1, \dots, x_d)$  – неизвестное местоположение размещаемого объекта (искомая точка),  $A_i=(a_{i1}, \dots, a_{id})$  – известное местоположение  $i$ -й точки-потребителя,  $i=1 \dots N$ ,  $w_i$  – неотрицательные константы (весовые коэффициенты), преобразующие расстояния в стоимость транспортировки в предположении, что стоимость транспортировки пропорциональна расстояния. Все объекты рассматриваются в  $d$ -мерном пространстве.

Для оценки длины дорожного пути исключительно на основе координат конечных точек исследователи в области теории размещения часто сводят задачу к одной из трех моделей.

Прямоугольное или манхетенское расстояние  $l_1$ , являющееся частным случаем метрики Минковского с параметром  $p=1$ :

$$l_1 = \sum_{i=1}^d |x_i - y_i|.$$

Евклидова метрика (прямая линия или  $l_2$ ):

$$l_2 = \sqrt{\sum_{i=1}^d |x_i - y_i|^2}.$$

Чебышёвская метрика (максимальное расстояние или  $l_\infty$ )

$$l_\infty = \max_i |x_i - y_i|$$

Тем не менее, наиболее популярной является евклидова метрика [441] и большая часть литературы посвящена задаче с этим типом метрики Минковского.

Следующая по популярности – прямоугольная метрика. Менее исследована чебышевская метрика, но и она полезна при решении практических полезных задач, включающих размещение объектов на автоматизированных складах и хранилищах [442, 443].

Извечной проблемой теории размещения является нереалистичность рассматриваемых ею моделей для большинства практических задач. Например, перечисленные выше метрики не подходят для решения задач, связанных с движением сырья и материалов на заводах [405], в системах с вращающимися механизмами, используемыми в качестве средств транспортировки (подъемные краны с телескопической стрелой, манипуляторы и др.) [296, 444], при размещении на сетях, например, дорожных [445, 441], а также при размещении объектов на очень больших расстояниях, когда поверхность Земли невозможно рассматривать как плоскость, и приходится использовать модели размещения на сферической поверхности [446-450, 438].

## 5.2 Классы функций расстояния

В нашей классификации ключевых групп функций расстояния принимаются во внимание способы их классификации, предложенные другими авторами [407] по типу используемой координатной системы и некоторым другим параметрам.

а) Функции расстояния, основанные на сферических координатах. Данные функции расстояния являются удобной моделью, когда точки-потребители географически разнесены на большие расстояния и модель размещения на плоскости не может быть применена.

Расстояния на сферической поверхности являются хорошим приближением для задачи размещения в крупном регионе таких объектов, как например, радиопередатчиков, радиолокационных станций, пунктов управления международными войсковыми группировками, крупных складских центров мирового уровня [448, 438].

В [449] рассмотрена задача, в которой точки-потребители располагаются на сфере, но при этом используются евклидовы расстояния. Также в [446] представлен унифицированный подход к решению задач множественного размещения (множественная задача Вебера) на сфере с евклидовыми, квадратичными евклидовыми расстояниями и собственно, расстояниями на поверхности сферы с доказательством сходимости предложенных методов. В [447] проведено сравнительное исследование трех норм для классической задачи Вебера на сферической поверхности и предложен алгоритм, аналогичный процедуре Вайсфелда.

В качестве последних работ можно отметить [450], в которой предложен альтернативный подход к моделированию расстояний для путешествия по воде. Вместо использования евклидовых расстояний в данной работе предложен алгоритм определения расстояний по географическим координатам, определенным географической сеткой широты и долготы и принципам сферической тригонометрии.

б) Функции расстояния, основанные на полярных координатах. Мотивацией развития данной группы функций явилось наблюдение о том, что дороги в старых городах часто образуют радиально-лучевую структуру.

б1) Метрика французской столицы (French metro). Некоторые типы не выпуклых мер расстояния предложены в контексте задач размещения в [451]. В [441] предложены

метрики на плоскости, применимые для размещения объектов в системе с радиальной структурой вокруг некоторого центра.

Метрика French metro описывается следующим образом [149, 88].

$$d_F(X, A_i) = \begin{cases} \|X - A_i\|, A_i = cX \\ \|A_i, 0\| + \|X, 0\|, A_i \neq cX \end{cases}$$

Здесь  $c$  – вещественный коэффициент,  $\|\cdot\|$  – евклидова норма. В полярных координатах [445] имеем:

$$d_F(X, A_i) = \begin{cases} |x_{R1} - a_{R1,i}|, x_{R2} = a_{R2,i} \\ a_{R1,i} + x_{R1}, x_{R2} \neq a_{R2,i} \end{cases}$$

Здесь  $x_{R1}$  – первая полярная координата (радиус – расстояние до начала координат) искомой точки  $X$ ,  $x_{R2}$  – вторая полярная координата (угловая координата) точки  $X$ . Аналогичные обозначения полярных координат применены и для точек-потребителей.

Данная модель может быть использована для оценки расстояний в экономических, транспортных и логистических задачах [445]. Подход дает весьма точные оценки расстояний, особенно при размещении объектов на сети дорог звездообразной структуры.

б2) Метрика Москвы. Изначально в метрике Москвы были рассмотрены в задачах распределения (автоматической группировки) объектов [421]. В данной работе был предложен метод построения диаграмм Вороного для группировки объектов в городах радиально-лучевой структуры, таких как Москва, Карлсруэ, Амстердам. Диаграмма Вороного для конечного множества точек является мощным механизмом группировки иных объектов (других точек) вокруг известных точек (центров).

Пластриа [451] отмечает, что данная метрика, введенная такими исследователями, как Миттель и Пальсуле, является непрерывным вариантом радиально-кольцевой метрики [441]. Данная метрика удобна, если движение в системе возможно либо вдоль лучей, исходящих из центра координат, либо вдоль концентрических окружностей, центр которых совпадает с центром координат. Именно таким образом рассоложены улицы в Москве (центр координат – Московский кремль) или Амстердаме (центр координат – Центральный вокзал).

б3) Метрика подъемного крана. Аналогична радиально-кольцевой метрике, но имеет некоторые отличия. Здесь также движение возможно вдоль лучей, ведущих от

центра координат и вдоль концентрических окружностей, но целью ставится сокращение времени транспортировки объекта из пункта  $X$  в пункт  $Y$  механизмом с вращающейся стрелой. При том, что механизм может одновременно совершать только один вид движения: изменение вылета стрелы с фиксированной скоростью либо вращение стрелы с фиксированной угловой скоростью. Модели оптимального размещения подъемного крана исследуются более 40 лет, см. Абдель-Халель и Шауки [452]. В [453] дана формула времени-расстояния, с помощью которого осуществляется количественная оценка возможных вариантов размещения подъемного крана. В [454] представлены иные подходы, в частности для оптимального размещения подъемного крана применены генетические алгоритмы. В [455] представлена модель, использующая геоинформационные системы для размещения подъемного крана. В [456] предложено решение другой задачи – нахождения центра  $N$  точек при применении данной функции расстояния (центр координат – местоположение условного подъемного крана в этом случае известно).

в) Простые функции расстояния в декартовых координатах, основанные на нормах [439].

в1) нормы прямых углов – нормы, контуры которых являются политопами. Например, таковыми являются  $l_1$  и  $l_\infty$ . Впервые были использованы при решении задач размещения Витцгаллом [457] и Уордом и Венделлом [458, 459]. Авторы определили нормы прямых углов следующим образом:

$$k(x) = \|x\|_B = \min \left\{ \sum_{g=1}^r |\lambda_g| : x = \sum_{g=1}^r \lambda_g b_g \right\}$$

Где точки  $b_g$  и  $-b_g$  при  $g = 1, \dots, r$  образуют экстремальные точки политопа, соответствующего единичному контуру («единичной окружности», которая в данных случаях в привычном смысле окружностью не является).

При использовании норм прямых углов в задачах размещения, таких как задача Вебера, задачи сводятся к задачам линейного программирования [460, 406]. Такие нормы могут быть использованы для представления единичной окружности настолько точно, насколько это необходимо, путем увеличения числа экстремальных точек ( $b_g$ ) соответствующего политопа. В предельном случае, при  $r \rightarrow \infty$ , норма прямых углов становится эквивалентной единичной окружности [406].



в2) Скругленные нормы. Данные нормы наиболее широко используются в задачах размещения (например, см. [439]). Нормы можно определить следующим образом (на плоскости):

$$l_{k,p} = d(x, y) = k \left[ \sum_{i=1}^2 |x_i - y_i|^p \right]^{\frac{1}{p}}, k > 0, p \geq 1, \forall x \in \mathbb{R}^2$$

Скругленные нормы (“round norms”) названы так [461] из-за формы единичного контура  $u$ , который для любого значения параметра  $p \in (1, \infty)$  и  $k > 0$  не содержит плоских участков.

в3) асимметричные меры [451]. Используются, например, для оценки расстояний при движении по односторонним улицам, вверх или вниз по рекам, при движении по наклонной плоскости и т.д. В [462, 463] авторы применили для решения задачи Вебера с такими асимметричными мерами расстояния алгоритм, подобный процедуре Вайсфелда, и доказали сходимость предложенных алгоритмов.

Меры, определяемые функционалом Минковского компактных выпуклых множеств, содержащих начало координат, имеют интересное свойство, полезное в задачах размещения [420]. В [405] автор отмечает, что расстояние между двумя точками в данном случае определяется лишь базовыми направлениями единичного контура.

#### в4) А-расстояние

Аналогичным образом симметричные многоугольные меры, иногда называемые в литературе нормами прямых углов (не путать с п. в1), называются А-расстояниями. Данный тип расстояний, определение которого дано Видмейером и др. [464], возникает в том случае, когда движение ограничено конечным множеством направлений (направления «туда» и «обратно» считаются различными).

Пусть  $a_1, a_2 \in \mathbb{R}^2$ . А-расстояние между этими двумя точками определяется следующим образом:

$$d_A(a_1, a_2) = \begin{cases} \|a_1 - a_2\|, & \text{если } [a_1, a_2] \text{ является } A\text{-направленным,} \\ \min_{a_3 \in \mathbb{R}^2} \{d_A(a_1, a_3) + d_A(a_3, a_2)\}, & \text{в иных случаях,} \end{cases}$$

где  $[a_1, a_2]$  обозначает отрезок между  $a_1, a_2 \in \mathbb{R}^2$ ,  $\|\bullet\|$  - Евклидова норма.

А-расстояния используются во многих задачах, таких как задача о минимальных остовных деревьях, минимаксные задачи размещения центров реагирования в чрезвычайных ситуациях (пожарные части станции скорой медицинской помощи и т.п.)

и соответствующие задачи распределения («прикрепления», «приписки») объектов между этими центрами, определение минимальных расстояний между выпуклыми многогранниками и другим множеством точек [423]. В частности, в [423] авторы рассматривают минисуммную задачу размещения с  $A$ -расстоянием и, изучив свойства оптимальных решений для данной задачи, предлагают «Алгоритм трассировки ребер» для ее решения. В [436] рассматривается стохастическая задача размещения единственного объекта с  $A$ -расстоянием. Авторы также представляют  $A$ -расстояния для конкурентной задачи размещения, переформулируют задачу в виде комбинаторной задачи оптимизации, решая ее генетическим алгоритмом.

г) Неевклидовы функции расстояния, сводимые к метрике  $l_1$ . Результаты с подобными мерами расстояния, а также их обзор представлены в [456, 158]. В данных работах соответствующие функции расстояния рассматриваются в 2-мерном пространстве, но могут быть обобщены и на 3-мерное.

г1) Метрика лифта.

Является частным случаем расстояний, основанных на нормах, которые были предложены для задач размещения Станимировичем и др. [158]. Расстояния используются при моделировании движения по дорожной сети с единственной главной улицей и множеством перпендикулярных ей улиц. Аналогичная ситуация возникает в здании с лифтом, выступающим в роли оси  $y$ . Кроме того, расстояния удобны при моделировании движения в шахтах и на складах с особыми видами погрузчиков в роли средства транспортировки. Метрика лифта для двух точек  $X=(x_1, x_2)$  и  $A_i=(a_{1,i}, a_{2,i})$  в декартовых координатах определяется следующим образом [88, 158]:

$$d_L(X, A_i) = \begin{cases} |x_1 - a_{1,i}|, & x_2 = a_{2,i} \\ |x_1| + |x_2 - a_{2,i}| + |a_{1,i}|, & x_2 \neq a_{2,i} \end{cases}$$

г2) Метрика британской железной дороги

Метрика является метрикой на  $\mathcal{R}^2$  (в общем случае  $\mathcal{R}^n$ ), определяемой как [88]:

$$\|x\| + \|y\|,$$

если  $x \neq y$ , и равно 0, если точки совпадают. Таким образом, для двух точек  $X$  и  $A_i$ :

$$d_{BR}(X, A_i) = \begin{cases} x_r + a_r^i, & X \neq A_i \\ 0, & X = A_i, \forall i = 1, \dots, m \end{cases}$$

Здесь  $x_r$  – расстояние от  $X$  до начала координат,  $a_r^i$  – расстояние от  $A_i$  до начала

координат.

Отметим, что в категорию  $\Gamma$  попадают и меры расстояния, указанные в категориях  $\mathcal{B}_1$ ,  $\mathcal{B}_2$ ,  $\mathcal{B}_3$ .

д) Функции расстояния специального назначения. Категория включает некоторые функции расстояния, не входящие в описанные выше категории, некоторые из которых широко используются в автоматической группировке нечисловых данных.

Иногда подобные функции являются достаточно сложными функциями координат, требующими дополнительной информации, такой как угол вращения координатных осей [456] или векторы возможных расстояний на типичной дороге [458, 459]. В последнем случае авторы рассматривают гибридные расстояния, такие как  $1 - \infty$  - нормы, которые в дальнейшем обобщены до норм прямых углов. Другая гибридная функция расстояния предложена Бринбергом и Лавом [434]. Она названа взвешенной 1-2-нормой, поскольку прямоугольные и евклидовы элементы преодолеваемого пути представлены взвешенными нормами  $l_1$  и  $l_2$ .

д1) Расстояния Жаккара. В задачах автоматической группировки функции расстояния используются как меры близости или различия между объектами данных, такими как документы, изображения и сигналы, если объекты данных представлены векторами [465]. Спат [466] вводит метрику расстояния Жаккара в задачу Вебера. Доказано, что данное расстояние является метрикой на множестве булевых векторов. Рассмотрим  $M$  как метрическое пространство и  $d(\cdot, \cdot)$  и  $d$  – одна из его метрик. Далее, пусть  $a_i \in M$  ( $i = 1, 2, \dots, m \geq 2$ ) известны. Минисуммная задача размещения (задача Вебера) с метрикой Жаккара определяется следующим образом:

$$\min z_J = \sum_{i=1}^m d_J^{(a)}(z, a_i)$$

где  $d_J^{(a)}(\cdot, \cdot)$  - метрика Жаккара. В [467] исследована вычислительная сложность задачи поиска медианы с метрикой Жаккара (собственно решения задачи Вебера с данной метрикой), доказано, что задача является  $NP$ -трудной. Уотсон [468] представил алгоритм спуска по вершинам для минисуммной задачи размещения с метрикой Жаккара и показал сходимость алгоритма к оптимальному решению.

д.2) Расстояние Хемминга. Ричард Хемминг представил названную его именем функцию расстояния как меру количеств ошибок, которые преобразуют одну строку бит в другую. С тех пор это нашло применение во многих прикладных задачах в различных

областях, таких как информационная теория, теория кодирования, криптография, комбинаторная оптимизация и др.[469]. Расстояние Хемминга применяется в задачах размещения на сети. Отметим, что данная метрика совпадает с метрикой  $l_1$  на булевом гиперкруге.

Задача улучшения расположения центральных мест была рассмотрена с расстоянием Хемминга. Задача, называемая CLISH - center location improvement problem under the sum-type Hamming distance – задача улучшения размещения центральных мест с расстоянием Хемминга типа суммы – состоит в улучшении сети таким образом, чтобы расстояние между вершиной  $S$  и другими вершинами сети не превышало установленного верхнего предела, и при этом стоимость модификации дуг сети достигало бы минимума. Модификации измеряются расстоянием Хемминга. Детально модели размещения с расстоянием Хемминга описаны в [470]. Безусловная задача размещения с расстоянием Хемминга определяется следующим образом:

$$\min z_H = \sum_{i=1}^n c_i d_H(w_i^\phi, w_i)$$

$$\text{где } d_H(w_i^\phi, w) = \begin{cases} 1, & w_i^\phi \neq w \\ 0, & w_i^\phi = w \end{cases}$$

Проблема использования расстояния Хемминга в минисуммных задачах размещения в общем случае является то, что функция  $d_H(.,.)$  является прерывной и невыпуклой, что делает известные методы, разработанные для  $l_p$  норм, не применимыми [471].

Кроме перечисленных выше двух специальных функций расстояния данная группа включает гильбертовы кривые, расстояния Махаланобиса, расстояния Хаусдорфа, расстояния Левенштейна и множество других мер расстояния [450, 405]. В [450] отмечено, что эти расстояния могут быть классифицированы как скругленные нормы с множеством параметров, нормы прямых углов и многоугольные расстояния, перечисленные в [420]. В [472] Казаковцев и др. представляют функцию расстояния, называемую метрикой такси или метрикой с минимальными расстояниями и предлагают алгоритм, использующий в своем составе процедуру Вайсфельда и специальную поисковую процедуру.

В таблице 5.1 подведены итоги литературного обзора функций расстояния, используемых в задачах размещения.

Таблица 5.1 – итоги литературного обзора применяемых функций расстояния

№ п/п	Функции расстояния	Тип задачи	Ссылки
1.	Сферические расстояния	Задачи размещения в крупном расстоянии	Aykin and Babu [446], Dhar and Rao [447], Katz and Cooper [449], Mwemezi and Huang [450], Wesolowsky [438].
2.	Френч метро	Задача Вебера, планирование расположения предприятия	Kazakovtsev and Stanimirovic [472], Osinuga et al. [445], Perreur and Thisse [441].
3.	Метрика Москвы	Задача Вебера, размещение центров реагирования в ЧС и контроль за состоянием преступности, поиск оптимального складирования грузов, перемещаемых подъемным краном или иным манипулятором	Kazakovtsev et al. [456], Klein [421], Perreur and Thisse [441].
4.	Метрика подъемного крана	Задача Вебера, задача размещения подъемных кранов, поиск оптимального складирования грузов, перемещаемых подъемным краном	Abdel-Khalel and Shawki [452], Alkriz and Mangin [454], Irizarry and Karan [455], Warzawski [453]
5.	Метрика лифта	Задача Вебера для систем транспортировки в шахтах, зданиях с лифтом, на складах со специальным погрузочно-разгрузочным оборудованием	Kazakovtsev and Stanimirovic [444], Kazakovtsev et al. [456], Stanimirovic et al. [158].
6.	Метрика британской железной дороги	Задачи являются тривиальными	Kazakovtsev et al. [456]
7.	Норма прямых углов	Задача Вебера, в том числе с ограничениями и барьерами	Brimberg [439], Fathali and Zaferanieh [460], Uster [440], Walter [406], Ward and Wendell [458, 459], Witzgall [457].
8.	Скругленные нормы	Задача Вебера, в том числе с ограничениями и барьерами	Brimberg [439], Thisse et al. [461], Walter [406], Zarimbal [405].
9.	Асимметричные меры	Планирование размещения предприятий, в том числе задачи с барьерами	Drezner and Wesolowsky [463], Hodgson et al. [462], Witzgall [457], Zarimbal [405].
10.	A-расстояние	Минимаксные задачи размещения пунктов реагирования в ЧС, разработка топологии микросхем, задача о минимальных остовных деревьях и др.	Kon and Kushimoto [423], Uno et al. [436], Widmayer et al. [464].
11.	Расстояние Жаккара	Задачи автоматической группировки	Chierichetti et al. [467], Spath [466], Watson [468].
12.	Расстояния Хемминга	Задача автоматической группировки, задача улучшения размещения центральных мест, квадратичная задача о назначениях	Kammerdiner et al. [469], Liu and Yao [471], Zhang et al. [470].
13	Прочие	Задача Вебера и множественная задача Вебера, размещение производственных мощностей и др.	Kazakovtsev and Stanimirovic [472], Zarimbal [405].

Среди источников, упомянутых в предыдущих параграфах, были отобраны 63 наиболее типичные работы по каждому из направлений исследований. Классификация функции расстояния приведена и в некоторых других исследованиях. Например, в нескольких статьях целью является оценка точности моделей с евклидовыми и прямоугольными расстояниями [430-433, 473, 428, 474], в то время как другие авторы лишь перечисляют используемые функции расстояния без какой-либо систематизации [427, 405]. Существуют работы, предлагающие схему систематизации различных областей исследования [475-479, 451]. В настоящем параграфе предлагается

таксономическое дерево с тремя уровнями, простым и понятным способом учитывающее широкий набор свойств функций расстояния. Поскольку, какая либо работа может принадлежать нескольким различным подкатегориям в одной из категорий, категории первого и второго классификационного уровня не являются строго взаимоисключающими. На первом уровне дерева классификации кратко представлены общие свойства функций расстояния с акцентом на тип координат. Далее располагаются несколько подкатегорий, в которых рассматриваются дальнейшие остальные компоненты и модели, а также области применения. Такой подход, однако, относится не ко всем категориям, полное классификационное дерево представлено ниже (в скобках дано буквенное обозначение класса функций расстояния, применяемое в таблице 5.2).

Типы функции расстояния:

1. По типу координатной системы:

1.1. Со сферическими координатами (a);

1.2. В полярных координатах (b):

1.2.1. Метрика французской столицы (French metro, c);

1.2.2. Метрика Москвы (d);

1.2.3. Метрика подъемного крана (e);

1.3. В декартовых координатах (f):

1.3.1. Неевклидовы, не основанные на нормах, сводимые к метрике  $l_1$

(g):

1.3.1.1. Метрика лифта (h);

1.3.1.2. Метрика британской железной дороги (i);

(1.3.1.3) Метрика французской столицы (French metro);

(1.3.1.4) Метрика Москвы;

(1.3.1.5) Метрика подъемного крана;

1.3.2. Основанные на нормах (j):

1.3.2.1. Норма прямых углов (k);

1.3.2.2. Скругленные нормы (l);

1.3.2.3. Асимметричные нормы (m);

1.3.2.4. A-расстояние (n);

2. Специальные функции расстояния (o):

1.4.1 Метрика Жаккара (p);

1.4.2 Расстояние Хемминга (q).

В таблице 5.2 представлены результаты исследования 63 статей по функциям расстояния. Знаком «X» обозначены метрики и меры расстояния, рассматриваемые в той или иной статье. Из таблицы видно, что большая часть литературы посвящена изучению единственной нормы, метрики или иной меры расстояния. Статей, содержащих обобщение, достаточно мало.

В следующих параграфах настоящей главы рассмотрены некоторые практически важные метрики и иные функции расстояния, для которых предложены алгоритмы решения задачи Вебера.

### 5.3 Свойства задачи Вебера с прямоугольной метрикой

Метрики, основанные на нормах Минковского –  $l_p$ -нормах – играют важнейшую роль в задачах автоматической группировки и теории. В частности, для  $l_p$ -норм были предложены эффективные алгоритмы [482, 113, 243]. Данные задачи обобщены для некоторых других мер расстояния на  $\mathbb{R}^n$  [113, 143, 114, 483], что позволяет использовать аналогичные модели и подходы для поиска оптимальных местоположений объектов в системах, использующих специальные средства транспортировки.

Задачи с нормой  $l_1$  – прямоугольной (иначе – манхэттенской) нормой обладают некоторыми характерными особенностями, позволяющими решать задачи быстрее, чем с более сложными метриками. Для задач Вебера с некоторыми метриками, являющимися удобными моделями оценки расстояний в системах со специальными механизмами в качестве средств транспортировки, могут быть построены эффективные алгоритмы, основанные на декомпозиции исходной задачи в задачи Вебера с прямоугольной метрикой. Таким задачам посвящены следующие параграфы.





Напомним, что задача Вебера заключается в поиске такой точки  $X$ , что сумма взвешенных расстояний от этой точки до некоторых известных точек  $A_1, \dots, A_N$  (точек-потребителей) достигает минимума [484]:

$$f(X, \mathcal{D}) = \sum_{i=1}^N w_i \mathcal{D}(A_i, X) \rightarrow \min. \quad (5.2)$$

Обозначение  $\mathcal{D}$  здесь представляет выбранную функцию расстояния, а вещественные значения  $w_i$  – неотрицательные весовые коэффициенты, преобразующие расстояние  $\mathcal{D}(A_i, X)$  в стоимость обслуживания точки-потребителя  $A_i$  из точки (центра)  $X$ . В «классической» постановке задачи  $\mathcal{D}(\cdot)$  определяется именно как евклидово расстояние, в более широкой трактовке допускается любая норма  $\mathbb{R}^n \rightarrow \mathbb{R}$  (точнее – метрика, основанная на норме, см. [485]), любая метрика или даже иная мера расстояния, метрикой не являющаяся.

Мы не рассматриваем здесь задачи Вебера с запрещенными областями, барьерами и т.д.

Если расстояние определяется метрикой  $l_1$  на плоскости  $\mathbb{R}^2$ , расстояние между  $A_i$  и  $X$  может быть оценено как

$$\mathcal{D}(A_i, X) = l_1(A_i, X) = |x_1 - a_1^i| + |x_2 - a_2^i|.$$

Здесь  $X = (x_1, x_2)$  и  $A_i = (a_1^i, a_2^i)$ ,  $i = 1, \dots, N$  – декартовы координаты.

Данная задача оптимизации может быть преобразована в две отдельные задачи с целевыми функциями

$$f_1(x_1) = \sum_{i=1}^N w_i |x_1 - a_1^i|, \quad (5.3)$$

$$f_2(x_2) = \sum_{i=1}^N w_i |x_2 - a_2^i|. \quad (5.4)$$

Обе задачи сводятся к обобщенной задаче

$$\min_x f(x) = \sum_{i=1}^N w_i |x - a^i|. \quad (5.5)$$

Для решения этой задачи может быть применен известный алгоритм [114, 243] (первый шаг является опциональным).

#### Алгоритм 5.1. Решить задачу Вебера с прямоугольной метрикой

Дано: Координаты  $a^1, \dots, a^N$  и соответствующие весовые коэффициенты  $w_1, \dots, w_N$ .

Шаг 1: Заменить каждое подмножество одинаковых элементов  $a^{i_1} = a^{i_2} = \dots =$

$a^{ij}$  единственным значением  $a^{i1}$  с весовым ко.эффициентом  $w_{i1} = w_{i1} + w_{i2} + \dots + w_{ij}$ . Таким образом, без потери общности можно предположить, что все элементы  $a^i$ ,  $i = 1, \dots, N$  различны.

Шаг 2: Упорядочить координаты  $a^i$ ,  $i = 1, \dots, N$  в порядке возрастания. Обозначим упорядоченную последовательность  $a^1 \leq a^2 \leq \dots \leq a^N$ . Пусть соответствующие им весовые коэффициенты обозначены  $\{w'_1, \dots, w'_N\}$ .

Обозначим частичные суммы весовых коэффициентов

$$S_w[0] = 0, \quad S_w[k] = \sum_{i=1}^k w'_i, \quad 1 \leq k \leq N.$$

Шаг 3: Если условие

$$S_w[j-1] < \frac{S_w[N]}{2} < S_w[j]$$

Выполняется для  $j$ -й координаты, то мы имеем оптимальное решение  $x = a^j$ .

Иначе, если условие

$$\frac{S_w[N]}{2} = S_w[j]$$

Выполняется для некоторого  $j$ , мы имеем множество оптимальных решений в интервале  $[a^j, a^{j+1}]$ . Ниже по тексту мы используем среднюю точку

$$x = \frac{a^j + a^{j+1}}{2}.$$

Шаг 4: ОСТАНОВ.

Для простоты изложения обозначим

$$W = \sum_{i=1}^N w_i. \quad (5.6)$$

В [144] показано, что в случае с евклидовой метрикой точка-потребитель  $A_k$  является точкой минимума целевой функции, если и только если значение ее весового коэффициента  $w_k$  превышает норму векторной суммы отдельных составляющих градиента целевой функции, преносимых другими токами-потребителями, в точке  $A_k$ , т.е.

$$\sqrt{\left(\sum_{i=1, i \neq k} \frac{w_i(a_1^i - a_1^k)}{r_i(a_1^k, a_2^k)}\right)^2 + \left(\sum_{i=1, i \neq k} \frac{w_i(a_2^i - a_2^k)}{r_i(a_1^k, a_2^k)}\right)^2} \leq w_k. \quad (5.7)$$

В неравенстве (5.7) использованы декартовы координаты. Через  $r_i(a_1^k, a_2^k)$  обозначим евклидово расстояние между  $i$ -й и  $k$ -й точками:

$$r_i(a_1^k, a_2^k) = \sqrt{(a_1^i - a_1^k)^2 + (a_2^i - a_2^k)^2}.$$

Задачи, удовлетворяющие условию (5.7), рассмотрены в [486].

Важным свойством как процедуры Вайсфельда [144], так и обобщенной процедуры Вайсфельда для множественной задачи Вебера [226], является то, что в случае, если решение задачи  $X$  (или одно из  $X_{j,j=\overline{1,n}}$  в случае множественной задачи, где  $n$  – число размещаемых объектов) совпадает с одной из точек — «потребителей»  $A_i, i=\overline{1,N}$ , то скорость сходимости данной процедуры к оптимальному решению крайне низка, что, впрочем, исправлено в усовершенствованной процедуре [148].

Здесь предлагается быстрый алгоритм (версия Алгоритма 5.1) для особого случая.

Лемма 5.1. Пусть  $a^1, \dots, a^N$  – координаты точек-потребителей (5.5), и  $w_1, \dots, w_N$  – соответствующие весовые коэффициенты. Выберем произвольно  $x^* \in \mathbb{R}$ . Обозначим  $Q_0 = \{i \mid a^i = x^*\}$ ,  $W^* = \sum_{i \in Q_0} w_i$ .

Тогда, если

$$W^* > \frac{W}{2}, \quad (5.8)$$

то  $x^*$  является точкой минимума функции (5.5).

*Доказательство.* Предположим, что  $a_2^1 = a_2^2 = \dots = a_2^N$  и обозначим  $a_1^i = a^i$ ,  $i = 1, \dots, N$ . В этом случае  $r_i(a_1^k, a_2^k) = \sqrt{(a_1^i - a_1^k)^2} = |a^i - a^k|$ , и задача Вебера с евклидовой метрикой сводится к рассматриваемой задаче Вебера с целевой функцией (5.5).

Поскольку  $w_i \geq 0$  для каждого  $i = 1, \dots, N$ , имеем

$$\begin{aligned} & \sqrt{\left(\sum_{i=1, i \neq k} \frac{w_i(a_1^i - a_1^k)}{r_i(a_1^k, a_2^k)}\right)^2 + \left(\sum_{i=1, i \neq k} \frac{w_i(a_2^i - a_2^k)}{r_i(a_1^k, a_2^k)}\right)^2} = \\ & = \sqrt{\left(\sum_{i=1, i \neq k} \frac{w_i(a^i - a^k)}{|a^i - a^k|}\right)^2} = \left|\sum_{i=1, i \neq k} \frac{w_i(a^i - a^k)}{|a^i - a^k|}\right| = \sum_{i=1, i \neq k} w_i. \end{aligned}$$

Таким образом, условие (5.7) сводится к

$$\sum_{i=1, i \neq k} w_i \leq w_k. \quad (5.9)$$

Следовательно, если (5.9) выполняется, то  $A_k$  является точкой минимума функции (5.5). Таким образом, можно доказать эквивалентность условий (5.8) и (5.9). Для этих целей рассмотрим три возможных случая:

Случай 1:  $Q_0 = \emptyset$ . В этом случае  $W^* = 0 \leq \frac{W}{2}$ , таким образом, (5.8) и (5.9) не выполняются, и в этом случае не достигается оптимальное решение  $x = x^*$ .

Случай 2:  $Q_0$  содержит 1 элемент:  $Q_0 = \{k\}$ . Тогда  $W^* = w_k$ , и

$$W^* = w_k > \frac{W}{2} = \frac{w_k + \sum_{i=1, i \neq k}^N w_i}{2} \Rightarrow 2w_k > w_k + \sum_{i=1, i \neq k}^N w_i \Rightarrow \sum_{i=1, i \neq k}^N w_i < w_k.$$

Таким образом, (5.8) эквивалентно (5.9).

Случай 3:  $Q_0$  содержит более одного элемента:  $Q_0 = \{k_1, k_2, \dots, k_n\}$ ,  $n \leq N$ .

Рассмотрим задачу

$$\min f'(x) = \sum_{i=1}^{N'} w'_i |x - a^i|,$$

где  $N' = N - n + 1$ ,  $w'_1 = \sum_{i \in Q_0} w_i$ ,  $a^1 = x^*$ ,  $w'_i = w_{j(i)}$ ,  $i = 2, \dots, N'$ ,  $a^i = a_{j(i)}$ ,  $i = 2, \dots, N'$ .

Обозначим

$$J(1) = \min\{j \mid j \in \{1, \dots, N\} \setminus Q_0\},$$

$$J(i) = \min\{j \mid j \in \{1, \dots, N\} \setminus Q_{0+}\},$$

$$Q_{0+} = Q_0 \cup \{J(1), \dots, J(i-1)\}.$$

В соответствии с нашими предположениями,

$$\begin{aligned} f'(x) &= w'_1 |x - a^1| + \sum_{i=2}^{N'} w'_i |x - a^i| = \sum_{i=1}^n w_{k_i} |x - a^{k_i}| + \sum_{i=1, i \notin Q_0}^N w_i |x - a^i| = \\ &= \sum_{i=1}^m w_i |x - a^i| = f(x). \end{aligned}$$

Новая задача эквивалентна (5.5), а Случай 3 эквивалентен Случаю 2.

Лемма 5.1 значительно сокращает вычислительную сложность Алгоритма 5.1 в частном случае (5.8). В этом случае алгоритм избегает выполнения процедуры упорядочивания координат асимптотической сложности  $O(N \log N)$  на Шаге 2. Если алгоритм включает проверку условия (5.8) перед Шагом 2, его вычислительная сложность линейна в рассматриваемом частном случае, и выражается формулой  $O(N \log N)$  в иных случаях.

#### 5.4 Алгоритм для задачи с метрикой лифта

Результаты настоящего параграфа опубликованы в [158].

Метрика лифта (иначе – метрика сборщика малины, метрика реки в джунглях) на плоскости  $R^2$  определяется как

$$\mathcal{L}(A, B) = \begin{cases} |x_1^A - x_1^B|, & x_2^A = x_2^B \\ |x_1^A| + |x_2^A - x_2^B| + |x_1^B|, & x_2^A \neq x_2^B, \end{cases} \quad (5.10)$$

где  $A(x_1^A, x_2^A)$  и  $B(x_1^B, x_2^B)$  - заданные точки (см., например, [487, 114]). Она может быть определена как минимальное евклидова длина пути, который может пролегать исключительно вдоль прямых, параллельных оси  $x$ , и отрезков на оси  $y$  [487, 114]. Таким образом, при  $x_2^A \neq x_2^B$ , расстояние между двумя точками  $A$  и  $B$  в метрике лифта равно сумме длин отрезков  $AA'$ ,  $A'B'$  и  $B'B$ , где  $A'$  и  $B'$  являются ортогональными проекциями точек  $A$  и  $B$  на ось  $y$  соответственно (рис. 5.1, слева). В противном случае,  $x_2^A = x_2^B$ , расстояние между  $A$  и  $B$  равно длине отрезка  $AB$  (Рис. 5.1, справа).

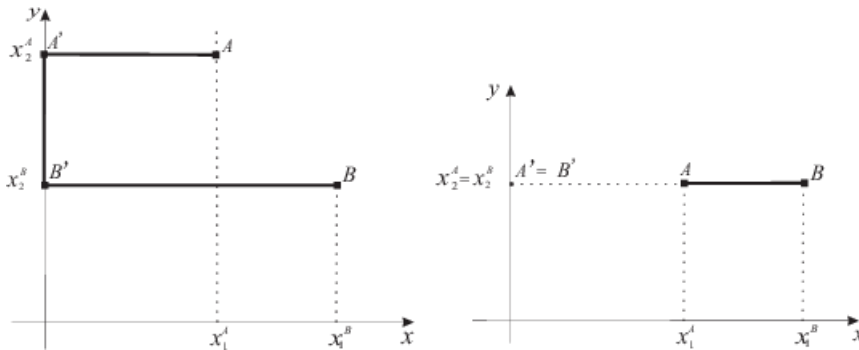


Рисунок 5.1 – Расстояния в метрике лифта. Слева – случай  $x_2^A \neq x_2^B$ , справа – случай  $x_2^A = x_2^B$

Метрика лифта может быть использована для оценки расстояний в городах с единственной главной улицей (соответствующей оси  $y$ ) и другими улицами, пересекающими ее под прямым углом. Такая структура уличного движения характерна, например, для Закинтоса – главного города одноименного греческого острова. Подобная же ситуация встречается в многоэтажных зданиях с единственным лифтом.

Для решения задачи с метрикой лифта множество  $C = \{A_1, \dots, A_N\}$  всех точек-потребителей разбивается на  $d$  классов эквивалентности, определенных как

$$C_L = \{C_{L_1} = \{A_i \mid a_2^i = y_1^*\}, C_{L_2} = \{A_i \mid a_2^i = y_2^*\}, \dots, C_{L_d} = \{A_i \mid a_2^i = y_d^*\}\}. \quad (5.11)$$

Полезно будет определить следующие множества для произвольной точки  $X = (x_1, x_2)$ :

$$Q_L(X) = \{i \mid x_2 = a_2^i, 1 \leq i \leq N\},$$

$$R_L(X) = \{a_2^i \mid i \in Q_L(X)\}.$$

Первая лемма позволяет найти решение для случая  $X \notin C_L$ .

Лемма 5.2. Пусть  $X^* = (x_1^*, x_2^*)$  является точкой минимума (1) при  $\mathcal{D} = \mathcal{L}$ . Пусть  $x_2^*$  – известное значение, удовлетворяющее  $x_2^* \neq a_2^i \forall i = \overline{1, N}$ . Тогда  $x_1^* = 0$ , а  $x_2^*$  является точкой минимума функции  $f_2(x_2)$ , определенной согласно (5.4).

*Доказательство.* Из (5.10) имеем

$$\mathcal{L}(X^*, A_i) = |x_1^*| + |a_2^i - x_2^*| + |a_1^i|$$

для каждого  $i = 1, \dots, N$ , и

$$\begin{aligned} f(X^*, \mathcal{L}) &= f_L(X^*) = \sum_{i=1}^N w_i (|x_1^*| + |a_2^i - x_2^*| + |a_1^i|) \\ &= \sum_{i=1}^N w_i |a_1^i| + \sum_{i=1}^N w_i |a_2^i - x_2^*| + \sum_{i=1}^N w_i |x_1^*|. \end{aligned} \quad (5.12)$$

Первое слагаемое в (5.12) является константой, и задача сводится к минимизации второй и третьей суммы. Для функции

$$f_1(x_1^*) = \sum_{i=1}^N w_i |x_1^*|$$

Единственной возможной точкой минимума очевидно является  $x_1^* = 0$ .

Тогда задача минимизации (5.12), согласно предположению  $x_1^* = 0$ , преобразуется в задачу минимизации целевой функции

$$\min_{x_2^*} f_2(x_2^*) = \sum_{i=1}^N w_i |x_2^* - a_2^i|. \quad (5.13)$$

Таким образом,  $x_2^*$  является точкой минимума функции  $f_2(x_2)$ , определенной в (5.3).

Лемма доказана.

Следствие 5.3 Если  $X^* = (x_1^*, x_2^*)$  является точкой минимума (5.2) с  $\mathcal{D} = \mathcal{L}$ , и  $x_2^* \neq y_j^* \forall j = \overline{1, d}$ , то  $x_1^* = 0$ .

Вторая лемма исследует возможные решения для случая  $X \in C_L$ .

Лемма 5.4 Пусть  $X^* = (x_1^*, x_2^*)$  является точкой минимума (5.2) с  $\mathcal{D} = \mathcal{L}$ . Пусть  $x_2^*$  – известное значение, удовлетворяющее условию  $x_2^* = y_j^*$  (что подразумевает  $X \in C_{L_j}$ ). Тогда  $x_1^*$  является точкой минимума функции

$$f_{L_1}(x_1) = \sum_{i=1}^N w_i |x_1 - a_1^{\beta(i)}|, \quad (5.14)$$

где

$$a_1^{\beta(i)} = \begin{cases} a_1^i, & a_2^i = y_j^* \\ 0, & a_2^i \neq y_j^*. \end{cases} \quad (5.15)$$

*Доказательство.* В соответствии с (5.10) целевая функция  $f_L(X^*) = f(X^*, \mathcal{L})$  распадается на две отдельные суммы:

$$f_L(X^*) = \sum_{i \notin Q_L(X^*)}^N w_i (|x_1^*| + |y_j^* - a_2^i| + |a_1^i|) + \sum_{i \in Q_L(X^*)}^N w_i |x_1^* - a_1^i|.$$

Сгруппировав первую часть первой суммы со второй суммой, имеем

$$f_L(X^*) = \sum_{i \notin Q_L(X^*)}^N w_i (|y_j^* - a_2^i| + |a_1^i|) + \sum_{i=1}^N w_i |x_1^* - a_1^{\beta(i)}|, \quad (5.16)$$

где  $a_1^{\beta(i)}$  определено в (5.15). Поскольку первой слагаемое в (5.16) является константой, задача нахождения  $x_1^*$  сводится к отысканию точки минимума, определенной в (5.14). Лемма доказана.

Лемма 5.5 При выполнении условий Леммы 5.4 точка минимума  $x_1^*$  функции  $f_{L_1}$ , определенной согласно (5.14) и (5.15), равна 0, если  $\sum_{i \notin Q_L(X^*)} w_i > \frac{W}{2}$ , где  $X^* = (x_1^*, x_2^*)$ .

*Доказательство.* Очевидно, что  $a_1^{\beta(i)} = 0 \quad \forall i \notin Q_L(X^*)$  Согласно Лемме 5.1, если

$$W^* = \sum_{i: a_1^{\beta(i)}=0} w_i = \sum_{i \notin Q_L(X^*)} w_i > \sum_{i=1}^N w_i / 2,$$

то 0 является точкой минимума функции (5.14). Лемма доказана.

Точка минимума функции  $f_2(x_2)$ , определенной в (5.3) и (5.4), как и точка минимума функции  $f_{L_1}(x_1)$ , определенной в (5.14), может быть найдена с помощью Алгоритма 5.1. В соответствии с этим предлагается следующий алгоритм для решения задачи Вебера на плоскости с метрикой лифта.

Алгоритм 5.2. Решение задачи Вебера с метрикой лифта.

Дано: Список координат точек-потребителей  $l_p = \{(a_1^1, a_2^1), \dots, (a_1^N, a_2^N)\}$  и список их весовых коэффициентов  $l_w = \{w_1, \dots, w_N\}$ .

Шаг 1: Разбить  $l_p$  на  $d$  подклассов  $C_{L_j}$  в соответствии с (5.11). Вычислить  $W = \sum_{i=1}^N w_i$ .

Шаг 2: Присвоить начальное значение множеству возможных решений  $\mathfrak{X}_L = \{(0, x_2^*)\}$ , где  $x_2^*$  - решение задачи (5.13), полученное с помощью Алгоритма 5.1.

Шаг 3: Для каждого  $j: 1 \leq j \leq d$  выполнять:

Шаг 3.1: Если  $W_j = \sum_{i: a_2^i = y_j^*} w_i < \frac{W}{2}$ , то присвоить  $x_1^* = 0$ .

Шаг 3.2: Иначе применить Алгоритм 5.1 для задачи (5.14) и получить решение  $x_1^*$ ; добавить  $(x_1^*, y_j^*)$  в множество возможных решений  $\mathfrak{X}_L$ .

Шаг 3.3: Следующая итерация цикла 3.

Шаг 4: Для каждого элемента  $X \in \mathfrak{X}_L$  вычислить выражение (5.2) и выбрать элемент из  $\mathfrak{X}_L$  с минимальным значением  $f_L(X)$ .

Шаг 5: Останов.

Лемма 5.6 *Множество возможных решений  $\mathfrak{X}_L$  содержит 1,2 или 3 элемента. Иными словами, Шаг 3.2 Алгоритма 5.2 (запускающий Алгоритм 5.1) выполняется 0,1 или 2 раза.*

*Доказательство.* На Шаге 2 добавляется первое возможное решение в множество  $\mathfrak{X}_L$ . Кроме того, возможное решение добавляется для каждого подкласса (пусть это будет  $j$ -й класс)  $C_{L_j}$  (см. Шаг 1), если

$$W_j = \sum_{i:a_2^i=y_j^*} w_i \geq \frac{W}{2}.$$

Пусть  $j'$ ,  $j''$  и  $j'''$  - индексы трех подклассов, такие, что  $j' \neq j''$ ,  $j' \neq j'''$ ,  $j'' \neq j'''$ , и  $W_j \geq W/2 \forall j \in \{j', j'', j'''\}$  (условие Шага 3.1).

Тогда  $W_{j'} + W_{j''} + W_{j'''} \geq 1.5W$ .

Обозначим  $Q''' = \{i | a_2^i \notin \{y_{j'}^*, y_{j''}^*, y_{j'''}^*\}\}$ . Тогда  $\sum_{i \in Q'''} w_i = W - W_{j'} - W_{j''} - W_{j'''} \leq -0.5W$ .

С другой стороны, поскольку весовые коэффициенты  $w_i$  неотрицательны, имеем  $\sum_{i \in Q'''} w_i \geq 0$ , что противоречит условиям Леммы. Очевидно, что условия Леммы выполняются, если количество подклассов не превышает 2. Лемма доказана.

### 5.5 Алгоритм для задачи с метрикой французской столицы (French Metro)

В настоящем разделе аналогичный подход применен к решению задачи с метрикой French metro. Идея алгоритма предложена в [296], [445]. Метрика French metro (иначе – радиальная метрика) на плоскости  $\mathbb{R}^2$  определяется следующим образом [88]:

$$\mathcal{F}(A, B) = \begin{cases} l_2(A, B), & A = cB \\ l_2(A, O) + l_2(B, O), & A \neq cB. \end{cases} \quad (5.17)$$

Здесь  $c$  - любое вещественное число,  $O$  – начало координат  $(0,0)$ , и



$$l_2(A, B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

– евклидово расстояние между точками  $A = (a_1, a_2)$  и  $B = (b_1, b_2)$ , что проиллюстрировано на рис. 5.2.

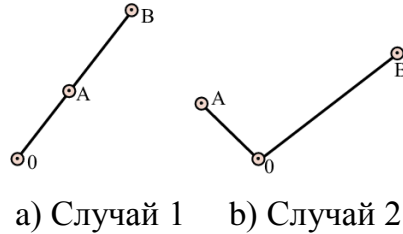


Рисунок 5.2. – Иллюстрация расстояний в метрике French Metro.

Возможны два случая. В Случае 1 обе точки  $A, B$  лежат на одной прямой (луче)  $A = cB$ , где  $c$  некоторая константа, и Случае 2, когда выполняется условие  $A \neq cB, \forall c \in \mathbb{R}$ . В метрике French metro, если обе точки лежат на одном луче (Случай 1), расстояние между ними равно евклидову расстоянию. В противном случае (Случай 2) расстояние вычисляется как сумма евклидовых расстояний от обеих точек до начала координат.

Для решения задачи Вебера с метрикой French metro прежде всего мы разобьем множество точек-потребителей  $\{A_1, \dots, A_N\}$  на  $d$  подклассов, таких, что все точки одного класса лежат на одном луче. Если множество  $\{A_1, \dots, A_N\}$  содержит начало координат  $(0,0)$ , такую точку выделим в отдельный подкласс. Для того, чтобы эту задачу выполнить было проще, перейдем от декартовых координат, в которых  $A_i = (a_1^i, a_2^i)$  к полярным. В полярных координатах каждая точка описывается углом  $a_\phi$  и радиальным расстоянием  $a_r$ . Условимся, что угол – это вторая координата, а радиальное расстояние – первая. Тогда в полярных координатах  $A_i = (a_r^i, a_\phi^i)$ , где

$$a_r^i = \sqrt{(a_1^i)^2 + (a_2^i)^2} \quad (5.18)$$

$$a_\phi^i = \begin{cases} -1, & a_1^i = 0, a_2^i = 0, \\ \frac{\pi}{2}, & a_1^i = 0, a_2^i > 0, \\ -\frac{\pi}{2}, & a_1^i = 0, a_2^i < 0, \\ \arctan\left(\frac{a_2^i}{a_1^i}\right), & a_1^i > 0, \\ \arctan\left(\frac{a_2^i}{a_1^i}\right) + \pi, & a_1^i < 0, \end{cases} \quad (5.19)$$

$$i = 1, \dots, N.$$

Здесь мы условно приняли угловую координату точки  $(0,0)$  равной  $-1$ .

Если угловая координата  $x_\phi$  искомой точки  $X$  задачи Вебера известна, то расстояние от  $X$  до  $i$ -й точки-потребителя равно

$$\mathcal{F}(A_i, X) = \begin{cases} |x_r - a_r^i|, & a_\phi^i = x_\phi \\ a_r^i + x_r, & a_\phi^i \neq x_\phi, \end{cases} \quad (5.20)$$

где  $X = (x_r, x_\phi)$  в полярных координатах.

Целевая функция задачи Вебера с метрикой French metro равна

$$f_F(X) = f(X, \mathcal{F}) = \sum_{i=1}^N w_i \mathcal{F}(A_i, X). \quad (5.21)$$

В полярных координатах наша задача очень близка к задаче Вебера с метрикой лифта. Разобьем множество  $C = \{A_1, \dots, A_N\}$  на  $d$  подмножеств с одинаковым значением угловой координаты:

$$C_F = \{C_{F_1} = \{A_i \mid a_\phi^i = \alpha_1\}, \dots, C_{F_d} = \{A_i \mid a_\phi^i = \alpha_d\}\}. \quad (5.22)$$

Обозначим

$$Q_F(X) = q_F(x_\phi) = \{i \mid x_\phi = a_\phi^i, i = 1, \dots, N\}, \quad (5.23)$$

$$R_F(X) = \{a_\phi^i \mid i \in Q_F(X)\}.$$

Докажем, что оптимальное решение принадлежит одному из классов  $C_F$ .

Лемма 5.7. Пусть  $X^* = (x_r^*, x_\phi^*)$  - точка минимума функции  $f_F$ , определенной согласно (5.21), где точки  $A_i$  описаны полярными координатами  $A_i = (a_r^i, a_\phi^i)$ , причем  $x_\phi^* \neq a_\phi^i \quad \forall i = \overline{1, N}$ . Тогда  $x_r^* = 0$ .

*Доказательство.* В случае  $x_\phi^* \neq a_\phi^i$  из (5.20) следует

$$\mathcal{F}(A_i, X^*) = a_r^i + x_r^*,$$

и целевая функция разбивается на 2 суммы:

$$f_F(X^*) = \sum_{i=1}^N w_i (a_r^i + x_r^*) = \sum_{i=1}^N w_i a_r^i + \sum_{i=1}^N w_i x_r^*.$$

Отметим, что первое слагаемое здесь – константа, и, принимая во внимание  $x_r^* \geq 0$  (как следует из (5.18)), точка минимума  $x_r^*$  этой функции равна 0. Лемма доказана.

Следствие 5.8. Если  $X^* = (x_r^*, x_\phi^*)$  является точкой минимума (5.21), и  $x_\phi^* \neq \alpha_j \quad \forall j = \overline{1, d}$ , где  $\alpha_j$  определены согласно (5.22), то  $x_r^* = 0$ .

Следующая лемма помогает найти решение в случае  $X \in C_{F_j}$ . Точнее, если угловая координата  $x_\phi^* = \alpha_j$  оптимальной точки  $X^*$  известна, решив задачу (5.24)–(5.25), мы получим координату  $x_r^*$ .

Лемма 5.9. *Предположим, что  $X^* = (x_r^*, x_\phi^*)$  - точка минимума функции (5.2) при  $\mathcal{D} = \mathcal{F}$ , причем значение  $x_\phi^*$  известно. Тогда  $x_r^*$  является точкой минимума функции*

$$f_{F_1}(x_r) = \sum_{i=1}^N w_i |x_r - a_r^{\beta(i)}|, \quad (5.24)$$

где

$$a_r^{\beta(i)} = \begin{cases} a_r^i, & a_\phi^i = x_\phi^* \\ 0, & a_\phi^i \neq x_\phi^*. \end{cases} \quad (5.25)$$

*Доказательство.* Целевая функция (5.2) разбивается на две суммы:

$$\begin{aligned} f(X^*, \mathcal{F}) &= f_F(X^*) = \sum_{i \notin Q_F(X^*)} w_i (a_r^i + x_r^*) + \sum_{i \in Q_F(X^*)} w_i |x_r^* - a_r^i| \\ &= \sum_{i \notin Q_F(X^*)} w_i a_r^i + \sum_{i \notin Q_F(X^*)} w_i x_r^* + \sum_{i \in Q_F(X^*)} w_i |x_r^* - a_r^i| \\ &= \sum_{i \notin Q_F(X^*)} w_i a_r^i + \sum_{i \notin Q_F(X^*)} w_i |x_r^* - 0| + \sum_{i \in Q_F(X^*)} w_i |x_r^* - a_r^i|. \end{aligned}$$

Принимая во внимание  $x_\phi^* = a_\phi^i \Leftrightarrow i \in Q(X^*)$ , и удалив константу (первое слагаемое) в  $f_F(X^*)$ , имеем новую эквивалентную целевую функцию  $f_{F_1}(x_r^*)$ , определяемую согласно (5.24) и (5.25). Здесь мы заменили множество наших точек-потребителей  $\{A_i\}$  на новое множество  $\{(a_r^{\beta(i)}, a_\phi^i)\}$ . Лемма доказана.

Лемма 5.10. *Точка минимума  $x_r^*$  функции (5.24) равна 0, если*

$$\sum_{i \notin Q_F(X^*)} w_i > \frac{W}{2},$$

где  $X^* = (x_r^*, x_\phi^*)$ .

*Доказательство.* По определению,  $a_r^{\beta(i)} = 0 \forall i \notin Q_F(X^*)$ . Таким образом, в соответствии с Леммой 5.1, если

$$W^* = \sum_{i: a_r^{\beta(i)}=0} w_i = \sum_{i \notin Q_F(X^*)} w_i > \sum_{i=1}^N w_i / 2,$$

то 0 является точкой минимума функции (5.24). Лемма доказана.

Для решения нашей задачи Вебера в полярных координатах предлагается следующий алгоритм.

Алгоритм 5.3. Решение задачи Вебера с метрикой French Metro.

Дано: Множество точки-потребителей в полярных координатах.

Шаг 1: Инициализация. Присвоить списку возможных решений значение  $\mathfrak{X}_F = \{(0,0)\}$ ; Вычислить  $W = \sum_{i=1}^N w_i$ ,

Шаг 2: Разобьем наше множество  $C = \{A_1, \dots, A_N\}$  на подмножества  $C_F$  с равной угловой координатой согласно (5.22). Обозначим через  $d$  количество этих подмножеств, а угловые координаты, им соответствующие, через  $\alpha_j, j=1, \dots, d$ .

Шаг 2.1. Для каждого  $j = 1, \dots, d$  вычислить

$$W_j = \sum_{i=1}^N w_i |\operatorname{sgn}(a_\phi^i - \alpha_j)|.$$

Шаг 2.2. Если  $W_j < W/2$ ,

Шаг 2.2.1. То перейти к Шагу 3 (в соответствии с Леммой 5.10, оптимальное решение задачи (5.24) равно 0, и это решение уже включено в  $\mathfrak{X}_F$ ).

Шаг 2.2.2. Иначе, принимая во внимание Лемму 5.9, определить значение координаты  $x_r$  точки минимума  $X = (x_r, \alpha_j)$ , лежащей на луче  $x_\phi = \alpha_j$ . Для этого заменяем все точки, не принадлежащие  $j$ -му подмножеству, на точку  $(0,0)$ , суммируя весовые коэффициенты заменяемых точек и выполнив для полученных новых данных Алгоритм 5.1. Иными словами, Алгоритм 5.1 применяется к задаче с множеством точек-потребителей  $\{(a_r^{\beta(i)}, a_\phi^i)\}$ , координаты которых вычисляются в соответствии с (5.25). Вместо всех точек, для которых  $a_\phi^i \neq \alpha_j$  мы вводим единственную точку  $a_r^{\beta(0)} = 0$  с весовым коэффициентом

$$w_{0'} = \sum_{i=1}^N w_i (1 - |\operatorname{sgn}(a_\phi^i - \alpha_j)|) = W - W_j, \quad j = 1, \dots, d.$$

Если Алгоритм 5.1 дает результат  $x_r \neq 0$ , то  $\mathfrak{X} = \mathfrak{X} \cup \{x_r, \alpha_j\}$ .

Шаг 3: Для каждого возможного решения  $X_k \in \mathfrak{X}_F, k \in \{1, \dots, r\}$ , вычислить значение целевой функции  $f_F(X)$ . Задача оптимизации

$$\text{minimize } f_F(X) = \sum_{i=1}^N w_i \mathcal{F}(A_i, X)$$

Решается полным перебором в множестве возможных решений  $\mathfrak{X}_F$ .

Шаг 4: СТОП.

Шаги 2, 2.1 и 2.2 Алгоритма 5.3 дают возможные решения, которые всегда принадлежат одному из подмножеств  $C_{F_1}, \dots, C_{F_d}$ , определенных согласно (5.22), т.е.,

$$x_\phi \in \{\alpha_1, \dots, \alpha_d\} \quad \forall X \in X_F.$$

Лемма 5.11. *Множество  $\mathfrak{X}_F$  возможных решений Алгоритма 5.3 содержит 1, 2 или 3 элемента (иными словами, Шаг 2.2.2 выполняется не более чем дважды).*

*Доказательство.* Ровно один элемент добавляется на Шаге 1, и одно возможное решение добавляется к  $\mathfrak{X}_F$  для каждого из подмножеств (пусть это будет  $j$ -е подмножество)  $C_{F_j}$  (см. Шаг 2.2.2), если  $W_j \geq W/2$ . Пусть  $j'$ ,  $j''$  и  $j'''$  – индексы трех подмножеств, таких, что  $j' \neq j''$ ,  $j' \neq j'''$ ,  $j'' \neq j'''$ , и  $W_j \geq W/2 \forall j \in \{j', j'', j'''\}$  (условие Шага 2.2).

$$\text{Тогда } W_{j'} + W_{j''} + W_{j'''} \geq 1.5W.$$

Используя обозначение  $Q''' = \{i | a_\phi^i \notin \{\alpha_{j'}, \alpha_{j''}, \alpha_{j'''}\}\}$ , имеем

$$\sum_{i \in Q'''} w_i = W - W_{j'} - W_{j''} - W_{j'''} \leq -0.5W.$$

Но  $\sum_{i \in Q'''} w_i \geq 0$ , вступает в противоречие с выражением выше. Если  $d \leq 2$ , условия леммы очевидно выполняются. Лемма доказана.

Замечание 5.12. *В случае  $\mathcal{D} = \mathcal{L}$  точка минимума  $X$  целевой функции задачи Вебера удовлетворяет условию  $X \in C_L$  или же лежит на оси  $x_2$ . В случае  $\mathcal{D} = \mathcal{F}$  точка минимума  $X$  всегда удовлетворяет условию  $X \in C_F$  или совпадает с началом координат. Таким образом, оба случая аналогичны: точка минимума  $X = (x_1, x_2)$  всегда лежит на одном из возможных путей, соединяющих одну из точек-потребителей с началом координат  $(0,0)$ . Эти возможные пути показаны на рис. 5.3.*

Метрика French Metro определена на плоскости. Тем не менее, определение и предложенный алгоритм могут быть легко адаптированы для трехмерного случая.

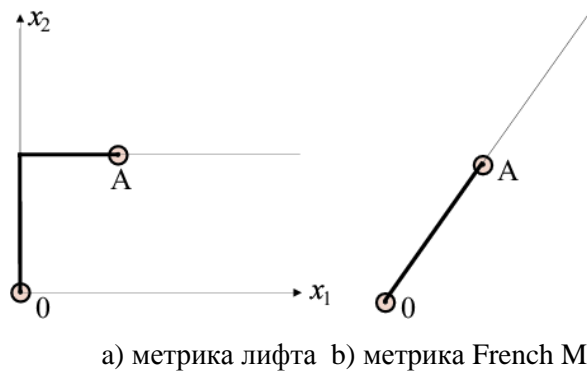


Рисунок 5.3 Возможные пути между началом координат 0 и точкой A в метриках лифта и French Metro

Для метрики French metro в трехмерном пространстве расстояния могут быть оценены по формуле (5.17) в случае использования декартовых координат. В трехмерном случае мы переходим к сферическим координатам [488], которые включают радиус – радиальное расстояние  $r$  и два угла: угол наклона  $\theta$  азимут  $\phi$ . И для преобразования декартовых координат точки-потребителя  $A = (a_1, a_2, a_3)$  в сферические координаты  $A = (a_r, a_\theta, a_\phi)$  будем использовать следующие формулы:

$$a_r^i = \sqrt{(a_1^i)^2 + (a_2^i)^2 + (a_3^i)^2},$$

$$a_\theta^i = \begin{cases} \arctan(a_2^i/a_1^i), & a_1^i > 0 \\ \arctan\left(\frac{a_2^i}{a_1^i}\right) + \pi, & a_1^i < 0 \\ \pi/2, & a_1^i = 0 \wedge a_2^i > 0 \\ -\pi/2, & a_1^i = 0 \wedge a_2^i < 0 \\ -1, & a_1^i = 0 \wedge a_2^i = 0 \end{cases}$$

$$a_\phi^i = \begin{cases} \arccos(a_3^i/a_r^i), & a_r^i \neq 0 \\ -1, & a_r^i = 0 \end{cases}$$

Для применения нашего алгоритма к трехмерному случаю заменим лишь три выражения – (5.22), (5.23) и (5.25) следующим образом:

$$C_F = \{C_{F_1} = \{A_i \mid a_\phi^i = \alpha_{\phi 1} \wedge a_\theta^i = \alpha_{\theta 1}\}, \dots, C_{F_d} = \{A_i \mid a_\phi^i = \alpha_{\phi d} \wedge a_\theta^i = \alpha_{\theta d}\}\};$$

$$Q_F(X) = \{i \mid x_\phi = a_\phi^i \wedge x_\theta = a_\theta^i, 1 \leq i \leq N\};$$

$$a_r^{\beta(i)} = \begin{cases} w_i, & a_\phi^i = \alpha_{\phi j} \wedge a_\theta^i = \alpha_{\theta j} \\ 0, & a_\phi^i \neq \alpha_{\phi j} \vee a_\theta^i \neq \alpha_{\theta j} \end{cases}, \quad i = 1, \dots, N, \quad j = 1, \dots, d.$$

## 5.6 Применение и примеры для моделей размещения с метриками лифта и French metro, оценка вычислительной сложности алгоритмов

В здании с единственной лестницей или лифтом и единственным коридором на каждом из этажей можно достичь точку А из точки В напрямую, если обе они лежат на одном этаже. В противном случае сперва придется достичь лифта, переместиться на нем, а затем достичь конечную точку. В этом случае метрика лифта – удобный способ оценки расстояния (длины пути).

Еще пример метрики лифта (заметим, что ее второе название – метрика сборщика малины [88]). Предположим, что складской погрузчик (управляемый человеком или же роботизированный) движется вдоль стеллажа. Если пол на складе не является идеально ровным, погрузчик не должен двигаться с высоко поднятым грузом, поскольку поддон с грузом может упасть от вибрации. В случае наиболее высоких стеллажей используются особые погрузчики, которые не могут двигаться с поднятым погрузочным механизмом. Для горизонтального движения поддон с грузом должен сперва быть опущен. И в этом случае поддон может быть перемещен из ячейки А в ячейку В напрямую, только если эти ячейки находятся одна под другой. В противном случае требуется опустить поддон, затем двигаться горизонтально, затем поднять.

Как следует из названия метрики французской столицы (French metro) [88, 114], Все наиболее важные дороги проходят через Париж. Поэтому, если два города не находятся на одной дороге, придется путешествовать через Париж. Аналогично, практически единственным центром авиационного сообщения в России является Москва. Если нужно попасть из города А в город В, Вам почти наверняка придется лететь через Москву. Например, через Москву придется лететь из Новосибирска в Омск, хотя города расположены рядом (в масштабах России). Исключение составляют длинные маршруты – например, Москва-Владивосток с посадкой в Иркутске. Здесь из Иркутска Во Владивосток можно попасть напрямую. Несколько иную структуру имеют городские дороги (улицы) таких городов, как Москва, Амстердам, Карлсруэ. В этих случаях более удобной мерой оценки расстояний является метрика Москвы, также описанная в настоящей главе ниже. Схемы метрополитенов таких городов, как Новосибирск, имеют единственную пересадочную станцию – центр координат. В системах, использующих такой метрополитен в качестве средства транспортировки,

также удобна метрика French Metro. Отметим, что в последнем случае возможно также использование в качестве модели задачи Вебера на графе.

Рассмотрим простой пример.

Пример 5.1. Решить задачу Вебера со следующими данными

$$A_1 = (8,8), w_1 = 7; \quad A_2 = (0,4), w_2 = 2; \quad A_3 = (0,2), w_3 = 3; \quad A_4 = (4,4), w_4 = 4$$

(рис.5.4).

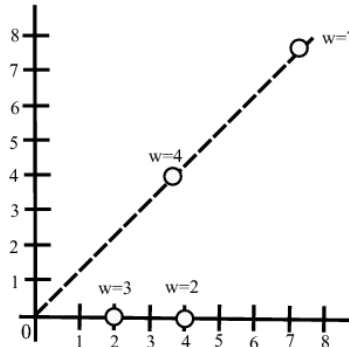


Рисунок 5.4 – Иллюстрация к примеру решения задачи Вебера

Имеем  $W=16$  (см. Шаг 1 Алгоритма 5.3).

В соответствии с (5.18) и (5.19), полярные координаты точек равны:

$$(11.313708, 0.78539819), (4.0, 0.0), (2.0, 0.0), (5.6568542, 0.78539819).$$

Разбиваем их на два подмножества (класса) в соответствии со значением угловой координаты согласно (5.22). Первое подмножество:  $\{(4.0, 0.0), (2.0, 0.0)\}$ , второе:  $\{(11.313708, 0.78539819), (5.6568542, 0.78539819)\}$ . Вычислим  $W_1=5$ ,  $W_2=11$  (Шаг 2). Включаем в множество возможных решений точку  $(0,0)$ .

Поскольку  $W_1 < W/2$ , пропускаем Шаг 2.2.2 и переходим к следующей итерации цикла 2.2. Поскольку  $W_2 > W/2$ , решим задачу Вебера для второго подмножества, добавив в него точку  $(0,0)$  с весом  $w_0 = W - W_1 = 5$ . Решение задачи с прямоугольной метрикой Алгоритмом 5.1 (см. Шаг 2.2.2 Алгоритма 5.3) ведет к результату  $x_r = a_r^4 = 5.6568542$ . Добавляем к множеству возможных решений точку  $(5.6568542, 0.78539819)$ , которая и станет решением алгоритма на его завершающем шаге.

В декартовых координатах имеем результат  $(4,4)$ .

Пример 5.2 Решим ту же задачу с метрикой лифта.

Опустим для краткости ход решения Алгоритмом 5.2. Отметим лишь, что в этом



случае результат равен  $X^* = A_2 = (0,4)$ .

Вычислительная сложность преобразования координат согласно (5.18) - линейна (асимптотическая сложность  $O(N)$ ). Так же линейна сложность вычисления  $W$  согласно (5.6). Сложность разбиения точек на классы эквивалентности (при использовании эффективных алгоритмов сортировки) с одновременным вычислением значений  $W_j$  можно оценить как  $O(N \log N)$ . Как доказано выше, запуск Алгоритма 5.1 выполняется не более двух раз. В этом алгоритме Шаг 1.2.2 (сортировка) имеет вычислительную сложность  $O(N_j \log N_j)$ . Поскольку  $N_j \leq N$ , сложность не превышает  $O(N \log N)$ . Далее, сложность вычисления частичных сумм (Шаг 1.3 Алгоритма 5.1) равна  $O(N_j)$ , чем можно пренебречь в асимптотической формуле сложности Алгоритма 5.1, поскольку  $O(N_j) < O(N_j \log N_j)$ . Конечный шаг Алгоритмов 5.2 и 5.3 – оценка значений целевой функции – имеет асимптотическую сложность  $O(N)$ , чем так же можно пренебречь. Таким образом, асимптотическая сложность всего алгоритма 5.2 или 5.3 равна  $O(N \log N) + O(N \log N) = O(N \log N)$ .

Результаты вычислительных экспериментов на задачах большой размерности показывают практически линейную зависимость времени счета от  $N$  для Алгоритмов 5.2 и 5.3. Задачи с метрикой French metro генерировались следующим алгоритмом.

Алгоритм 5.4. Генерация тестовых примеров задачи Вебера с метрикой French metro.

Дано: число точек-потребителей  $N$ , число подклассов ("лучей", на которых лежат эти точки)  $N_r$ .

Шаг 1. Присвоить  $i = 1$ ;  $C = \emptyset$ ;  $\mathfrak{W} = \emptyset$ ;  $N_{items} = 0$ .

Шаг 2. Присвоить  $\alpha = 2\pi f_{random}()$ .

Шаг 3. Если  $i = d$ , то  $n_p = N - N_{items}$ , иначе, если  $i = N_r - 1$ , то присвоить  $n_p = 1 + [(N - N_{items} - 1)f_{random}()]$ ;

иначе присвоить  $n_p = 1 + [2(N - N_{items}) / (d - i) f_{random}()]$ .

Присвоить  $a = 0$ ;  $\delta = 0.001 + 100f_{random}()$ ;  $j = 1$ .

Шаг 4. Присвоить  $a = a + \delta f_{random}()$ ;  $w = (f_{random}() + 0.001)^2$ ;  $C = C \cup \{a \sin \alpha, a \cos \alpha\}$ ;  $\mathfrak{W} = \mathfrak{W} \cup w$ ;  $N_{items} = N_{items} + 1$ ;  $j = j + 1$ ;

Шаг 6. Если  $j \leq n_p$ , то вернуться к Шагу 4;

Шаг 7. Присвоить  $i = i + 1$ ;

Шаг 8. Если  $i \leq d$ , то вернуться к Шагу 1;

Шаг 9. «Перетасовать» - расположить в случайном порядке элементы множеств  $\mathcal{C}$  и  $\mathcal{W}$ ; ОСТАНОВ.

Здесь  $f_{random}()$  - генератор случайных чисел в интервале  $[0; 1)$ . Возвращаемые значения алгоритма – это пары. При фиксированном количестве классов точек-потребителей (здесь  $d=75$ ) мы имеем зависимость времени выполнения Алгоритма 5.3 от  $m$ , изображенную на рис. 5.5.

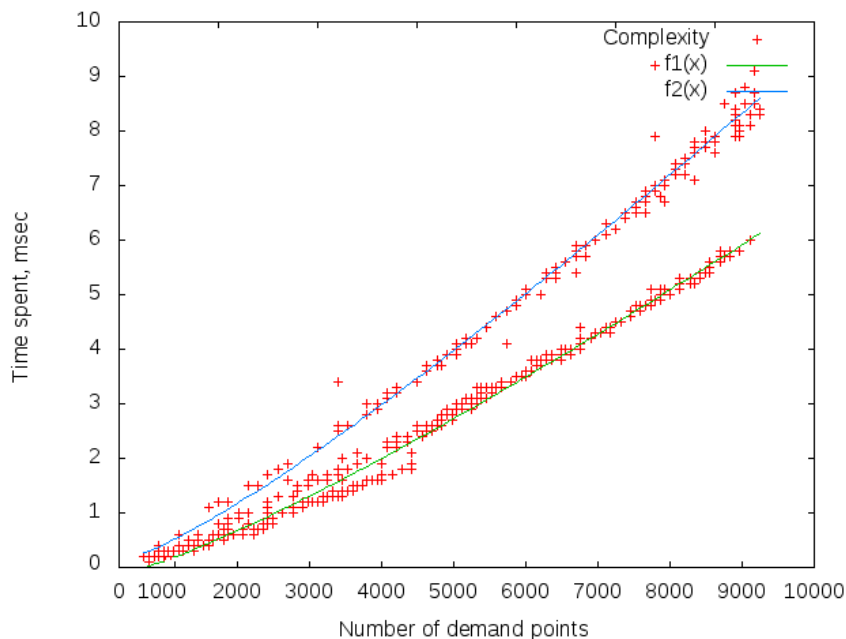


Рисунок 5.5 – Иллюстрация вычислительной сложности Алгоритма 5.3. По оси абсцисс – число точек-потребителей  $m$ , по оси ординат – время в миллисекундах.

Нижняя линия иллюстрирует случаи, когда решением является точка  $(0,0)$ , и выполнение Алгоритма 5.1 не требуется ( $W_j < W/2 \forall j: 1 = \overline{1, d}$ ). Верхняя линия соответствует иным случаям. На нижней линии в действительности заметны два участка, различия между которыми можно объяснить различным поведением связки Центральный процессор – кэш-память в зависимости от объема данных (данные малых задач целиком укладываются в кэш).

Таким образом, и задача Вебера с метрикой лифта, и задача с метрикой French метро фактически сводятся к задаче с прямоугольной метрикой на прямой, для которой давно известны эффективные алгоритмы решения. Предложенные здесь алгоритмы

обладают асимптотической сложностью  $O(m \log m)$ , причем вычислительные примеры показывают практически линейную зависимость времени решения от объема данных.

Совокупность возможных путей от точек-потребителей до начала координат (см. рис.5.3) с расположенными на них точками-потребителями образуют некую сеть (граф). Поскольку доказано, что оптимальное решение задачи Вебера лежит на одном из возможных путей, можно решать задачу как задачу Вебера на графе. Длины ребер этого графа будут соответствовать расстояниям между соседними точками-потребителями при движении вдоль возможных путей. На рис. 5.4 точки-потребители (белые кружки), разбитые на два класса (классам соответствуют значения угловых координат) и начало координат (черный кружок) образуют сеть из пяти узлов (рис. 5.6), которая преобразована в двоичное дерево в правой части рис.5.6 путем добавления дополнительных вершин с нулевым весом (дополнительные черные кружки).

Число таким образом добавляемых вершин равно  $N_{added} = (N + N_{01} - d)$ .

Здесь  $N_{01} = 1$ , если начало координат не входит в число точек-потребителей (в противном случае  $N_{01} = 0$ ),  $d$  - число классов ("лучей"). Длины дополнительных ребер можно выбрать произвольно, например, равными 0.

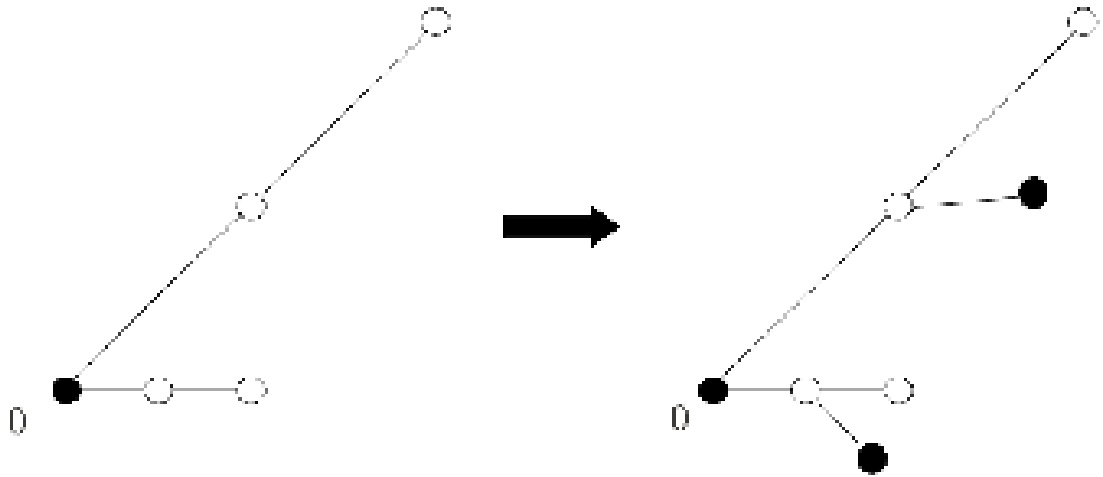


Рисунок 5.6 – Преобразование задачи в задачу на графе – двоичном дереве

Для решения такой задачи можно применить известный алгоритм [489, 218]. Для сетей с максимальной степенью узлов, равной 3 (наш случай), алгоритм, описанный в [218], имеет вычислительную сложность  $O(p^2 n^2)$ , где  $p$  - число искомых точек на сети (1 в нашем случае),  $n$  - число вершин ( $n = N + N_{added}$ ). Для применения подобного алгоритма, вначале все-равно придется исполнить первый шаг Алгоритма 5.3 (разбиение

точек-потребителей на классы), который и определяет асимптотическую сложность всего алгоритма. Таким образом, подобные приемы преобразования задач Вебера в соответствующие задачи на графе не снижают общую асимптотическую сложность решения.

### 5.7 Обобщение метрик лифта и французской столицы, алгоритм для множественной задачи Вебера

Множественная задача Вебера формулируется следующим образом:

$$\arg \min f(X_1, X_2, \dots, X_n, D) = \sum_{i=1}^m \sum_{j=1}^n w_{i,j} D(A_i, X_j) + \sum_{j=1}^{n-1} \sum_{k=j+1}^n v_{j,k} D(X_j, X_k) \quad (5.26)$$

Обозначение  $D$  здесь представляет выбранную функцию расстояния. Вещественные неотрицательные коэффициенты  $w_{i,j}$  преобразуют расстояния в стоимость обслуживания соответствующего ( $i$ -го) потребителя  $j$ -м размещаемым объектом, вещественные коэффициенты  $v_{j,k}$  преобразуют расстояние между  $j$ -м и  $k$ -м размещаемым объектом в стоимость транспортировки между ними. Расположение  $j$ -го объекта описывается координатами  $X_j = (x_1^j, x_2^j)$ ,  $j = \overline{1, n}$ . Как и задача Вебера с единственным размещаемым объектом, в случае метрики  $l_1$ , данная задача разбивается на 2 отдельных обобщенных задачи оптимизации

$$\arg \min_{x_r^1, \dots, x_r^n} f_2(x_r^1, \dots, x_r^n) = \sum_{j=1}^n \sum_{i=1}^m w_{i,j} |x_r^j - a_r^i| + \sum_{j=1}^{n-1} \sum_{k=j+1}^n v_{j,k} |x_r^j - x_r^k|, r \in \{1, 2\} \quad (5.27)$$

Известный алгоритм для этой задачи [243] определяется следующим образом (первый шаг — опциональный). Поскольку задачи при  $r=1$  и  $r=2$  полностью аналогичны, индекс  $r$  — номер координаты — при описании алгоритма во всех выражениях опустим.

Алгоритм 5.5 *Найти решение задачи (5.27).*

Дано: Координаты  $a^1, \dots, a^m$  и соответствующие весовые коэффициенты  $w_{ik}$ ,  $i = \overline{1, m}$ ,  $k = \overline{1, n}$  и  $v_{kl}$ ,  $k = \overline{1, (n-1)}$ ,  $l = \overline{(k+1), n}$ .

Шаг 1: Заменить каждый набор одинаковых элементов множества координат  $a^{i1} = a^{i2} = \dots = a^{ij}$  единственным значением  $a^{i1}$ , имеющим вес  $w_{i1,k} = w_{i1,k} + w_{i2,k} + \dots + w_{ij,k}$  для всех  $k = \overline{1, (n-1)}$ . Таким образом, после такой замены можно считать, что все координаты

$a^i, i = \overline{1, m}$  различны.

Шаг 2: Отсортировать координаты  $a^i, i = \overline{1, m}$  в неубывающем порядке. Обозначим отсортированную последовательность  $a^{1^1} \leq a^{2^2} \leq \dots \leq a^{m^m}$ , а соответствующие им весовые коэффициенты  $w'_{1k}, \dots, w'_{mk}, k = \overline{1, n}$ .

Введем обозначение частичных сумм этого списка коэффициентов

$$S_w[0, k] = 0, S_w[K, k] = \sum_{i=1}^K w'_{ik}, 1 \leq K \leq m \forall k = \overline{1, n}.$$

Присвоим  $j=1$ .

Шаг 3: Если условие  $S_w[j, k] \leq S_w[m, k]/2$  выполняется для всех  $k = \overline{1, n}$ , то присвоим  $j=j+1$  и повторим этот шаг.

Шаг 4: Если  $S_w[m, k]/2 \leq S_w[j, k]$  выполняется для всех  $k = \overline{1, n}$ , то оптимальным решением является  $x_k = a_j, k = \overline{1, n}$ .

Шаг 5: Иначе решим задачу линейного программирования

$$\arg \min \left( \sum_{i=1}^m \sum_{k=1}^n w_{ik} s_{ik} + \sum_{k=1}^{n-1} \sum_{l=k+1}^n v_{kl} t_{kl} \right);$$

$$-s_{ik} \leq x_k - a_i \leq s_{ik}, i = \overline{1, m}, k = \overline{1, n};$$

$$-t_{kl} \leq x_k - x_l \leq t_{kl}, k = \overline{1, (n-1)}, l = \overline{(k+1), n}.$$

Здесь  $s_{ik}$  и  $t_{kl}$  — дополнительные переменные. Эффективный алгоритм для решения таких задач линейного программирования предложен в [243].

Шаг 6: СТОП.

В дальнейшем будем использовать обозначение

$$W = \sum_{i=1}^m w_i.$$

Рассмотрим задачу с обобщенной метрикой

$$G(A_i, X) = \begin{cases} |x_1 - a_1^i|, & x_2 = a_2^i, \\ |x_1| + v|x_2 - a_2^i| + |a_1^i|, & x_2 \neq a_2^i \end{cases} \quad (5.28)$$

Если коэффициент  $v=1$ , эта метрика совпадает с метрикой лифта. При  $v=0$ , как будет показано ниже, данная метрика идентична метрике French metro при условии

использования полярных координат.

Для решения задачи размещения этой метрике  $G$ , как и в случае с метриками лифта и French metro, введем для множества точек требования  $C=\{A_1, \dots, A_m\}$  множество классов эквивалентности:

$$C_G = \{C_{G1} = \{A_i \mid a_2^i = y_1^*\}, C_{G2} = \{A_i \mid a_2^i = y_2^*\}, \dots, C_{Gd} = \{A_i \mid a_2^i = y_d^*\}\}. \quad (5.29)$$

Для точки  $X$  обозначим следующие множества:

$$Q_G(X) = \{i \mid x_2 = a_2^i, 1 \leq i \leq m\};$$

$$R_G(X) = \{a_2^i \mid i \in Q_G(X)\}.$$

Предлагаемый алгоритм основан на двух леммах. Доказательство первой из них для метрики лифта приведено в [158]. Первая лемма ограничивает поиск возможных решений для случая  $X_k \notin C_G, n = \overline{1, n}$ . Сформулируем лемму для метрики  $G$  (доказательство аналогично приведенному в [158]).

Лемма 5.13. Пусть  $X^* = (x_1^{k*}, x_2^{k*})$  - точка минимума функции (5.26) при  $D=G$ . Предположим, что  $x_2^{k*}$  - известное значение, такое, что  $x_2^{k*} \neq a_2^i \forall i = \overline{1, m}$ . Тогда  $x_1^{k*} = 0$ , при этом  $x_2^{k*}$  - точка минимума функции  $f_2(x_2^k)$ , определенной в (5.27).

Вторая лемма ограничивает круг возможных решений для случая  $X_k \in C_G, n = \overline{1, n}$ .

Доказывается аналогично Лемме 5.2.

Лемма 5.14. Пусть при  $X_1=X_1^*, X_2=X_2^*, \dots, X_n=X_n^*$  достигается минимум функции (5.26),  $X_k^* = (x_1^{*k}, x_2^{*k})$ ,  $k=1, n$ ,  $D=G$ . Предположим, что все  $x_2^{*k}$ ,  $k=1, n$  известны и  $x_2^{*k_l} = y_j^* \forall l = \overline{1, z}, k_l \in \{\overline{1, n}\}, 0 \leq z \leq n$ . Введем новые коэффициенты

$$a_1^{\beta(i)} = \begin{cases} a_1^i, & a_2^i = y_j^*, \forall i = \overline{1, m}, \\ 0, & a_2^i \neq y_j^*, \end{cases} \quad (5.30)$$

и новые переменные

$$x_1^{\gamma(k)} = \begin{cases} x_1^i, & x_2^k = y_j^*, \forall k = \overline{1, n}, \\ 0, & x_2^k \neq y_j^*, \end{cases} \quad (5.31)$$

Тогда  $x_1^{*k_l} = x_1^{\gamma(*k_l)} \forall k \in \{i \mid x_2^{*i} = y_j^*\}$ , где  $x_1^{\gamma(*1)}, \dots, x_1^{\gamma(*n)}$  - значения, при которых достигается минимум функции

$$f_{G_1}^M = \sum_{k=1}^n \sum_{i=1}^m w_{ik} |x_1^{\gamma(k)} - a_1^{\beta(i)}| + \sum_{k=1}^{n-1} \sum_{l=k+1}^n v_{kl} |x_1^{\gamma(k)} - x_1^{\gamma(l)}|. \quad (5.32)$$

*Доказательство.* Обозначим множества индексов

$$Q_A = \{i \mid a_2^i = y_j^*, i \in \overline{1, m}\},$$

$$Q_X = \{k \mid x_2^{*k} = y_j^*, k \in \overline{1, n}\}.$$

В соответствии с (5.28),

$$\begin{aligned} f_G(X_1, \dots, X_n) = & \sum_{k \in Q_X} \sum_{i=1}^n w_{ik} G(A_i, X_k) + \sum_{k \in Q_X} \sum_{i \in Q_A} w_{ik} |x_1^k - a_1^i| + \\ & + \sum_{k \in Q_X} \sum_{i \in Q_A} w_{ik} |x_1^k| + \sum_{k \in Q_X} \sum_{i \in Q_A} w_{ik} |a_1^i| + \\ & + \sum_{k \in Q_X} \sum_{i \in Q_A} w_{ik} |a_2^i - x_2^k| + \sum_{k \in Q_X} \sum_{l \in Q_X, l > k} v_{kl} G(X_k^*, X_l^*) + \\ & + \sum_{k \in Q_X} \sum_{l \in Q_X, l > k} v_{kl} |x_1^k| + \sum_{k \in Q_X} \sum_{l \in Q_X, l > k} v_{kl} |x_1^l| + \sum_{k \in Q_X} \sum_{l \in Q_X, l > k} v_{kl} |x_2^k - x_2^l| + \\ & + \sum_{k \in Q_X} \sum_{l \in Q_X, l > k} v_{kl} |x_1^k| + \sum_{k \in Q_X} \sum_{l \in Q_X, l > k} v_{kl} |x_1^l| + \\ & + \sum_{k \in Q_X} \sum_{l \in Q_X, l > k} v_{kl} |x_2^k - x_2^l| + \sum_{k \in Q_X} \sum_{l \in Q_X, l > k} v_{kl} |x_1^k - x_1^l|. \end{aligned} \quad (5.33)$$

Здесь  $\overline{Q_X} = \overline{1, n} \setminus Q_X$ ,  $\overline{Q_A} = \overline{1, m} \setminus Q_A$ .

Предположим, что  $x_1^k \forall k \in \overline{Q_X}$  известны. В этом случае 1-е, 4-е, 5-е, 6-е, 7-е, 9-к, 11-е и 12-е слагаемые в (5.33) - константы. Удалив их, в соответствии с (5.30) и (5.31), получаем эквивалентную целевую функцию:

$$\begin{aligned} f_{G_0}(X_1, \dots, X_n) = & \sum_{k \in Q_X} \sum_{i \in Q_A} w_{ik} |x_1^k - a_1^i| + \sum_{k \in Q_X} \sum_{i \in Q_A} w_{ik} |x_1^k - 0| + \\ & + \sum_{k \in Q_X} \sum_{l \in Q_X, l > k} v_{kl} |x_1^l - 0| + \sum_{k \in Q_X} \sum_{l \in Q_X, l > k} v_{kl} |x_1^k - 0| + \sum_{k \in Q_X} \sum_{l \in Q_X, l > k} v_{kl} |x_1^k - x_1^l| = \\ = & \sum_{k \in Q_X} \sum_{i \in Q_A} w_{ik} |x_1^{\gamma(k)} - a_1^{\beta(i)}| + \sum_{k \in Q_X} \sum_{i \in Q_A} w_{ik} |x_1^{\gamma(k)} - a_1^{\beta(i)}| + \sum_{k \in Q_X} \sum_{l \in Q_X, l > k} v_{kl} |x_1^{\gamma(l)} - x_1^{\gamma(k)}| + \\ & + \sum_{k \in Q_X} \sum_{l \in Q_X, l > k} v_{kl} |x_1^{\gamma(k)} - x_1^{\gamma(l)}| + \sum_{k \in Q_X} \sum_{l \in Q_X, l > k} v_{kl} |x_1^{\gamma(k)} - x_1^{\gamma(l)}| = \\ = & \sum_{k \in Q_X} \sum_{i=1}^m w_{ik} |x_1^{\gamma(k)} - a_1^{\beta(i)}| + \sum_{k \in Q_X} \sum_{l \in Q_X, l > k} v_{kl} |x_1^{\gamma(l)} - x_1^{\gamma(k)}| + \sum_{k \in Q_X} \sum_{l=k+1}^n v_{kl} |x_1^{\gamma(k)} - x_1^{\gamma(l)}|. \end{aligned} \quad (5.34)$$

В соответствии с (5.31),

$$x_1^{\gamma(k)} = 0 \forall k \in \overline{Q_X} \Rightarrow \sum_{k \in Q_X} \sum_{i=1}^m |x_1^{\gamma(k)} - a_1^{\beta(i)}| = const, \quad (5.35)$$

$$x_1^{\gamma(k)} = 0 \forall k \in \overline{Q_X}, x_1^{\gamma(l)} = 0 \forall l \in \overline{Q_X} \Rightarrow \sum_{k \in Q_X} \sum_{l \in Q_X} |x_1^{\gamma(k)} - x_1^{\gamma(l)}| = 0. \quad (5.36)$$

Добавив (5.35) и (5.36) к (5.34), получим эквивалентную задачу оптимизации с

целевой функцией (5.32). Лемма доказана.

Таким образом, с использованием алгоритма, предложенного в [243], с учетом Леммы 5.13 и Леммы 5.14, получаем следующий алгоритм для решения задачи с метрикой (5.28).

Алгоритм 5.6. Решить множественную задачу Вебера (5.26) с метрикой  $G$ .

Дано: список координат точек требования  $l_p = \{(a_1^1, a_1^2), \dots, (a_m^1, a_m^2)\}$ , список их весовых коэффициентов  $l_w = \{w_{ik} \mid i = \overline{1, m}, k = \overline{1, n}\}$  и список коэффициентов  $l_v = \{v_{kl} \mid k = \overline{1, n-1}, l = \overline{k+1, n}\}$ .

Шаг 1. Разбить  $l_p$  на  $d$  подклассов согласно (5.29).

Рассчитать  $W_k = \sum_{i=1}^m w_{ik} \forall k = \overline{1, n}$ .

Шаг 2. Присвоить списку  $X_G$  значение  $X_G = \{(0, x_2^{*1}), \dots, (0, x_2^{*n})\}$ , где  $x_2^{*k}$  – решение задачи (5.27) при  $r=2$  с использованием Алгоритма 5.5.

Шаг 3. Для каждого  $j: 1 \leq j \leq d$  выполнить следующие шаги:

Шаг 3.1. Если  $\exists k: W_{jk} = \sum_{i: a_2^i = y_j^*} w_{ik} \geq W_k / 2, k \in \{\overline{1, n}\}$ , то

Шаг 3.1.2. Применить Алгоритм 5.5 для задачи (5.27) при  $r=1$  с координатами точек требования, вычисленными согласно (5.30) для получения списка значений  $x_1^{\gamma(k)}, k = \overline{1, n}$ . Заменить элементы  $X_G$  соответствующими значениями  $(x_1^{*k}, y_j^*)$  для всех  $k: x_1^{\gamma(k)} \neq 0, k = \overline{1, n}$ .

Шаг 4. Стоп. Возврат  $X_G$ .

Условие Шага 3.1 основано на следующем утверждении [486]. Если

$$\exists j, k: w_{jk} \geq \sum_{i=1}^m w_{i,k} / 2, k \in \{\overline{1, n}\},$$

то  $x^{\gamma(k)} = a^{\beta(j)}$  – оптимальное решение. Поскольку  $a^{\beta(i)} = 0 \forall i: a_2^i \neq y_j^*$ , и все точки с одинаковыми (в данном случае — нулевыми) координатами заменяются единственной точкой (Шаг 1 Алгоритма 5.5) с общим весом  $w_{0,k} = \sum_{i: a_1^i \neq y_j^*} w_{i,k}, \forall k \in \{\overline{1, n}\}$ , при невыполнении условия Шага 3.1 решением Алгоритма 5.5 будет нулевая точка.

Рассмотрим следующий пример. Пусть автопогрузчик на складе движется вдоль



высоких стеллажей. В случае наиболее высоких стеллажей используются особые погрузчики, которые перед подъемом груза должны специальным образом зафиксировать свое положение. Таким образом, горизонтальное движение с грузом возможно только при опущенной платформе погрузчика. Поддон с грузом из ячейки  $A$  в ячейку  $B$  может быть доставлен по прямой, только если ячейка  $A$  расположена над ячейкой  $B$  (или наоборот), в одном вертикальном ряду. В противном случае вначале груз должен быть опущен с ячейки  $A$  до минимальной высоты погрузчика, затем совершено горизонтальное перемещение погрузчика, и лишь затем — вновь подъем до ячейки  $B$ .

Предположим, в некоторых частично свободных ячейках комплектуются  $n$  заказов покупателей. В этом случае требуется выбрать  $n$  таких ячеек, чтобы суммарное взвешенное расстояние в метрике лифта от них до исходных ячеек с товаром было минимальным. При этом в качестве весовых коэффициентов используются требуемые количества товаров, находящиеся в исходных ячейках, подсчитанные в поддонах (за одну операцию погрузчик перемещает один поддон). В этом случае мы имеем задачу в метрике лифта.

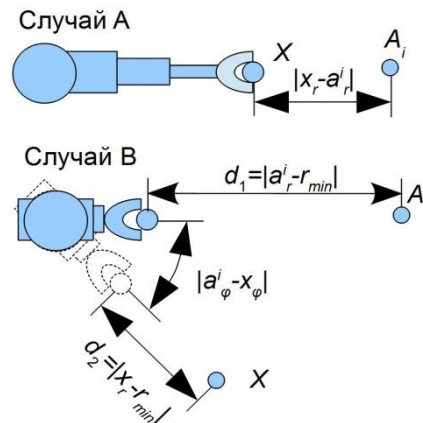


Рисунок 5.7 – Иллюстрация к задаче с обобщенной метрикой

Выбор метрики зависит от свойств пространства и способа перемещения в нем. Задача с универсальной метрикой  $G$  может быть показана на следующем примере. Пусть манипулятором с телескопической стрелой, вращающейся вокруг неподвижной оси (рис. 5.7) требуется собрать груз из нескольких местоположений в единое место. Пусть количество груза, выраженное в ходках манипулятора, находящееся в  $i$ -й точке с полярными координатами  $A_i = (a_r^i, a_\phi^i)$  при центре координат на оси стрелы, равно  $w_i$ . Задача состоит в поиске координат для складирования общего количества груза

$X = (x_r, x)$ , такого, чтобы суммарная стоимость операций манипулятора с грузом была минимальной. При условии, если манипулятор способен вращать стрелу только в случае, если стрела находится в положении минимальной длины (полностью сокращена), данная задача может быть решена предлагаемым алгоритмом. В отсутствие данного условия задача может быть решена алгоритмами, предложенными в работах [490] и [296], результаты которых изложены в следующих параграфах.

Таким образом, предложен алгоритм для решения круга задач размещения с системами транспортировки, использующими особые виды погрузочного оборудования в качестве средства транспортировки грузов. Эффективность алгоритма доказана аналитически.

### 5.8 Алгоритм для простейшей множественной задачи размещения

В настоящем параграфе дается постановка простейшей из возможных множественных задач Вебера, в которой имеются лишь две точки — «потребителя». Показано, что в данной постановке задача в евклидовой метрике эквивалентна задаче в метрике  $l_1$  (прямоугольной). Исследуются ее свойства, позволяющие получить ее частичное решение неитеративным путем. Предложен алгоритм частичного решения асимптотической сложности  $O(n \cdot n')$ , где  $n'$  — число переменных, значение которых определено алгоритмом. Результаты опубликованы в [491,492].

Пусть в задаче (5.37) всего две точки-«потребителя»,  $A_1=(a_{1,1}, a_{2,1})$  и  $A_2=(a_{1,2}, a_{2,2})$ ,  $A_1 \neq A_2$ . Выберем такую систему декартовых координат, в которой  $A_1=(0,0)$ ,  $a_{1,2}=0$ ,  $a_{2,2}>0$ . Для простоты дальнейшего изложения введем обозначение  $a=a_{2,2}$ .

Для множественной задачи Вебера с метриками, основанными на  $l_p$ -нормах, предусмотрены алгоритмы, основанные на процедуре Вайсфелда, сложность которых экспоненциально растет с ростом  $n$ . Задачи Вебера (в том числе множественные), в которых все точки  $A_i$  лежат на одной прямой, вырождаются в задачу поиска взвешенной медианы [493], и решение задачи, соответственно, лежит на этой же прямой. Кроме того, решения множественной задачи Вебера лежат внутри выпуклого многоугольника, для которого справедливо, что все  $A_i$  лежат в его пределах [494]. В случае  $m=2$  многоугольник вырождается в отрезок  $A_1A_2$ . Таким образом, в пределах отрезка  $A_1A_2$  расстояния в евклидовой метрике совпадают с расстояниями в метрике  $l_1$ , и, введя

обозначения  $x_j = x_{1,j} \forall j = \overline{1, n}$ , нашу задачу возможно записать следующим образом:

$$\begin{aligned} \arg \min_{x_1, \dots, x_n \in [0, a]} f(x_1, \dots, x_n) &= \arg \min_{x_1, \dots, x_n \in [0, a]} \sum_{j=1}^n w_{1,j} |x_j - 0| + \sum_{j=1}^n w_{2,j} |x_j - a| + \sum_{j=1}^{n-1} \sum_{k=j+1}^n v_{j,k} |x_j - x_k| \\ &= \arg \min_{x_1, \dots, x_n \in [0, a]} \sum_{j=1}^n (w_{1,j} - w_{2,j}) x_j + \sum_{j=1}^n w_{2,j} a + \sum_{j=1}^{n-1} \sum_{k=j+1}^n v_{j,k} |x_j - x_k|. \end{aligned} \quad (5.37)$$

Для данной задачи разработан эффективный алгоритм (см. Алгоритм 5.5), гарантирующий получение точного решения [243] асимптотической сложности  $O(m n^3)$ . Многие задачи размещения с неевклидовыми метриками, рассмотренные в настоящей главе, сводятся к данной задаче [158, 445, 490, 296]. Для задачи Вебера предложен метод неитеративного решения [486], позволяющий в определенных случаях избежать выполнения процедуры Вайсфельда или аналогичной процедуры для в случае нормы  $l_1$  и получить точное решение за линейное время.

Целевая функция  $f(\cdot)$  данной задачи — кусочно-линейная. Следовательно, если значения  $x_1^*, \dots, x_n^*$  — ее решения, то каждое из них либо совпадает с одной из точек 0 либо  $a$ , либо оптимальными являются все значения из диапазона  $[0; a]$ , поэтому без потери общности можно сказать, что существует решение данной задачи  $x_1^*, \dots, x_n^*$ , такое, что  $x_j^* \in \{0, a\} \forall j = \overline{1, n}$ . Заменяв неотрицательные весовые коэффициенты  $w_{1,j}$  и  $w_{2,j}$  вещественными коэффициентами  $w'_j = w_{1,j} - w_{2,j}$ , имеем эквивалентную целевую функцию:

$$f'(x_1, \dots, x_n) = \sum_{j=1}^n w'_j x_j + \sum_{j=1}^{n-1} \sum_{k=j+1}^n v_{j,k} |x_j - x_k|. \quad (5.38)$$

При  $a=1$  мы имеем задачу псевдоболевой оптимизации:

$$\arg \min_{x_1, \dots, x_n \in \{0, 1\}} \sum_{j=1}^n w'_j x_j + \sum_{j=1}^{n-1} \sum_{k=j+1}^n v_{j,k} (x_j \oplus x_k),$$

где  $\oplus$  — обозначение логической операции "исключающее или".

Путем ввода дополнительных вещественных переменных  $t_{j,k}$ ,  $j = \overline{1, (n-1)}$ ,  $k = \overline{(j+1), n}$ , задача (5.38) преобразуется к стандартной форме задачи линейного программирования [243]:

$$\arg \min_{x_1, \dots, x_n \in [0, a]} \sum_{j=1}^n w'_j x_j + \sum_{j=1}^{n-1} \sum_{k=j+1}^n v_{j,k} t_{j,k}, \quad (5.39)$$

$$-t_{j,k} \leq x_j - x_k \leq t_{j,k} \quad \forall j=\overline{1,(n-1)}, \quad k=\overline{(j+1),n}, \quad (5.40)$$

$$x_j \geq 0 \quad \forall j=\overline{1,n}, \quad (5.41)$$

$$-t_{j,k} \geq 0 \quad \forall j=\overline{1,(n-1)}, \quad k=\overline{(j+1),n}. \quad (5.42)$$

Известные методы решения [243], имеют вычислительную сложность  $O(mn^3)$ .

**Лемма 5.15.** Если в задаче (5.38) для произвольно выбранного индекса  $j' \in \{\overline{1,n}\}$  справедливо неравенство

$$w'_{j'} - \sum_{j=1}^{j'-1} |v_{j,j'}| - \sum_{j=j'+1}^n |v_{j',j}| > 0, \quad (5.43)$$

то минимум функции  $f(x_1, \dots, x_j, \dots, x_n)$  достигается при  $x_{j'}=0$ .

Если же справедливо неравенство

$$w'_{j'} + \sum_{j=1}^{j'-1} |v_{j,j'}| - \sum_{j=j'+1}^n |v_{j',j}| < 0, \quad (5.44)$$

то минимум функции  $f(x_1, \dots, x_j, \dots, x_n)$  достигается при  $x_{j'}=a$ .

*Доказательство.* Пусть при значениях  $x_j=x_j^*, j=\overline{1,n}$  достигается минимум функции (5.38), причем значения  $x_j^*, j \in \{\overline{1,n}\} \setminus \{j'\}$  известны. Обозначим множество  $Q_{j'} = \{\overline{1,n}\} \setminus \{j'\}$ . Тогда

$$\begin{aligned} f'(x_1^*, \dots, x_{j'}^*, \dots, x_n^*) = & \sum_{j \in Q_{j'}} w'_{j'} x_j^* + w'_{j'} x_{j'} + \sum_{j \in Q_{j'} \cap \{\overline{1,(n-1)}\}} \sum_{k \in \{\overline{(j+1),n}\} \setminus \{j'\}} v_{j,k} |x_j^* - x_k^*| + \\ & + \sum_{j=1}^{j'-1} v_{j,j'} |x_j^* - x_{j'}| + \sum_{j=j'+1}^n v_{j',j} |x_j^* - x_{j'}| \end{aligned}$$

Заметим, что значения первого и последнего слагаемых при фиксированных  $x_j^*, j \in \{\overline{1,n}\} \setminus \{j'\}$  являются константами. Исключив их, получим новую целевую функцию:

$$f_{j'}^*(x_{j'}) = w'_{j'} x_{j'} + \sum_{j=1}^{j'-1} v_{j,j'} |x_j^* - x_{j'}| + \sum_{j=j'+1}^n v_{j',j} |x_j^* - x_{j'}|.$$

Поскольку  $x_j^* \in [0, a] \quad \forall j=\overline{1,n}$ , оптимальное значение  $x_{j'}$  также лежит в интервале  $[0; a]$ . Таким образом,  $|x_j^* - x_{j'}| \in [0, a] \quad \forall j \in Q_{j'}$ . Поскольку  $a > 0$ , очевидно, что  $|v_{j,j'} x_j^* - x_{j'}| \in [0, a/v_{j,j'}]$ ,  $|v_{j',j} x_j^* - x_{j'}| \in [0, a/v_{j',j}] \quad \forall j \in Q_{j'}$ . Следовательно,

$$\left| \sum_{j=1}^{j'-1} v_{j,j'} |x_j^* - x_{j'}| + \sum_{j=j'+1}^n v_{j',j} |x_j^* - x_{j'}| \right| \in \left[ 0, a \sum_{j=1}^{j'-1} |v_{j,j'}| + a \sum_{j=j'+1}^n |v_{j',j}| \right].$$

Введем обозначения

$$V_j^{max} = \sum_{j=1}^{j'-1} |v_{j,j'}| + \sum_{j=j'+1}^n |v_{j',j}|, \quad V_j(x_j) = \sum_{j=1}^{j'-1} v_{j,j'} |x_j^* - x_j| + \sum_{j=j'+1}^n v_{j',j} |x_j^* - x_j|.$$

Определим значения целевой функции  $f_j^*$  при  $x_j=0$  и  $x_j=a$ .

$$f_j^*(0) = aV_j(0), \quad f_j^*(a) = w'_{j'}a + V_j(a).$$

Очевидно, что  $V_j(x_j) \in [a, V_j^{max}] \forall x_j \in [0, a]$ . Следовательно,  $f_j^*(0) \in [-aV_j^{max}, aV_j^{max}]$ ,  $f_j^*(a) \in [-aV_j^{max}, aV_j^{max}]$ .

Отсюда следует, что если

$$w'_{j'}a > aV_j^{max}, \quad (5.45)$$

то  $f_j^*(a) > f_j^*(0)$ , т.е. минимум функции  $f_j^*(x_j)$  и эквивалентной ей исходной функции  $f(\cdot)$  достигается при  $x_j=0$  независимо от значений  $x_j^*, j \in Q_j$ .

Аналогично, если

$$w'_{j'}a < aV_j^{max}, \quad (5.46)$$

то  $f_j^*(a) < f_j^*(0)$ .

Сократив множитель  $a$  в выражениях (5.45) и (5.46), получим неравенства (5.43) и (5.44). Лемма доказана.

Согласно условиям Леммы 5.15, оптимальные значения некоторых переменных  $x_j$  могут быть определены путем подсчета частичных сумм коэффициентов  $v_{j,k}$ , без запуска алгоритма решения задачи линейного программирования. В этом случае может быть снижена размерность (число переменных) в решаемой задаче линейного программирования. Более того, после исключения некоторого числа переменных из исходной задачи (5.38) и исключения констант из целевой функции, процесс проверки условий (5.43) и (5.44) может быть продолжен.

Пусть путем проверки условий (5.43) и (5.44) установлены значения некоторых переменных  $x_j$ ,  $j \in Q^*$ ,  $Q^* = (Q_0 \cup Q_a)$ , где

$$Q_0 = \left\{ j' \in \{1, n\} \left| w'_{j'} - \sum_{j=1}^{j'-1} |v_{j,j'}| - \sum_{j=j'+1}^n |v_{j',j}| > 0 \right. \right\}, \quad Q_a = \left\{ j' \in \{1, n\} \left| w'_{j'+1} + \sum_{j=1}^{j'-1} |v_{j,j'}| + \sum_{j=j'+1}^n |v_{j',j}| < 0 \right. \right\}.$$

Запишем целевую функцию (5.38) в следующем виде:

$$\begin{aligned}
f'(x_1, \dots, x_n) = & \sum_{j \in \{1, n\} \setminus Q^*} w'_j x_j + \sum_{j \in Q^*} w'_j x_j + \sum_{j \in \{1, (n-1)\} \setminus Q^*} \sum_{k \in \{(j+1), n\} \setminus Q^*} v_{j,k} |x_j - x_k| + \\
& + \sum_{j \in Q^* \setminus \{n\}} \sum_{k \in \{(j+1), n\} \setminus Q^*} v_{j,k} |x_j - x_k| + \sum_{j \in \{1, (n-1)\} \setminus Q^*} \sum_{k \in \{(j+1), n\} \cap Q^*} v_{j,k} |x_j - x_k| + \\
& + \sum_{j \in Q^* \setminus \{n\}} \sum_{k \in \{(j+1), n\} \cap Q^*} v_{j,k} |x_j - x_k|.
\end{aligned}$$

Второе и последнее слагаемые при известных  $x_j : j \in Q^*$  являются константами.

Исключив их, получив эквивалентную целевую функцию:

$$\begin{aligned}
f''(x_1, \dots, x_n) = & \sum_{j \in \{1, n\} \setminus Q^*} w'_j x_j + \sum_{j \in \{1, (n-1)\} \setminus Q^*} \sum_{k \in \{(j+1), n\} \setminus Q^*} v_{j,k} |x_j - x_k| + \\
& + \sum_{j \in (Q_0 \cup Q_1) \setminus \{n\}} \sum_{k \in \{(j+1), n\} \setminus Q^*} v_{j,k} |x_j - x_k| + \sum_{j \in \{1, (n-1)\} \setminus Q^*} \sum_{k \in \{(j+1), n\} \cap (Q_0 \cap Q_1)} v_{j,k} |x_j - x_k| = \\
= & \sum_{j \in \{1, n\} \setminus Q^*} w'_j x_j + \sum_{j \in \{1, (n-1)\} \setminus Q^*} \sum_{k \in \{(j+1), n\} \setminus Q^*} v_{j,k} |x_j - x_k| + \sum_{j \in Q_0 \setminus \{n\}} \sum_{k \in \{(j+1), n\} \setminus Q^*} v_{j,k} |0 - x_k| + \\
& + \sum_{j \in Q_1 \setminus \{n\}} \sum_{k \in \{(j+1), n\} \setminus Q^*} v_{j,k} |a - x_k| + \sum_{j \in \{1, (n-1)\} \setminus Q^*} \sum_{k \in \{(j+1), n\} \cap Q_0} v_{j,k} |0 - x_j| + \\
& + \sum_{j \in \{1, (n-1)\} \setminus Q^*} \sum_{k \in \{(j+1), n\} \cap Q_1} v_{j,k} |x_j - a|.
\end{aligned}$$

Поскольку  $x_j \in [0, a] \forall j \in Q^*$ , имеем

$$\begin{aligned}
f''(x_1, \dots, x_n) = & \sum_{j \in \{1, n\} \setminus Q^*} w'_j x_j + \sum_{j \in \{1, (n-1)\} \setminus Q^*} \sum_{k \in \{(j+1), n\} \setminus Q^*} v_{j,k} |x_j - x_k| + \sum_{j \in Q_0 \setminus \{n\}} \sum_{k \in \{(j+1), n\} \setminus Q^*} v_{j,k} x_k + \\
& + \sum_{j \in Q_1 \setminus \{n\}} \sum_{k \in \{(j+1), n\} \setminus Q^*} v_{j,k} (a - x_k) + \sum_{j \in \{1, (n-1)\} \setminus Q^*} \sum_{k \in \{(j+1), n\} \cap Q_0} v_{j,k} x_j + \sum_{j \in \{1, (n-1)\} \setminus Q^*} \sum_{k \in \{(j+1), n\} \cap Q_1} v_{j,k} (a - x_j) = \\
= & \sum_{j \in \{1, n\} \setminus Q^*} w'_j x_j + \sum_{j \in \{1, (n-1)\} \setminus Q^*} \beta_j \cdot x_j + a \cdot \alpha + \sum_{j \in \{1, (n-1)\} \setminus Q^*} \sum_{k \in \{(j+1), n\} \setminus Q^*} v_{j,k} |x_j - x_k| + \\
& + \sum_{j \in Q_0 \setminus \{n\}} \sum_{k \in \{(j+1), n\} \setminus Q^*} v_{j,k} x_k - \sum_{j \in Q_1 \setminus \{n\}} \sum_{k \in \{(j+1), n\} \setminus Q^*} v_{j,k} x_k,
\end{aligned}$$

где

$$\begin{aligned}
\alpha = & \sum_{j \in Q_1 \setminus \{n\}} \sum_{k \in \{(j+1), n\} \setminus Q^*} v_{j,k} + \sum_{j \in \{1, (n-1)\} \setminus Q^*} \sum_{k \in \{(j+1), n\} \cap Q_1} v_{j,k}, \\
\beta_j = & \sum_{k \in \{(j+1), n\} \cap Q_0} v_{j,k} - \sum_{k \in \{(j+1), n\} \cap Q_1} v_{j,k} \quad \forall j \in \{1, (n-1)\} \setminus Q^*, \quad (5.47)
\end{aligned}$$

Отметим, что  $\alpha$  и  $\beta_j, j \in \{1, (n-1)\} \setminus Q^*$  при фиксированных  $x_j, j \in Q_j$  являются константами. Дополнительно обозначим:

$$\beta_n = 0, \quad (5.48)$$

$$\zeta_j = \begin{cases} 1, & j \in Q_0, \\ -1, & j \in Q_1 \end{cases} \quad \forall j \in Q^*.$$

Тогда

$$\begin{aligned}
f''(x_1, \dots, x_n) = & \sum_{j \in \{1, n\} \setminus Q^*} (w'_j + \beta_j) \cdot x_j + \sum_{j \in \{1, (n-1)\} \setminus Q^*} \sum_{k \in \{(j+1), n\} \setminus Q^*} v_{j,k} |x_j - x_k| + \\
& + \sum_{k \in \{2, n\} \setminus Q^*} \sum_{j \in \{1, (k-1)\} \setminus Q^*} v_{j,k} \cdot \zeta_j \cdot x_k + a \cdot \alpha.
\end{aligned}$$

Обозначив константы

$$\gamma_k = \sum_{j \in \overline{1, (k-1)} \setminus Q^*} v_{j,k} \cdot \zeta_j = \sum_{j \in \overline{1, (k-1)} \cap Q_0} v_{j,k} - \sum_{j \in \overline{1, (k-1)} \cap Q_1} v_{j,k} \quad \forall k \in \overline{2, n} \setminus Q^*, \quad (5.49)$$

$$\gamma_1 = 0, \quad (5.50)$$

$$w''_j = w'_j + \beta_j + \gamma_j \quad \forall j \in \overline{1, n} \setminus Q^*, \quad (5.51)$$

удалив константу  $a \cdot \alpha$ , имеем эквивалентную целевую функцию:

$$f'''(x_1, \dots, x_n) = \sum_{j \in \overline{1, n} \setminus Q^*} w''_j x_j + \sum_{j \in \overline{1, (n-1)} \setminus Q^*} \sum_{k \in \overline{(j+1), n} \setminus Q^*} v_{j,k} |x_j - x_k|.$$

В результате мы имеем задачу, полностью аналогичную исходной задаче (5.38) с эквивалентной целевой функцией, из которой исключены переменные, определенные согласно условиям Леммы 5.15.

Таким образом, неитеративный метод состоит в проверке условий Леммы 5.15 для каждой из переменных, фиксации ее значения при выполнении условий и пересчете коэффициентов согласно (5.51). Полученная задача вновь может быть подвергнута проверке на соответствие условиям Леммы 5.15 и дальнейшему частичному решению – таким образом неитеративный метод включается в итеративный процесс, который каждый раз снижает вычислительную сложность оставшейся части задачи.

Основываясь на Лемме 5.15 и эквивалентности функций (5.37) и (5.51), предлагается следующий алгоритм.

#### Алгоритм 5.7. Решение задачи (5.37).

Дано: количество переменных  $n$ , вещественные коэффициенты  $v_{j,k}$  и  $w_{1,i}$ ,  $w_{2,i}$ .

1. Присвоить  $w'_j = w_{1,j} - w_{2,j} \quad \forall j = \overline{1, n}; Q = \overline{1, n}$ .
2. Присвоить  $Q_0 = \emptyset, Q_1 = \emptyset$ .
3. Для каждого  $j' \in Q$  цикл:
4. Если справедливо (5.43), то присвоить  $x'_{j'} = 0, Q_0 = Q_0 \cup \{j'\}$ , перейти к 6.
5. Если справедливо (5.44), то присвоить  $x'_{j'} = 1, Q_1 = Q_1 \cup \{j'\}$ .
6. Следующая итерация цикла 3.
7. Если  $Q_0 = \emptyset$  и  $Q_1 = \emptyset$ , то решить задачу (5.39)–(5.42) для оставшихся переменных, **ОСТАНОВ**.
8. Для каждого  $j \in (Q_0 \cup Q_1)$  рассчитать  $\beta_j, \gamma_j$  согласно (5.47)–(5.50), присвоить

$$w'_j = w'_j + \beta_j + \gamma_j.$$

9. Перейти к 2.

Оценим вычислительную сложность алгоритма. Основные шаги (шаг 4 и 5) сводятся к сравнению суммы  $n''$  коэффициентов, где  $n''$  — мощность множества оставшихся переменных  $Q$ ,  $n'' \leq n$ . Данные шаги выполняются  $n'$  раз ( $n'$  — общее число переменных, значение которых определено данным алгоритмом) не более  $n$  раз, так как каждый раз исключается хотя бы одна переменная. Следовательно, вычислительная сложность лежит в пределах  $O(n n')$ . При этом алгоритм не расходует память на ввод дополнительных переменных. Отметим, что вычислительная сложность известного алгоритма [243] лежит в пределах  $O(m n^3)$ , что в случае фиксированного  $m$  эквивалентно  $O(n^3)$ .

Множественная задача Вебера с двумя имеющимися объектами может быть сформулирована в виде задачи псевдодобулевой оптимизации. Задача может быть частично решена предлагаемым алгоритмом, благодаря чему снижается число переменных, для определения которых требуется решение задачи линейного программирования.

### 5.9 Задачи размещения с метриками, основанными на угловых расстояниях

Рассмотрим пример на рис.5.8 [495]. Подъемный кран или иной манипулятор на неподвижной платформе 1 с вращающейся стрелой 2 с подвижным подъемным механизмом 3, перемещающимся вдоль стрелы, служит для перемещения грузов. Система имеет 3 степени свободы: высота крюка  $h$ , угол поворота стрелы  $\varphi$  и положение подъемного механизма (радиус) на стреле  $r$ .

Для описания положения и движения крюка или иного исполнительного элемента нашего крана или манипулятора удобно использовать полярные или цилиндрические координаты [496]. Начало координат должно совпадать с осью вращения стрелы.

Мы будем использовать следующие обозначения.  $Y = (y_r, y_\varphi, y_h)$  в нашей координатной системе описывается тремя координатами: высота  $y_h$ , радиус  $y_r$ , угол  $y_\varphi$ . Радиус  $y_r$  — евклидово расстояние между точкой  $Y$  и осью вращения, высота  $y_h$  — расстояние между  $Y$  и некоторой плоскостью, перпендикулярной оси вращения. Для преобразования декартовых координат точки  $Y = (y_1, y_2)$  вна плоскости в полярные



координаты будем использовать следующие выражение:

$$y_r = \sqrt{(y_1)^2 + (y_2)^2}$$

$$y_\varphi = \begin{cases} 0, & y_1 = 0, y_2 = 0 \\ \frac{\pi}{2}, & y_1 = 0, y_2 > 0 \\ -\frac{\pi}{2}, & y_1 = 0, y_2 < 0 \\ \arctan\left(\frac{y_2}{y_1}\right), & y_1 \neq 0. \end{cases}$$

В 3-мерном пространстве для  $Y = (y_1, y_2, y_3)$  вводится дополнительная координата .

Если механизм перемещает груз из точки  $A = (a_r, a_\varphi, a_h)$  в точку  $B = (b_r, b_\varphi, b_h)$ , затрачивается некоторая энергия на то, чтобы изменить высоту  $\Delta h = |a_h - b_h|$ , угол  $\Delta\varphi = \delta_\varphi(a_\varphi, b_\varphi) = \min_{k \in \mathbb{Z}} \{|a_\varphi - b_\varphi - 2k\pi|\}$  и положение (радиус) подъемного механизма  $\Delta r = |a_r - b_r|$ .

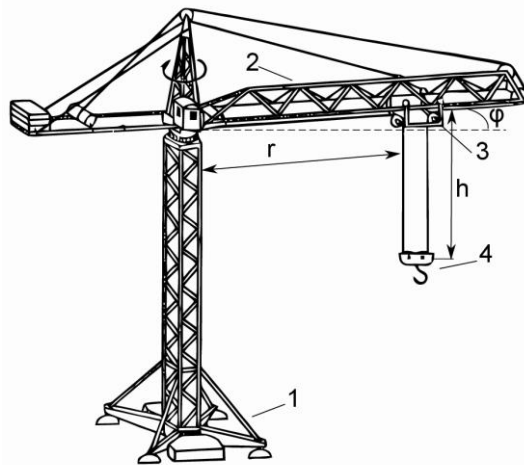


Рисунок 5.8 – Схема подъемного механизма

В некоторой области, доступной манипулятору ( $h < H, r < R$ , где  $H$ - высота крана,  $R$ - эффективная длина стрелы) имеется множество из  $N$  точек требования (потребителей)  $A_i, i = 1, \dots, N$ . Из произвольно выбранной  $i$ -й точки  $A_i = (a_r^i, a_\varphi^i, a_h^i)$  требуется доставить  $w_i$  единиц груза (например, поддонов со строительными материалами). Задача размещения состоит в том, чтобы найти точку  $X = (x_r, x_\varphi, x_h)$ , такую, чтобы стоимость размещения всех грузов из всех точек требования достигала минимума. В общем случае, задача заключается в нахождении подходящей точки

$X$  путем решения задачи минимизации

$$\operatorname{argmin}_{X \in \mathbb{R}^n} F(X) = \sum_{i=1}^N w_i L(X, A_i). \quad (5.52)$$

здесь,  $w_i, i = \overline{1, N}$  – неотрицательные весовые коэффициенты,  $L(X, A_i)$  – функция расстояния, определяющая стоимость передвижения единицы груза из точки  $X$  в точку  $A_i$ . В классической задаче Вебера подразумевается евклидово расстояние  $L(X, A_i) = \|X - A_i\|_2$ . В тоже время, затраты механизма (энергия, время и т.д.) не пропорциональны евклидову расстоянию. В зависимости от способа определения этих затрат могут быть предложены как минимум 4 стратегии управления манипулятором и 4 соответствующих постановки задачи размещения.

*Стратегия 1.* Минимизировать стоимость движения механизмов. Если стоимость вертикального передвижения крюка на 1 м составляет  $C_h$ , стоимость вращения стрелы на 1 радиан составляет  $C_\varphi$ , и стоимость передвижения подъемного механизма вдоль стрелы (изменение радиуса) на 1 м составляет  $C_r$ , то условное расстояние между  $X$  и  $A_i$ .

$$L(X, A_i) = C_r |x_r - a_r^i| + C_\varphi \delta_\varphi(x_\varphi, a_\varphi^i) + C_h |x_h - a_h^i| \quad \forall i = \overline{1, N},$$

и целевая функция задачи (5.52) принимает вид

$$F(X) = \sum_{i=1}^N w_i \left( C_r |x_r - a_r^i| + C_\varphi \min_{K \in \mathbb{Z}} \{ |a_\varphi - b_\varphi - 2K\pi| \} + C_h |x_h - a_h^i| \right). \quad (5.53)$$

Тогда условное расстояние (фактически – стоимость задействования механизмов) назовем метрикой подъемного крана. Задача Вебера, соответствующая целевой функции (5.53), описана в [296].

*Стратегия 2.* Минимизировать длину пути, проделываемого грузом, при условии, что одновременно может производиться только один тип движения (вертикальное передвижение, вращение стрелы, изменение радиуса подъемного механизма). С данным предположением мы имеем смешанную норму (подобные смешанные нормы рассмотрены в [497]).

$$L(X, A_i) = |x_h - a_h^i| + \begin{cases} x_r + a_r^i, \delta_\varphi(a_\varphi^i, x_\varphi) \geq 2 \\ \min\{x_r; a_r^i\} \delta_\varphi(a_\varphi^i, x_\varphi) + |a_r^i - x_r|, \delta_\varphi(a_\varphi^i, x_\varphi) < 2 \end{cases} \\ \forall i = \overline{1, N}.$$

Вертикальная компонента пути  $|x_h - a_h^i|$  не зависит от других его компонент. Длина горизонтальной компоненты соответствует расстоянию в метрике Москвы (иначе называемой метрикой Карлсруэ) [88]. В городах, подобных Москве, движение транспорта возможно либо вдоль «лучей» от центра, либо «колец» вокруг центра. Из

точки  $A_i$  в точку  $X$  можно попасть тремя путями:

1. Двигаясь вдоль «кольцевой» улицы из  $A_i$ , а затем двигаясь вдоль «луча» до  $X$  (Случай 2 на рис.5.9). Данный путь является кратчайшим, если  $\varphi = \delta_\varphi(a_\varphi^i, x_\varphi) \leq 2$  и  $x_r \geq a_r^i$ .

2. Двигаясь вдоль «луча» из  $A_i$ , а затем вдоль «кольца» до  $X$  (Случай В на рис. 5.9). Данный путь оптимален, если  $\varphi = \delta_\varphi(a_\varphi^i, x_\varphi) \leq 2$ , и  $x_r \leq a_r^i$ .

3. Двигаясь из  $A_i$  вдоль «луча» до центра (начала координат), а затем вдоль другого «луча» до  $X$ . Данный путь оптимален, если  $\varphi = \delta_\varphi(a_\varphi^i, x_\varphi) \geq 2$  (см. случай С на рис. 5.9) [445]. В последнем случае расстояние совпадает с метрикой французской столицы (French metro).

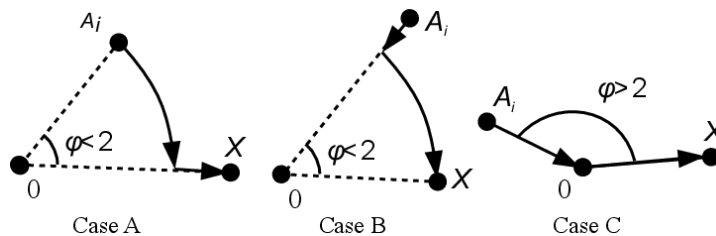


Рисунок 5.9. Различные варианты пути при управлении по Стратегии 2.

*Стратегия 3.* Минимизировать длину пути крюка при условии, что вращение стрелы возможно только в определенном положении подъемного механизма, когда радиус подъемного механизма равен 0. Если точка-потребитель не доступна из текущей точки без вращения механизма, то вначале следует сократить эффективный радиус стрелы, затем стрела вращается и подъемный механизм вновь перемещается на требуемое расстояние вдоль стрелы. Данное предположение актуально для некоторых манипуляторов с телескопической стрелой. Для данного типа манипуляторов вертикальное движение как правило не возможно, либо возможно путем вращения стрелы в вертикальной плоскости (изменение азимута). Для простоты изложения приводим формулу для 2-мерного случая:

$$L(X, A_i) = \begin{cases} x_r + a_r^i, & a_\varphi^i \neq x_\varphi \\ |x_r - a_r^i|, & a_\varphi^i = x_\varphi \end{cases} \forall i = \overline{1, N}.$$

*Стратегия 4.* Эта стратегия похожа на предыдущую, за исключением одного важного условия. Груз проходит через начало координат в любом случае, при этом не имеет значения, требуется ли вращение стрелы. Расстояние между  $X$  и  $A_i$  для этого

случая:

$$L(X, A_i) = \begin{cases} x_r + a_r^i, & X \neq A_i \\ 0, & X = A_i \end{cases} \forall i = \overline{1, N}. \quad (5.54)$$

В случае применения стратегии 4 расстояние, преодолеваемое исполнительным элементом манипулятора, может быть описано метрикой британской железной дороги (British Rail metric), другое название – метрика цветочного киоска [88], при этом точка вращения рассматривается как начало координат. Функция расстояния в Стратегии 3 соответствует метрике French metro metric в полярных координатах. Алгоритм для этого случая изложен в [296, 445]. Дальнейшее развитие этого метода описано в [444].

Алгоритмы для метрики подъемного крана (Стратегия 1), метрики Москвы (Стратегия 2) и метрики британской железной дороги (Стратегии 4) описаны в следующих трех параграфах. Представленные алгоритмы расширяют арсенал методов теории размещения, предлагая модели, являющиеся более реалистичными для систем, в которых средством транспортировки является манипулятор с вращающейся стрелой.

Все алгоритмы завершают свою работу за полиномиальное время. Их эффективность доказана аналитически и продемонстрирована примером.

### 5.10 Алгоритм для задачи Вебера с метрикой подъемного крана

Расстояние между точками  $A_i = (a_1^i, a_2^i, a_3^i)$  и  $X = (x_1, x_2, x_3)$  в прямоугольной метрике определяется формулой:

$$L(A_i, X) = l_1(A_i, X) = |x_1 - a_1^i| + |x_2 - a_2^i| + |x_3 - a_3^i|. \quad (5.55)$$

Задача оптимизации (5.52) с функцией расстояния (5.55) разбивается на три независимые задачи оптимизации с целевыми функциями

$$f_1(x_1) = \sum_{i=1}^N w_i |x_1 - a_1^i|, f_2(x_2) = \sum_{i=1}^N w_i |x_2 - a_2^i|, f_3(x_3) = \sum_{i=1}^N w_i |x_3 - a_3^i|.$$

Эти задачи могут быть сведены к задаче более общего вида

$$f(x) = \sum_{i=1}^N w_i |x - a^i|. \quad (5.56)$$

Задача (5.56) может быть решена известными алгоритмами (см., например, [114, 498]).

В случае Стратегии 1 (см. предыдущий параграф), целевая функция (5.53) сумму трех независимых функций:

$$f_r(x_r) = \sum_{i=1}^N w_i |a_r^i - x_r|, \tag{5.57}$$

$$f_\varphi(x_\varphi) = \sum_{i=1}^N w_i \delta_\varphi(a_\varphi^i, x_\varphi) = \sum_{i=1}^N w_i \min_{K \in \mathbb{Z}} |x_\varphi - a_\varphi^i + 2K\pi|, \tag{5.58}$$

$$f_h(x_h) = \sum_{i=1}^N w_i |a_h^i - x_h|. \tag{5.59}$$

Решением задачи (5.53) является точка  $X^* = (x_r^*, x_\varphi^*, x_h^*)$ , координаты которой  $x_r^*$ ,  $x_\varphi^*$  и  $x_h^*$  являются решениями (точками минимума) функций (5.57), (5.58) и (5.59), соответственно. Очевидно, что задачи (5.57) и (5.59) являются частными случаями задачи (5.56) и могут быть решены соответствующими методами [114, 498].

Лемма 5.16. Пусть  $\mathcal{X}$  - множество точек минимума целевой функции (5.58), и  $\mathcal{A}_\varphi = (a_\varphi^1, \dots, a_\varphi^N)$  - множество угловых координат известных точек (точек-потребителей). Тогда существует  $a_\varphi^i \in \mathcal{A}_\varphi$ , такая что  $a_\varphi^i \in \mathcal{X}$ , или  $(a_\varphi^i + \pi) \in \mathcal{X}$ .

*Доказательство.* Пусть  $x_\varphi^* \in \mathcal{X}$  - точка минимума функции (5.58), и

$$x_\varphi^* \neq a_\varphi^i + K\pi \quad \forall i = \overline{1, N}, \quad K \in \{0, 1\}.$$

Перейдем к полярной системе координат, такой, что  $x_\varphi^* = 0$  (рис. 5.10), при этом будем считать, что значения всех углов таковы, что  $-\pi < a_\varphi^i \leq \pi$  (случай  $a_\varphi^i = 0$  исключается, поскольку  $x_\varphi^* \neq a_\varphi^i$ ). Такое преобразование возможно, поскольку значения  $x_\varphi^* + K\pi$  можно вычесть из значений всех углов  $a_\varphi^i$ , при том что  $x_\varphi^*$ , независимо от выбора  $x_\varphi^*$ . Значения функции  $\delta_\varphi(a_\varphi^i, x_\varphi) = \min_{K \in \mathbb{Z}} |x_\varphi - a_\varphi^i - 2K\pi|$  в новой координатной системе остаются неизменными. Таким образом, и значения целевой функции (5.58) неизменны.

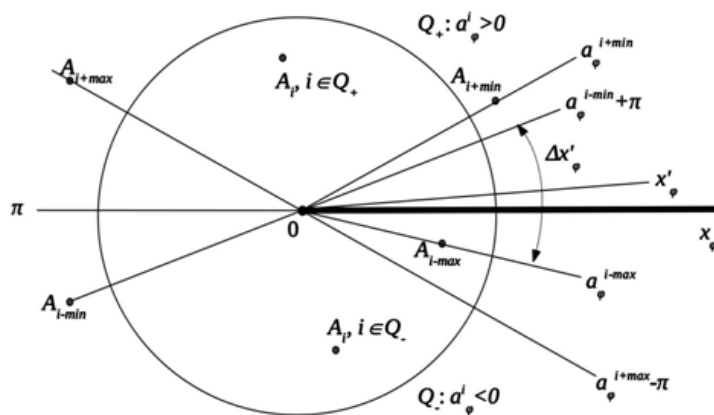


Рис 5.10 - Иллюстрация к Лемме 5.16

Разбиваем множество индексов координат  $\{a_\varphi^i\}$  на два подмножества:

$$Q_+ = \{i \mid a_\varphi^i > 0\}, \quad Q_- = \{i \mid a_\varphi^i < 0\}.$$

Тогда

$$L(x_\varphi^*, A_i) = |x_\varphi^* - a_\varphi^i - 2K\pi| = a_\varphi^i \quad \forall i \in Q_+,$$

$$L(x_\varphi^*, A_i) = |x_\varphi^* - a_\varphi^i - 2K\pi| = -a_\varphi^i \quad \forall i \in Q_-.$$

Если  $Q_+ = \emptyset$ , то  $f_\varphi(x_\varphi^*) = -\sum_{i=1}^N w_i a_\varphi^i$ .

Из  $a_\varphi^i < 0 \quad \forall i \in Q_-$  и  $\delta_\varphi(a_\varphi^i, a_\varphi^{i'}) \leq \pi \quad \forall i, i' \in Q_-$ , следует, что

$$L(A_1, A_i) = \min_{K \in \mathbb{Z}} \{|a_\varphi^1 - a_\varphi^i + 2K\pi|\} = |a_\varphi^i - a_\varphi^1| \quad \forall i = \overline{1, N}.$$

Поскольку  $a_\varphi^i \neq 0 \quad \forall i = \overline{1, N}$ , из этого следует  $|a_\varphi^i - a_\varphi^1| < |a_\varphi^i - 0| \quad \forall i = \overline{1, N}$ , т.е.  $\delta_\varphi(a_\varphi^1, a_\varphi^i) < \delta_\varphi(x_\varphi, a_\varphi^i)$ . Далее, поскольку  $w_i \geq 0 \quad \forall i = \overline{1, N}$ , если  $\exists i: w_i > 0$ , то  $f_\varphi(a_\varphi^1) < f_\varphi(x_\varphi)$ .

Если  $w_i = 0 \quad \forall i \in \overline{1, N}$ , то  $f_\varphi(x_\varphi) = 0 \quad \forall x_\varphi \in \mathbb{R}$ , и  $a_\varphi^i \in \mathcal{X} \quad \forall i = \overline{1, N}$ .

Таким образом, если  $Q_+ = \emptyset$ , то  $x_\varphi^*$  не является точкой минимума функции (5.58), если только все значения  $a_\varphi^i$  не являются точками минимума.

Аналогичным образом можно доказать, что, если  $Q_- = \emptyset$ , то  $x_\varphi^*$  не является точкой минимума функции (5.58), если только все значения  $a_\varphi^i$  не являются точками минимума.

Таким образом, если  $x_\varphi^*$  - точка минимума целевой функции (5.58), и все значения  $a_\varphi^i$  не являются ее точками минимума, то оба подмножества  $Q_-$  и  $Q_+$  являются не пустыми.

Выберем четыре индекса, по два для каждого из подмножеств  $Q_-$  и  $Q_+$  (выбранные индексы могут совпадать, если подмножество содержит единственный индекс точки):

$$i^{+min} \in Q_+: a^{i^{+min}} \leq a_\varphi^i \quad \forall i \in Q_+, \quad i^{+max} \in Q_+: a^{i^{+max}} \geq a_\varphi^i \quad \forall i \in Q_+,$$

$$i^{-min} \in Q_-: a^{i^{-min}} \leq a_\varphi^i \quad \forall i \in Q_-, \quad i^{-max} \in Q_-: a^{i^{-max}} \geq a_\varphi^i \quad \forall i \in Q_-.$$

Тогда можно выбрать произвольное значение  $x'_\varphi$ , такое, что

$$\max(a_\varphi^{i^{+max}} - \pi, a_\varphi^{i^{-min}}) < x'_\varphi < \min(a_\varphi^{i^{+min}}, \pi + a_\varphi^{i^{-max}}).$$

В этом случае  $L(x'_\varphi, A_i) = a_\varphi^i - x'_\varphi \quad \forall i \in Q_+$ , и  $L(x'_\varphi, A_i) = -a_\varphi^i + x'_\varphi \quad \forall i \in Q_-$ .

Таким образом, для каждого

$$x'_\varphi \in \left[ \max(a_\varphi^{i+\max} - \pi, a_\varphi^{i-\min}), \min(a_\varphi^{i+\min}, \pi + a_\varphi^{i-\max}) \right]$$

значение целевой функции (5.58) равно

$$\begin{aligned} f_\varphi(x'_\varphi) &= \sum_{i \in Q_+} w_i(a_\varphi^i - x'_\varphi) + \sum_{i \in Q_-} w_i(x'_\varphi - a_\varphi^i) = \\ &= \sum_{i \in Q_+} w_i a_\varphi^i - \sum_{i \in Q_-} w_i a_\varphi^i + x'_\varphi \left( \sum_{i \in Q_-} w_i - \sum_{i \in Q_+} w_i \right) = C_0 + \Delta W x'_\varphi. \end{aligned}$$

Здесь

$$\begin{aligned} \Delta W &= \sum_{i \in Q_-} w_i - \sum_{i \in Q_+} w_i; C_0 = \sum_{i \in Q_+} w_i(a_\varphi^i - x'_\varphi) + \sum_{i \in Q_-} w_i(x'_\varphi - a_\varphi^i) = \\ &= \sum_{i \in Q_+} w_i a_\varphi^i. \end{aligned}$$

Из  $x_\varphi^* = 0$  следует, что  $f(x_\varphi^*) = C_0$ .

Если  $\Delta W > 0$ , то для  $x''_\varphi$ , для которого выполняется

$$\max(a_\varphi^{i+\max} - \pi, a_\varphi^{i-\min}) < x''_\varphi < x_\varphi,$$

справедливо следующее:

$$f_\varphi(x''_\varphi) = C_0 + \Delta W x''_\varphi < f_\varphi(x_\varphi^*).$$

Это означает, что  $x_\varphi^*$  не является точкой минимума функции (5.58). Аналогично, если  $\Delta W < 0$ , то для  $x''_\varphi$ , удовлетворяющего

$$x_\varphi < x''_\varphi < \min(a_\varphi^{i+\min}, \pi + a_\varphi^{i-\max}),$$

справедливо следующее выражение:

$$f_\varphi(x''_\varphi) = C_0 + \Delta W x''_\varphi < f_\varphi(x_\varphi^*).$$

Если  $\Delta W < 0$ , то для  $x''_\varphi$ , удовлетворяющего условию

$$x''_\varphi \in \left[ \max(a_\varphi^{i+\max} - \pi, a_\varphi^{i-\min}), \min(a_\varphi^{i+\min}, \pi + a_\varphi^{i-\max}) \right]$$

справедливо  $f_\varphi(x''_\varphi) = C_0 = f_\varphi(x_\varphi^*)$ .

Таким образом, в этом случае угол  $x_\varphi^*$  является точкой минимума, и то же самое справедливо для всех углов от  $\max(a_\varphi^{i+\max} - \pi, a_\varphi^{i-\min})$  до  $\min(a_\varphi^{i+\min}, \pi + a_\varphi^{i-\max})$  включительно.

Значение  $x_\varphi^*$  выбрано произвольно. Таким образом, если  $x_\varphi^* \neq a_\varphi^i + K\pi \quad \forall i = \overline{(1, N)}$ ,  $K \in \mathbb{Z}$ , то  $x_\varphi^*$  не является точкой минимума целевой функции за исключением

случая  $\Delta W = 0$ . Но и в этом случае значения  $\max(a_\varphi^{i+max} - \pi, a_\varphi^{i-min})$  и  $\min(a_\varphi^{i+min}, \pi + a_\varphi^{i-max})$  являются точками минимума. Значения  $a_\varphi^{i+max}$ ,  $a_\varphi^{i-min}$ ,  $a_\varphi^{i+min}$  и  $a_\varphi^{i-max}$  выбраны из одного из трех множеств:  $\{a_\varphi^i \mid i = \overline{1, N}\}$ ,  $\{a_\varphi^i + \pi \mid i = \overline{1, N}\}$  или  $\{a_\varphi^i - \pi \mid i = \overline{1, N}\}$ . Заметим, что углы  $a_\varphi^i + \pi$  и  $a_\varphi^i - \pi$  эквивалентны и дают одинаковые значения целевой функции.

Лемма доказана.

Таким образом, если требуется найти точку минимума  $f_\varphi(x_\varphi)$ , следует искать в двух множествах:  $\{a_\varphi^i \mid i = \overline{1, N}\}$  и  $\{a_\varphi^i + \pi \mid i = \overline{1, N}\}$ . Для решения нашей задачи может быть предложен следующий алгоритм.

#### Алгоритм 5.8 Решение задачи Вебера с "метрикой подъемного крана".

Шаг 1: Решить задачу (5.57), используя известный алгоритм для решения задачи (5.56). Сохранить результат в  $x_r^*$ .

Шаг 2: Таким же образом решить задачу (5.59). Сохранить результат в  $x_h^*$ .

Шаг 3: Если  $f_\varphi(a_\varphi^1 + \pi) < f_\varphi(a_\varphi^1)$ , то  $x_\varphi^* = a_\varphi^1 + \pi$ , иначе  $x_\varphi^* = a_\varphi^1$ .

Шаг 4:  $f = f_\varphi(x_\varphi^*)$ .

Шаг 5: Для каждого  $i: i = \overline{2, N}$  выполнять:

Шаг 5.1: Если  $f_\varphi(a_\varphi^i) < f$ , то  $x_\varphi^* = a_\varphi^i$ ;  $f^* = f_\varphi(a_\varphi^i)$ .

Шаг 5.2: Если  $f_\varphi(a_\varphi^i + \pi) < f$ , то  $x_\varphi^* = a_\varphi^i + \pi$ ;  $f^* = f_\varphi(a_\varphi^i + \pi)$ .

Шаг 5.3: Следующая итерация цикла 5.

Шаг 6: Возвратить результат  $X^* = (x_r^*, x_\varphi^*, x_h^*)$ . СТОП.

Оценим сложность Алгоритма 5.8. Пусть  $N$  - количество известных точек (точек-потребителей). Алгоритм для задачи с метрикой  $l_1$  (см. [114, 498]) включает сортировку координат (поиск медианного значения координаты) с асимптотической сложностью  $O(N \log N)$  и суммирование значений координат линейной сложности. Таким образом, общая вычислительная сложность Шагов 1 и 2 описывается асимптотической формулой  $O(N \log N)$ . На Шагах 5 - 5.3 в цикле выполняется вычисление значений целевой функции для каждой из  $N - 1$  координат  $a_\varphi^i$  и  $a_\varphi^i + \pi$ . Согласно Шагам 3 и 4, значение целевой функции оценивается  $2N$  раз. Целевая функция



линейна (асимптотическая сложность вычисления ее значения равна  $O(N)$ ), поэтому асимптотическая сложность Шагов 3-5.3 и всего алгоритма равна  $O(N^2)$ .

В изложенном далее примере проиллюстрировано решение задачи Вебера с «метрикой подъемного крана». Полярные координаты известных точек (точек-потребителей) и соответствующие весовые коэффициенты даны в Таблице 5.3.

Таблица 5.3 – Исходные данные для задачи с метрикой подъемного крана

$i$	$a_r^i$	$a_h^i$	$a_\varphi^i$	$w_i$
1	10	5	0	3
2	20	3	0	2
3	10	5	$\pi/4$	4
4	20	5	$\pi/4$	3
5	30	3	$\pi/4$	4

Решение.

Шаг 1.  $x_r^* = \operatorname{argmin}_{x_r} \sum_{i=1}^N w_i |a_r^i - x_r| = 20$ .

Шаг 2.  $x_h^* = \operatorname{argmin}_{x_h} \sum_{i=1}^N w_i |a_h^i - x_h| = 5$ .

Шаг 3.  $f_\varphi(a_\varphi^1 + \pi) = f_\varphi(0 + \pi) = \sum_{i=1}^N \delta_\varphi(\pi, a_\varphi^i) = 3,75\pi$ ;

$f_\varphi(a_\varphi^1) = f_\varphi(0) = \sum_{i=1}^N \delta_\varphi(0, a_\varphi^i) = 1,25\pi$ .

Условие  $f_\varphi(a_\varphi^1 + \pi) < f_\varphi(a_\varphi^1)$  не выполняется, и  $x_\varphi^* = a_\varphi^1 = 0$ .

Шаг 4.  $f = f_\varphi(x_\varphi) = 1,25\pi$ .

Шаг 5. Для каждого  $i = 2$  до 5 выполнять (результаты шагов 5.1-5.3 сведены в Табл. 5.4).

Шаг 6.  $X^* = (x_r^*, x_\varphi^*, x_h^*) = (20, 1\frac{1}{4}\pi, 5)$ .

Таблица 5.4 – Ход решения (Шаги 5.1 – 5.3 Алгоритма 5.8)

Шаг, показатель	$i = 2$	$i = 3$	$i = 4$	$i = 5$
Шаг 5.1, значение $f_\varphi(a_\varphi^i)$ .	$1,25\pi$	$\pi$	$1,25\pi$	$1,25\pi$
Шаг 5.1 условие $f_\varphi(a_\varphi^i) < f^*$	ложь	ложь	ложь	ложь
Шаг 5.1, значение $f^*$	$1,25\pi$	$\pi$	$\pi$	$\pi$
Шаг 5.1, значение $x_\varphi^*$	0	$0,25\pi$	$0,25\pi$	$0,25\pi$
Шаг 5.2, значение $f_\varphi(a_\varphi^i + \pi)$	$3,75\pi$	$5\pi$	$6\pi$	$6\pi$
Шаг 5.2, условие $f_\varphi(a_\varphi^i + \pi) < f^*$	ложь	ложь	ложь	ложь
Шаг 5.2, значение $f^*$	$1,25\pi$	$\pi$	$\pi$	$\pi$
Шаг 5.2, значение $x_\varphi^*$	0	$0,25\pi$	$0,25\pi$	$0,25\pi$

Алгоритм 5.8 решения задачи Вебера может использоваться в составе алгоритмов автоматической группировки (см. Алгоритм 3.1 и использующие его в своем составе Алгоритмы 3.4-3.14 и 3.17-3.19). Форма получаемых при этом кластеров продемонстрирована на рис. 5.11.

### 5.11 Алгоритм для задачи Вебера с метрикой Москвы

В случае реализации Стратегии 2 управления манипулятором, целевая функция соответствующей задачи Вебера (5.52) представляет собой сумму

$$F(X) = f_h(x_h) + f_{MK}(x_r, x_\varphi) \quad (5.60)$$

Двух независимых функций

$$f_h(X) = \sum_{i=1}^N w_i |x_h - a_h^i|, \quad (5.61)$$

$$f_{MK}(X) = \sum_{i=1}^N w_i \begin{cases} x_r + a_r^i, & \delta_\varphi(a_\varphi^i, x_\varphi) \geq 2 \\ \min(x_r, a_r^i) \delta_\varphi(a_\varphi^i, x_\varphi) + |a_r^i - x_r|, & \delta_\varphi(a_\varphi^i, x_\varphi) < 2. \end{cases} \quad (5.62)$$

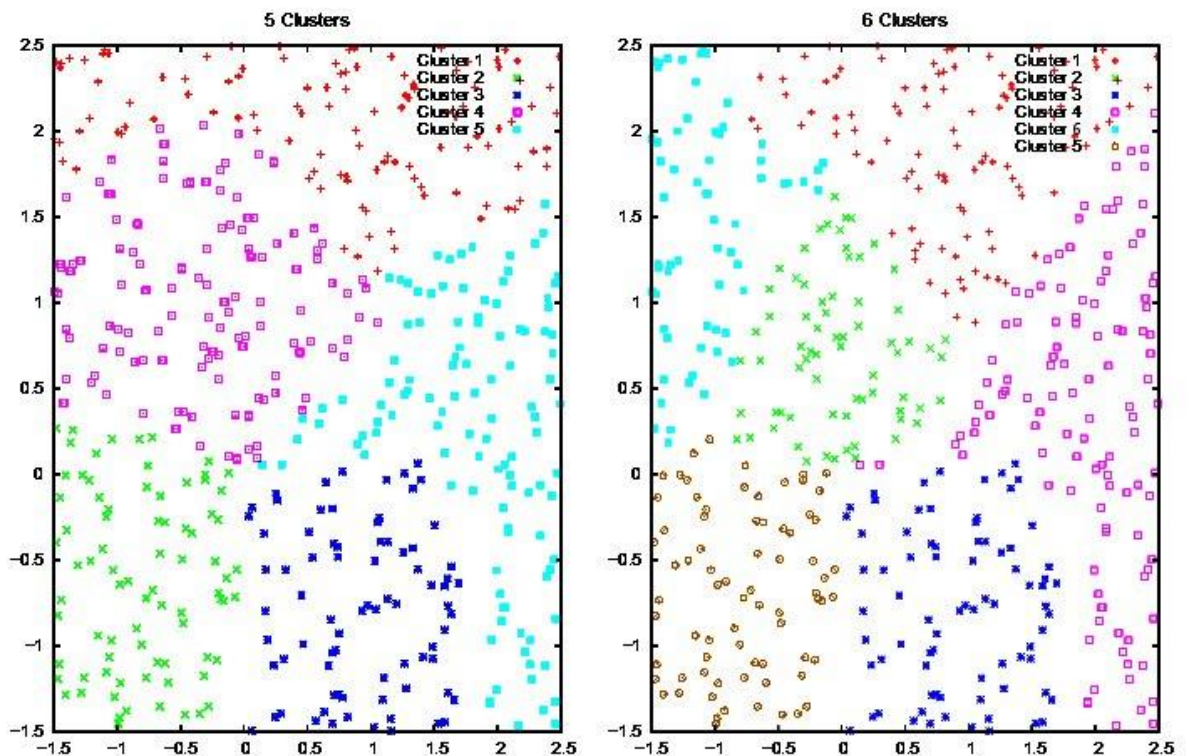


Рисунок 5.11 – Результат группировки объектов на плоскости с использованием метрики подъемного крана (меры, основанной на угловых расстояниях)

Минимум целевой функции (5.60) достигается в точке  $X^* = (x_r^*, x_\varphi^*, x_h^*)$ , где  $x_h^*$  - точка минимума функции (5.61),  $x_\varphi^*$  -  $x_r^*$  точки минимума функции (5.62). Таким

образом, задача (5.60) соответствует обобщенной задаче (5.56), которая может быть решена соответствующим алгоритмом [114, 498, 243].

Лемма 5.17 ограничивает область поиска точек минимума функции (5.62).

Лемма 5.17. Пусть  $\mathcal{X}$  – множество точек минимума целевой функции (5.62), и  $\mathcal{A} = \{A_1, \dots, A_N\}$  – множество известных точек-потребителей в полярных координатах,  $A_i = (a_r^i, a_\varphi^i)$ .

Тогда существует  $X^* = (x_r^*, x_\varphi^*) \in \mathcal{X}$ , и  $A_i = (a_r^i, a_\varphi^i) \in \mathcal{A}$ , что выполняется хотя бы одно из следующих трех условий:

$$(x_r^*, a_\varphi^i) \in X^*, \text{ или}$$

$$(x_r^*, a_\varphi^i + 2) \in X^*, \text{ или}$$

$$(x_r^*, a_\varphi^i - 2) \in X^*.$$

*Доказательство.* Предположим, что точка минимума  $X^* = (x_r^*, x_\varphi^*)$  функции (5.62) удовлетворяет условию  $x_\varphi^* \neq a_\varphi^i \pm 2 \forall i = \overline{1, N}$ .

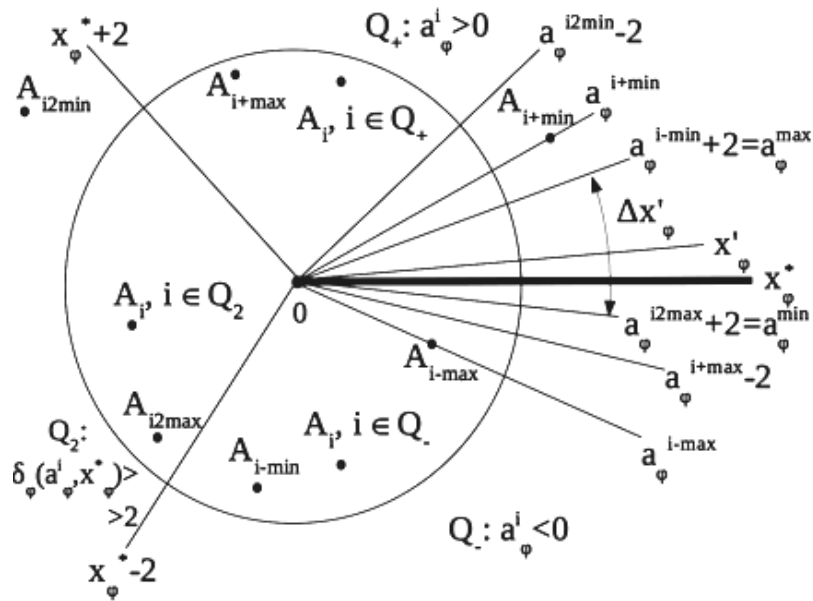


Рис. 5.12 – Иллюстрация к Лемме 5.17

После перехода к полярной системе координат, в которой  $x_\varphi^* = 0$  (Рис. 5.12), значения всех углов  $a_\varphi^i$  можно выразить таким образом, чтобы выполнялось  $-\pi < a_\varphi^i \leq \pi$  (случай  $a_\varphi^i = 0$  исключается, поскольку  $x_\varphi^* \neq a_\varphi^i$ ). Такое преобразование возможно,

поскольку, независимо от выбора  $x_\varphi^*$ , можно вычесть значение  $x_\varphi^*$  из значений всех углов  $a_\varphi^i$  и собственно из угла  $x_\varphi^*$ , получив тем самым значения новых координат. Значения  $\delta_\varphi(a_\varphi^i, x_\varphi) = \min_{K \in \mathbb{Z}} \{x_\varphi - x_\varphi^i - 2\pi K\}$  остаются в новой координатной системе неизменными. Таким образом, значение целевой функции (5.62) не меняется.

Разобьем множество индексов координат  $\{a_\varphi^i \mid i = \overline{1, N}\}$  на три подмножества:

$$Q_+^* = \{i \mid 0 < a_\varphi^i < 2\}, \quad Q_-^* = \{i \mid -2 \leq a_\varphi^i < 0\}, \quad Q_2^* = \{i \mid |a_\varphi^i| > 2\}.$$

Кроме того, рассмотрим три подмножества индексов, определяемых относительно произвольно выбранного угла  $x_\varphi \in (-\pi, \pi]$ :

$$Q_+(x_\varphi) = \{i \mid \exists K \in \mathbb{Z}: 0 < a_\varphi^i - x_\varphi + 2\pi K \leq 2\},$$

$$Q_-(x_\varphi) = \{i \mid \exists K \in \mathbb{Z}: -2 \leq a_\varphi^i - x_\varphi + 2\pi K < 0\},$$

$$Q_2(x_\varphi) = \{i \mid \exists K \in \mathbb{Z}: |a_\varphi^i - x_\varphi + 2\pi K| > 2\}.$$

Тогда

$$\delta_\varphi(x_\varphi^*, a_\varphi^i) = \begin{cases} a_\varphi^i, & \forall i \in Q_+^*, \\ -a_\varphi^i, & \forall i \in Q_-^*, \\ a_\varphi^i - x_\varphi, & \forall i \in Q_+(x_\varphi)^*, \\ -a_\varphi^i + x_\varphi, & \forall i \in Q_-(x_\varphi)^*. \end{cases}$$

Значение целевой функции (5.62) в точке  $x_\varphi^* = 0$  равно

$$\begin{aligned} f_{MK}(x_r^*, x_\varphi^*) &= \sum_{i \notin Q_2^*} w_i |x_r^* - a_r^i| + \sum_{i \in Q_+^*} w_i \min\{x_r^*, a_r^i\} a_\varphi^i - \\ &- \sum_{i \in Q_-^*} w_i \min\{x_r^*, a_r^i\} a_\varphi^i + \sum_{i \in Q_2^*} w_i (x_r^* + a_r^i). \end{aligned}$$

Для произвольного угла  $x_\varphi$  выполняется

$$\begin{aligned} f_{MK}(x_r^*, x_\varphi) &= \sum_{i \notin Q_2(x_\varphi)} w_i |x_r^* - a_r^i| + \sum_{i \in Q_+(x_\varphi)} w_i \min\{x_r^*, a_r^i\} (a_\varphi^i - x_\varphi) + \\ &+ \sum_{i \in Q_-(x_\varphi)} w_i \min\{x_r^*, a_r^i\} (x_\varphi - a_\varphi^i) + \sum_{i \in Q_2(x_\varphi)} w_i (x_r^* + a_r^i) = \\ &= \sum_{i \in Q_2(x_\varphi)} w_i (x_r^* + a_r^i) + \sum_{i \notin Q_2(x_\varphi)} w_i |x_r^* - a_r^i| + \\ &+ \sum_{i \in Q_+(x_\varphi)} w_i \min\{x_r^*, a_r^i\} a_\varphi^i - \sum_{i \in Q_+(x_\varphi)} w_i \min\{x_r^*, a_r^i\} x_\varphi - \end{aligned}$$

$$-\sum_{i \in Q_-(x_\varphi)} w_i \min\{x_r^*, a_r^i\} x_\varphi - \sum_{i \in Q_-(x_\varphi)} w_i \min\{x_r^*, a_r^i\} a_\varphi^i. \quad (5.63)$$

Существует некоторый интервал (Рис. 5.12)  $[x_\varphi^{\min}, x_\varphi^{\max}]$ , такой, что

$$Q_+(x_\varphi) = Q_+^* \wedge Q_-(x_\varphi) = Q_+^* \wedge Q_2(x_\varphi) = Q_2^* \quad \forall x_\varphi \in [x_\varphi^{\min}, x_\varphi^{\max}].$$

Если  $x_\varphi \in [x_\varphi^{\min}, x_\varphi^{\max}]$ , некоторые слагаемые (5.63) являются константами.

Обозначим

$$\begin{aligned} C_0 &= \sum_{i \in Q_2(x_\varphi)} w_i (x_r^* + a_r^i) + \sum_{i \notin Q_2(x_\varphi)} w_i |x_r^* - a_r^i| + \\ &+ \sum_{i \in Q_+(x_\varphi)} w_i \min\{x_r^*, a_r^i\} a_\varphi^i - \sum_{i \in Q_-(x_\varphi)} w_i \min\{x_r^*, a_r^i\} a_\varphi^i, \\ \Delta W &= \sum_{i \in Q_-(x_\varphi)} w_i \min\{x_r^*, a_r^i\} - \sum_{i \in Q_+(x_\varphi)} w_i \min\{x_r^*, a_r^i\}. \end{aligned}$$

Целевая функция линейна для  $x_\varphi \in [x_\varphi^{\min}, x_\varphi^{\max}]$ , поскольку

$$\Delta W = \sum_{i \in Q_-(x_\varphi)} w_i \min\{x_r^*, a_r^i\} - \sum_{i \in Q_+(x_\varphi)} w_i \min\{x_r^*, a_r^i\} = w_i x_r - w_i x_r = 0$$

подразумевает

$$f_{MK}(x_r^*, x_\varphi) = C_0 + \Delta W x_\varphi = C_0.$$

Рассмотрим три случая.

Случай 1:  $\Delta W > 0$ . Выберем  $x'_\varphi$ , удовлетворяющее  $x_\varphi^{\min} < x'_\varphi < x_\varphi^*$ . Таким образом,

$$f_{MK}(x_r^*, x'_\varphi) = C_0 + \Delta W x'_\varphi < C_0 = f_{MK}(x_r^*, x_\varphi^*),$$

т.е.  $x'_\varphi$  не является точкой минимума функции (5.62).

Случай 2:  $\Delta W < 0$ . Выберем  $x'_\varphi$ , такое, что  $x_\varphi^{\max} > x'_\varphi > x_\varphi^*$ . Тогда

$$f_{MK}(x_r^*, x'_\varphi) = C_0 + \Delta W x'_\varphi < C_0 = f_{MK}(x_r^*, x_\varphi^*),$$

и  $x'_\varphi$  также не является точкой минимума функции (5.62).

Случай 3:  $\Delta W = 0$ . В этом случае  $f_{MK}(x_r^*, x'_\varphi) = C_0 = f_{MK}(x_r^*, x_\varphi^*)$ . Таким образом,  $x'_\varphi$  является точкой минимума функции (5.62). Но  $x_\varphi^*$  также является точкой минимума.

Оценим значения  $x_\varphi^{\min}$  и  $x_\varphi^{\max}$ .

Из  $a_\varphi^i \in Q_+(x_\varphi) \wedge a_\varphi^i \in Q_+^* \quad \forall i \in \overline{\{1, N\}}$  непосредственно следует, что  $0 < a_\varphi^i - x_\varphi \leq 2 \quad \forall i \in \overline{\{1, N\}}$ , что подразумевает  $0 > x_\varphi \geq a_\varphi^i - 2 \quad \forall i \in \overline{\{1, N\}}$ . Отсюда можно

заклучить, что  $x_\varphi \geq \min_{i \in Q_+^*} \{a_\varphi^i\} - 2$ .

В случае, если  $a_\varphi^i \in Q_-(x_\varphi) \wedge a_\varphi^i \in Q_-^* \quad \forall i \in \overline{\{1, N\}}$ , справедливо  $0 > a_\varphi^i - x_\varphi \geq -2 \quad \forall i \in \overline{\{1, N\}}$ , и далее  $0 < x_\varphi \leq a_\varphi^i + 2 \quad \forall i \in \overline{\{1, N\}}$ . Следовательно,  $x_\varphi \leq \max_{i \in Q_-^*} \{a_\varphi^i\} + 2$ .

В третьем случае,  $a_\varphi^i \in Q_2(x_\varphi) \wedge a_\varphi^i \in Q_2^* \quad \forall i \in \overline{\{1, N\}}$ . Аналогичным образом,  $|a_\varphi^i - x_\varphi| < 2 \quad \forall i \in \overline{\{1, N\}}$ , и

$$0 < x_\varphi \leq a_\varphi^i + 2 \vee 0 > x_\varphi \geq a_\varphi^i - 2 \quad \forall i \in \overline{\{1, N\}},$$

из чего следует, что  $x_\varphi \leq \max_{i \in Q_2^*} \{a_\varphi^i\} + 2 \vee x_\varphi \geq \min_{i \in Q_2^*} \{a_\varphi^i\} - 2$ . Таким образом,

$$x_\varphi^{\min} = \max \left\{ \max_{i \in Q_+^*} \{a_\varphi^i\} - 2, \min_{i \in Q_2^*} \{a_\varphi^i\} - 2 \right\},$$

$$x_\varphi^{\max} = \min \left\{ \min_{i \in Q_-^*} \{a_\varphi^i\} + 2, \max_{i \in Q_2^*} \{a_\varphi^i\} + 2 \right\}.$$

В случае, когда  $\Delta W = 0$ , если  $x_\varphi^* \notin \{a_\varphi^i\}$  является точкой минимума функции (5.62), одно из значений

$$\max_{i \in Q_+^*} \{a_\varphi^i\} - 2, \quad \min_{i \in Q_2^*} \{a_\varphi^i\} - 2, \quad \min_{i \in Q_-^*} \{a_\varphi^i\} + 2, \quad \max_{i \in Q_2^*} \{a_\varphi^i\} + 2$$

также является точкой минимума. Лемма доказана.

Предположим, что  $X^* = (x_r^*, x_\varphi^*)$  является точкой минимума функции (5.62), и значение координаты  $x_\varphi^*$  известно. Тогда значение  $x_r^*$  может быть вычислено следующим образом.

Разобьем множество индексов известных точек-потребителей  $\{A_i\}$  на два подмножества, определенных следующим образом:

$$Q_{<}^* = \{i \mid \delta_\varphi(a_\varphi^i, x_\varphi^*) \leq 2\}, \quad Q_{>}^* = \{i \mid \delta_\varphi(a_\varphi^i, x_\varphi^*) > 2\}.$$

Целевая функция (5.62) может быть выражена как

$$\begin{aligned} f_{MK}(x_r, x_\varphi) &= \sum_{i \in Q_{>}} w_i (a_r^i + x_r^*) + \sum_{i \in Q_{<}} w_i |x_r^*, a_r^i| + \sum_{i \in Q_{<}} w_i \min\{x_r^*, a_r^i\} \delta_\varphi(a_\varphi^i, x_\varphi^*) \\ &= \sum_{i \in Q_{>}} w_i (a_r^i + x_r^*) + \sum_{i \in Q_{<}} w_i |x_r^* - a_r^i| + \frac{\sum_{i \in Q_{<}} w_i a_r^i \delta_\varphi(a_\varphi^i, x_\varphi^*)}{2} \\ &\quad + \frac{\sum_{i \in Q_{<}} w_i a_r^i \delta_\varphi(a_\varphi^i, x_\varphi^*) (|x_r - a_r^i| + x_r)}{2}. \end{aligned}$$

Введем обозначения

$$C_1 = \sum_{i \in Q_{>}} w_i a_r^i + \frac{\sum_{i \in Q_{<}} w_i a_r^i \delta_\varphi(a_\varphi^i, x_\varphi^*)}{2},$$

$$C'_1 = \sum_{i \in Q_{>}} w_i + \frac{\sum_{i \in Q_{<}} w_i \delta_\varphi(a_\varphi^i, x_\varphi^*)}{2},$$

$$C_i = w_i + w_i \delta_\varphi(a_\varphi^i, x_\varphi^*)/2 \quad \forall i = \overline{1, N}.$$

$$\text{Тогда } f_{MK}(x_r, x_\varphi^*) = C_1 + C'_1 x_r + \sum_{i \in Q_{<}} C_i |x_r - a_r^i|.$$

Эта функция кусочно линейна. Границы каждого ее интервала  $a_r^i, i = \overline{1, N}$  – ее возможные точки минимума. Если производная данной функции на каком-то интервале  $[a_r^i, a_r^{i'}]$  равна 0, то все точки интервала могут являться точками минимума (5.62), в том числе конечные его точки  $a_r^i$  и  $a_r^{i'}$ . Таким образом, для нахождения точки минимума (5.62) достаточно определить значение функции в точках, входящих в множество

$$\begin{aligned} \chi = \{ & (a_r^i, a_\varphi^j) | i = \overline{1, N}, j = \overline{1, N} \} \cup \{ (a_r^i, a_\varphi^j + 2) | i = \overline{1, N}, j = \overline{1, N} \} \cup \\ & \cup \{ (a_r^i, a_\varphi^j - 2) | i = \overline{1, N}, j = \overline{1, N} \} \cup \{ (0, 0) \}. \end{aligned}$$

Мощность данного множества равна  $|\chi| = 3N^2 + 1$ .

Алгоритм 5.9 предлагается как способ решения задачи (5.60)-(5.62).

Алгоритм 5.9. Решение задачи Вебера с метрикой Москвы (Карлсруэ)

Шаг 1: Решить задачу (5.61), используя любой алгоритм для решения задачи (5.56). Сохранить результат в  $x_h^*$ .

Шаг 2:  $x_\varphi^* = 0; x_r^* = 0; f^* = f_{MK}(0, 0)$ .

Шаг 3: Для каждого  $i = \overline{1, N}$  выполнять:

Шаг 3.1: Для каждого  $j = \overline{2, N}$  выполнять:

Шаг 3.1.1: Присвоить  $f = f_{MK}(a_r^i, a_\varphi^j)$ .

Шаг 3.1.2: Если  $f < f^*$ , то  $x_r^* = a_r^i; x_\varphi^* = a_\varphi^j; f^* = f$ .

Шаг 3.1.3: Если  $f = f_{MK}(a_r^i, a_\varphi^j + 2)$ .

Шаг 3.1.4: Если  $f < f^*$ , то  $x_r^* = a_r^i; x_\varphi^* = a_\varphi^j + 2; f^* = f$ .

Шаг 3.1.5: Присвоить  $f = f_{MK}(a_r^i, a_\varphi^j - 2)$ .

Шаг 3.1.6: Если  $f < f^*$ , то  $x_r^* = a_r^i; x_\varphi^* = a_\varphi^j - 2; f^* = f$ .

Шаг 3.1.7: Следующая итерация 3.1.

Шаг 3.2: Следующая итерация 3.

Шаг 4: Возвратить результат  $X^* = (x_r^*, x_\varphi^*, x_h^*)$ ; ОСТАНОВ.

Оценим вычислительную сложность Алгоритма 5.9. Пусть  $N$  – число точек-потребителей. Шаг 1 Алгоритма 5.9 находит решение задачи Вебера с метрикой  $l_1$  (см. [114, 498]), что включает в себя сортировку (упорядочивание) значений координат – процедура асимптотической сложности  $O(N \log N)$  – и суммирование значений координат (меньшая, линейная сложность). Таким образом, асимптотическая сложность Шага 1 описывается асимптотической формулой  $O(N \log N)$ . Шаги 3–3.2 задают вложенный цикл, в котором значения целевой функции вычисляются для каждого из  $N$  значений координат  $a_\varphi^i$  и  $a_\varphi^i \pm 2$ . На шагах 3 и 4 значение целевой функции оценивается  $1 + 3N^2$  раз. Целевая функция линейна (ее вычисление имеет асимптотическую сложность  $O(N)$ ), таким образом, вычислительная сложность Шагов 3-5.3 и всего Алгоритма 5.9 – в пределах  $O(N^3)$ .

### 5.12 Алгоритм для задачи размещения с метрикой британской железной дороги

Лемма 5.18 призвана сократить множество рассматриваемых возможных решений задачи (решений-кандидатов) в случае задачи Вебера с метрикой британской железной дороги (см. Стратегию 4).

Лемма 5.18. Если  $\exists i' \in \overline{\{1, N\}}: w_{i'} > \sum_{i=1}^N \frac{w_i}{2}$ , то точка  $O = (0,0)$  является точкой минимума целевой функции (5.52) с функцией расстояния (5.54).

*Доказательство.* Предположим, что  $X^* = (x_r, x_\varphi)$  является решением задачи (5.52) с функцией расстояния (5.54). Здесь мы используем полярные координаты. Если  $X^* \neq A_i = (a_r^i, a_\varphi^i) \quad \forall i \in \overline{\{1, N\}}$ , и  $X^* \neq (0,0)$ , то  $F(X^*) = \sum_{i=1}^N w_i a_r^i + \sum_{i=1}^N w_i x_r^*$ .

Сравним это значение со значением целевой функции (5.52) в точке  $O$ :

$$F(O) = \sum_{i=1}^N w_i a_r^i.$$

Поскольку  $x_r^* > 0$ , с учетом  $X^* \neq (0,0)$ , и  $w_i \geq 0 \forall i \in \overline{\{1, N\}}$ , имеем  $F(X^*) > F(O)$ , т.е.  $X^*$  не является точкой минимума, за исключением случаев  $X^* \in \{O\} \cup \{A_i | i \in \overline{\{1, N\}}\}$ , либо



$$w_i = 0 \forall i = \overline{1, N}. \quad (5.64)$$

Если выполняется (5.64), любая точка (включая  $O$ ) является точкой минимума.

Предположим, что  $X^* = A_{i'}, i' \in \overline{1, N}$ , и  $X^* \neq O = (0,0)$ . Тогда

$$\begin{aligned} F(X^*) &= F(A_{i'}) = \sum_{i \in \overline{1, N} \setminus \{i'\}} w_i (a_r^i + x_r) = \sum_{i=1}^N w_i a_r^i + x_r \sum_{i=1}^N w_i - w_{i'} a_r^{i'} - \\ & \quad w_{i'} x_r = \\ &= \sum_{i=1}^N w_i a_r^i + a_r^{i'} \sum_{i=1}^N w_i - 2 \cdot w_{i'} a_r^{i'} F(O) + a_r^{i'} \sum_{i=1}^N w_i - 2 \cdot w_{i'} a_r^{i'}. \end{aligned}$$

Таким образом,  $F(A_{i'}) < F(O)$ , если и только если

$$a_r^{i'} \sum_{i=1}^N w_i - 2 \cdot w_{i'} a_r^{i'} \equiv a_r^{i'} \sum_{i=1}^N w_i < 2w_{i'} a_r^{i'} \equiv \sum_{i=1}^N w_i / 2 < w_{i'},$$

что соответствует нашему изначальному предположению. Лемма доказана.

Результат Леммы 5.18 согласуется с главным результатом работы [486]. В [486] Р.Чен доказывает, что  $A_i$  является решением классической задачи Вебера с евклидовой метрикой, если  $\exists i' \in \overline{1, N}: w_{i'} \geq \sum_{i=1}^m w_i$ . Если это условие не выполняется в случае с метрикой британской железной дороги, единственным возможным решением является  $(0,0)$ .

Предложим следующий простой Алгоритм 5.10.

Алгоритм 5.10. *Решение задачи Вебера с метрикой британской железной дороги.*

Шаг 1: Рассчитать  $W = \sum_{i=1}^N w_i$ .

Шаг 2: Для каждого  $i = \overline{1, N}$  выполнять:

Если  $w_i \geq W/2$ , то вернуть  $X^* = A_i$ . ОСТАНОВ.

Шаг 3: вернуть результат  $X^* = (0,0)$ . ОСТАНОВ.

Очевидно, что этот примитивный алгоритм имеет асимптотическую сложность  $O(N)$ .

### 5.13 Задача Вебера с мерой расстояния, включающей минимальную стоимость транспортировки

В настоящем параграфе рассматривается задача Вебера со специальной мерой расстояния, в которой установлены минимальные фиксированные расстояния между объектами. Предлагается алгоритм для быстрого решения данной задачи с заранее

известной точностью. Вычислительные эксперименты показывают более высокую эффективность предложенного алгоритма по сравнению с известными методами выпуклой оптимизации.

Введем следующую меру расстояния:

$$d_1(X, Y) = \max\{1, \|X - Y\|_2\} \forall X, Y \in \mathbb{R}^n. \quad (5.65)$$

Отметим, что данная мера не является метрикой. В то же время, следующая ее модификация метрикой является:

$$d'_1(X, Y) = \begin{cases} \max\{1, \|X - Y\|_2\} \forall X, Y \in \mathbb{R}^n, X \neq Y, \\ 0, & X = Y. \end{cases}$$

В различных задачах автоматической группировки аналогичная метрика используется для описания расстояния между векторами данных (объектами) и ядром группы (кластера) [499] при фиксированном диаметре ядра. Мера расстояния, при которой игнорируются все расстояния, меньшие, чем некий известный предел точности измерений  $\varepsilon$  эквивалентна нашей метрике  $d_1$ :

$$d_{clusterCore} = \max\{0, \|X - Y\|_2 - \varepsilon\}.$$

Расстояния могут быть элементарным образом масштабированы:

$$d_{clusterCore} = \varepsilon \cdot d_1(X/\varepsilon, Y/\varepsilon) - \varepsilon.$$

В системах общественного транспорта цена перевозки или проезда может зависеть от расстояния. В то же время, часто устанавливается некий нижний предел стоимости. Например, начальная цена (цена посадки) в такси включает определенное расстояние. Если требуется перевозка на меньшее расстояние, все равно взимается эта минимальная стоимость. После соответствующего масштабирования расстояний, функция (5.65) описывает стоимость перевозки между точками  $X$  и  $Y$ , и задача Вебера с мерой расстояния  $d_1$  становится задачей поиска точки, такой, что суммарная стоимость перевозки из нее до известных точек-потребителей  $A_1, \dots, A_N$  достигает минимума:

$$X^* = \operatorname{argmin}_{X \in \mathbb{R}^2} f_1(X) = \operatorname{argmin}_{X \in \mathbb{R}^2} \sum_{i=1}^N w_i d_1(A_i, X). \quad (5.66)$$

Оптимизационные методы прямого поиска (метод Нелдера-Мида [500], Пауэлла [501] и др.) способны решить задачу (5.66) с заданной точностью (5.66). Тем не менее, алгоритм, предлагаемый ниже, решает ее быстрее. Кроме того, в некоторых случаях наш алгоритм позволяет получить точное решение.

Наиболее популярной процедурой для задачи Вебера с  $l_p$ -нормой является итеративная процедура Вайсфелда [144]. Данная процедура была предложена для задачи

с евклидовой нормой ( $p = 2$ ) и обобщена для  $l_p$ -норм с параметром  $p \in [1; 2]$  [143].

Для классической задачи Вебера с евклидовыми расстояниями процедура Вайсфелда реализует градиентный спуск следующим образом [144, 502, 503].

Алгоритм 5.11 Процедура Вайсфелда.

Дано: координаты точек-потребителей  $A_1 = (a_{1,1}, a_{1,2}), \dots, A_N = (a_{N,1}, a_{N,2})$ , весовые коэффициенты  $w_1, \dots, w_N$ , точность (допуск)  $\varepsilon \in \mathbb{R}, \varepsilon > 0$ .

1: Выбрать начальную точку  $X = (x_1, x_2)$ . Обычно используется точка с усредненным значением каждой из координат:

$$x_r = \sum_{i=1}^N w_i a_{i,r} / N, r = 1, 2.$$

2: Если  $\exists i \in \{1, N\}: X = A_i$ , то присвоить  $X' = X$ , иначе

$$x'_r = \sum_{i=1}^N \frac{a_{i,r} w_i / \|X - A_i\|_2}{\sum_{j=1}^N w_j / \|X - A_j\|_2}, r = 1, 2.$$

3: Если  $\|X' - X\| < \varepsilon$ , то ОСТАНОВ, иначе перейти к 2.

Эта простая процедура сходится крайне медленно, если расстояние от начального решения до одной из точек-потребителей  $A_i$  мало [504, 293]. Но многие авторы предлагают модификации данной процедуры, которые ускоряют сходимость, не меняя основной идеи метода [505, 502].

Очень похожая метрика, называемая метрикой экрана радара [88], является метрикой, основанной на норме, определяемой как

$$d_{rs}(X, Y) = \|X - Y\|_{rs} = \min\{1, \|X - Y\|_2\} \forall X, Y \in \mathbb{R}^n.$$

В [506, 507] авторы предлагают алгоритмы решения задачи Вебера с аналогичной метрикой (задача названа задачей Вебера с ограниченными расстояниями):

$$d_{limited}(X, y) = \min\{D_{max}, \|X - Y\|_2\},$$

где  $D_{max} \in \mathbb{R}$  – фиксированное значение. Задача с такой метрикой не является задачей выпуклой оптимизации.

Наша целевая функция выпукла (см. ниже), соответственно, могут быть применены методы локального поиска.

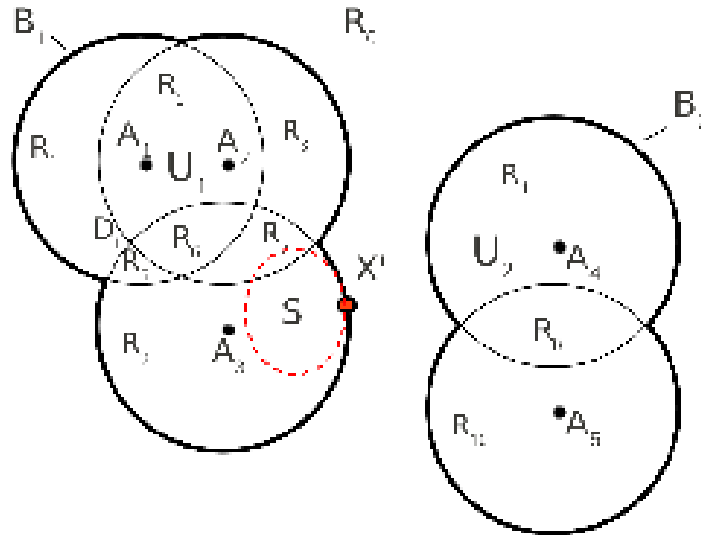


Рисунок 5.13 – Иллюстрация задачи (5.65), ее регионов  $\mathcal{R}_k, k = \overline{1, M}$  и объединения  $U_1, U_2$  этих регионов

В [508] и в следующем параграфе рассматривается решение задачи Вебера с евклидовыми расстояниями и ограничением

$$\begin{aligned} \|X - A_i\|_2 &\leq 1 \forall A_i \in Q_<, \\ \|X - A_i\|_2 &> 1 \forall i \in Q_>, \end{aligned} \quad (5.67)$$

где  $Q_<$  и  $Q_>$  – множества индексов точек требования,  $Q_> \cup Q_< = \{\overline{1, N}\}$ . Очевидно, что наша целевая функция (5.66) совпадает с целевой функцией (5.1) в пределах каждого из регионов (рис.5.13), ограниченных условиями (5.67):

$$\begin{aligned} f_1(X) &= \sum_{i \in Q_>} w_i \|A_i - X\|_2 + C \forall X \in \mathbb{R}^2: \\ \|X - A_i\|_2 &\leq 1 \forall A_i \in Q_<, \|X - A_i\|_2 > 1 \forall i \in Q_>, \end{aligned} \quad (5.68)$$

где  $C = \sum_{i \in Q_<} w_i$ .

Решив задачу (5.68) на каждом из регионов, можно получить решение задачи (5.66). В то же время, верхняя оценка количества регионов возрастает с в квадратичной зависимости от числа точек требования [506], соответственно, требуется решить до  $N^2$  задач (5.68).

Целевая функция (5.1) недифференцируема в точках  $A_i, i = \overline{1, N}$ . Кроме того, наша целевая функция (5.66) – недифференцируема в точках  $X \in \mathbb{R}^2: \exists i \in \{\overline{1, N}\}: \|X - A_i\| = 1$ .

Для простоты изложения, без потери общности предположим, что  $w_i > 0 \forall i \in$

$\overline{\{1, N\}}$ . Точки-потребители с нулевым весом могут быть просто удалены из задачи.

Целевая функция (5.1) выпукла (Рис.5.14).

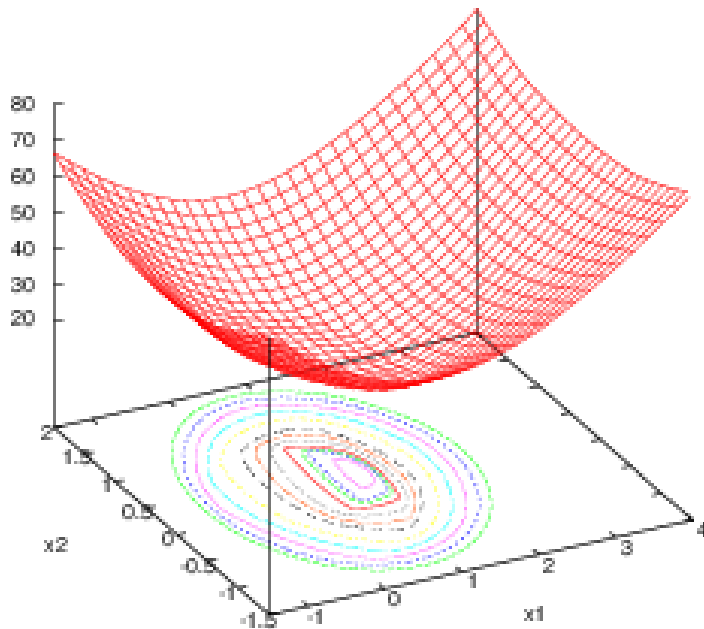
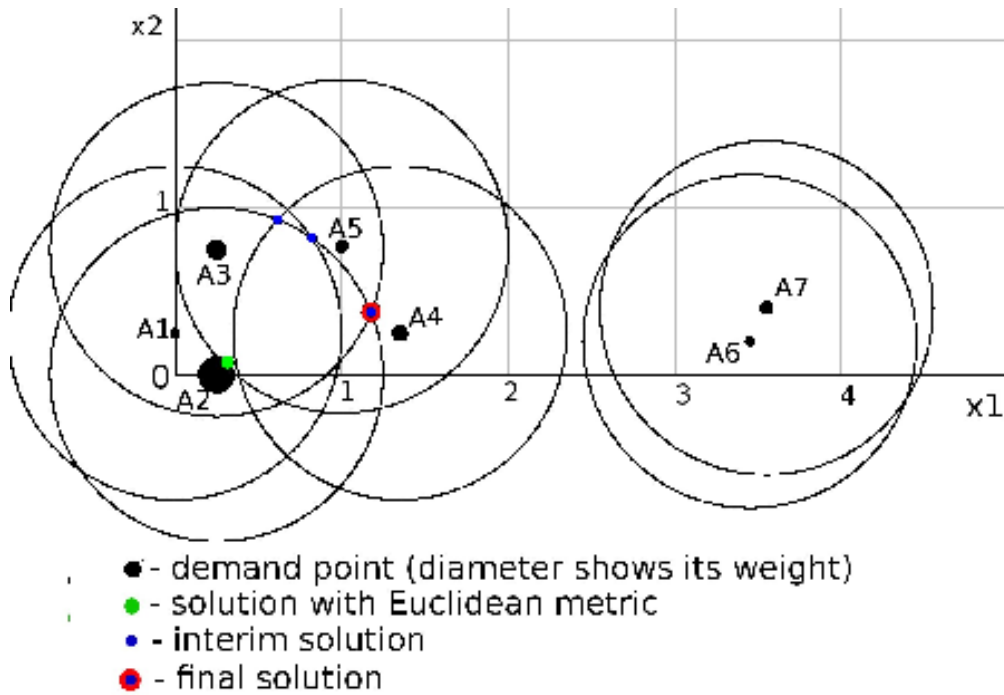


Рисунок 5.14 – Пример схемы задачи и график ее целевой функции.

Лемма 5.19 Целевая функция (5.66) выпукла и непрерывна.

*Доказательство.* Сумма непрерывных выпуклых функций является непрерывной

выпуклой функцией. Лемма доказана.

Лемма 5.20 Если  $\exists \mathcal{S} \subset \mathbb{R}^2: \|X - A_i\|_2 \leq 1 \forall X \in \mathcal{S}, i \in \overline{\{1, N\}}$ , то любая точка  $X^* \in \mathcal{S}$  является точкой минимума функции (5.66). Более того, если  $X^*$  - точка минимума функции (5.66), то  $X^* \in \mathcal{S}$ .

*Доказательство.* Пусть  $X' \notin \mathcal{S}$ . Тогда  $\exists Q_{>} \subset \overline{\{1, N\}}: \|A_i - X'\|_2 > 1 \forall i \in Q_{>}$ .

Таким образом,

$$f_1(X') = \sum_{i=1}^N w_i + \sum_{i \in Q_{>}} w_i (\|X' - A_i\|_2 - 1).$$

Отметим, что  $(\|X' - A_i\|_2 - 1) > 0 \forall i \in Q_{>}$ .

Следовательно,  $f_1(X^*) = \sum_{i=1}^N w_i < f_1(X')$ .

Для любой точки  $X' \in \mathbb{R}^2$  обозначим множества

$$Q_{>}(X') = \{i \in \overline{\{1, N\}} \mid \|A_i - X'\|_2 > 1\}$$

и

$$\mathcal{R}(X') = \{X \in \mathbb{R}^2 \mid \|X - A_i\|_2 \leq 1 \equiv \|X' - A_i\|_2 \leq 1 \forall i \in \overline{\{1, N\}}\}.$$

Кроме того, введем обозначения для окружностей

$$O_i = \{X \in \mathbb{R}^2 \mid \|X - A_i\|_2 = 1\} \forall i \in \overline{\{1, N\}}.$$

Пусть  $\mathcal{C}$  - объединение всех окружностей:

$$\mathcal{C} = \bigcup_{i=1}^N O_i.$$

Обозначим также множество

$$\mathcal{R}_0 = \{X \in \mathbb{R}^2 \mid \|X - A_i\|_2 > 1 \forall i \in \overline{\{1, N\}}\}.$$

Лемма 5.21 Если  $X^* \in \mathbb{R}^2$  - точка минимума функции (5.66), и  $X^* \notin \mathcal{C}$ , то  $X^*$  является точкой минимума функции

$$f^*(X) = \sum_{i \in Q_{>}(X^*)} w_i \|X - A_i\|_2. \quad (5.69)$$

*Доказательство.* Очевидно, что,  $X^* \in \mathcal{R}(X^*)$ . Соответственно,

$$f_1(X) = \sum_{i \in Q_{>}(X^*)} w_i \|X - A_i\|_2 + \sum_{i \in \overline{\{1, N\}} \setminus Q_{>}(X^*)} w_i \forall X \in \mathcal{R}(X^*).$$

Второе слагаемое является константой. Первое слагаемое является целевой функцией классической задачи Вебера с евклидовыми расстояниями. Удалив второе

слагаемое, получаем эквивалентную целевую функцию

$$f_1^*(X) = \sum_{i \in Q_{>}(X^*)} w_i \|X - A_i\|_2. \quad (5.70)$$

Решение задачи (5.66) совпадает с решением задачи (5.1) с ограничением  $X \in \mathcal{R}(X^*)$ .

Целевая функция задачи Вебера выпукла. Для задач оптимизации с выпуклыми целевыми функциями решение либо совпадает с решением безусловной задачи оптимизации, либо лежит на границе запрещенной области [509] (более того, решение задачи условной оптимизации «видимо» из точки, являющейся решением задачи безусловной оптимизации, т.е. существует отрезок, соединяющий решения задач условной и безусловной оптимизации, не пересекающий границу запрещенной области).

Пусть  $B$  - граница области  $\mathcal{R}(X^*)$ . Тогда  $B \subset \mathcal{C}$ . Поскольку  $X^* \notin \mathcal{C}$ ,  $X^*$  является точкой минимума функции (5.70).

Отметим, что, если  $Q_{>}(X^*) = \emptyset$ , то  $f_1^*(X) = 0$ , и точкой минимума является любая точка. Таким образом, любая точка  $X \in \mathcal{R}(X^*)$  является точкой минимума функции (5.66), что соответствует случаю, рассматриваемому Леммой 5.20.

Если  $X^*$  является точкой оптимума задачи (5.66), и  $X^* \notin \mathcal{C}$ , то, выбрав любую начальную точку  $X_0 \in \mathcal{R}^2$ , метод градиентного спуска в пределах области  $\mathcal{R}(X_0)$  сойдется к  $X^*$  либо приведет к границе области  $\mathcal{R}(X_0)$ .

Рассмотрим случай  $X^* \in \mathcal{C}$ . Предположим, что  $X^* \in O_{i^*}$ . Тогда

$$f_1(X) = w_{i^*} + \sum_{i \in \overline{\{1, N\}} \setminus \{i^*\}} w_i \min\{1, \|X - A_i\|_2\} \forall X \in O_{i^*}.$$

Первое слагаемое является константой. Удалив его, получаем эквивалентную функцию на  $O_{i^*}$ :

$$f_1^{**}(X) = \sum_{i \in \overline{\{1, N\}} \setminus \{i^*\}} w_i \min\{1, \|X - A_i\|_2\} \forall X \in O_{i^*}. \quad (5.71)$$

Обозначим дугу  $\mathcal{A}(X^*, i) = \mathcal{R}(X^*) \cap O_i \forall X \in \mathcal{C}, i = \overline{1, N}$  и точки пересечения

$$J_{i^*} = \bigcup_{i \in \overline{\{1, N\}} \setminus \{i^*\}} O_i \cap O_{i^*}.$$

Лемма 5.22 Пусть  $X \in O_{i^*}$  является точкой минимума функции (5.71), причем  $X^* \notin J_{i^*}$ . Тогда  $X^*$  является решением условной задачи оптимизации с целевой функцией

$$f^*(X) = \sum_{i \in Q_{>}(X^*)} w_i \|X - A_i\|_2. \quad (5.72)$$

и ограничением  $X \in O_{i^*}$

*Доказательство* аналогично Лемме 5.21.

В случае условной оптимизации с дополнительным ограничением  $X \in O_{i^*}$ ,  $i^* = \overline{1, N}$  вместо градиентного спуска можно использовать спуск вдоль кривой  $O_{i^*}$ , оценивая значение ковариантной производной (5.72) вдоль кривой  $O_{i^*}$ . Начав из точки  $X_0 \in O_{i^*}$ , такой спуск вдоль кривой сойдется к точке  $X^*$  либо приведет к точке пересечения с другой кривой.

Таким образом, для точки минимума  $X^*$  функции  $f_1(X)$  возможны четыре случая.

Случай 1.  $X^* \in \mathcal{S}$  (см. Лемму 5.20).

Случай 2.  $X^* \notin \mathcal{C}$ . В этом случае  $X^*$  является решением задачи (5.69).

Случай 3.  $X^* \in \mathcal{C}$ ,  $X^* \notin \bigcup_{i=1}^N J_i$  (точка минимума лежит на окружности, но не совпадает с точкой пересечения с другой окружностью). Из  $X^* \in \mathcal{C}$  следует, что  $\exists i^* \in \overline{1, N}$ :  $X^* \in O_{i^*}$  и дуга  $\mathcal{A}(X^*, i^*)$ , на которой функция (5.72) дифференцируема. Начав из любой произвольной точки  $X'' \in \mathcal{A}(X^*, i^*)$ , можно использовать метод спуска вдоль дуги  $\mathcal{A}(X^*, i^*)$ .

Случай 4.  $X^* \in \bigcup_{i=1}^N J_i$  (т.е. точка минимума является точкой пересечения окружностей). Если известно, что  $X \in O_{i^*}$ , то все точки пересечения  $J_{i^*}$  окружности  $O_{i^*}$  могут быть оценены в качестве решений-кандидатов.

В [510] и [482] авторы доказывают, что каждая следующая итерация процедуры Вайсфелда (Алгоритм 5.12) дает результат  $X'$ , который как минимум не хуже результата  $X$ , достигнутого на предыдущей итерации:

$$f_w(X') \leq f_w(X).$$

Таким образом, если  $X, X' \in \mathcal{R}(X)$ , то  $f_1(X') \leq f_1(X)$ .

Из этого можно сделать следующее заключение.

Следствие 5.23 Пусть  $X \in \mathbb{R}^2$  – произвольная точка,  $X \notin \mathcal{C}$ . Пусть координаты точки  $X' = (x'_1, x'_2)$  рассчитываются следующим образом:

$$x'_r = \sum_{i \in Q_{>}(X)} \frac{a_{i,r} w_i / \|X - A_i\|_2}{\sum_{j \in Q_{>}(X)} w_j / \|X - A_j\|_2}, r = 1, 2.$$

Если  $f_1(X') > f_1(X)$ , то  $X' \notin \mathcal{R}(X)$ .

Иными словами, если  $f_1(X') > f_1(X)$ , то отрезок  $\overline{XX'}$  пересекает одну из



окружностей  $O_i$ :

$$\exists i \in \overline{\{1, N\}}: \overline{XX'} \cap O_i \neq \emptyset.$$

Если  $X' \neq X$ , то направление вектора  $\overline{XX'}$  совпадает с направлением градиента  $\nabla f_1(X)$ . Поскольку  $X \notin \mathcal{C}$ , существует окрестность  $\dot{O}(X) \subset \overline{XX'}$ , такая, что  $f_1(X'') \leq f_1(X) \forall X'' \in \dot{O}(X)$ . Таким образом, заменяя  $X'$  на средней точкой отрезка  $\overline{XX'}$ , после нескольких итераций получим отрезок  $\overline{XX'}$ :  $f_1(X') \leq f_1(X)$ . Чтобы ускорить спуск, можно сперва оценивать значение целевой функции  $f_1(X)$  в точках пересечения  $\bigcup_{i=1}^N (O_i \cap \overline{XX'})$ . При спуске вдоль кривой  $O_{i^*}$ , можно применять аналогичный подход для обнаружения точек пересечения окружностей.

Следующий алгоритм основан на стандартной процедуре Вайсфелда.

Алгоритм 5.12 Алгоритм для решения задачи (5.66).

Дано: координаты точек-потребителей  $A_1 = (a_{1,1}, a_{1,2}), \dots, A_N = (a_{N,1}, a_{N,2})$ , весовые коэффициенты  $w_1, \dots, w_N$ , точность (допуск)  $\varepsilon \in \mathbb{R}$ ,  $\varepsilon > 0$ .

1: Вычислить координаты начальной точки  $X = (x_1, x_2)$ :  $x_r = \sum_{i=1}^N w_i a_{i,r} / N, r = 1, 2$ .

2: Если  $x_r \in \mathcal{C}$ , то перейти к 3.

2.1: Вычислить  $X' = (x'_1, x'_2)$  как

$$x'_r = \sum_{i \in Q_{>}(X)} \frac{a_{i,r} w_i / \|X - A_i\|_2}{\sum_{j \in Q_{>}(X)} w_j / \|X - A_j\|_2}, r = 1, 2.$$

2.2: Если  $f_1(X') \leq f_1(X)$ , то перейти к 4.

2.3: Для  $i \in \overline{\{1, N\}}$  выполнять:

2.3.1: Если  $[a_{i,1} \pm 1] \cap [x_1; x'_1] \neq \emptyset$  и  $[a_{i,2} \pm 1] \cap [x_2; x'_2] \neq \emptyset$ , то:

2.3.1.1: Вычислить множество точек пересечения  $\mathcal{S}_{intersect} = \overline{XX'} \cap O_i$ .

2.3.1.2: Если  $\mathcal{S}_{intersect} \neq \emptyset$  и  $\exists X'' \in \mathcal{S}_{intersect} : f_1(X'') < f_1(X)$ , то присвоить  $X' = X''$

и перейти к 4.

2.3.2: Следующая итерация цикла 2.3.

2.4: Если  $f_1(X') > f_1(X)$ , то:

2.4.1: Вычислить  $x'_1 = (x_1 + x'_1)/2, x'_2 = (x_2 + x'_2)/2$ .

2.4.2 Если  $f_1(X') < f_1(X)$  или  $\|X' - X\|_2 < \varepsilon$ , то перейти к 4.

2.4.3: Перейти к 2.4.1.

2.5: Перейти к 4.

3: Для каждого  $i \in \{\overline{1, N}\}$ :  $\|X - A_i\| = 1$  выполнять:

3.1: Вычислить проекцию  $X'' = (x''_1, x''_2)$  точки  $X'$  на кривую  $O_i$ , решив систему уравнений

$$(a_{i,1} - x''_1)^2 + (a_{i,2} - x''_2)^2,$$

$$x''_2 - a_{i,2} = \frac{x'_2 - a_{i,2}}{x'_1 - a_{i,1}}(x''_1 - a_{i,1})$$

и выбрав ее решение, ближайшее к  $X'$ .

3.2: Если  $f_1(X'') < f_1(X)$ , то присвоить  $X' = X''$ , перейти к 4.

3.3: Для  $j \in \{\overline{1, N}\}$  выполнять:

3.3.1: Если  $[a_{j,1} \pm 1] \cap [x_1; x''_1] \neq \emptyset$  и  $[a_{j,2} \pm 1] \cap [x; x''_2] \neq \emptyset$ , то:

3.3.1.1: Вычислить множество точек пересечения  $\mathcal{S}_{intersect} = (\overline{XX''} \cap O_j) \setminus \{X, X''\}$ .

3.3.1.2: Если  $|\mathcal{S}_{intersect}| \neq 1$ , перейти к 3.3.4.

3.3.2: Вычислить координаты точки пересечения  $X'''$  окружностей  $O_i$  и  $O_j$ .

3.3.3: Если  $f_1(X''') < f_1(X)$ , то присвоить  $X' = X'''$ , перейти к 4.

3.3.4 Следующая итерация цикла 3.3.

3.5: Вычислить  $x'_1 = (x_1 + x''_1)/2$ ,  $x'_2 = (x_2 + x''_2)/2$  и перерассчитать проекцию  $X'' = (x''_1, x''_2)$  точки  $X'$  на кривую  $O_i$ , решив систему уравнений (см. Шаг 3.1).

3.6: Если  $f_1(X'') < f_1(X)$ , то присвоить  $X' = X''$  и перейти к 4.

3.7: Если  $\|X - X''\| \geq \varepsilon$ , то перейти к 3.5.

3.8: Следующая итерация цикла 3.

4: Если  $f(X') > f(X)$ , или  $\|X - X'\|_2 < \varepsilon$ , то ОСТАНОВ.

5: Присвоить  $X = X'$  и перейти к 2.

Шаги 2.1, 4 и 5 данного алгоритма реализуют стандартную процедуру Вайсфелда. Шаги 2.3–2.3.3 оценивают значение целевой функции в точках пересечения отрезка  $XX'$  с окружностями  $O_i$ . Если отсутствует точка пересечения с лучшим значением  $f_1(X)$ , то на Шагах 2.4.1–2.4.3 алгоритм сокращает рассматриваемый отрезок  $XX'$ .

Шаги 3–3.8 реализуют спуск вдоль кривой  $O_i$  и оценку значений целевой функции в точках пересечения окружностей.

С целью тестирования алгоритма были сгенерированы 110 наборов данных. Каждый набор содержит  $N = 10, 100, 1000$  или  $10000$  точек-потребителей. Координаты точек-потребителей равномерно распределены на квадрате размером  $5 \times 5$  или  $50 \times 50$ :  $a_{i,1}, a_{i,2} \in [0; 5)$  или  $a_{i,1}, a_{i,2} \in [0; 50) \forall i = \overline{1, N}$ . Весовые коэффициенты равномерно распределены:  $w_i \in [1, 10) \forall i = \overline{1, N}$  либо равны 1 для всего набора данных (равные весовые коэффициенты). Для каждой комбинации параметров (число точек-потребителей  $N$ , распределение координат  $A_i$ , распределение весовых коэффициентов  $w_i$ ) было сгенерировано по 10 наборов данных. Параметры сгенерированных наборов данных приведены в Таблице 5.5. Полностью наборы данных доступны в сети Интернет<sup>1</sup>.

Было произведено сравнение результатов с результатами популярных алгоритмов: метода спуска Нелдера-Мида [500], Пауэлла [501] и нелинейного алгоритма сопряженных градиентов (СГ) [511]. Для всех алгоритмов было использовано одинаковое значение допуска  $\varepsilon = 0,00001$ . Усредненные результаты представлены в Табл. 5.5. В Табл. 5.6 представлены некоторые характерные примеры решений для наборов данных с различными параметрами.

Наилучшие (по точности) в общем случае дает метод Нелдера-Мида. Метод Пауэлла быстрее, однако он намного менее точен. Алгоритм СГ во многих случаях позволяет получить весьма точные решения (см. Таблицу 5.6). Тем не менее, если результат  $X^*$  расположен на окружности  $\|A_i - X^*\|_2 = 1$  (такие результаты обозначены буквой "b" в колонке 5 таблицы результатов), то этот алгоритм обладает крайне медленной сходимостью и требует очень большого количества оценок целевой функции и ее производных.

Алгоритм, предлагаемый здесь («новый» алгоритм в таблице) является самым быстрым. Его результаты менее точны, чем результаты метода Нелдера-Мида, но новый алгоритм более точен, чем метод Пауэлла и алгоритм СГ. Если результат лежит в точке пересечения окружностей (обозначение "i" в колонке 5 таблицы результатов), то новый алгоритм дает точные результаты в отличие от остальных методов.

---

<sup>1</sup> <http://dl.dropbox.com/u/20889097/weber1data.zip>

Таблица 5.5 – Сравнение результатов работы Алгоритма 5.12 и известных алгоритмов

Номера наборов данных	Число точек-потребителей $N$	Распределение точек-потребителей $A_i$ , и весов $w_i$	Метод	Усредненная ошибка (сравнительная, в % к лучшему результату)	Среднее время вработки, ус.	Среднее число оценок целевой функции и ее производных
1–10	10	$a_{i,1}, a_{i_2} \in [0; 5), w_i \in [1; 10)$	Нелдер-Мид	0.0000000137	0.209-5	180.1
			Пауэлл	0.0000104146	0.06600	68.2
			СГ	0.0000053567	2.80023	1748
			Новый алг.	0.0000001202	0.02994	31.6
11–20	10	$a_{i,1}, a_{i_2} \in [0; 50), w_i \in [1; 10)$	Нелдер-Мид	0.0000000151	0.23022	179.9
			Пауэлл	0.0000020391	0.07229	66.6
			СГ	0.0000058669	3.07406	1697.1
			Новый алг.	0.0000001322	0.03278	28.1
21–30	10	$a_{i,1}, a_{i_2} \in [0; 5), w_i = 1$	Нелдер-Мид	0.0000000166	0.25282	179.2
			Пауэлл	0.0000020756	0.07935	66.1
			СГ	0.0000064536	3.38138	1863.6
			Новый алг.	0.0000001455	0.03593	25.6
31–40	100	$a_{i,1}, a_{i_2} \in [0; 5), w_i \in [1; 10)$	Нелдер-Мид	0.0000000079	0.27405	178.3
			Пауэлл	0.0000015044	0.08595	66.4
			СГ	0.0000070588	3.68216	1895.2
			Новый алг.	0.0000001601	0.03899	25.6
41–50	100	$a_{i,1}, a_{i_2} \in [0; 50), w_i \in [1; 10)$	Нелдер-Мид	0.0000000087	0.30146	196.2
			Пауэлл	0.0000016549	0.09454	73
			СГ	0.0000077647	4.05038	2084.7
			Новый алг.	0.0000001760	0.04289	28,2
51–60	100	$a_{i,1}, a_{i_2} \in [0; 50), w_i \in [1; 10)$	Нелдер-Мид	0.0000000062	0.32683	197.1
			Пауэлл	0.0000018203	0.10103	66
			СГ	0.0000084797	4.37678	1993.9
			Новый алг.	0.0000001016	0.04676	28.9
61–70	1000	$a_{i,1}, a_{i_2} \in [0; 5), w_i \in [1; 10)$	Нелдер-Мид	0.0000000068	0.31815	181.2
			Пауэлл	0.0000019977	0.09706	61.7
			СГ	0.0000088688	4.18989	1938.6
			Новый алг.	0.0000000862	0.04752	27
71–80	1000	$a_{i,1}, a_{i_2} \in [0; 5), w_i = 1$	Нелдер-Мид	0.0000000075	0.31691	184
			Пауэлл	0.0000021581	0.09597	62.9
			СГ	0.0000097557	4.60229	2129.7
			Новый алг.	0.0000000948	0.04672	27.1
81–90	1000	$a_{i,1}, a_{i_2} \in [0; 50), w_i \in [1; 10)$	Нелдер-Мид	0.0000000082	0.31355	186.2
			Пауэлл	0.0000023720	0.09160	62.8
			СГ	0.0000059605	4.01122	1909.7
			Новый алг.	0.0000001017	0.04877	28.6
91–100	10000	$a_{i,1}, a_{i_2} \in [0; 50), w_i \in [1; 10)$	Нелдер-Мид	0.000000010	0.28625	184.9
			Пауэлл	0.000002812	0.08802	65.7
			СГ	0.000007383	3.84634	1925.3
			Новый алг.	0.000000121	0.04178	28.9
101–110	10000	$a_{i,1}, a_{i_2} \in [0; 5), w_i = 1$	Нелдер-Мид	0	0.34718	161.1
			Пауэлл	0.0000003826	0.11746	54.1
			СГ	0.0000159116	5.91964	2455.6
			Новый алг.	0.0000000646	0.03677	17.3

Таблица 5.6 – Результаты работы Алгоритма 5.12 и известных алгоритмов

Номер набора данных	Число точек требования $N$	Распределение координат точек требования $A_i$ , весов $w_i$	Метод	Результат $f_1(X^*)$ , тип результата $X^*$	Ошибка (относительно лучшего результата), В %	Время, с.	Количество произведенных оценок целевой функции и ее производных
4	10	$a_{i,1}, a_{i,2} \in [0; 5)$ , $w_i \in [1; 10)$	Нелдер-Мид	108.8540	0	0.000525	181
			Пауэлл	108.8623	0.007582	0.000525	182
			СГ	108.8581	0.003786	0.0211	7077
			Новый алг.	108.8540    b	0	8.75e-05	36
13	10	$a_{i,1}, a_{i,2} \in [0; 50)$ , $w_i \in [1; 10)$	Нелдер-Мид	1120.0664	0	0.00055	190
			Пауэлл	1120.0667	0.000024	0.000175	52
			СГ	1120.0664	0	7.5e-05	26
			Новый алг.	1120.0664    e	0	0.0002	77
37	100	$a_{i,1}, a_{i,2} \in [0; 5)$ , $w_i \in [1; 10)$	Нелдер-Мид	1065.6038	7e-7	0.00408	182
			Пауэлл	1065.6071	0.00031	0.00217	97
			СГ	1065.6053	0,00015	0.17053	6485
			Новый алг.	1065.6037    i	0	0.00053	26
52	100	$a_{i,1}, a_{i,2} \in [0; 5)$ , $w_i = 1$	Нелдер-Мид	193.05004	5.2e-10	0,004	182
			Пауэлл	193.05023	0.00009	0.0032	84
			СГ	193.05005	2.5e-7	0.06193	2257
			Новый алг.	193.05004    b	0	0.0016	67
49	100	$a_{i,1}, a_{i,2} \in [0; 50)$ , $w_i \in [1; 10)$	Нелдер-Мид	10284.211	0	0.0042	187
			Пауэлл	10284.211	1.7e-6	0.00163	72
			СГ	10284.211	0	0.00085	32
			Новый алг.	10284.211    g	0	0.00128	53
71	1000	$a_{i,1}, a_{i,2} \in [0; 5)$ , $w_i = 1$	Нелдер-Мид	1925.2776	3.4e-8	0.0477	187
			Пауэлл	1925.2776	0	0.02968	143
			СГ	1925.2776	6.1e-6	0.78645	2993
			Новый алг.	1925.2776    b	9.2e-7	0.0042	21
90	1000	$a_{i,1}, a_{i,2} \in [0; 50)$ , $w_i \in [1; 10)$	Нелдер-Мид	107319.71	0	0.04015	183
			Пауэлл	107319.71	4.7e-9	0.01083	50
			СГ	107319.71	0	0.00973	39
			Новый алг.	107319.71    g	0	0.0089	37
93	10000	$a_{i,1}, a_{i,2} \in [0; 50)$ , $w_i \in [1; 10)$	Нелдер-Мид	1059221.9	0	0.4277	188
			Пауэлл	1059221.9	0	0.10833	48
			СГ	1059221.9	0	0.19378	79
			Новый алг.	1059221.9    g	0	0.08955	35
116	10000	$a_{i,1}, a_{i,2} \in [0; 5)$ , $w_i = 1$	Нелдер-Мид	19585.035	0	0.35053	162
			Пауэлл	19585.035	1.8e-8	0.1397	64
			СГ	19585.045	0.00005	10.513025	4330
			Новый алг.	19585.035    b	2.6e-8	0.02621	13

Примечание к таблице 5.6: тип результата: e – результат совпадает с решением задачи Вебера с евклидовыми расстояниями ( $\exists i \in \overline{1, N}: \|A_i - X^*\|_2 \leq 1$ ),

b – результат лежит на окружности ( $\exists i \in \overline{1, N}: \|A_i - X^*\|_2 = 1$ ),

i – результат совпадает с точкой пересечения окружностей ( $\exists i, j \in \overline{1, N}, i \neq j: \|A_i - X^*\|_2 = 1, \|A_j - X^*\|_2 = 1$ ),

g – иной результат ( $\exists i \in \overline{1, N}: \|A_i - X^*\|_2 = 1$ ).

### 5.14 Задача Вебера на области, ограниченной дугами

Отметим, что целевая функция задачи Вебера с мерой расстояния, включающей минимальную стоимость транспортировки (5.65)-(5.66), рассмотренной в предыдущем параграфе, является дифференцируемой в областях, ограниченных выражениями

$$\|X - A_i\|_2 \leq 1 \forall i \in S_<, \quad (5.73)$$

$$\|X - A_i\|_2 \geq 1 \forall i \in S_>, \quad (5.74)$$

где  $S_<, S_> \in \{\overline{1, N}\}$ ,  $S_< \cap S_> = \emptyset$  - некие подмножества индексов точек-потребителей  $A_i$ . В пределах каждой из этих областей целевая функция является целевой функцией классической задачи Вебера, из которой исключены часть исходных точек-потребителей. В настоящем параграфе приводится алгоритм на основе процедуры Вайсфелда [144, 148] для задачи Вебера с ограничениями (5.73)-(5.74).

Следует также отметить, что для задач Вебера с запрещенными зонами, барьерами и т.д. рассмотрены лишь частные случаи. Так, в [512, 509, 513] авторы предлагают методы решения задач Вебера с ограничениями и минимаксных задач размещения. Алгоритмы для задач Вебера на разрешенной области в виде многоугольника или многогранника предложены в [512, 514] and many other papers. Многие частные случаи рассмотрены, в том числе, в работах Кламрот [419, 420, 143, 416, 160], Забудского [153, 513, 515]. Большинство подобных работ ограничиваются рассмотрением одной или двух запрещенных областей строго определенной формы, хорошо изученными можно считать лишь задачи размещения с прямоугольными запрещенными областями [516].

Простейшим, но достаточно эффективным алгоритмом для задач Вебера с метриками, основанными на  $l_p$ -нормах, является процедура Вайсфелда (см. Алгоритм 5.11). В качестве условия останова может быть выбрано выполнение определенного числа итераций или достижение

Допустимая область, заданная ограничениями (5.73) и (5.74) в общем случае не выпукла, целевая функция  $f(X)$  задачи Вебера выпукла [514].

Для задач условной оптимизации с выпуклыми целевыми функциями решение совпадает с решением соответствующей задачи безусловной оптимизации либо лежит на границе запрещенной зоны [509] (более того, решение задачи условной оптимизации является «видимым» из точки, являющейся решением задачи безусловной

оптимизации). Если  $X^*$  - решение задачи (5.2) с евклидовой метрикой и ограничениями (5.73) и (5.74), то оно же является решением безусловной задачи (5.2), либо же  $\exists i \in \overline{\{1, N\}}: \|A_i - X^*\|_2 = 1$ .

Способ выбора начального решения процедуры Вайсфелда (Алгоритм 5.11) может быть любым, но выбор точки-потребителя при этом нежелателен из-за недифференцируемости целевой функции [504, 293]. Выбор точки со средневзвешенным значением по каждой из координат – пожалуй, наиболее популярный подход.

Шаг 2 Алгоритма 5.11 может привести в выбору нового промежуточного решения за пределами допустимой области  $R_f$ , определяемой ограничениями (5.73)-(5.74). Условимся, что  $R_f \neq \emptyset$ .

Для произвольной точки  $X \in \mathbb{R}^2$  обозначим ближайшую к ней точку в  $R_f$ :

$$C(X) = \arg \min_{X' \in R_f} \|X - X'\|_2 = \begin{cases} X, & X \in R_f, \\ \arg \min_{X' \in R_f} \|X - X'\|_2, & X \notin R_f. \end{cases}$$

Сходимость процедуры Вайсфелда доказана для норм  $l_p$  [114], где  $p \in [1, 2]$ . Несмотря на медленную сходимость в особых случаях [504, 293], эта процедура градиентного спуска, предложенная 16-летним математиком задолго до появления самого понятия «градиентный спуск», долго остававшаяся неизвестной для широких кругов математиков и несколько раз открытая заново, остается весьма эффективной для задач безусловной оптимизации.

Предлагаемый алгоритм для нашей задачи основывается на следующей гипотезе.

Гипотеза 5.24. Если Шаг 2 Алгоритма 5.11 выбирает новое промежуточное решение  $X' \notin R_f$ , то, добавив к Шагу 2 Алгоритма 5.11 замену  $X'$  этого решения на  $C(X')$ , мы получим алгоритм для задачи условной оптимизации с ограничениями (5.73)-(5.74).

Обозначим множества точек

$$R_{Ai} = \begin{cases} \{X \in \mathbb{R}^2 \mid \|X - A_i\|_2 \leq 1\}, & i \in S_< \\ \{X \in \mathbb{R}^2 \mid \|X - A_i\|_2 \geq 1\}, & i \in S_> \end{cases} \quad \forall i \in S_< \cup S_>.$$

Очевидно, что,  $R_f = \bigcap_{i \in (S_< \cup S_>)} R_{Ai}$ .

Для заданного  $i \in S_< \cup S_>$  обозначим через  $C_{Ai}(X)$  множества  $R_{Ai}$ , ближайшую к заданной точке  $X$ .

Если  $X^{**} \notin R_{A_i}$ ,  $i \in S_{<} \cup S_{>}$ ,  $X^{**} \neq A_i$ , то точка  $C_{A_i}(X^{**})$ , ближайшая к  $X^{**}$  является точкой пересечения окружности с центром в  $A_i$  с отрезком, соединяющим  $A_i$  и  $X^{**}$  [508].

Для заданного  $i \in (S_{<} \cup S_{>})$  обозначим функцию на  $\mathbb{R}^2$

$$G_{A_i}(X) = \begin{cases} 0, & X \in R_{A_i} \\ \|X - C_{A_i}(X)\|_2, & X \notin R_{A_i} \end{cases} \quad (5.75)$$

Выберем два индекса  $i, j \in S_{<} \cup S_{>}$ :  $R_{A_i} \cap R_{A_j} \neq \emptyset$ ,  $A_i \neq A_j$  и произвольную точку  $X' \notin R_{A_i} \cap R_{A_j}$ . Точка  $C_{A_i, A_j}(X') \in R_{A_i} \cap R_{A_j}$ , ближайшая к точке  $X'$ , совпадает с  $C_{A_i}(X')$ ,  $C_{A_j}(X')$ , либо с точкой пересечения двух окружностей радиуса 1 с центрами в  $A_i$  и  $A_j$ .

*Доказательство.* Поскольку функция  $f_R(X) = \|X - X'\|$  выпукла, и  $X' \notin R_{A_i} \cap R_{A_j}$ , ее минимум в области  $R_{A_i} \cap R_{A_j}$  находится на границе области  $R_{A_i} \cap R_{A_j}$ , которая состоит из двух дуг окружностей с центрами  $A_i$  и  $A_j$ , соединенных в двух точках пересечения окружностей. Обозначим эти точки пересечения  $I_{i,j}$  и  $I'_{i,j}$ .

Если  $C_{A_i, A_j}(X')$  принадлежит окружности с центром  $A_i$  ( $\|X' - A_i\| = 1$ ), и  $C_{A_i, A_j}(X') \notin \{I_{i,j}, I'_{i,j}\}$ , то  $C_{A_i, A_j}(X') = C_{A_i}(X')$ . Аналогично, если  $C_{A_i, A_j}(X')$  находится на окружности с центром  $A_j$ , то  $C_{A_i, A_j}(X') \notin \{I_{i,j}, I'_{i,j}\}$ , и  $C_{A_i, A_j}(X') = C_{A_j}(X')$ .

Обозначим штрафную функцию на  $\mathbb{R}^2$ :

$$G(X) = \max_{i \in S_{<} \cup S_{>}} G_{A_i}(X). \quad (5.76)$$

Очевидно,  $G(X) \geq 0 \forall X \in \mathbb{R}^2$ , и  $G(X) = 0 \forall X \in R_f$ .

Из выпуклости каждой из функций  $G_{A_i}(\cdot)$ , следует выпуклость функции  $G(\cdot)$ .

Таким образом, задача поиска точки, удовлетворяющей  $X' \in R_f$ , сводится к задаче минимизации  $G(X)$ .

Основываясь на Гипотезе 5.24, предлагается следующий алгоритм.

Алгоритм 5.13. Модифицированная процедура Вайсфелда для задачи Вебера с ограничениями (5.73)-(5.74).

Дано: Координаты и весовые коэффициенты  $A_i = (a_1^i, a_2^i)$ ,  $w_i$ ,  $i = \overline{1, N}$ , требуемая точность  $\varepsilon$ , ограничения (5.73) и (5.74), заданные множествами  $S_{<}$  и  $S_{>}$ , начальное решение  $X^*$ .

Шаг 1: Присвоить  $\Delta = +\infty$ .

Шаг 2: пока  $\Delta > \varepsilon$  выполнять:



Шаг 2.1: Присвоить  $n_{iter} = n_{iter} + 1$ ;  $d_{denom} = \sum_{i=1}^N w_i / \|A_i - X^*\|_2$ .

Шаг 2.2: Присвоить  $x_r^{**} = \sum_{i=1}^N \frac{x_r^* w_i}{\|X^* - A_i\|_2 \cdot d_{denom}} \forall r \in \{1, 2\}$ ;  $X^{**} = (x_1^{**}, x_2^{**})$ .

Шаг 2.3: если  $G(X^{**}) > 0$ , то присвоить  $X^{***} = C(X^{**})$ , иначе  $X^{***} = X^{**}$ .

Шаг 2.4: Присвоить  $\Delta = \|X^* - X^{***}\|$ ;  $X^* = X^{***}$ .

Шаг 2.5: Продолжит цикл 2.

Шаг 3: ОСТАНОВ, возврат значения  $X^{**}$ .

Для данного алгоритма мы должны определить две вспомогательные процедуры – поиска начального решения  $X^* \in R_f$  и поиска ближайшей точки  $C(X^{**}) \in R_f$ .

Мы используем начальную допустимую точку ближайшую к средней.

Алгоритм 5.14. Вычисление начальной точки  $X^* \in R_f$ .

Шаг 1: Присвоить  $x_r = \frac{\sum_{i=1}^N w_i a_r^i}{\sum_{i=1}^N w_i} \forall r \in \{1, 2\}$ . Обозначим  $X = (x_1, x_2)$ .

Шаг 2: Присвоить  $X^* = C(X)$ .

Шаг 3: ОСТАНОВ, возврат  $X^*$ .

Алгоритм 5.15. Вычисление  $C(X)$  (Шаг 2 Алгоритма 5.14 and Шаг 2.3 Алгоритма 5.14).

Шаг 1: Вычисляем  $g_i = G_{A_i}(X) \forall i \in S_{<} \cup S_{>}$ ;  $G = \sum_{i \in S_{<} \cup S_{>}} g_i$ .

Шаг 2: Пока  $G > 0$  выполнять:

Шаг 2.1: Выбрать два индекса  $i' = \arg \max_{i \in S_{<} \cup S_{>}} g_i$ ;  $i'' = \arg \max_{i \in (S_{<} \cup S_{>}) \setminus \{i'\}} g_i$ .

Шаг 2.2: Если  $g_{i''} = 0$ , то присвоить  $X = C_{A_{i'}}(X)$ ; Вычислить  $g_i = G_{A_i}(X) \forall i \in S_{<} \cup S_{>}$ ;  $G = \sum_{i \in S_{<} \cup S_{>}} g_i$ ; Продолжить цикл 2.

Шаг 2.3: Вычислить координаты двух точек :  $I_1, I_2$  пересечения окружностей с центрами  $A_{i'}$  и  $A_{i''}$ :  $I_1, I_2 = P_{intersect}(A_{i'}, A_{i''})$ ; Вычислить  $C_1 = C_{A_{i'}}(X)$ ;  $C_2 = C_{A_{i''}}(X)$ .

Шаг 2.4: Присвоить  $X = \arg \min_{X \in \{C_1, C_2, I_1, I_2\}} G(X)$ ;  $G = G(X)$ .

Шаг 2.5: Продолжить цикл 2.

Шаг 3: ОСТАНОВ, возврат значения  $X$ .

Здесь мы используем выражения (5.75) и (5.76). Для определения  $C_{A_i}(X)$  (точки пересечения отрезка, соединяющего  $X$  и  $A_i$  и окружности с центром  $A_i$ ) для заданных  $X' = (x'_1, x'_2)$  и  $A_i = (a_1^i, a_2^i)$  мы используем формулы.

$$b = \frac{x'_1 - a_1^i}{x'_2 - a_2^i}, a = a_1^i - ba_2^i.$$

Из уравнения

$$(x_1 - a_1^i)^2 + (x_2 - a_2^i)^2 = 1$$

окружности с центром  $A_i$  имеем уравнение

$$(1 + b^2)x_2^2 + (2b(a - a_1^i) - 2a_2^i)x_2 + (a_2^i)^2 + (a - a_1^i)^2 - 1 = 0.$$

Решив его, предположив, что  $C_{A_i}(X') = (c_1, c_2)$ , получаем:

$$\begin{aligned} c'_2 &= -(2b(a - a_1^i) - 2a_2^i) + D, c''_2 = -(2b(a - a_1^i) - 2a_2^i) + D, \\ c'_1 &= a_1^i + \frac{(c'_2 - a_2^i)(x'_1 - a_2^i)}{x'_2 - a_2^i} + a_1^i, c''_1 = a_1^i + \frac{(c''_2 - a_2^i)(x'_1 - a_2^i)}{x'_2 - a_2^i} + a_1^i, \\ (c_1, c_2) &= \arg \min_{X \in \{(c'_1, c'_2), (c''_1, c''_2)\}} \|X - X'\|, \end{aligned} \quad (5.77)$$

$$\text{где } D = \sqrt{(2b(a - a_1^i) - 2a_2^i)^2 - 4(1 + b^2)(a_2^i)^2 + (a - a_1^i)^2 - 1}.$$

При этом следует учитывать два особых случая. Если  $X' = A_i$ , расстояния от  $X'$  до всех точек окружности с центром  $X'$  равны, в качестве точки с минимальным расстоянием можно использовать любую. Мы будем использовать  $(c_1, c_2) = (x'_1, x'_2 - 1)$ .

В случае, если  $a_2^i = x'_2$ , мы будем пользоваться формулами

$$\begin{aligned} c'_1 &= c''_1 = x'_1, \\ c'_2 &= x'_2 + 1, c''_2 = x'_2 - 1 \end{aligned}$$

при этом для выбора  $(c_1, c_2)$  будем использовать (5.77).

Рассмотрим задачу на рис. 5.15, Случай 1. Здесь  $A_1 = (0, 0.75)$ ,  $w_1 = 3$ ,  $A_2 = (0.3, 0.5)$ ,  $w_2 = 2$ ,  $A_3 = (0.6, 0.5)$ ,  $w_3 = 3$ ,  $A_4 = (1, 2)$ ,  $w_4 = 6$ . Допустимая область  $R_f$  ограничена четырьмя неравенствами:  $\|X - A_1\|_2 \leq 1$ ,  $\|X - A_2\|_2 \geq 1$ ,  $\|X - A_3\|_2 \geq 1$ ,  $\|X - A_4\|_2 \leq 1$ . То есть  $S_< = \{1, 4\}$ ,  $S_> = \{2, 3\}$ .

Данная начальная точка  $X^*$  лежит вне  $R_f$ . На Шаге 1 Алгоритма 5.14 имеем  $g_1 = 0$ ,  $g_4 = 0$ ,  $g_2 = 0.241704046$ ,  $g_3 = 0.30357143$ ,  $G = 0.545275476 > 0$ . Таким образом, Шаг 2 Алгоритма 5.14 в данном случае выполняется.

На Шаге 2.1 Алгоритма 5.14 имеем  $i' = 3$ ,  $i'' = 2$ . На Шаге 2.2,  $g_{i''} = g_2 \neq 0$ .

На Шаге 2.3 точки пересечения окружностей с центрами  $A_2$  и  $A_3$  равны  $I_1 = (0.45, 1.4886856)$  и  $I_2 = (0.45, -0.4886856)$ . Точками, ближайшими к этим

окружностям, являются  $C_1 = C_{A3} = (0.6, 1.5)$  и  $C_2 = C_{A2} = (0.6956239, 1.41841261)$ .

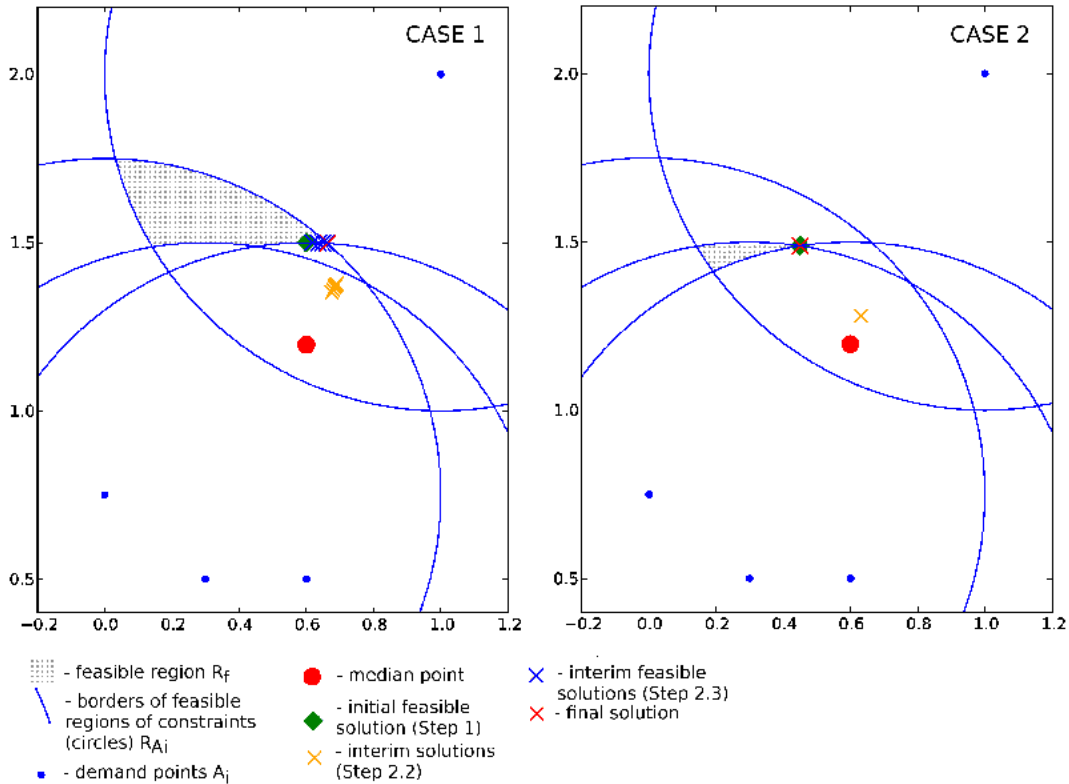


Рисунок 5.15 – Примеры задач на областях, ограниченных дугами, и ход решения

Решим ее с точностью  $\varepsilon = 0.00001$ .

Начальное решение (Шаг 1 Алгоритма 5.14) равно

$$X^{**} = \frac{\sum_{i=1}^4 A_i w_i}{\sum_{i=1}^4 w_i} = (0.6, 1.19642857).$$

На Шаге 2.4 Алгоритма 5.14 имеем  $G(C_1) = 0$ ,  $G(C_2) = 0.07662269$ ,  $G(I_1) = 0$ ,  $G(I_2) = 1.86662994$ . Таким образом, новое значение  $X = C_1 = (0.6, 1.5)$ . Поскольку  $G = 0$  это значение  $X$  возвращается Алгоритмом 5.14 в качестве начального значения  $X^* = (0.6, 1.5)$  Алгоритма 5.13.

На Шагах 2.1 и 2.2 Алгоритма 5.13 имеем  $d_{denom} = 17.4095535$ ,  $X^{**} = (0.67463657, 1.35220493)$ . Данная точка лежит вне  $R_f$ .

На Шагах 2.3 и 2.4, после применения Алгоритма 5.15 имеем  $X^{***} = (0.61937139, 1.49981236)$ ,  $\Delta = 0.01628994$ .

Промежуточные результаты дальнейших итераций приведены в Таблице 5.7

Таблица 5.7: Промежуточные результаты Алгоритма 5.13

Номер итерации	$X^*$	$d_{denom}$	$X^{**}$	$\Delta$
1	(0.6, 1.5)	17.4095535	(0.67463657, 1.35220493)	0.01937229
2	(0.619371390, 1.49981236)	17.5361094	(0.67960047, 1.36051086)	0.01628994
3	(0.63565516, 1.49936415)	17.64033129	(0.68362543, 1.367251455)	0.01360489
4	(0.64924779, 1.49878659)	17.7251354	(0.68687416, 1.37269171)	0.01129761
5	(0.66052837, 1.49816648)	17.79357911	(0.689489, 1.37706744)	0.003214
6	(0.66373616, 1.49796678)	17.81267195	(0.69021857, 1.3782875)	0

Задача решена за 6 итераций.

Задача на рис.5.15, случай 2 решается за одну итерацию. Здесь  $\|X - A_1\|_2 \leq 1$ ,  $\|X - A_2\|_2 \leq 1$ ,  $\|X - A_3\|_2 \geq 1$ ,  $\|X - A_4\|_2 \leq 1$ ,  $S_< = \{1,2,4\}$ ,  $S_> = \{3\}$ .

В этом случае начальное решение, возвращаемое Алгоритмом 5.14 и окончательное решение  $X^{***} = (0.45000325, 1.4886855)$  совпадают.

Поскольку состоятельность Гипотезы 5.24 аналитически не доказана, наш алгоритм – эвристический. Для экспериментального доказательства применимости алгоритма решим автоматически сгенерированные задачи двумя способами. Тестовые задачи создадим следующим алгоритмом.

#### Алгоритм 5.16. Генерация тестовых примеров

Дано: Число точек-потребителей  $N$ .

Шаг 1: Присвоить  $S_< = \emptyset$ ;  $S_> = \emptyset$ .

Шаг 2: для каждого  $i \in \overline{\{1, N\}}$  выполнять:

Шаг 2.1:  $x_1 = 4 \cdot \text{Random}()$ ;  $x_2 = 4 \cdot \text{Random}()$ ;  $X = (x_1, x_2)$ ; если  $\exists A \in \{A_1, \dots, A_{i-1}\}$ :  $\|X - A\|_2 < 0.1$ , то повторить Шаг 2.1.

Шаг 2.2:  $A_i = (x_1, x_2)$ ;  $w_i = 9 \cdot \text{Random}() + 1$ .

Шаг 2.3:  $r = \text{Random}()$ ; если  $r < 0.5$ , то  $S_< = S_< \cup \{i\}$ , иначе  $S_> = S_> \cup \{i\}$ .

Шаг 2.4: Продолжить цикл 2.

Шаг 3: Выбрать точку  $X = (4 \cdot \text{Random}(), 4 \cdot \text{Random}())$ ; Выполнить Алгоритм 5.15 для вычисления  $X_r = C(X)$ ; если  $X_r \in R_f$  не получено (т.е. допустимая область

пуста), удалить один случайным образом выбранный элемент из  $S_<$  и повторить Шаг 3.

Шаг 4: Останов, возврат  $\{A_i\}, \{w_i\}, S_<, S_>$ .

Алгоритм 5.15 работает, если допустимая область не пуста, иначе он выходит в бесконечный цикл. Во избежание этого, здесь используется модифицированная версия Алгоритма 5.15 с дополнительным шагом:

Шаг 2.3а: Если  $\min_{X \in \{C_1, C_2, J_1, J_2\}} G(X) \geq G$ , то ОСТАНОВ, возврат без результата.

Все задачи были решены Алгоритмом 5.13.

Целевая функция задачи Вебера - липшицева [227]. Для получения липшицевой константы  $\lambda$  мы использовали следующее неравенство [517]:

$$\begin{aligned} \lambda \|X - X'\| &\geq |f(X) - f(X')| = \left| \sum_{i=1}^N w_i \|A_i - X\| - \sum_{i=1}^N w_i \|A_i - X'\| \right| \\ &\leq \sum_{i=1}^N w_i \left| \|A_i - X\| - \|A_i - X'\| \right|. \end{aligned} \quad (5.78)$$

Из правила треугольника имеем:

$$\sum_{i=1}^N w_i \left| \|A_i - X\| - \|A_i - X'\| \right| \leq \sum_{i=1}^N w_i \|X' - X\|.$$

Таким образом,  $\lambda \|X - X'\| \geq \sum_{i=1}^N w_i \|X' - X\|$ .

С константой  $\lambda = \sum_{i=1}^N w_i$  неравенство (5.78) верно.

В тестовых задачах (см. Алгоритм 5.16) справедливо  $1 < w_i < 10$ . Следовательно,  $N < \lambda < 10 \cdot N$ .

Для задач выпуклой оптимизации медленным, но эффективным является метод поиска (полного перебора) вариантов решений на сетке с шагом  $s$  вдоль каждой из координат (также известен как метод сэмпинга [518]). При заданном значении шага  $s$  при этом мы получим точность  $\lambda s = 10 \cdot N \cdot s$ . Поскольку координаты в тестовых задачах находятся в интервале  $[0,4]$ , оценка целевой функции в  $n$  значений каждой координаты даст точность  $40 \cdot N/n$ .

Сгенерировав 30 задач с  $N = 3,4,5,7,10,15,20$  и решив их как Алгоритмом 5.13, так и методом сэмпинга при  $n = 1000$ ) везде получаем различия в результатах в

пределах  $\lfloor 40 \cdot \frac{N}{1000} \rfloor$ . При этом сэмплинг-метод требует 1000000 вычислений целевой функции, а Алгоритм 5.13 – не более 129 во всех тестовых задачах.

Таким образом, результатом исследования, изложенного в данном параграфе, является эвристический алгоритм, эффективный при решении практических задач, что продемонстрировано вычислительными экспериментами.

Таким образом, созданы алгоритмы для задачи Вебера с единственным размещаемым объектом для неевклидовых (и не- $l_p$ ) метрик, имеющих практическую значимость с точки зрения их применения в задачах размещения и автоматической группировки объектов в системах с особыми средствами транспортировки. Разработанные алгоритмы для задачи Вебера были использованы как часть процедуры локального поиска при решении задач автоматической группировки с соответствующими метриками (см. Таблицу 3.5 в Главе 3).

Для задач Вебера и множественных задач Вебера с метриками, основанными на угловых расстояниях, впервые предложены алгоритмы, позволяющие за полиномиальное время получать точное решение. Разработанный алгоритм, основанный на модифицированной процедуре Вайсфелда в комбинации с методом бисекций, для задачи Вебера с функцией расстояния  $L(X, Y) = \max\{\|X - Y\|_2, 1\}$  позволяет решать задачу быстрее известных методов, причем в определенных случаях достигается точный результат.

Разработанные алгоритмы, основанные на декомпозиции задач в задачи с прямоугольной метрикой, могут успешно использоваться как составная часть алгоритма решения соответствующей  $p$ -медианной задачи и применяться для решения логистических задач автоматической группировки объектов с целью снижения стоимости их транспортировки при использовании специальных средств транспортировки, в частности, манипуляторов с вращающейся стрелой.

## Выводы к Главе 5

В результате проведенных исследований созданы алгоритмы для задачи Вебера с единственным размещаемым объектом для неевклидовых (и не- $l_p$ ) метрик, имеющих практическую значимость с точки зрения их применения в задачах размещения и автоматической группировки с особыми средствами транспортировки. Разработанные

алгоритмы для задачи Вебера были использованы как часть процедуры локального поиска при решении задач автоматической группировки с соответствующими метриками (некоторые результаты решения задач автоматической группировки даны в Главе 3 в таблице 3.5).

Для задач Вебера и множественных задач Вебера с метриками, основанными на угловых расстояниях, впервые предложены алгоритмы, позволяющие за полиномиальное время получать точное решение. Разработанный алгоритм, основанный на модифицированной процедуре Вайсфелда в комбинации с методом бисекций, для задачи Вебера с мерой расстояния  $L(X, Y) = \max\{\|X - Y\|_2, 1\}$  позволяет решать задачу быстрее известных методов, причем в определенных случаях достигается точный результат.

Разработанные алгоритмы, основанные на декомпозиции задач в задачи с прямоугольной метрикой, могут успешно использоваться как составная часть алгоритма решения соответствующей  $p$ -медианной задачи и применяться для решения логистических задач автоматической группировки объектов с целью снижения стоимости их транспортировки при использовании специальных средств транспортировки, в частности, манипуляторов с вращающейся стрелой.

## ГЛАВА 6. ЗАДАЧИ АВТОМАТИЧЕСКОЙ ГРУППИРОВКИ И РАЗМЕЩЕНИЯ С ПРОИЗВОЛЬНОЙ МЕРОЙ РАССТОЯНИЯ

В настоящей главе рассмотрен метод решения задач автоматической группировки и множественного размещения (непрерывной  $p$ -медианной задачи) в наиболее сложных случаях, когда расстояния между группируемыми объектами описываются произвольной функцией, например, заданной алгоритмически. В данной главе для решения подобных задач предлагается алгоритм, основанный на идеях метода изменяющихся вероятностей [234] (МИВЕРА), аналогичный алгоритму для решения задач на сети, описанному во главе 2. Решение практической задачи предложенным методом проиллюстрировано на примере задачи об оптимальном размещении точек доступа беспроводной сети Ethernet 803.11(WiFi).

### 6.1 Постановка задач

Алгоритмы решения задач автоматической группировки и множественного размещения ( $k$ -средних,  $k$ -медоид,  $p$ -медианной задачи), рассмотренные в главе 3, предполагают наличие алгоритма решения соответствующей задачи Вебера—простейшей задачи размещения [225, 226], т.е. поиска точки  $X$ , такой, чтобы сумма взвешенных расстояний от нее до  $N$  известных точек  $A_1, \dots, A_N$  (т.н. точки-потребители или векторы данных) достигала минимума [484, 227].

$$F(X) = \sum_{i=1}^N w_i L(X, A_i) \rightarrow \min. \quad (6.1)$$

Здесь  $w_i$  - весовые коэффициенты известных точек,  $L(A, B)$  — расстояние между точками, рассчитанное каким-либо образом.

Функция  $L()$  может в принципе быть любой. При этом для решения задачи автоматической группировки алгоритмы главы 3 предполагают наличие некоторой процедуры нахождения центра каждой группы векторов данных (каждого кластера) — фактически требуется способ решения соответствующей задачи Вебера.

Задача Вебера (с любой метрикой или мерой расстояния), являясь обобщением задачи Ферма, в свою очередь имеет множество обобщений, например — множественная задача Вебера (с множеством размещаемых объектов):



$$F(X_1, X_2, \dots, X_M) = \sum_{i=1}^N \sum_{j=1}^p w_{ij} L(X_j, A_i) + \sum_{i=1}^p \sum_{j=1}^{i-1} v_{ij} L(X_i, X_j) \rightarrow \min.$$

Здесь  $w_{ij}$ ,  $v_{ij}$  – весовые коэффициенты (как правило, неотрицательные). Задача состоит в поиске  $p$  местоположений (центров) для размещения новых мощностей (в экономической интерпретации задачи)  $X_j$ ,  $j = \overline{1, p}$ .

Минисуммная задача ( $p$ -медианная,  $k$ -средних) имеет иную целевую функцию:

$$F(X_1, X_2, \dots, X_p) = \sum_{i=1}^N w_i \min_{j, 1 \leq j \leq p} L(X_j, A_i) \rightarrow \min. \quad (6.2)$$

Задача может иметь дополнительные условия — запрещенные зоны, барьеры. В случае наличия запрещенных зон появляется условие:

$$X_j \notin R_Z, \quad (6.3)$$

где  $R_Z$  — множество запрещенных местоположений.

В случае задачи с барьерами расстояние – длина кратчайшего пути – может складываться из длин нескольких сегментов (рис.6.1).

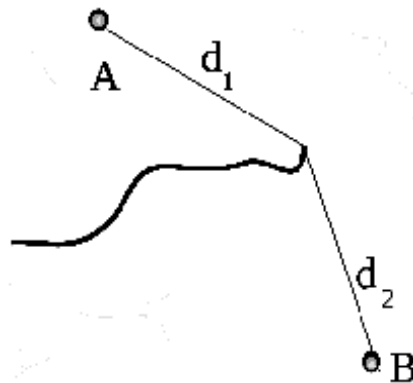


Рисунок 6.1 – Задача с барьерами

Здесь кратчайший путь состоит из отрезков, длины которых равны  $d_1$  и  $d_2$ .

Дальнейшее обобщение задачи — региональная задача, заключающаяся в поиске медианы для континуума точек-потребителей. В частности, рассматриваются версии непрерывной  $k$ -медианной задачи, где целью является нахождение одной или нескольких точек, таких, что взвешенное расстояние от них до множества точек «региона требования» минимальна. Если «регион требования» ограничен областью  $\Omega \subset R^n$ , то задача с одним размещаемым объектом [159] (региональная задача Вебера) заключается в поиске такой точки  $X$ , что

$$F(X) = \sum_{i=1}^N w_i \int_{\Omega} L(X, A_i) d\mu(A_i) \rightarrow \min,$$

где  $\mu(A_i)$  - математическое ожидание того, что  $i$ -й потребитель находится в некоторой точке «региона требования».

Если это ожидание одинаково для какой-либо области, т.е., если  $i$ -й потребитель единообразно распределен в области  $\Omega_i$ , то

$$F(X) = \sum_{i=1}^N w_i \int_{\Omega_i} L(X, A) \rho_i(A) dA \rightarrow \min,$$

где

$$\rho_i(A) = \begin{cases} 1, A \in \Omega_i \\ 0, A \notin \Omega_i. \end{cases}$$

В случае непрерывной (региональной) множественной задачи Вебера,

$$F(X_1, \dots, X_M) = \sum_{i=1}^N w_i \int_{\Omega_i} \min_{j, 1 \leq j \leq M} L(X_j, A) \rho_i(A) dA \rightarrow \min. \quad (6.4)$$

Здесь  $M$  – число искомых точек (центров),  $N$  – число «регионов требования».

Соответствующая задача автоматической группировки может быть сформулирована как задача разбиения некоторого множества  $\Omega = \bigcup_{i=1}^N \Omega_i \in R^n$ , каждому подмножеству  $\Omega_i$  которого поставлен в соответствие его «вес»  $w_i$ , на несколько непересекающихся подмножеств (регионов), для каждого из которых определен элемент, называемый центром  $X_i$ , таким образом, чтобы минимизировать значение функции (6.3). Отметим, что при известных центрах  $X_1, \dots, X_M$  задача нахождения соответствующих им подмножеств множества  $\Omega$  сводится к построению диаграммы Вороного [519, 520, 421].

Ниже предлагается алгоритм для приближенного решения задачи (6.4) с ограничениями (6.3), где функция расстояния  $L()$  — произвольная монотонная функция. В приведенном в данной главе примере используется известная функция потерь на трассе (затухания) для задач прогнозирования уровня приема радиосигнала. Результаты приведенного в настоящей главе исследования опубликованы в [156, 521-524, 286, 316, 317, 230, 229].

## 6.2 Существующие методы

Напомним, что  $p$ -медианная задача изначально имела дискретную постановку. В [217, 525] впервые была предложена постановка аналогичной задачи для нахождения медианы (точки на графе, минимизирующей сумму взвешенных расстояний до вершин) в сетях или на графах, доказана возможность дискретного представления непрерывной задачи размещения на сетях, обобщена задача абсолютной медианы как поиск  $p$  медиан на графе ( $p$ -медианы). Для этой задачи, относимой к первичным задачам дискретной теории размещения [526], предложено множество методов решения [527], в том числе – методы, представленные в Главах 2 и 3. Медианы ищутся в известном конечном множестве точек. В случае дискретных координат, полученных путем округления непрерывных координат, рассматривая все точки дискретного пространства как допустимые, мы имеем  $p$ -медианную задачу. Но размерность такой задачи чрезвычайно велика в случае достаточно точной дискретизации.

В Главе 5 дан подробный обзор используемых метрик и некоторых других функций расстояния. Задачи с применением функций расстояния, основанных на измененных нормах, рассмотрены в [428, 528]. Смешанные нормы исследованы в [487], в [461] нормы классифицируются на круговые и блочные, взвешенные суммы  $p$ -порядка в задачах размещения исследованы в [529, 440]. Асимптотическим расстояниям посвящена работа [462]. Безусловные задачи со смешанной координатной системой (дискретные и непрерывные) рассмотрены в [158]. В [160] приведены две эвристики для множественной задачи Вебера с барьерами, которые могут достичь результата определенной точности за ограниченное время. Обзор литературы позволяет прийти к выводу, что для обобщенной множественной задачи размещения с запрещенными зонами и барьерами рассмотрены лишь некоторые частные случаи.

Преобразовав непрерывную в задачу с дискретными координатами, мы получаем задачу комбинаторной оптимизации. Из подходов, гарантирующих точное решение задач дискретной оптимизации («рюкзачные», «задача коммивояжера» и др.), наиболее распространенные основаны на сведении задачи к задаче целочисленного линейного программирования, либо на ином применении метода ветвей и границ [234, 235, 530]. Большинство задач в этом случае относятся к классу  $NP$ -трудных и требуют поиска в дереве с экспоненциальным ростом числа ветвей, так что даже параллельные алгоритмы

для высокопроизводительных систем не позволяют решать задачи большой размерности за приемлемое время.

Изначально разработанный для решения безусловных задач псевдодволевой оптимизации, метод изменяющихся вероятностей (МИВЕР) — это метод случайного поиска, общая схема которого описана Алгоритмом 2.4 [234, 235]. При решении задачи большой размерности приходится оперировать миллионами переменных. Например, задача ассортиментного планирования розничной торговой компании [531] может включать выбор тысяч наименований товаров у сотен поставщиков с прогнозированием продаж при 2-10 вариантах устанавливаемых розничных цен для каждого товара. Подобные задачи успешно решаются методами случайного поиска. Для применения в задачах условной оптимизации метод должен быть изменен. Модификации, предложенные в следующих параграфах, а также в [531, 242], позволяют решать задачи размерности до миллионов переменных.

### 6.3 Постановка задачи в дискретных координатах

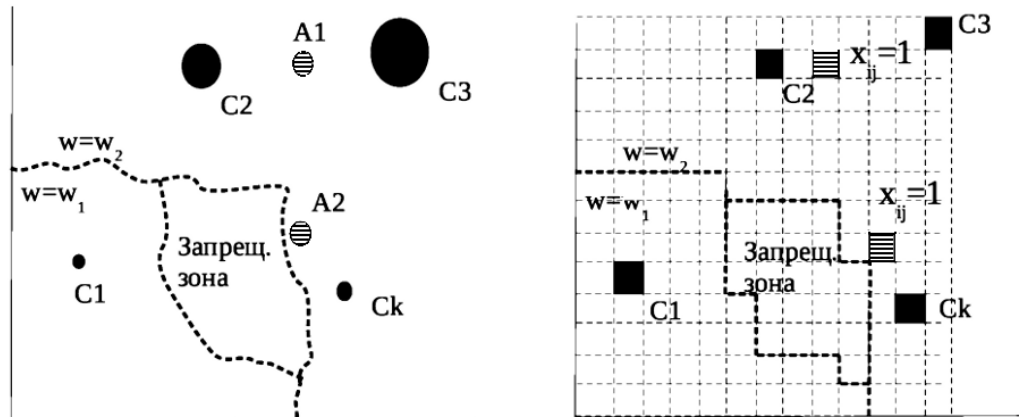
Рассмотрим задачу (6.4) с ограничениями (6.3) и с условием, что все координаты во входных данных задачи и в результате ее решения задаются дискретными значениями с определенным шагом дискретизации. Для практических задач такое приближенное решение приемлемо: точность измерения расстояний и разрешение карт всегда конечны, что превращает любую такую задачу в дискретную и на этапе сбора исходных данных, и на этапе реализации полученного решения. Превращение непрерывных координат в дискретную координатную сетку показано на рис.6.2. Область разбита на  $N_x$  столбцов и  $N_y$  строк так, что вся область представляет собой набор ячеек. В этом случае интеграл в формуле (6.4) региональной задачи превращается в сумму (6.2) обычной множественной задачи размещения. Задача состоит в выборе  $p$  ячеек для размещения новых объектов таких, что

$$F_1(X) = \sum_{k=1}^{N_x} \sum_{l=1}^{N_y} w_{ij} \min_{\substack{1 \leq i \leq N_x \\ 1 \leq j \leq N_y \\ x_{ij}=1}} x_{ij} L((i, j), (k, l)) \rightarrow \min. \quad (6.5)$$

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1N_y} \\ x_{21} & x_{22} & \dots & x_{2N_y} \\ \dots & \dots & \dots & \dots \\ x_{N_x1} & x_{N_x2} & \dots & x_{N_xN_y} \end{pmatrix} \quad (6.6)$$

с ограничениями

$$x_{ij} = 0 \forall (i, j) \in R_z; \quad \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} x_{ij} \leq N_F. \quad (6.7)$$



а) непрерывная система координат б) Дискретная система координат

Рисунок 6.2 – Переход к дискретным координатам

Здесь  $X$  — матрица булевых переменных,  $R_z$  — множество ячеек, запрещенных для размещения объектов (множество пар координат),  $L$  — функция расстояния, в общем случае произвольная, но монотонная. Если мы имеем задачу с барьерами, функция вычисляется по принципу, показанному на рис.6.1.  $w_{ij}$  — весовой коэффициент ячейки  $(i, j)$ ,  $N_F$  — число размещаемых объектов.

Задача может иметь другие ограничения.

В качестве примера задачи (6.5-6.7) рассмотрим задачу размещения антенн. В данной задаче мы имеем карту с дискретными координатами, каждой ячейке поставлен в соответствие весовой коэффициент — мера важности обеспечения уверенного приема в этом регионе от любой из антенн (вероятность нахождения пользователя в данной ячейке). Ячейки могут содержать различные виды препятствий прохождению сигнала (стены, деревья и т.п.). В качестве функции расстояния используем известную функцию потерь трассы [532], вычисляемую как

$$L((i, j)(k, l)) = 20 \log \|(i, j), (k, l)\| + L_{OBS}((i, j), (k, l)). \quad (6.8)$$

Здесь  $L_{OBST}$  — функция потерь трассы, вычисляемая как сумма потерь от прохождения через объекты (в зависимости от материала и толщины) по пути между ячейками  $(i,j)$  и  $(k,l)$ . Поглощающие свойства элементов среды известны из справочных таблиц [533] и могут быть уточнены, если единственное препятствие размещается между существующим передатчиком и приемником или в зоне Френеля [532]. Мы не учитываем потери, не зависящие от расположения антенн.

Мы имеем задачу условной псевдоболевой оптимизации (6.5-6.8) размерности  $N_x \times N_y$ . Даже в случае дискретизации исходной области до координатной сетки  $100 \times 100$  (грубое приближение), имеем задачу с 10000 переменными.

Отметим, что приведенная здесь постановка задачи в дискретных координатах отлична от задач Вебера в дискретных постановках [534, 535, 536, 537], близкие к задаче  $k$ -медоид, в которых предполагается размещение некоторого графа, причем множество возможных позиций (местоположений) его вершин изначально является конечным.

В случае задач большой размерности даже вычисление линейной целевой функции требует значительных вычислительных мощностей. Здесь функция задана алгоритмически. Таким образом, в случае применения МИВЕРа, требующего большого количества итераций, требуются значительные вычислительные мощности. Поэтому распределение вычислений между процессорами системы с общей памятью — актуальная задача.

Здесь мы не рассматриваем «жадные» («greedy») алгоритмы (которые также могут быть стохастическими или же детерминированными [538]), предполагая их использование для улучшения результата на завершающем этапе поиска. Также мы не рассматриваем параллельные генетические алгоритмы [320], хотя некоторые подходы могут быть применены и в случае параллельного МИВЕРа.

#### **6.4 Последовательная реализация алгоритма и его OpenMP-параллелизация**

Последовательная реализация алгоритма представлена на рис. 6.3, вариант 1.

На шаге инициализации все переменные  $p_i$  (компоненты матрицы вероятностей  $P$ ) устанавливаются в начальные значения ( $0 < p_i < 1$ ). Затем генерируются матрицы  $X$  булевых переменных целевой функции. Большие значения элементов матрицы  $P$  приводят к генерации экземпляров  $X$  за пределами области допустимых решений из-за

ограничений (6.7), благодаря которым оптимальное начальное значение элементов матрицы вероятностей не превосходит  $N_F/(N_x N_y)$  [531, 242].

Наш алгоритм основан на Алгоритме 6.1. Здесь вместо вектора вероятностей используется матрица  $P$ . Также алгоритм оперирует матрицей булевых переменных  $X$  вместо вектора. Это не меняет общей схемы алгоритма, но такое представление данных упрощает дальнейшее описание.

Вместо предельного числа шагов ( $M$  на рис. 6.3) можно использовать предельное время работы алгоритма в качестве условия останова. Также уместно использовать в качестве условия останова достижение предельного числа шагов, не дающих улучшения значения целевой функции.

В цикле ( $i=1:N$ ) генерируются  $N$  экземпляров матрицы  $X_{ki}$  в соответствии с распределениями, заданными элементами матрицы вероятностей  $P$ . После этого для каждого  $X_{ki}$  вычисляется значение целевой функции.

Чтобы принять во внимание условия (6.7) на этапе генерации экземпляров  $X$ , мы применяем измененную по сравнению с МИВЕРом процедуру генерации  $X$ .

Алгоритм 6.1 – алгоритм схемы МИВЕР для региональной задачи размещения

1.  $X_{SET} = \emptyset$ ;  $n=0$ ;
2. пока  $n < N_F$  цикл:
  - 2.1. для каждого  $i$  в интервале  $(1, N_x)$ :  $S_i = \sum_{j=1}^{N_y} p_{ij}$ ;
  - 2.2.  $r_x = \text{Random}()$ ;
  - 2.3.  $S_x = r_x \sum_{i=1}^{N_x} S_i$ ;
  - 2.4. ищем минимальное  $i$ , такое, что  $\sum_{k=1}^i S_k \geq S_x$ ;
  - 2.5.  $r_y = \text{Random}()$ ;
  - 2.6.  $S_y = r_y S_x$ ;
  - 2.7. ищем минимальное  $j$ , такое, что  $\sum_{l=1}^j p_{il} \geq S_y$ ;
  - 2.8. если  $(i, j) \in R_z$ , то переход к шагу 2.2
  - 2.9. иначе  $X_{SET} = X_{SET} \cup (i, j)$ ;  $n=n+1$ ; переход к шагу 2.

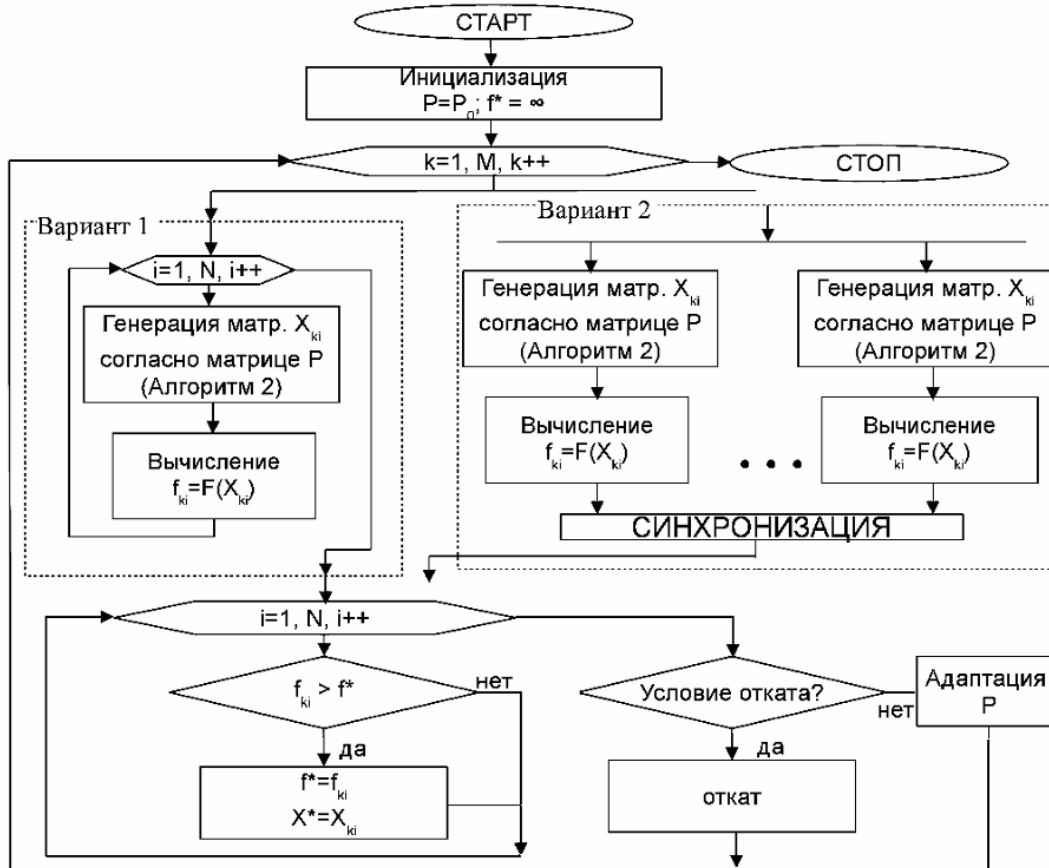


Рисунок 6.3 – Последовательный (вар.1) и параллельный (вар.2) алгоритмы

Здесь  $X_{SET}$  — множество координат (номера столбцов и строк координатной сетки) полученных матриц  $X$ , принимающие значения 0 или 1,  $N_F$  — число размещаемых объектов,  $Random()$  - функция, возвращающая в соответствии с равномерным распределением случайные значения в диапазоне  $[0, 1)$ .

В качестве результата Алгоритма 6.1 мы имеем матрицу  $X$  из множества допустимых решений.

Практика решения задач (численные эксперименты) показывает, что лучшие результаты достигаются при использовании мультипликативной адаптации элементов матрицы  $P$  и процедуры отката (возврата) [235]. В этом случае компоненты матрицы  $P$  никогда не достигают значений 0 или 1, что может приводить к генерации повторяющихся значений  $X$ .

В Алгоритме 2.4 все булевы переменные (элементы вектора  $X$ ) рассматриваются как независимые. После перехода от векторов к матрицам, при использовании мультипликативной адаптации значение элемента  $p_{ij}$  матрицы  $P$  на  $k$ -м шаге может быть вычислено как



$$p_{k,i,j} = \frac{p^{(k-1),i,j} \cdot d_{k,i,j}}{\sum_{l=1}^{N_x} \sum_{m=1}^{N_y} p^{(k-1),i,j} \cdot d_{k,i,j}} \sum_{l=1}^{N_x} \sum_{m=1}^{N_y} p^{(k-1),i,j} \cdot$$

Здесь  $d_{k,i,j}$  — коэффициент адаптации. В случае мультипликативной адаптации  $d_{k,i,j}$  не зависит от номера шага  $k$ , абсолютное значение шага адаптации зависит от текущего значения элемента  $p_{kj}$ .

В случае задачи Вебера мы должны рассматривать соседние (и расположенные в некоторой окрестности) элементы матрицы  $x_{ij}$  как зависимые друг от друга. Как следует из здравого смысла (и подтверждается численными экспериментами), если целевая функция достигает меньшего (минимального среди сгенерированных точек) значения в точке пространства решений, заданной матрицей  $X^{min}$ , причем  $x_{i^*,j^*,k}^{min} = 1$ , то, с некоторой вероятностью, результаты и на следующих шагах будут высокими при условии  $x_{i,j,k}^{min} = 1$ , а также в некоторой окрестности точки  $(i^*, j^*)$ :

$$x_{i^*,j^*,k}^{min} = 1, i^* - N_S \leq i \leq i^* + N_S, j^* - N_S \leq j \leq j^* + N_S,$$

где  $N_S$  — ширина и высота окрестности (исчисляемая в числе столбцов и строк матрицы, мы использовали  $N_S = \max\{N_x, N_y\}/10$ ). Соответственно, матрица вероятностей должна подвергаться адаптации и в этих ее элементах, при этом значение коэффициента адаптации  $d$  должно быть больше для точек, максимально близких к  $(i, j)$ , для дальних точек оно должно стремиться к 1.

Мы использовали следующие формулы:

$$d_{k,i,j} = d_{k,i,j}^* / d_{k,i,j}^w;$$

$$d_{k,i,j}^* = \begin{cases} 1 + d_0 / (1 + L((i^*, j^*), (i, j))), & i \in [j^* \pm N_S] \wedge j \in [j^* \pm N_S], \\ d_{k,i,j}^* = 1, & i \notin [j^* \pm N_S] \vee j \notin [j^* \pm N_S], \end{cases}$$

$$d_{k,i,j}^w = \begin{cases} 1 + d_0 / (1 + L((i^w, j^w), (i, j))), & i \in [j^w \pm N_S] \wedge j \in [j^w \pm N_S] \\ d_{k,i,j}^w = 1, & i \notin [j^w \pm N_S] \vee j \notin [j^w \pm N_S]. \end{cases}$$

Здесь  $(i^*, j^*)$  и  $(i^w, j^w)$  — ближайшие к точке  $(i, j)$  точки такие, что  $x_{i^*,j^*,k}^{min} = 1$  и  $x_{i^w,j^w,k}^{worst} = 1$  соответственно,  $X^{min} = (x_{ij}^{min})$  и  $X^{worst} = (x_{ij}^{worst})$  — экземпляры матрицы  $X$  из сгенерированного множества экземпляров, в которых достигается лучшее и худшее значение целевой функции.

Отметим, что способ адаптации матрицы вероятностей в целом аналогичен

способу адаптации вектора вероятностей алгоритма схемы МИВЕР для задач на сети, рассматриваемой в Главе 2.

После нескольких шагов адаптации значения элементов матрицы  $P$  становятся близки к 0 или 1, что ведет к генерации мало различающихся экземпляров  $X$ , близких к точке локального минимума целевой функции. Во избежание этого полезна процедура отката (возврата). Она устанавливает элементы матрицы  $P$  в начальное (или другое) значение. В простейшем случае возврат (откат) производится при условии достижения предельного числа шагов, не улучшающих значение целевой функции.

Лучшие результаты алгоритм демонстрирует при использовании процедуры частичного отката, при которой новое значение зависит от достигнутого на предыдущих шагах. Для данной процедуры мы использовали формулу:

$$p_{kij} = (p_{k-1ij} + q_k p_0) / (1 + q_k), \text{ если } p_{k-1ij} < p_0.$$

Здесь  $p_0$  — начальное значение вероятности, в качестве которого мы использовали среднее арифметическое элементов матрицы  $P$ . Коэффициент  $q_k$  может быть постоянным или меняться в зависимости от результатов предыдущих шагов. Например, он может зависеть от числа шагов  $s_m$ , прошедших с момента последнего улучшения достигнутого минимума целевой функции.

$$q_k = w / s_m.$$

Коэффициент  $w$  подбирается экспериментально, мы использовали  $w=100$ .

Адаптация алгоритма для многопроцессорных систем с общей памятью может быть выполнена путем параллельной генерации экземпляров матрицы  $X$  и оценки целевой функции. Схема этой версии показана на рис. 6.3, вариант 2. Если в нашей системе  $N_p$  процессоров, цикл генерации  $N$  экземпляров матрицы  $X$  может быть разделен между процессорами. Каждый процессор должен будет сгенерировать  $N/N_p$  экземпляров вектора  $X$  ( $N$  подбираем кратным  $N_p$ ) и вычислить для них значение целевой функции.

## 6.5 Результаты экспериментов

Для целей тестирования мы использовали задачу Вебера на плоскости (6.5-6.8) с дискретными координатами  $N_x=200$ ,  $N_y=400$ . Схема задачи представлена на рис.6.4 (часть А). Темные области соответствуют участкам с большим весовым коэффициентом (важные области), белые — с нулевым весовым коэффициентом (где радиопокрытие не

важно). На схеме имеются 3 препятствия.

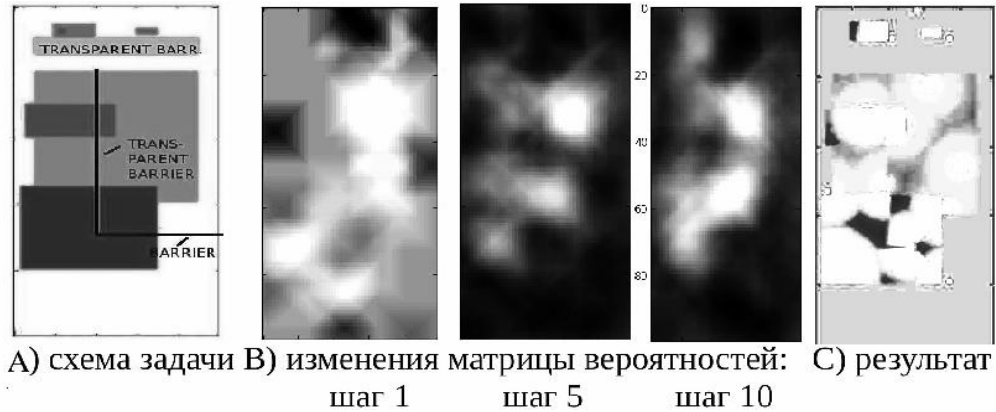


Рисунок 6.4 – Схема задачи и визуализация ее решения

Поскольку процедура отката представляет собой модификацию мультистарта, возможна организация параллельного мультистарта на независимых узлах. В случае задачи (6.8) система из 8 независимых узлов достигает результата, достигнутого единственным узлом, приблизительно в 2 раза быстрее (за 47%-59% времени) при данной схеме. Коэффициент параллельной эффективности, таким образом, составляет 0,21-0,27.

Для параллельной версии алгоритма (рис. 6.3, вариант 2), значение коэффициента параллельной эффективности составляет 0,92-0,95 для задач размерности 100x100 и растет с ростом размерности. Для нашей задачи (200x400) значение равно 0,95. Среднее значение коэффициента параллельной эффективности вычислено по итогам 10 запусков для 5 различных целевых функций (разные схемы задачи, аналогичные рис. 6.4).

Визуализация матрицы вероятностей для  $N_F=10$  представлена на рис. 6.4, часть B. Результаты (прогнозируемое покрытие на плоскости) также легко может быть визуализировано (рис. 6.4, часть C). Темные пятна - области с недостаточным прогнозируемым уровнем сигнала.

Для подтверждения оптимальности получаемого результата проведен ряд экспериментов для достаточно простых задач ( $N_F=3$ , схема рис. 6.4). Решения, полученные методом полного перебора (более 72ч счета), сравнивались с результатами алгоритма (рис. 6.5). Из графика видно, что для достижения приемлемых результатов требуется менее 1000 вычислений целевой функции (выполнено 100 шагов с генерацией 10 экземпляров  $X$  на каждом шаге).

Таким образом, предложенный алгоритм на базе метода изменяющихся

вероятностей может быть использован для приблизительного решения обобщенной задачи Вебера с неевклидовой монотонной функцией расстояния. Современные возможности вычислительной техники (многопроцессорные системы, недорогие кластеры) позволяют решать задачи с достаточной точностью.

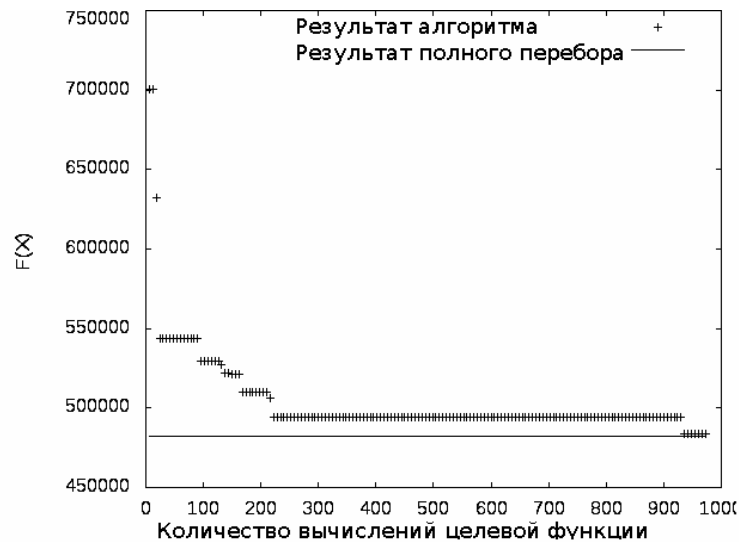


Рисунок 6.5 – Сравнение результатов алгоритма с результатами полного перебора

## 6.6 Практический пример: задача размещения точек доступа беспроводной сети

Прогресс в области телекоммуникаций влечет за собой расширение инфраструктуры и все более широкий охват отдаленных и малонаселенных регионов, где, в силу объективных причин, предпочтение все чаще отдается беспроводным сетям.

В настоящей работе мы рассматриваем метод поиска оптимального способа развития беспроводной сети (мест новых точек доступа или базовых станций), основанный на критериях охвата (покрытия) надежным доступом важных территорий и минимальной стоимости оборудования. Проблема поиска оптимального расположения и подбора типов элементов оборудования растущей сети сформулирована как многокритериальная дискретная задача оптимизации с ограничениями. Проблема моделирования радиопоглощающих свойств среды распространения также сформулирована и решена как дискретная задача оптимизации.

Непрерывное быстрое развитие инфраструктуры телекоммуникационных сетей,

включающее качественный рост полосы пропускания каналов связи и качество обслуживания, главным образом типично для больших городов и плотно населенных территорий, но наименее очевидно в отдаленных и малонаселенных регионах, где беспроводные сети часто даже более важны. С одной стороны, отсутствие какой-либо инфраструктуры препятствует развитию традиционных кабельных линий. С другой стороны, потребность этих территорий в качественном доступе к сети определяется современными стандартами (как правило, стандартами де-факто) офисной работы и бизнес-процессов различных производственных, торговых, правительственных, образовательных и исследовательских организаций. Кроме того, отдаленные территории нуждаются в качественных коммуникациях, которые делают их виртуально ближе к центральным областям, так как в некоторых случаях это единственный доступный способ установить контакты с партнерами, обеспечить централизованное управление и даже выполнить простые операции, такие как банковский перевод. В ближайшем будущем только у беспроводных технологий есть серьезные перспективы в арктическом регионе. Конечно, спутниковые технологии позволяют устанавливать доступ в Интернет, обеспечить телефонную связь и другие сервисы в любой точке, но они дороги. Поэтому локальные сети (часто покрывающие большие территории) обычно развиваются с использованием технологии WiFi (часто, в дополнение к кабелю или оборудованию VSAT как шлюз к всемирной сети). Хотя технологии WiFi спроектированы для локального использования, применение специальных антен, усилителей и другого оборудования позволяет охватить весьма значительные территории [539] (рекорд-более 150 км).

Зачастую такие сети развиваются хаотично, по мере необходимости охватывают новые территории. Излишнее количество точек доступа не увеличивает качество из-за проблем интерференции, но эта проблема менее важна для малонаселенных территорий. Выполнение полевых испытаний радиопоглощающих свойств среды и ее геометрии требует применения специального оборудования, работы квалифицированного персонала и существенных затрат времени. В то же время, максимальное использование доступной (возможно, весьма приблизительной) информации об уровнях сигнала и шума в различных точках и радиопоглощающих свойства среды, полученной от работающих аппаратных средств беспроводной сети, в состоянии помочь с поиском оптимальной конфигурации новых элементов растущей сети. Та же самая проблема

актуальна и для беспроводных сетей, хаотично развивающихся на больших сложноструктурированных территориях (например, реконструируемых территориях бывших промышленных зон).

В настоящей работе мы рассматриваем метод поиска оптимального способа развития беспроводной сети (мест новых точек доступа или базовых станций), основанный на критериях охвата (покрытия) надежным доступом важных территорий и минимальной стоимости оборудования, который не требует проведения специальных полевых испытаний и определения точной геометрии элементов среды радиовещания и ее радиопоглощающих свойств окружающего пространства, но позволяет принять во внимание минимальную доступную информацию, полученную от встроенных измерительных приборов беспроводных аппаратных средств и приблизительных данных о форме элементов среды. Проблема поиска оптимального расположения и подбора типов элементов оборудования растущей сети сформулирована как многокритериальная дискретная задача оптимизации с ограничениями (множественная задача о рюкзаке), разрешимая методом изменяющихся вероятностей [234]. Проблема моделирования радиопоглощающих свойств среды распространения также сформулирована и решена как дискретная задача оптимизации.

Рассматриваемая проблема оптимального размещения точек доступа - предмет дискуссий инженеров и сетевых администраторов на интернет-форумах и в научных публикациях. Некоторые из них предлагают готовое к использованию программное обеспечение, которое в состоянии предсказывать распространение сигнала в соответствии с физическими свойствами элементов окружающей среды. Другие научные работы предлагают различные методы поиска нового размещения точек доступа с оптимальным распространением сигнала [540] или (более сложный случай) размещение, которое предусматривает лучшую локализацию мобильного положения оборудования WLAN (вычисление его координат) в области вещания сети [541, 542]. Прогноз уровня сигнала в каждом пункте, учитывающем потери трассы от стен, потолков и других элементов, является известной технической проблемой [532].

Литература предлагает много способов постановки и решения аналитическим путем задачи предсказания уровня распространяемого сигнала (в уличных условиях и внутри зданий) и различные детерминированные и стохастические методы оптимизации для нахождения оптимального размещения. В работе [540] автор предлагает версию

симплекс-метода Нелдера-Мида и паттерн-алгоритма, который рассматривает задачу как задачу минимизации доли непокрытых устойчивым сигналом точек во всем множестве точек.

Используя эмпирическую внутреннюю модель распространения волны Мотли-Кинана с учетом типа стен и потолков для вычисления потери трассы, работа [238] представляет технику размещения точек доступа для оптимального внутреннего покрытия сигналом. Генетический алгоритм используется для нахождения конфигурации с меньшим значением максимальной потери пути для каждой точки пространства. В [533] авторы также предлагают строго типизированное генетическое программирование, чтобы решить проблему конфигурации точек доступа с оптимальным расположением точек доступа, предоставляющих лучшее покрытие. В [541] авторы используют генетические алгоритмы для нахождения расположения точек доступа, обеспечивающего оптимальную локализацию мобильного устройства в некоторой области, охваченной покрытием точек доступа. В [542] предлагается система для локализации такого устройства.

В [544] авторы предлагают метод проектирования крупномасштабной сети, где размещение точек доступа гарантирует отсутствие непокрытых областей. Оптимальную зону охвата сигналом точек доступа рассматривают как цилиндр, положения точек оптимизированы с использованием геометрических схем.

В нашем случае оптимальное покрытие - одна из целей. Второй важный критерий - минимальная стоимость беспроводного оборудования и его установки. Результаты вышеупомянутых методов трудно интерпретировать, если оптимальное место для точки доступа расположено далеко от электрической проводки, стен или других конструктивных элементов, где мы в состоянии закрепить и запитать оборудование.

Большинство вышеупомянутых работ основано на модели распространения сигнала, которая рассматривает логарифмическую потерю от каждого препятствия (стены, потолки) по некоторым информационным таблицам, но часто мы не знаем даже точного строительного материала наших элементов среды и их точной геометрии, и реальное поглощение может отличаться очень существенно. В [541] предлагается использовать поиск методом наименьших квадратов на экспериментальных данных, чтобы определить реальную потерю трассы, обусловленную препятствиями. Но у нас могут быть достаточно или недостаточно экспериментальных данных, которые

необходимы для метода наименьших квадратов, и может быть трудно получить дополнительные данные.

Для задачи [114] с запрещенными зонами, барьерами и неевклидовой функцией расстояния рассмотрены лишь некоторые частные случаи [160]. В параграфах 6.1-6.5 настоящей диссертации рассмотрен метод приближенного решения таких задач, основанный на идеях метода изменяющихся вероятностей. Ниже работе мы показываем практическую применимость данного метода к сформулированной задаче оптимального размещения точек доступа беспроводной сети.

### **6.7 Математическая модель размещения точек доступа**

Информацию об уровне сигнала и шума в точках, где сеть уже работает, можно получить от сетевого оборудования (например, в Linux утилита `iwconfig` дает нам такую информацию). Минимальная информация о геометрии окружающей среды (расположение стен, деревьев и других объектов) всегда доступна из карт, схем или непосредственного визуального наблюдения. Радиопоглощающие свойства элементов окружающей среды доступны из информационных таблиц [533] и могут быть определены более точно, если элемент расположен между существующим передатчиком и приемником или в зоне Френеля между ними. Методика оценки зоны Френеля в использованной нами дискретной системе координат и затухания сигнала при прохождении через зону Френеля приведена в [545]. В сложных случаях поглощающие свойства могут быть определены как решение вложенной задачи оптимизации.

Кроме того, у нас есть информация о возможности и приблизительной стоимости размещения новой точки доступа в каждом месте. Так, места, где имеется электрическая проводка (по крайней мере, места с существующими аппаратными средствами беспроводного клиента) лучше подходят для установки новых точек доступа, чем отдаленные от инфраструктуры электропитания и зданий. Кроме того, лицо, принимающее решения, в состоянии определить некоторое число возможных мест новых точек доступа в соответствии с причинами, которые очевидны для него, но трудно формализуемы.

Так, у нас есть информация о приблизительной геометрии окружающей среды (расположение существующего оборудования, расположение препятствий для



распространения сигнала и их возможных поглотительных свойств) и данные сигнала и уровня шума в некоторых точках.

Чтобы упростить наши вычисления, разделим нашу область на ячейки. Как правило, реальные задачи не позволяют нам игнорировать вертикальные позиционные соотношения элементов окружающей среды и инфраструктуры. Именно поэтому мы должны рассматривать нашу среду как трехмерное пространство. Для простоты описания и представления на бумаге мы рассматриваем 2-мерный случай.

Допустим, что оборудование расположено в центрах ячеек, а границы элементов окружающей среды (препятствия, такие как стены, окна, деревья и т.д.) совпадают с границами любого набора клеток. Так, мы рисуем схему нашей среды с существующими и новыми точками доступа и препятствиями и существующие и перспективные зоны вещания нашей сети, как будто мы отмечаем элементы в тетради в клетку (рис.6.6). Если у нас есть информация об уровне сигнала в некотором месте, то мы предполагаем, что этот уровень одинаков для целой клетки.

Если наша сеть содержит направленные антенны, то мы предполагаем тогда, что сигнал такой антенны одинаков по мощности в области, ограниченной некоторым углом (исключая пары антенн направленной передачи точка-точка, в этом случае мы предполагаем, что область распространения включает только 2 клетки, где расположены 2 антенны).

Пусть в нашей схеме есть несколько зон уверенного приема. Мы определяем весовой коэффициент для каждой из этих областей (мера ее «важности»). Кроме того, возможно установить некоторый минимальный битрейт (скорость передачи информации, определяемый уровнем сигнала) для той или иной области. Таким образом, у нас есть  $N_c$  ячеек, которым мы должны сопоставить весовые коэффициенты  $v_j$  и минимальные битрейты  $b_j$ ,  $1 \leq j \leq N_c$ .

Пусть есть  $N_p$  точек (клеток), где мы в состоянии поместить точки доступа. Для каждой из этих клеток мы рассматриваем  $N_t$  типов аппаратных средств (различные антенны, усилители и т.д.). Для каждого из этих мест мы знаем также приблизительную стоимость вспомогательного оборудования (такого, как провода электропитания или автономное электропитание, кронштейны антенны и т.д.)  $C_i$ ,  $1 \leq i \leq N_p$ , и стоимость каждого вида аппаратных средств  $C_k$  также известна,  $1 \leq k \leq N_c$ .



Рисунок 6.6 – Фрагмент схемы с располагаемым оборудованием и элементами окружения

Определим матрицу  $X$  булевых переменных  $x_{ik}$ . Полагаем, что значение переменной  $x_{ik}$ , равное 1 означает, что мы решили поместить аппаратные средства точки доступа  $k$ -го типа в  $i$ -й возможной клетке, а установка значения  $x_{ik}=0$  означает, что мы решили поместить туда другой вид оборудования или оставить ячейку незанятой.

Наша цель состоит в том, чтобы поставлять максимальный битрейт в максимальном количестве клеток в соответствии с их весовыми коэффициентами с при минимальной стоимости установки оборудования.

$$\begin{aligned}
 & b_j(X) v_i \rightarrow \max, \quad b_j \geq b_{minj} \\
 & \sum_{i=1}^{N_p} \sum_{k=1}^{N_t} (C_i + C_k) x_{ik} \rightarrow \min, \\
 & \sum_{k=1}^{N_t} x_{ik} \leq 1 \forall 1 \leq i \leq N_p, \\
 & x_{ik} \in \{0,1\} \forall 1 \leq i \leq N_p, 1 \leq k \leq N_t.
 \end{aligned}$$

Здесь  $b_j(X)$  - максимальный возможный битрейт в  $j$ -й точке приема (хотя все беспроводные устройства – дуплексные, для простоты описания мы называем точки доступа передающими, а клиентские устройства - принимающими). Точки доступа

установлены в соответствии с вектором  $X$  логических переменных,  $v_j$  - коэффициент веса  $j$ -th получающей точки,  $N_p$  - общее количество возможных мест для точек доступа в будущей системе,  $N_t$  - число типов оборудования точки доступа (точки доступа или беспроводные маршрутизаторы, оборудованные соответствующими типами антенн),  $b_{min j}$  - минимальные гарантируемые битрейты, необходимые для некоторых точек приема.

Определим поглощающие свойства каждого препятствия, показанного в нашей схеме. Пусть  $\Pi_1$  - поглощение (в децибелах) сигнала, проходящего через слой 1-го препятствия в слое ячеек. Начальные значения могут быть получены из информационных таблиц [533]. Например, для стены  $\Pi_1 = -7$  дБ, для окна -2 дБ, для растущих деревьев -3 дБ на каждый метр (Значение для ячейки зависит от ее размера). Ниже мы рассматриваем метод, разрешающий нам определить более точно значение  $\Pi_1$  в соответствии с экспериментальными данными.

Далее, уровень сигнала (см. рис. 6.7)  $i$ -й точки доступа, которую мы можем получить в  $j$ -й клетке, может быть вычислен как

$$P_{rij}(X) = P_{ti} + G_{ti} + G_r - L_{ij}(S_{OBST}) - 40 - 20 \log D_{ij}.$$

Здесь  $P_{rij}(X)$  - уровень сигнала (дБ)  $i$ -й точки доступа, полученной в  $j$ -й принимающей точке,  $G_{ti}$  - усиление антенны точки доступа (включая потерю всех кабелей),  $G_r$  - усиление антенны пункта получения (также, включая все кабели),  $L_{ij}(S_{OBST})$  является потерей пути между  $i$ -й точкой доступа и  $j$ -й получающей точкой с конфигурацией препятствий (стены, окна, деревья и т.д.), описанной множеством  $S_{OBST}$ .  $D_{ij}$  - расстояние (в метрах) между  $i$ -й точкой доступа и  $j$ -й получающей точкой.

Битрейт, который обеспечивается этогоэтого уровня сигнала,

$$b_{ij}(X) = \begin{cases} 54 \text{mb} / s, P_{rij}(X) - N_j > 66 \text{dBm}, \\ 18 \text{mb} / s, P_{rij}(X) - N_j > 66 \text{dBm}, \\ 1 \text{mb} / s, P_{rij}(X) - N_j > 93 \text{dBm}, \\ 0, P_{rij}(X) - N_j \leq 93 \end{cases}$$

Здесь,  $b_{ij}(X)$  - максимальный доступный битрейт связи между  $i$ -й точкой доступа и  $j$ -й получающей точкой.

Учитывая несколько возможных вариантов беспроводного оборудования для каждой точки доступа, уровень сигнала

$$P_{rij} = \sum_{k=1}^{N_t} P_{ti} + \sum_{k=1}^{N_t} G_{ti} x_{ik} + G_r - L_{ij}(OBST) - 40 - 20 \log D_{ij}$$

и доступный битрейт для  $j$ -й ячейки приема равен

$$b_{ij}(X) = \begin{cases} 54Mб / c, & P_{rij}(X) - N_j < 66дб \\ 18Mб / c, & 78д8 > P_{rij}(X) - N_j > 66дб \\ 1Mб / c, & 93д3 > P_{rij}(X) - N_j > 78д8 \\ 0, & P_{rij}(X) - N_j > 93д3 \end{cases} .$$

Чтобы вычислить уровень сигнала, принимаемого в некоторой ячейке от некоторой точки доступа, мы должны определить, какие препятствия лежат между точкой доступа и покрытой ячейкой, учитывая зону Френеля.

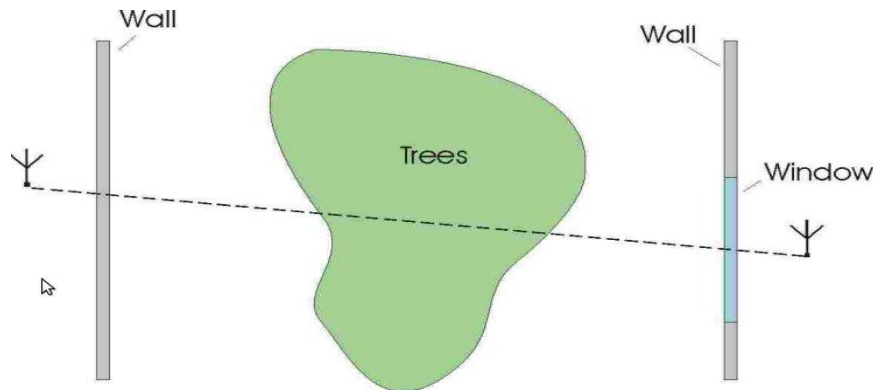


Рисунок 6.7 – Потеря трассы

Чтобы определить клетки, которые мы рассматриваем как принадлежащие линии прямой видимости, мы выполняем алгоритм рисования линии (который применяется в программном обеспечении компьютерной графики) [546] (рис. 6.8).

$$R_F = \frac{17.3}{d_{cell}} \sqrt{\frac{1}{(1000F)} \frac{D_i D_j}{D_i + D_j}} = \frac{0.3531}{d_{cell}} \sqrt{\frac{D_i D_j}{D_i + D_j}} .$$

Здесь  $R_F$  - диаметр зоны Френеля,  $d_{cell}$  - размер клетки в дискретной схеме координат,  $D_i$  - расстояние до  $i$ -й точки доступа,  $D_j$  - расстояние до  $j$ -й принимающей точки,  $F$  является частотой (допустим, 2.44 ГГц).

Давайте определим (в клетках) толщину зоны Френеля в срединной точке. Если она не превышает 1.5 ячейки, тогда мы предполагаем, что она равна 1 клетке, и зона распространения волны лежит в пределах линии, показанной на рис. 6.8.

$$R_F = \begin{cases} 1, & \frac{0.3531}{d_{cell}} \sqrt{\frac{D_i D_j}{D_i + D_j}} < 1.5, \\ \frac{0.3531}{d_{cell}} \sqrt{\frac{D_i D_j}{D_i + D_j}}, & \frac{0.3531}{d_{cell}} \sqrt{\frac{D_i D_j}{D_i + D_j}} \geq 1.5 \end{cases}$$

Иначе, для каждой клетки линии, показанной на рис. 6.8, где толщина зоны Френеля превышает 1.5 клетки, мы 'чертим' линейный сегмент, перпендикулярный к линии прямой видимости с длиной, равной диаметру зоны Френеля. Вычислим количество клеток этого линейного сегмента, занятого препятствием.

$$r_{Fmij} = \frac{N_{OBSTm}}{N_{Fm}}.$$

Здесь,  $r_{Fmij}$  - отношение числа клеток  $m$ -го линейного сегмента, занятого препятствием ( $N_{OBSTm}$ ) и общее количество клеток в линейном сегменте ( $N_{Fm}$ ).

Если препятствие занимает меньше чем 25% клеток линейного сегмента, тогда мы предполагаем, что поглощение препятствия при этом сегменте пропорционально части зоны Френеля, если она превышает 25%, тогда мы предполагаем, что потеря пути, вызванная препятствием, равна той, которая была бы, если бы препятствие полностью перекрывало путь.

$$L_{mij} = \begin{cases} L_{OBSTq}, r_{Fmij} \leq 0.25 \wedge OBST_q \cap SEGmij \neq \emptyset, \\ L_{OBSTq}, r_{Fmij} > 0.25 \wedge OBST_q \cap SEGmij \neq \emptyset, \\ 0, OBST_q \cap SEGmij = \emptyset \end{cases}$$

Здесь  $L_{OBSTq}$  - потеря пути, вызванная с  $q$ -м препятствием, у которого размер в 1 клетку,  $OBST_q$  - конфигурация  $q$ -го препятствия (множество клеток, которые оно занимает),  $SEGmij$  - ряд клеток  $m$ -го линейного сегмента, нормального к линии от  $i$ -й точки доступа до  $j$ -й точки приема.

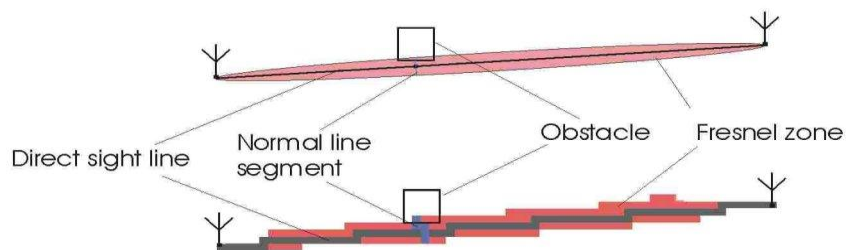


Рисунок 6.8 – Зона Френеля и ее представление в дискретных координатах

Если зона Френеля занята препятствием больше чем на 30%, тогда мы

предполагаем, что препятствие поглощает сигнал, как будто оно занимает 100% зоны Френеля [533]. Так, мы знаем значение потери, вызванной препятствиями, расположенными между точкой доступа и покрытой клеткой.

$$L_{ij} = \sum_{m=1}^{N_{ij}} L_{mij}(OBST)$$

Здесь  $L_{ij}$  - полная потеря пути, вызванная препятствиями, расположенными между  $i$ -й точкой доступа и  $j$ -й ячейкой приема. Таким образом, уровень сигнала от  $i$ -й точки доступа, оборудованной аппаратными средствами  $k$ -го типа принимаемого в  $j$ -th клетке,

$$P_{rj}(X) = \sum_{k=1}^{N_t} (P_{ti} + G_{ti})x_{ik} + G_r - \sum_{m=1}^{N_{ij}} L_{mij}(OBST) - 40 - 20\log N_{ij}d_{cell}$$

Радиооборудование клиента, расположенное в каждой из покрытых клеток, в состоянии установить связь с любой точкой доступа, но оно выбирает точку доступа с максимальным уровнем сигнала. Уровень сигнала “лучшей” точки доступа, полученной в  $j$ -й клетке, может быть вычислен как

$$P_{rij}(X^{real}, S_{OBST}) = \sum_{k=1}^{N_t} (P_{ti} + G_{ti})x_{ik} + G_r - \sum_{m=1}^{N_{ij}} L_{mij}(S_{OBST}) - 40 - 20\log N_{ij}d_{cell} .$$

Для точек, в которых требуется обеспечить приемлемый уровень сигнала, введем дополнительную функцию, учитывающую корректирующую штрафную составляющую при недостаточном уровне сигнала.

$$B_{ij}(X) = \begin{cases} b_{ij}(X), & b_{ij}(X) > 0, \\ -100, & b_{ij}(X) = 0. \end{cases}$$

Таким образом, первый критерий преобразуется в следующую целевую функцию:

$$F(X) = \sum_j B_{ij}(X) \rightarrow \max.$$

Таким образом у нас есть дискретная задача оптимизации с псевдодобулевыми функциями с ограничениями, имеющая 2 критерия. Для решения задач этого вида накоплен большой опыт применения алгоритма на базе метода изменяющихся вероятностей, предложенного в [234] в его версии, описанной в [230, 156]. Для задачи сокращения 2-ого критерия и получения Парето-оптимального набора решений мы используем метод, предложенный в [523] и применяемый для задач оптимизации телекоммуникационных систем [523], таким образом, возможно предположить его эффективность для нашей проблемы. Заключительный шаг алгоритма - сравнение

Парето-оптимальных решений, показанных в схеме с со значениями возможных битрейтов каждой ячейки для различных значений стоимости достижения такого покрытия. Максимальный битрейт, доступный в  $j$ -й ячейке, равен

$$b_j(X) = \begin{cases} 54mb/s, P_{rj}(X) - N_j > 66dBm, \\ 18mb/s, P_{rj}(X) - N_j > 66dBm, \\ 1mb/s, P_{rj}(X) - N_j > 93dBm, \\ 0, P_{rj}(X) - N_j \leq 93 \end{cases}$$

## 6.8 Настройка параметров модели, результаты

Мы можем предположить поглощение сигнала каждым из препятствия в нашей схеме (рис. 6.1). Но обычно мы не знаем точную структуру строительных материалов, точную толщину стен и других параметров, которая обуславливает значительную разницу (даже в логарифмической шкале) между реальными значениями поглощения и значениями, взятыми из информационных таблиц. Иногда есть препятствия, которые не показаны в изначальной схеме, но проявляют существенное влияние на распространение сигнала. Попытаемся решить задачу более точного определения значений поглощения для каждого препятствия и обнаружения 'невидимых' препятствий.

Пусть истинное значение уровня сигнала в  $j$ -й клетке и расчетное значение равны

$$P_{rj}(X^{real}, S_{OBST}) = \sum_{k=1}^{N_t} (P_{ti} + G_{ti}) x_{ik} + G_r - \sum_{m=1}^{N_{ij}} L_{mij}(S_{OBST}) - 40 - 20 \log N_{ij} d_{cell}$$

Здесь  $X^{real}$  - вектор, характеризующий конфигурацию текущего размещения точек доступа ( $x_{ikreal} = 1$ , если  $k$ -й тип оборудования уже расположен в  $i$ -й ячейке).

В этом расчете мы учитываем только существующие точки доступа. Таким образом, чтобы определить более точные значения коэффициентов  $L_{mij}$ , мы должны решить задачу минимизации

$$\sum_{j=1}^{N_p^{real}} |P_j^{real} - P_{rj}(X^{real}, OBST)| \rightarrow \min$$

Здесь  $N_p^{real}$  - число точек доступа, которые уже функционируют. Здесь переменные оптимизации - значения поглощения "невидимых" препятствий. Для простоты мы предполагаем, что эти препятствия расположены между точкой доступа и принимающей ячейкой, и их форма - сфера или круг с центром между точкой доступа и

ячейкой-приемником сигнала, так как реальная форма неизвестна. Мы предполагаем, что такое невидимое препятствие существует, если прямая две ячейки находятся в прямой видимости и зона Френеля свободна, но реальный уровень сигнала отличается от расчетного значительно (10 дБ в нашем примере)

$$L_{ij}^{inv} = \begin{cases} 0, & P_{rj}(X^{real}, OBST) - P_j^{real} < 10dB, \\ P_{rj}(X^{real}, OBST) - \frac{P_j^{real}}{D_{ij}} / d_{cell}, & P_{rj}(X^{real}, OBST) - P_j^{real} \geq 10dB \end{cases}$$

Здесь  $L_{ij}^{inv}$  – потеря трассы от ‘невидимого’ препятствия между  $i$ -й точкой доступа и  $j$ -й ячейкой приема.

Мы рассмотрели задачу выбора мест для дополнительных точек доступа растущей сети и их типов как задачу оптимизации, которая основана на адаптации модели окружающей среды распространения радиоволн. Хотя наша модель учитывает только наиболее важную информацию о взаимодействии сигнала и окружающей среды (фактически, только поглощение и шум без определения его характера), эта модель позволяет нам предсказывать уровень сигнала и битрейт каждой перспективной области, по крайней мере, как грубую оценку. Этот метод не нуждается ни в каких полевых испытаниях, но он учитывает результаты их, если они были выполнены. В основном наш метод совпадает с обычной практикой инженерных вычислений, но позволяет собирать всю информацию о нашей среде в единую модель и находить одно или более Парето-оптимальных решений, принимающих во внимание всю доступную информацию об окружающей среде и стоимость размещения нового оборудования.

Результаты нашего метода могут легко интерпретироваться специалистом, так как возможные места точек доступа определены в первом шаге. Практика показывает необходимость проверки реального уровня сигнала после первоначального решения задачи предлагаемым методом и повторного применения метода с новыми значениями измерений. Для системы, подобной показанной на рис.6.6, различие между реальными и расчетными значениями сигнала составляет до 11 дБ, второй шаг дает максимальную ошибку не более 6 дБ, что приемлемо для инженерных вычислений. На рис.6.9 приведен пример задачи и ее решения предложенным методом.

На рис.6.10 приведен пример задачи, в которой области возможного размещения точек доступа ограничены зданиями.



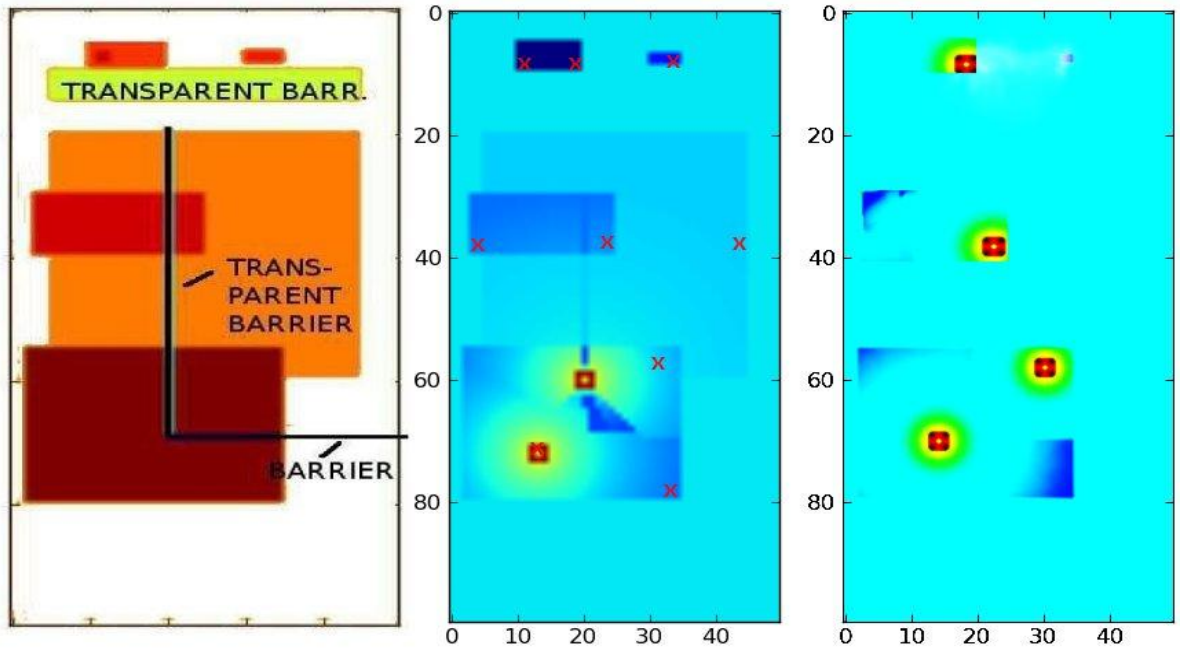


Рисунок 6.9 – Пример решения

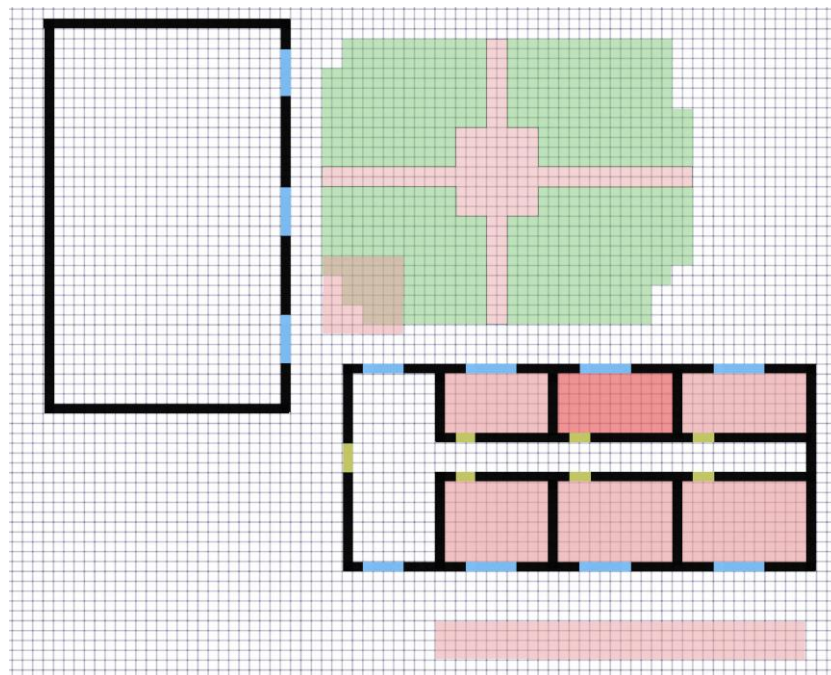


Рисунок 6.10 – Фрагмент схемы с требуемыми зонами покрытия и элементами окружения

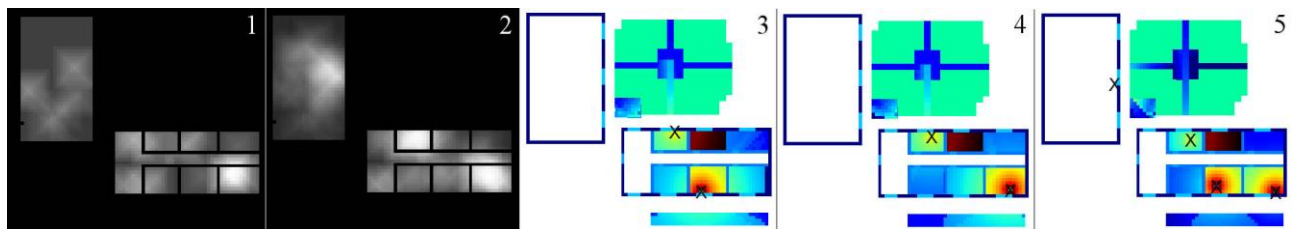


Рисунок 6.11 – Ход решения и результаты

На рисунке 6.11 изображены ход решения задачи и результаты, полученные при различных ограничивающих условиях. По условиям задачи, точки доступа могут располагаться в пределах двух строений, остальные области составляют запрещенную зону задачи. Части 1 и 2 отражают матрицу вероятностей применяемого метода изменяющихся вероятностей [316] после первого и четвертого шагов алгоритма соответственно. Результаты алгоритма для двух и четырех размещаемых точек доступа изображены на частях 4 и 5. В красных и желтых зонах покрытие достаточно для уверенной работы оборудования, зоны с недостаточным уровнем сигнала обозначены темно-синим. Такой подход позволяет специалисту, занимающемуся размещением оборудования, получить более широкий спектр вариантов, различающихся зонами охвата и стоимостью (количеством) размещаемого оборудования, и принять более продуманное решение. Для сравнения на части 3 приведен результат полного перебора для двух точек доступа.

Таким образом, современное оборудование сетей Ethernet 802.11 (WiFi) предоставляет информацию, достаточную для решения задачи оптимального размещения на основе приблизительной пространственной схемы радиопоглощающей среды, скорректированной с использованием встроенных датчиков уровня сигнала. Предложенный в настоящей главе метод решения задач автоматической группировки и размещения с произвольной мерой расстояния, основанный на идеях метода изменяющихся вероятностей, позволяет решать подобные задачи. Несмотря на высокие вычислительные затраты метода, эффективные способы распараллеливания алгоритмов на базе метода изменяющихся вероятностей позволяют решать возникающие инженерные задачи.

## **Выводы к Главе 6**

Проведенные эксперименты показали, что алгоритм, основанный на идеях метода изменяющихся вероятностей, может быть применен для получения приблизительного решения задачи автоматической группировки или задачи размещения (р-медианной) в непрерывном пространстве в наиболее сложных случаях, когда в качестве меры расстояния используется произвольная функция, например - заданная алгоритмически.

При этом разработанный алгоритм [547] может эффективно применяться и для сложных задач размещения с барьерами и запрещенными зонами, что продемонстрировано на примере решения задачи о размещении точек доступа беспроводной сети.

Также впервые разработаны параллельные версии алгоритмов на основе метода изменяющихся вероятностей (параллельные версии алгоритма для задач на сети даны в Главе 2), что позволяет задействовать ресурсы вычислительных кластеров и многопроцессорных/многоядерных систем при использовании этих требовательных к вычислительным ресурсам алгоритмов для решения широкого круга задач автоматической группировки или иных задач псевдодулевой оптимизации.

## ЗАКЛЮЧЕНИЕ

В диссертации предложен новый метод решения задачи автоматической группировки и размещения объектов с использованием различных метрик и мер сходства, состоящий в составлении эффективных комбинаций агломеративных жадных эвристик с различными схемами глобального и локального поиска, позволяющий успешно решать широкий круг задач автоматической группировки с повышенной точностью и стабильностью результата при большом объеме входных данных.

Цель диссертации достигнута путем решения поставленных задач. А именно:

1. Проведен анализ существующих методов и алгоритмов решения задач автоматической группировки, размещения и иных задач оптимизации. Разработан новый алгоритм, основанный на идеях МИВЕРа, который может быть применен как для решения задач автоматической группировки на сети (как самостоятельный алгоритм либо в качестве алгоритма инициализации для генетических алгоритмов), так и для решения задачи автоматической группировки объектов в непрерывном пространстве с произвольной мерой расстояния с барьерами и запрещенными зонами. Алгоритм в комбинации с генетическим алгоритмом с жадной эвристикой повышает как быстродействие, так и точность последнего, что доказано экспериментально на различных примерах задач большой размерности. Алгоритм применен, в частности, при решении задачи размещения пунктов реагирования в чрезвычайных ситуациях на сети автодорог. В диссертации разработаны эффективные параллельные версии алгоритмов МИВЕРа.

2. Впервые предложены новые модификации жадных эвристических процедур в составе генетического алгоритма для задачи автоматической группировки на сети. Доказано, что разработанные модификации позволяют в случае большой размерности практических задач получать стабильно более точные результаты за фиксированное приемлемое время, либо получать точные результаты быстрее в 2-55 раз в сравнении с известными эволюционными алгоритмами с жадной агломеративной эвристической процедурой.

Доказана применимость и эффективность жадной агломеративной эвристической процедуры для решения широкого круга задач монотонной псевдодобулевой оптимизации, что показано на примере задачи оптимальной загрузки производственных мощностей

литейно-прокатных и химических производств, для которой впервые предложен алгоритм, позволяющий получать решения в интерактивном режиме.

3. Предложены новые модификации жадных эвристических процедур, изначально разработанных для решения дискретных задач размещения, а также новая жадная эвристическая процедура с вещественным алфавитом для задачи автоматической группировки данных в непрерывном пространстве характеристик, которые могут быть эффективно использованы как самостоятельные алгоритмы, а также в комбинации с различными стратегиями глобального поиска, такими как эволюционные алгоритмы и адаптивный алгоритма МИВЕРа для решения практических задач большой размерности с широким кругом метрик и иных мер расстояния. Эвристические процедуры комбинируются с различными известными методами локального поиска и методами агрегирования данных, что позволяет задействовать передовые подходы для повышения быстродействия алгоритмов.

При этом новая жадная эвристическая процедура с вещественным алфавитом является компромиссным подходом по времени счета и точности результата и позволяет решать практические задачи группировки данных большой размерности (число векторов данных – от нескольких тысяч при числе измерений пространства характеристик от 2 до нескольких тысяч) в интерактивном режиме с точностью, не уступающей точности других известных методов, при этом быстрее (в 1.4-220 раз) получать наиболее стабильный результат (наименьшее среднеквадратичное отклонение получаемого значения целевой функции), что продемонстрировано на примере задачи автоматической группировки электрорадиоизделий по производственным партиям. При этом разработанный метод классификации электрорадиоизделий космического применения по классам качества и производственным партиям в условиях специализированных тестовых центров позволил запустить производство изделий нового класса качества – специальные партии электронной компонентной базы космического применения.

4. Исследованы и систематизированы меры расстояния, применяемые в задачах размещения и автоматической группировке. Для задач Вебера и множественных задач Вебера с метриками, основанными на угловых расстояниях, впервые предложены алгоритмы, позволяющие за полиномиальное время получать точное решение. Разработанные алгоритмы, основанные на декомпозиции в задачи с прямоугольной

метрикой, могут успешно использоваться как составная часть алгоритма решения соответствующей р-медианной задачи и применяться для решения логистических задач автоматической группировки объектов с целью снижения стоимости их транспортировки при использовании специальных средств транспортировки, в частности, манипуляторов с вращающейся стрелой.

Также предложен новый алгоритм для задачи Вебера с мерой расстояния  $L(X, Y) = \max\{\|X - Y\|_2, 1\}$ , основанный на модифицированной процедуре Вайсфелда в комбинации с методом бисекций, который позволяет решать задачу быстрее известных методов, причем, в отличие от них, в определенных случаях достигается точный результат.

Проведенная модернизация арсенала точных методов теории размещения, а именно – методов решения задачи Вебера с широким кругом мер расстояния, позволила создать основу для синтеза новых моделей и методов.

5. В целом, разработан новый эффективный метод потокового решения широкого круга задач автоматической группировки и размещения большой размерности (от сотен до сотен тысяч векторов данных) с широким кругом мер расстояния (при условии существования эффективного алгоритма решения соответствующей задачи Вебера) и задач монотонной псевдобулевой оптимизации – метод жадных эвристик, основанный на применении новых агломеративных жадных эвристических процедур и их модификаций в составе различных схем глобального поиска (эволюционные алгоритмы, МИВЕР, детерминированный, мультистарт) в сочетании со сформулированными условиями применимости эвристик, их модификаций, схем глобального поиска в зависимости от параметров задачи. Определены границы эффективного использования нового метода жадных эвристик при создании алгоритмов решения практических задач, создана система практических рекомендаций (оформленная в виде алгоритма) по синтезу эффективных алгоритмов потокового решения задач автоматической группировки, размещения и монотонной псевдобулевой оптимизации с большим объемом входных данных, позволяющих повысить точность решения таких задач с большим объемом входных данных за ограниченное время, а также повысить стабильность получаемых результатов.

**СПИСОК ЛИТЕРАТУРЫ**

1. Бериков, В.Б. Современные тенденции в кластерном анализе / В.Б.Бериков, Г.С. Лбов // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению "Информационно-телекоммуникационные системы". Новосибирск: Институт математики им. С.Л. Соболева СО РАН.- 2008.- 26 с. [Электронный ресурс] Режим доступа URL <http://www.ict.edu.ru/ft/005638/62315e1-st02.pdf> (дата обращения 21.07.2014).
2. Gantz, J.F. The diverse and exploding digital universe. IDC White Paper /Gantz, J.F.- Framingham: IDC.- 2008 [Электронный ресурс] Режим доступа URL <http://www.emc.com/collateral/analyst-reports/diverse-exploding-digitaluniverse.pdf> (дата обращения: 01.12.2014)
3. Jain, A.K. Data clustering: 50 years beyond K-means /A.K. Jain // Pattern Recognition Letters.- 2010.- Vol.31.- P. 651-666.
4. Tukey, J.W. Exploratory Data Analysis / J.W. Tukey.- Addison-Wesley.- 1977.- P.688
5. Tabachnick, B.G. Using Multivariate Statistics, fifth ed. / B.G.Tabachnick, L.S. Fidell.- Boston:Allyn and Bacon.- 2007.- P.980
6. Duda, R. Pattern Classification, second ed. / R. Duda., P. Hart, D. Stork.- New York:John Wiley and Sons.- 2001.- P.680
7. Semi-Supervised Learning / O. Chapelle, B. Schoelkopf.,A.Zien (Eds.).- Cambridge:MIT Press.- 2006.- P.508
8. Lange, T. Learning with constrained and unlabelled data /T. Lange, M.H. Law, A.K. Jain, J. Buhmann // IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognition.- 2005.- Vol.1.- P.730-737.
9. Merriam-Webster Online Dictionary. Cluster analysis.- 2008. [Электронный ресурс] Режим доступа URL <http://www.merriam-webster-online.com> (дата обращения 21.07.2014)
10. JSTOR.- JSTOR.- 2009.- [Электронный ресурс] Режим доступа URL <http://www.jstor.org> (дата обращения 21.07.2014).
11. Jain, A.K. Image segmentation using clustering /A.K. Jain, P. Flynn // Advances in Image Understanding.- IEEE Computer Society Press.- 1996.- P.65–83.
12. Арлазаров, В.В. Структурный анализ текстовых полей в системах потокового ввода оцифрованных документов /В.В. Арлазаров, В.М. Кляцкин, О.А. Славин // Труды ИСА РАН.- 2015.- Т. 65, вып. 1.- С.75-81.
13. Shi, J. Normalized cuts and image segmentation / J. Shi, J. Malik // IEEE Trans. Pattern Anal. Machine Intell.- 2000.- Vol.22.- P. 888–905.

14. Борисенко, В.И. Сегментация изображения (состояние проблемы) / В.И.Борисенко, А.А.Златопольский, И.Б. Мучник// Автомат. и телемех.- 1987.- вып. 7.- С. 3-56.
15. Iwayama, M.Cluster-based text categorization: A comparison of category search strategies / M. Iwayama, T. Tokunaga // Proc. 18th ACM Internat. Conf. on Research and Development in Information Retrieval.- 1995.- P. 273–281.
16. Барахнин, В.Б. Кластеризация текстовых документов на основе составных ключевых термов / В.Б. Барахнин, Д.А.Ткачев// Вестник НГУ. Серия: Информационные технологии.- 2010.- Т.8, вып.2.- С. 5-14.
17. Барахнин, В.Б. О задании меры сходства для кластеризации текстовых документов / В.Б. Барахнин, В.А. Нехаева, А.М. Федотов // Вестник НГУ. Серия: Информационные технологии.- 2008.- Т.6, вып.1.- С.3-9.
18. Bhatia, S. Conceptual clustering in information retrieval / S. Bhatia, J. Deogun // IEEE Trans. Systems Man Cybernet.- 1998.- Vol. 28 (B).- P. 427–436.
19. Berry, M.J.A. Data Mining techniques: for marketing, sales, and customer relationship management, 2nd ed. /Berry M.J.A., Linoff G.S.- [s.l.]: Wiley.- 2004.- P.464.
20. Галямов, А.Ф. Управление взаимодействием с клиентами коммерческой организации на основе методов сегментации и кластеризации клиентской базы / А.Ф. Галямов, С.В. Тархов // Вестник УГАТУ.- 2014.- Т.18, № 4(65).- С.149-156
21. Hu, J. Statistical methods for automated generation of service engagement staffing plans / J. Hu, B.K. Ray, M. Singh // IBM J. Res. Dev. 2007. Vol. 51, issue 3. P. 281–293.
22. Baldi, P. DNA Microarrays and Gene Expression /P. Baldi., G. Hatfield.- [s.l.]: Cambridge University Press.- 2002.- P.208.
23. Андреев, В.Л. Классификационные построения в экологии и систематике / В.Л. Андреев.- М.:Наука.- 1980.- С.142.
24. Славин, О.А. Алгоритмы распознавания шрифтов в печатных документах / О.А. Славин // Информационные технологии и вычислительные системы.- 2010.- №3.- С. 27-38.
25. Connell, S.D. Writer adaptation for online handwriting recognition / S.D. Connell, A.K. Jain//IEEE Trans. Pattern Anal. Machine Intell.- 2002.- Vol.24, issue 3, P.329–346.
26. Frank, I.E. Data Analysis Handbook / I.E. Frank, R. Todeschini.- [s.l.]: Elsevier Science Inc.- 1994.- P. 227–228.
27. Ester, M. Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise / M. Ester, H.-P. Kriegel, J. Sander, X. Xu // KDD-96 Proceedings.- [s.l.]:[s.n.]- 1996.- P. 226-231.
28. Рубан А.И. Идентификация и чувствительность сложных систем / А.И. Рубан.- Томск:Изд-



- во Томск, гос. Ун-та.- 1982.- С.302.
29. Справочник по прикладной статистике / Под ред. Э. Ллойда, У. Ледермана, С.А. Айвазяна и др.- М.: Финансы и статистика.- 1989. - Т.1. - С.510.
  30. Parzen, E. On the Estimation of Probability Density Function and the Mode / E. Parzen / E. Parzen // Ann. Math. Statist.- 1962.- Vol. 33.- P. 1065.
  31. Parzen, E. On Estimation of a Probability Density, Function and Mode / E. Parzen // II IEEE Transactions on Information Theory.- 1982.- Vol. 4, No. 6.- P. 663-666.
  32. Епанечников, В.А. Непараметрическая оценка многомерной плотности вероятности / В.А. Епанечников // ТВиП.- 1969.- Т.14.- С.156-161.
  33. Медведев, А.В. Непараметрические оценки плотности вероятности и ее производных / А.В. Медведев // Автоматизация промышленного эксперимента.- Фрунзе: Илим.- 1973.- с. 22-31.
  34. Надарая Э.А. Об оценке регрессии / Э.А. Надарая // ТВиП.- 1964.- Т. 9, №1.- С. 157-159.
  35. Sage, A.P. System Identification / A.P. Sage, J.L. Melsa.- [s.l.]:Chu I.- 1971.- P.221.
  36. Иванова, Н.В. Определение параметров сглаживания в непараметрических оценках функции плотности по выборке / Н.В. Иванова, К.Т. Протасов // Математическая статистика и ее приложения. Томск: изд-во Томск, гос. Ун-та.- 1982.- Вып. 8.- С. 50-65.
  37. Bellman, R.E. Dynamic Programming / R.E.Bellman.- NJ,Princeton: Princeton University Press.- 1957.- P.392.
  38. McLachlan, G.L. Mixture Models: Inference and Applications to Clustering / G.L. McLachlan, K.E. Basford.- New York: Marcel Dekker.- 1987.- P.253.
  39. Blei, D.M.Latent dirichlet allocation / D.M. Blei, A.Y. Ng, M.I. Jordan // J. Machine Learn. Res.- 2003.- Vol.3.- P. 993–1022.
  40. Коплярова, Н.В. О непараметрических моделях в задаче диагностики электрорадиоизделий / Н.В.Коплярова, В.И.Орлов, Н.А.Сергеева, В.В.Федосов // Заводская лаборатория: диагностика материалов.- 2014.- №80(7).- С.37-77.
  41. Li, W. Pachinko allocation: Dag-structured mixture models of topic correlations / W. Li, A. McCallum // Proc. 23rd Internat. Conf. on Machine Learning.- [s.l.]:[s.n.]- 2006.- P. 577–584.
  42. Welling, M. Exponential family harmoniums with an application to information retrieval / M. Welling, M. Rosen-Zvi, G. Hinton // Adv. Neural Inform. Process. Systems.- 2005.- Vol.17.- P. 1481–1488.
  43. Mohd, W.M.B.W. An Improved Parameter less Data Clustering Technique based on Maximum Distance of Data and Lioyd k-means Algorithm / W.M.B.W. Mohd, A.H. Beg, T. Herawan, K.F. Rabbi // First World Conference on Innovation and Computer Sciences (INSODE 2011).- [s.l.]:[s.n.]-2012.- Vol.1.- P. 367–371, DOI: 10.1016/j.protcy.2012.02.076

44. Долгушев, А.В. К вопросу об алгоритмической сложности одной задачи кластерного анализа / А.В. Долгушев, А.В. Кельманов // Дискретный анализ и исследование операций.- 2010.- Т.17, № 2.- С. 39–45
45. Bandyopadhyay, S. An evolutionary technique based on K-Means algorithm for optimal clustering / S. Bandyopadhyay, U. Maulik // Information Sciences.- 2002.- Vol. 146.- P.221-237.
46. Zhou, H. Accurate integration of multi-viewrange images using k-means clustering / H. Zhou, Y. Liu // Pattern Recognition.- 2008.- Vol. 41.- P.152-175.
47. Dunham, M.H. Data Mining: Introductory and advance topics / M.H. Dunham.- N.J.:Prentice Hall.- 2003.- P.315.
48. Lloyd, S.P. Least Squares Quantization in PCM / S.P. Lloyd // IEEE Transactions on Information Theory.- 1982.- Vol. 28.- P. 129-137.
49. Загоруйко, Н.Г. Прикладные методы анализа данных и знаний / Н.Г. Загоруйко.- Новосибирск: ИМ СО РАН.- 1999. С. 270.
50. Chiang, M. A time-efficient pattern reduction algorithm for k-means clustering / M. Chiang, C. Tsai, C. Yang // Information Sciences.- 2011.- Vol. 181.- P.716-731.
51. Xu, R. Survey of clustering algorithms / R. Xu, D.Wunsch // IEEE Transaction on Neural Netowrks.- 2005.- Vol.16, issue 3.- P. 645-678.
52. Кельманов, А.В. Приближенный полиномиальный алгоритм для одной задачи бикластеризации последовательности / А.В. Кельманов, С.А. Хамидуллин // Журнал вычислительной математики и математической физики.- 2015.- Т.55, № 6.- с.1076–1085.
53. Кельманов, А.В. Квадратичные евклидовы задачи би-разбиения множества и последовательности: сложность и алгоритмы с оценками качества решения / А.В. Кельманов // Материалы VI международной конференции «Проблемы оптимизации и экономические приложения», Омск, 28 июня-4 июля 2015, Омск: Омский государственный университет.- 2015.- С. 36-41.
54. Hagen, L. New spectral methods for ratio cut partitioning and clustering / L. Hagen, A.B. Kahng // IEEE Trans. Comput.-Aid. Des. Integrated Circuits Systems.- 1992.- Vol. 11, issue 9.- P. 1074–1085.
55. Ng, A.Y. On spectral clustering: Analysis and an algorithm / A.Y. Ng, M.I. Jordan, Y. Weiss // Adv. Neural Inform. Process. Systems.- [s.l.]:MIT Press.- 2001.- Vol. 14.- P. 849–856.
56. Hofmann, T. Pairwise data clustering by deterministic annealing / T. Hofmann, J.M. Buhmann // IEEE Trans. Pattern Anal. Machine Intell.- 1997.- Vol.19, issue 1.- P. 1–14.
57. Pavan, M. Dominant sets and pairwise Clustering / M.Pavan, M.Pelillo // IEEE Trans. Pattern Anal. Machine Intell.- 2007.- Vol.29, issue 1.- P.167–172.

58. Motzkin, T.S. Maxima for graphs and a new proof of a theorem of Turan / T.S. Motzkin, E.G. Straus // Canadian J. Math.- 1965.- Vol.17.- P.533–540.
59. Сергиенко, И.В. Математические модели и методы решения задач целочисленной оптимизации / И.В. Сергиенко, 2-е изд., доп. и перераб.- Киев: Наукова думка.- 1988.- С.472.
60. Береснев, В.Л. Экстремальные задачи стандартизации / В.Л. Береснев, Э.Х. Гимади, В.Т. Дементьев.- Новосибирск: Наука, 1978.- 333 С.
61. Гимади, Э. Х. Задача стандартизации с данными произвольного знака и связными, квазивыпуклыми и почти квазивыпуклыми матрицами / Э.Х. Гимади // Управляемые системы. Сб. науч. тр. Вып. 27. — Новосибирск: Ин-т математики СО АН СССР.- 1987.- С. 3-11.
62. Гимади, Э.Х. Эффективные алгоритмы для решения многоэтапной задачи размещения на цепи / Э.Х.Гимади // Дискретн. анализ и исслед. Опер..- 1995.- том 2, № 4.- С. 13–31.
63. Васильев, И.Л. Новые нижние оценки для задачи размещения с предпочтениями клиентов / И.Л. Васильев, К.Б. Климентова, Ю.А. Кочетов // Журнал вычислительной математики и математической физики.- 2009.- т.49, вып. 6.- С.1055-1066.
64. Алексеев, О.Г. Некоторые алгоритмы решения задачи о покрытии и их экспериментальная проверка на ЭВМ / О.Г. Алексеев, В.Ф. Григорьев // Журнал вычислительной математики и математической физики. - 1984. - Т.24, №10.- С. 1565-1570.
65. Агеев, А.А. Полиномиальный алгоритм решения задачи размещения на цепи с одинаковыми производственными мощностями предприятий / А.А. Агеев, Э.Х. Гимади, А.А. Курочкин // Дискретный анализ и исследование операций.- 2009.- Т.16, № 5.- С. 3–18.
66. Браверман, Э.М. Структурные методы в обработке эмпирических данных / Э.М. Браверман, И.Б. Мучник.- М.: Наука.- 1983.- С.464.
67. Кочетов, Ю.А. Методы локального поиска для дискретных задач размещения: дис. ... доктора физ.-мат. Наук: 05.13.18: защищена 19.01.2010.- Новосибирск: Институт математики им.Соболева.- 2010.- С.259.
68. Ausiello, G. Local Search, Reducibility and Approximability of NP-optimization Problems / G. Ausiello, M. Protasi // Information Processing Letters.- 1995.- Vol.54.- P. 73-79.
69. Ganti, V. Clustering large datasets in arbitrary metric spaces / V. Ganti, R. Ramakrishnan, J. Gehrke, A. Powell, J. French. // Proc. 15th Int. Conf. Data Engineering.- [s.l.]:[s.n.]- 1999.- P. 502-511.
70. Pelleg, D. X-means: Extending k-means with efficient estimation of the number of clusters / D. Pelleg, A. Moore // Seventeenth Internat. Conf. on Machine Learning.- 2000.- P. 727–734.
71. Kaufman, L. Finding groups in data: an introduction to cluster analysis / L. Kaufman, P.J.

- Rousseeuw.- New York:Wiley.- 1990.- P.368.
72. Roberts, S.J. Minimum-entropy data clustering using reversible jump Markov chain Monte Carlo / S.J. Roberts, C. Holmes, D. Denison // Proc. Internat. Conf. Artificial Neural Networks.- [s.l]:[s.n.]- 2001.- P. 103–110.
  73. Still, S. Geometric Clustering using the Information Bottleneck method / S. Still, W. Bialek, L. Bottou // Advances In Neural Information Processing Systems 16 / Eds.:S. Thrun,L. Saul, and B. Scholkopf.- Cambridge:MIT Press.- 2004 [Электронный ресурс] Режим доступа URL <http://papers.nips.cc/paper/2361-geometric-clustering-using-the-information-bottleneck-method.pdf> (дата обращения 19.06.2015)
  74. Tishby, N. The information bottleneck method / N. Tishby, F.C. Pereira, W. Bialek // Proc. 37th Allerton Conf. on Communication, Control and Computing.- Monticello:[s.n.]-1999.- P. 368–377.
  75. Slonim, N. Document clustering using word clusters via the information bottleneck method / N. Slonim, N. Tishby // ACM SIGIR 2000.- 2000.- P. 208–215.
  76. Kanungo, T. An efficient K-means clustering algorithm: analysis and implementation / T. Kanungo, D. Mount, N.S. Netanyahu, C. Piatko, R. Silverman, A. Wu // IEEE Trans. Pattern Anal. Mach. Intell.- 2002.- Vol.24, issue 7.- P. 881-892.
  77. Likas, A. The global k-means clustering algorithm / A. Likas, M. Vlassis, J. Verbeek // Pattern Recognition.- 2003.- Vol.36.- P. 451-461.
  78. Charalampidis, D. A modified K-means algorithm for circular invariant clustering / D. Charalampidis // IEEE Trans. Pattern Anal. Mach. Intell.- 2005.- Vol.27, issue 12.- P.1856-1865.
  79. Selim, S.Z. K-means type algorithms: a generalized convergence theorem and characterization of local optimality / S.Z. Selim, M.A. Ismail // IEEE Trans. Pattern Anal. Mach. Intell.- 1984.- Vol. 6.- P.81-87.
  80. Spath, H. Cluster Analysis Algorithms / H. Spath.- Chichester: Ellis Horwood.- 1989.- P. 226.
  81. Chang, D. A genetic algorithm with gene rearrangement for K-means clustering / D. Chang, D. Xian, W. Chang // Pattern Recognition.- 2009.- Vol. 42.- P.1210-1222.
  82. Sheng, W. A genetic k-medoids clustering algorithm / W. Sheng, X. Liu //Journal of Heuristics.- 2006.-Vol.12, No.6.- P. 447-466.
  83. Lim, A. A fixed-length subset genetic algorithm for the p-median problem / A. Lim, Z. Xu // Lecture notes in computer science.- 2003.- Vol. 2724.- P. 1596-1597.
  84. Maulik, U. Genetic Algorithm-Based Clustering Technique / U. Maulik, S. Bandyopadhyay // Pattern Recognition.- 2000.- Vol. 33.- P. 1455-1465.
  85. Kalyani, S. Particle swarm optimization based K-means clustering approach for security assessment in power systems / S. Kalyani, K.S. Swarup // Expert Systems with Applications.-

- 2011.- Vol.38.- P.10839-10846.
86. Bagirov, A.M. Fast modified global k-means algorithm for incremental cluster construction / A.M. Bagirov, J. Ugon, D. Webb // *Pattern Recognition*.- 2011.- Vol. 44.- P. 866-876.
  87. Bagirov, A.M. Modified global k-means algorithm for sum-of-squares clustering problem / A.M. Bagirov // *Pattern Recognition*.- 2008.- Vol. 41.- P.3192–3199.
  88. Deza, M.M. *Encyclopedia of Distances* / M. M. Deza, E. Deza.- Berlin, Heidelberg: Springer-Verlag.- 2009.- P.590.
  89. Mao, J. A self-organizing network for hyper-ellipsoidal clustering (HEC) / J. Mao, A.K. Jain // *IEEE Trans. Neural Networks*.- 1996.- Vol. 7 (January).- P. 16–29.
  90. Linde, Y. An algorithm for vector quantizer design / Y. Linde, A. Buzo, R. Gray // *IEEE Trans. Comm.*- 1980.- Vol.28.- P. 84–94.
  91. Banerjee, A. Clustering with bregman divergences / A. Banerjee, S. Merugu, I. Dhillon, J. Ghosh // *J. Machine Learn. Res.*- 2004.- P. 234–245.
  92. Capoleas, V. *Geometric Clusterings* / V. Capoleas, G. Rote, G.J. Woeginger // *J. Algorithms*.- 1991.- Vol.12.- P.341-356.
  93. Ball, G.H. ISODATA. An iterative method of multivariate analysis and pattern classification / G.H. Ball, D.J. Hall // 1966 *IEEE Int. Commun. Conf.*, Philadelphia.- New York: Digest of Techn. Pap.- 1966.- P.116–117.
  94. Гамаюн, И. П. Некоторые методы решения задач классификации объектов и группировки параметров / И.П. Гамаюн, О.Н. Безменова // *Вестник Нац. техн. ун-та ХПИ: сб. науч. тр. Темат. вып. : Системный анализ, управление и информационные технологии*.- Харьков : НТУ ХПИ.- 2013.- № 62 (1035).- С. 70-77.
  95. Pelleg, D. Accelerating exact k-means algorithms with geometric reasoning / D. Pelleg, A. Moore // *Proc. Fifth Internat. Conf. on Knowledge Discovery in Databases*.- [s.l.]: AAAI Press.- 1999.- P. 277–281.
  96. Scholkopf, B. Nonlinear component analysis as a kernel eigenvalue problem / B. Scholkopf, A. Smola, K.-R. Muller // *Neural Comput.*- 1998.- Vol, 10, issue 5.- P. 1299–1319.
  97. Миркин, В.Г. Шкалы упорядочения / В.Г. Миркин, Н.В. Высоцкая // *Моделирование в экономических исследованиях*.- Новосибирск: Наука, Сибирское отделение, 1978. -С.109-120.
  98. Diday, E. The dynamic clusters methods in nonhierarchical clustering / E. Diday // *Int. J. Comput. and Inf. Sci.*- 1973.- Vol.2, No 1.- P.61–68.
  99. Datta, R. Image retrieval: Ideas, influences, and trends of the new age / R. Datta, D. Joshi, J. Li, J.Z. Wang // *ACM Computing Surveys*.- 2008.- Vol. 40, issue 2.- Article 5.- P.5:1-5.60, DOI:

- 10.1145/1348246.1348248.
100. Lowe, D.G. Distinctive image features from scale-invariant keypoints / D.G. Lowe // *Internat. J. Comput. Vision.*- 2004.- Vol. 60, issue 2.- P. 91–110.
  101. Moore, A.W. Very fast EM-based mixture model clustering using multiresolution kd-trees / A.W. Moore // *NIPS II.*- [s.l.]:MIT Press.- 1998.- P. 543–549.
  102. Muja, M. Fast approximate nearest neighbors with automatic algorithm configuration / M. Muja, D.G. Lowe // *Proc. Internat. Conf. on Computer Vision Theory and Applications (VISAPP'09).*- [s.l.]:INSTICC Press.- 2009. P.331-340.
  103. Zhang T. BIRCH: An Efficient Data Clustering Method for Very Large Databases / T. Zhang, R. Ramakrishnan, M. Livny // *Proceedings of the 1996 ACM SIGMOD international conference on Management of data (SIGMOD '96).*- New York: ACM.- 1996, P. 103-114, DOI: 10.1145/233269.233324.
  104. Steinbach, M. A comparison of document clustering techniques / M. Steinbach, G. Karypis, V. Kumar // *KDD Workshop on Text Mining.*- 2000 [Электронный ресурс] Режим доступа URL [http://www.cs.cmu.edu/~dunja/KDDpapers/Steinbach\\_IR.pdf](http://www.cs.cmu.edu/~dunja/KDDpapers/Steinbach_IR.pdf) (дата обращения 19.06.2015).
  105. Karypis, G. A fast and high quality multilevel scheme for partitioning irregular graphs / G.Karypis, U.Fayyad, C.Reina // *SIAM J. Sci. Comput.*- 1998.- Vol.20, issue 1.- P. 359-392, DOI: 10.1137/S1064827595287997.
  106. Bradley, P.S. Scaling clustering algorithms to large databases / P.S. Bradley P.S., U. Fayyad, C. Reina // *Proceedings of the 4th Int. Conf. on Knowledge Discovery and Data Mining.*- New York: [s.n.].-1998.- P. 9-15.
  107. Fisher, D.H. Knowledge acquisition via incremental conceptual clustering / D.H. Fisher // *Machine Learn.*- 1987.- P. 139–172.
  108. Guha, S. CURE: An efficient clustering algorithm for large databases / S. Guha, R. Rastogi, K. Shim // *SIGMOD '98 Proceedings of the 1998 ACM SIGMOD international conference on Management of data .*- New York: ACM.- 1998. P. 73–84.
  109. Kollios, G. Efficient biased sampling for approximate clustering and outlier detection in large data sets / G. Kollios, D. Gunopulos, N. Koudas, S. Berchtold // *IEEE Trans. Knowledge Data Eng.*- 2003.- Vol. 15, No.5.- P. 1170–1187.
  110. Ng, R. CLARANS: a method for clustering objects for spatial data mining / R. Ng, J. Han // *IEEE Trans. Knowl. Data Eng.*- 2002.- Vol.14.- P.1003–1016.
  111. Eisenbrand, F. Approximating connected facility location problems via random facility sampling and core detouring / F. Eisenbrand, F. Grandoni, T. Rothvoss, G. Schafer // *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*

- 2008).- New York: ACM.- 2008.- P. 1174–1183.
112. Bubeck, S. How the Initialization Affects the Stability of the k-Means Algorithm / S. Bubeck, M. Meila, U. von Luxurg // *Probability and Statistics*.- 2012.- Vol.16.- P. 436-452.
  113. Drezner, Z. Facility location: applications and theory / Z. Drezner, H. Hamacher.- Berlin:Springer-Verlag.- 2004.- P.460.
  114. Farahani, R. Facility location: Concepts, models, algorithms and case studies / R. Z. Farahani and M. Hekmatfar (eds.).- Berlin Heidelberg:Springer-Verlag.- 2009.- P.549.
  115. Бельц, Е.А. Оптимизация размещения предприятий с учетом минимально допустимых расстояний / Е.А. Бельц, А.А. Колоколов // *Вестн. Ом. ун-та*.- 2012.- No 4.- С. 13–16.
  116. Кочетов, Ю.А. Двухуровневые задачи размещения / Кочетов Ю.А. // *Труды ИВМ и МГ / Серия Информатика*.- Новосибирск: Изд. ИВМиМГ СО РАН.- 2007.- Вып. 7.- С. 97–104.
  117. Климентова, К.Б. Оценки оптимальных значений и методы решения задач размещения с предпочтениями клиентов: дис. ... канд. физ.-мат. наук.- Иркутск: Институт динамики систем и теории управления СО РАН.- 2010.- С.124.
  118. Ottaviano, G.I.P. New economic geography: what about the N? / G.I.P. Ottaviano, J.-F. Thisse // *Environment and Planning A*.- 2005.- Vol 37, issue 10.- P. 1707–1725.
  119. Hansen, P. Cluster analysis and mathematical programming / P. Hansen, B. Jaumard // *Mathematical Programming*.- 1997.- Vol. 79.- P.191-215.
  120. Hansen, P. Variable neighborhood search for the p-median / P. Hansen, N. Mladenovic // *Location Science*.- 1997.- Vol. 5, No. 4.- P. 207-226.
  121. Rosing, R.E. Towards the solution of the (generalized) Weber problem / R.E. Rosing// *Environment and Planning B: Environment and Design*.- 1991.- Vol.18.- P.347-360.
  122. Hall, R.W. Median mean and optimum as facility locations / R.W. Hall // *Journal of Regional Science*.-1988.- Vol. 28.- P. 65-81.
  123. Boltyanski, Y. Geometric Methods and Optimization Problems (Combinatorial Optimization) / Boltyanski Y., Martini H., Soltan V.- vol. 4.- Dordrecht: Kluwer Academic Publishers.- 1999.- P.432.
  124. Volek, J. Location analysis - Possibilities of use in public administration / J. Volek // *Verejna sprava*.- Pardubice: Univerzita Pardubice.- 2006.- P. 84-85.
  125. Teodorovic, D. Transportne mreze, Poglavlje 9: Lokacijski problem / D. Teodorovic.- Beograd: Saobranajni fakultet.- 2009.- P. 389-399.
  126. Watanabe, D.Generalized Weber Model for Hub Location of Air Cargo / D. Watanabe, T. Majima, K. Takadama, M. Katuhara // *The Eighth International Symposium on Operations Research and Its Applications (ISORA'09)*.- Zhangjiajie.- 2009.- P. 124–131.

127. Гимади, Э. Х. Обоснование априорных оценок качества приближенного решения задачи стандартизации / Э.Х. Гимади // Управляемые системы: Сб. науч. тр.- Новосибирск: Ин-т математики СО АН СССР.- 1987.- Вып. 27.- С. 12-27.
128. Гончаров, Е.Н. Поведение вероятностных жадных алгоритмов для многостадийной задачи размещения / Е.Н. Гончаров, Ю.А. Кочетов // Дискретный анализ и исследование операций. Серия 2.- 1999.- Т.6, № 1.- С.12-32.
129. Cooper, L. The transportation-location problem / L.Cooper //Oper. Res.- 1972.- Vol.20, No.1.- P. 94-108.
130. Hansen, P. Analysis of global k-means, an Incremental Heuristic for Minimum Sum of Squares / P. Hansen, E. Ngai, B. Cheung, N. Mladenovic // Journal of Classification.- 2005.- Vol. 22(3).- P. 287-310.
131. Kuehn, A.A. A heuristic program for locating warehouses/A.A.Kuehn, M.J.Hamburger// Management Science.- 1963.- 9(4).- P.643-666.
132. Brazil, M. On the History of the Euclidean Steiner Tree Problem / M.Brazil, R.L.Graham, D.A.Thomas, M.Zachariasen // Archive for History of Exact Sciences.- Berlin Heidelberg: Springer.- 2014.- Vol. 68.- P. 327-354.
133. Torricelli, E. Opere de Evangelista Torricelli/E.Torricelli, G.Loria, G.Vassura // English edition.-Vol I.- Part 2.- Fa?nza.- 1919.- P. 90-97.
134. Mladenovic, N. Kontinualni lokacijski problemi / N. Mladenovic.- Beograd: Matematicki institut SANU.- 2004.- P.50.
135. Kirszenblat, D. Dubins networks: Thesis / D. Kirszenblat.- Melbourne: Department of Mathematics and Statistics of the University of Melbourne.- 2011. -P.56.
136. Pergamo de, A. Ad lib de max et min, Apendix Monitvm. Probl.I, Prop VI. / A. de Pergamo, V. Viviano. // De maximis et minimis geometrica divinatio in quintum conicorum Apollonio Pergaei.- Florentiae: Apud Iosef Cocchini1659.- P. 143-150.
137. Heinrich, D. 100 Great Problems Of Elementary Mathematics their history and solution/ D. Heinrich.- New York: Dover Publication, Inc.- 1965.- P. 361-363.
138. Sandor, J. On Certain Inequality by Visschers / J. Sandor // Octogon Math. Mag.- 2005.- Vol. 13, issue 2.- P. 1053-1054.
139. Региональная экономика и управление. Учебное пособие в 2 – х частях / Под ред. А.И. Гаврилова. - Н. Новгород: Изд-во ВВАГС.- 2005.- С.260.
140. Hale, T.S. Location science research: a review / T. S. Hale, C. R. Moberg // Annals of Operations Research.- 2003.- Vol. 123.- P.21-35.
141. Kupitz, Y.S. Geometric aspects of the generalized Fermat-Torricelli problem / Y.S. Kupitz, H.



- Martini // *Intuitive Geometry* (Budapest, 1995) / I. Barany and K. Boroczky (Eds.).- Budapest: Bolyai Society Mathematical Studies.- 1997, issue 6.- P. 55-127.
142. Eiselt, H.A. Pioneering developments in location analysis / H.A. Eiselt, V. Marianov // *Foundations of Location Analysis: International Series in Operations Research & Management Science* /Eiselt, H.A., Marianov, V. (eds.) .- Berlin:Springer.- 2011.- Vol. 155.- P. 3-22.
143. Drezner, Z. The Weber problem / Z. Drezner, K. Klamroth, A. Schobel, G. O. Wesolowsky // *Facility Location: Applications and Theory* / Z. Drezner and H. W. Hamacher (eds.).- Berlin:Springer-Verlag.- 2004.-P. 1-36
144. Weiszfeld, E. Sur le point sur lequel la somme des distances de n points donnees est minimum/ E. Weiszfeld // *Tohoku Mathematical Journal*.- 1937.- Vol. 43, No.1.- P.335–386.
145. Vazsonyi, A. Pure mathematics and the Weiszfeld algorithm/A. Vazsonyi // *Decis. Line*.- 2002.- Vol. 33.- P.12-13.
146. Sturm, R. Ueber den Punkt kleinster Entfernungssumme von gegebenen Punkten / R. Sturm // *J. Rein. Angew. Math*.- 1884.- Vol.97.- P. 49–61.
147. Beck, A. Weiszfeld’s Method: Old and New Results / A. Beck, S. Sabach // *J. Optim. Theory Appl*.- 2014.- [Электронный ресурс] Режим доступа DOI 10.1007/s10957-014-0586-7 (дата обращения 19.06.2015).
148. Drezner, Z. The fortified Weiszfeld algorithm for solving the Weber problem / Z. Drezner // *IMA Journal of Management Mathematics*.- 2013.- Vol.26.- P.1-9. DOI: 10.1093/imaman/dpt019
149. Deza, M. Distances in pattern recognition / Deza M.- 2007 [Электронный ресурс]. Режим доступа URL [www.picb.ac.cn/pattern07/SLIDES/2007\\_summerschool\\_Shanghai\\_Deza.pdf](http://www.picb.ac.cn/pattern07/SLIDES/2007_summerschool_Shanghai_Deza.pdf) (дата обращения 09.08.2013)
150. Deza, M.M. Metrics on Normed Structures / M. M. Deza, E. Deza. // *Encyclopedia of Distances*.- Berlin Heidelberg:Springer.- 2013.- P.89-99, DOI: 10.1007/978-3-642-30958-85.
151. Masuyama, S. The Computational Complexity of the m-Center Problems on the Plane / S. Masuyama, T. Ibaraki, T. Hasegawa // *The Transactions of the Institute of Electronics and Communication Engineers of Japan*.- 1981.- Vol. 64E.- P. 57-64
152. Megiddo, N. On the Complexity of Some Common Geometric Location Problems / N. Megiddo, K. Supowit // *SIAM Journal on Computing*.- 1984.- Vol. 13.- P. 182-196.
153. Забудский, Г.Г. Решение задачи размещения в евклидовом пространстве с запрещенной областью / Г.Г. Забудский, И.В. Нежинский // *Вестник Омского университета*.- 1999.- Т.2.- С. 17-19.
154. Morris, J.G. Convergence of the Weiszfeld algorithm for Weber problems using a generalized

- "distance" function / J. G. Morris // *Operations Research*.- 1981.- Vol.29.- P.37-48.
155. Wesolowsky, G.O. A Nonlinear Approximation Method for Solving a Generalized Rectangular Distance Weber Problem / G.O. Wesolowsky, R.F. Love // *Management Science*.- 1972.- Vol. 18, No.11.- P. 656-663.
  156. Kazakovtsev, L.A. Adaptation of the Probability Changing Method for Weber Problem with an Arbitrary Metric / L.A.Kazakovtsev // *Facta Universitatis (Nis) Ser. Math. Inform.*- 2012.- Vol. 27, No.2.- P.239-254.
  157. Antamoshkin, A.N. Random Search Algorithm for the p-Median Problem / A.N. Antamoshkin, L.A. Kazakovtsev // *Informatika*.- 2013.- Vol. 37, issue 3.- P.267–278.
  158. Staminirovic, P.S. Single-facility Weber location problem based on the Lift metric / Predrag S. Staminirovic, Maria Ciric, Lev A. Kazakovtsev, Idowu A. Osinuga // *Facta Universitatis (Nis), Ser. Math. Inform.*- 2012.- Vol. 27, issue 2.- P. 175-190.
  159. Gugat, M. Weber problems with mixed distances and regional demand / M. Gugat, B. Pfeiffer // *Math. Meth. Oper. Res.*- 2007.- Vol. 66.- P. 419–449.
  160. Bischoff, M. The multi-facility location-allocation problem with polyhedral barriers / M. Bischoff, T. Fleischmann, K. Klamroth // *Computers and Operations Research*.- 2009.- Vol. 36.- P. 1376-1392.
  161. Саватеев, А.В. Задача многомерного размещения и ее приложения: теоретико-игровой подход/ А.В. Саватеев: дис. .. д-ра физ.-мат. наук.- Москва: Центральный экономико-математический институт.- 2013.- С.268.
  162. Liao, K. A clustering-based approach to the capacitated facility location problem / K. Liao, D. Guo // *Transactions in GIS*.- 2008.- Vol. 12, issue 3.- P.323-339.
  163. Rebreyend, P. A Computational Comparison of Different Algorithms for Very Large p-median Problems / P. Rebreyend, L. Lemarchand, R. Euler // *LNCS*.- 2015.- Vol. 9026 (EvoCOP 2015).- P. 13–24, DOI: 10.1007/978-3-319-16468-7\_2.
  164. Goodchild, M.F. ILACS: A location-allocation model for retail site selection / M.F. Goodchild // *Journal of Retailing*.- 1984.- Vol. 60.- P. 84-100.
  165. Ceselli, A. A branch-and-price algorithm for the capacitated p-median problem / A. Ceselli, G. Righini // *Networks*.- 2005.- Vol. 45, issue 3.- P. 125–42.
  166. Horner, M. W. A combined harvesting and transport planning within a sugar value chain / M.W. Horner, W.E. O’Kelly // *Journal of the Operational Research Society*.- 2005.- Vol. 57.- P.367–376.
  167. Haghifam, M. R. Optimal location and sizing of HV/MV substations in uncertainty-load environment using genetic algorithm / M. R. Haghifam, M. Shhabi // *Electric Power Systems*

- Research.- 2002.- Vol. 63.- P. 37–50.
168. Kalvenes, J. Hierarchical cellular network design with channel allocation / J. Kalvenes, E.O. Kennington // *European Journal of Operational Research*.- 2005.- Vol.160.- P.3–18.
  169. Kratica, J. Genetic algorithm for solving uncapacitated multiple allocation hub location problem / J. Kratica, Z. Stanimirovic, D. Tomic, V. Filipovic // *Computing and Informatics*.- 2005.- Vol.24.- P.414–26.
  170. Ehrgott, M. Location of rescue helicopters in South Tyrol / M. Ehrgott // *International Journal of Industrial Engineering – Theory, Applications and Practice*.- 2002.- Vol.9.- P.16–22.
  171. Seden-Noda, A. An algorithm for the biobjective integer minimum cost flow problem / A. Seden-Noda, C. Gonzalez-Martin // *Computers and Operations Research*.- 2001.- Vol.28.- P.139–156.
  172. Bertsimas, D. Robust discrete optimization and network flows / D. Bertsimas, M. Sim // *Mathematical Programming*.- 2003.- Vol. 98.- P. 49–71.
  173. Vinod, H.D. Integer programming and the theory of grouping / H.D. Vinod // *Journal of the American Statistical Association*.- 1969.- Vol.64.- P. 506–519.
  174. ReVelle, C.S. Central facilities location / C.S. ReVelle, R.W. Swain // *Geographical Analysis*.- 1970.- Vol.2.- P. 30-42.
  175. Алексеева, Е.В. Генетический локальный поиск для задачи о р-медиане с предпочтениями клиентов / Е.В. Алексеева, Ю.А. Кочетов // *Дискретный анализ и исследование операций. Серия 2*.- 2007.- Т.14, № 1.- С.3–31.
  176. Garey, M.R. *Computers and Intractability: A Guide to the Theory of NP-Completeness* / M.R. Garey, D.S. Johnson // *A Series of Books in the Mathematical Sciences*.- San Francisco: W. H. Freeman and Co.- 1979.- 338 P.
  177. Wu, L.-Y. Capacitated facility location problem with general setup cost / L.-Y. Wu., X.-S. Zhang, J.-L. Zhang // *Computers and Operations Research*.- 2006.- Vol. 33.- P.1226–1241.
  178. Franca, P.M. An adaptive tabu search algorithm for the capacitated clustering problem / P.M. Franca, N.M. Sosa, V. Pureza // *International Transactions in operational Research*.- 1999.- Vol. 6.- P. 665–678.
  179. Mulvey, J.M. Solving capacitated clustering problems / J.M. Mulvey, M.P. Beck // *European Journal of Operational Research*.- 1984.- Vol.18.- P.339–348.
  180. Brusco, M.J. Optimal Partitioning of a Data Set Based on the p-Median Model / M.J. Brusco, H.F. Kohn // *Psychometrica*.- 2008.- Vol. 73, No. 1.- P.89–105, DOI: 10.1007/S11336-007-9021-4.
  181. Teitz, M.B. Heuristic methods for estimating the generalized vertex median of a weighted graph

- / M.B. Teitz, P. Bart // *Operations Research*.- 1968.- Vol.16.- P. 955-961.
182. Mulvey, J.M. Cluster analysis: An application of Lagrangian relaxation / J.M. Mulvey, H.P. Crowder // *Management Science*.- 1979.- Vol.25.- P. 329-340.
183. Klastorin, T.D. The p-Median Problem for Cluster Analysis: A Comparative Test Using the Mixture Model Approach / T.D. Klastorin // *Management Science*.- 1985.- Vol.31, No.1.- P. 84-95.
184. Hansen P. Solving large p-median clustering problems by primaldual variable neighborhood search / P. Hansen, J. Brimberg, D. Urosevic, N. Mladenovic // *Data Mining and Knowledge Discovery*.- 2009.- 19, No. 3.- P. 351–375.
185. Mladenovic, N. Variable neighborhood search / N. Mladenovic, P. Hansen // *Comput. Oper. Res.* 1997.- Vol.24.- P.1097–1100.
186. Hansen, P. Variable neighborhood search: principles and applications / P. Hansen, N. Mladenovic // *Eur. J. Oper. Res.*- 2001.- Vol.130.- P.449–467.
187. Hansen, P. Variable Neighborhood Search / P. Hansen, N. Mladenovic // *Search Methodology* / E.K.Brucke, G.Kendall [eds].- Springer US.- 2005.- P. 211-238, doi: 10.1007/0-387-28356-0\_8.
188. Lu, Y. Incremental genetic k-means algorithm and its application in gene expression data analysis / Y. Lu, S. Li, F. Fotouhi, Y. Deng, S. Brown // *BMC Bioinformatics*.- 2004. - Vol.5, No.1.- Article 172.- [Электронный ресурс] режим доступа DOI: 10.1186/1471-2105-5-172 (жата обращения 20.06.2015).
189. Vidyasagar, M. Statistical learning theory and randomized algorithms for control / M. Vidyasagar // *IEEE Control Systems*. -1998.- No. 12. -P. 69-85.
190. Граничин, О. Н. Рандомизированные алгоритмы оценивания и оптимизации при почти произвольных помехах / О.Н. Граничин, Б.Т. Поляк.- М.: Наука. -2003.-С. 291.
191. Goldberg, D.E. Genetic algorithm in search, optimization and machine learning / D.E. Goldberg.- MA: Addison-Wesley.- 1989.-P. 432.
192. Kohonen, T. Self-Organization and Associative Memory, 3rd ed. / T. Kohonen // *Springer information sciences series*.- New York:Springer-Verlag.- 1989.- P.312.
193. Kirkpatrick, S. Optimization by simulated annealing / S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi // *Science*.- 1983.- Vol. 220(4598).- P. 671–680.
194. Krishna, K. Genetic K-means algorithm / K. Krishna, M. Murty // *IEEE Transaction on System, Man and Cybernetics - Part B*.- 1999.- Vol.29.- P. 433-439.
195. Holland, J. Adaptation in Natural and Artificial Systems. Ann / J. Holland.- Arbor: University of Michigan Press.- 1975.- P.183.
196. Еремеев, А.В. Генетический алгоритм с турнирной селекцией как метод локального

- поиска / А.В. Еремеев // Дискретный анализ и исследование операций.- 2012.- Т. 19, № 2  
.- С. 41-53.
197. Akhmedova, Sh.A. New optimization metaheuristic based on co-operation of biology related algorithms / Sh.A. Akhmedova, E.S. Semenkin // Vestnik SibGAU.- No. 4 (50).- 2013.- P. 92-99.
  198. Семенкин, Е.С. Коэволюционный генетический алгоритм решения сложных задач условной оптимизации / Е.С. Семенкин, Р.Б. Сергиенко // Вестник СибГАУ.- 2009.- №2.- С. 17-21.
  199. Neema, M.N. New Genetic Algorithms Based Approaches to Continuous p-Median Problem / M.N. Neema, K.M. Maniruzzaman, A. Ohgai // Netw. Spat. Econ.- 2011.- Vol.11.- P.83–99, DOI:10.1007/s11067-008-9084-5.
  200. Hong, Y. To combine steady-state genetic algorithm and ensemble learning for data clustering / Y. Hong, S. Kwong // Pattern Recognition Letters.- 2008.- Vol. 29.- issue 9.- P.1416-1423.
  201. Hosage, C.M. Discrete Space LocationAllocation Solutions from Genetic Algorithms / C.M. Hosage, M.F. Goodchild // Annals of Operations Research. -1986.- Vol. 6.- P.35–46.
  202. Chiou, Y. Genetic clustering algorithms / Y. Chiou, L.W. Lan // European Journal of Operational Research.- 2001.- Vol. 135.- P. 413-427.
  203. Bozkaya, B.A. Genetic Algorithm for the p-Median Problem / B. Bozkaya, J. Zhang, E.Erkut // Facility Location: Applications and Theory / Z. Drezner, H. Hamacher [eds].- New York: Springer.- 2002.- P. 179-205.
  204. Alp, O. An Efficient Genetic Algorithm for the p-Median Problem / O. Alp, E. Erkut, Z. Drezner // Annals of Operations Research.- 122 (1-4).- 2003.- P. 21–42, doi 10.1023/A:1026130003508
  205. Dvoretz, J. Compatibility-based genetic algorithm: A new approach to the p-median problem : Technical report / J. Dvoretz.- Evanston,IL: Department of Industrial Engineering and Management Sciences, Northwestern University.- 1999. 162 P.
  206. Lorena, L.A.N. Constructive genetic algorithm for clustering problems / L.A.N. Lorena, J.C. Furtado // Evolutionary Computation.- 2001.- Vol. 9.- P. 309-328.
  207. Алексеева, Е.В. Алгоритмы локального поиска для задачи о р-медиане с предпочтениями клиентов: дис. ... канд. физ.-мат. наук.- Новосибирск: Институт математики им. С.Л.Соболева СО РАН.- 2007.-С. 92.
  208. Boese, K.D. A new adaptive multi-start technique for combinatorial global optimizations / K.D. Boese, A.B. Kahng, S. Muddu // Oper. Res. Lett.- 1994.- Vol. 16, No 2.- P. 101-114.
  209. Drezner, Z. New heuristic algorithms for solving the planar p-median problem / Z. Drezner, J.

- Brimberg, N. Mladenovic, S. Salhi // *Computers & Operations Research*.- 2013.- Vol. 40, issue 1.- 427-437.
210. Resende, M. On the implementation of a swap-based local search procedure for the p-median problem / M. Resende, R. Werneck // *Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments (ALENEX'03)* / [Edited by] Richard E. Ladner.- Philadelphia: SIAM.- 2003.- P. 119-127.
211. Reeves, C.R. Genetic algorithms for the operations researcher / C.R. Reeves // *INFORMS Journal of Computing*.- 1997.- Vol.9, issue 3.- P.231-250.
212. Agarwal, C.C. Optimized crossover for the independent set problem/ C.C.Agarwal, J.B.Orlin, R.P. Tai // *Operations research*.- 1997.- Vol. 45, issue 2.- P. 226 - 234.
213. Ereemeev, A.V. Optimal recombination in genetic algorithms for combinatorial optimization problems, part 1 / A.V. Ereemeev, J.V. Kovalenko // *Yugoslav Journal of Operations Research*.- 2014.- Vol.24, issue 1.- P. 1-20, DOI:10.2298/YJOR130731040E.
214. Kochetov, Y. Large Neighborhood Local Search for the p-Median Problem / Y. Kochetov, E. Alekseeva, T. Levanova, M. Loresh // *Yugoslav Journal of Operations Research*.- 2005.- Vol.15, issue 1.- P. 53-63.
215. Muhlenbein, H. The Equation for Response to Selection and Its Use for Prediction / H. Muhlenbein // *Evolutionary Computation*.- 1998.- Vol.5, No. 3.- P. 303-346.
216. Семёнкин, Е.С. Эволюционные методы моделирования и оптимизации сложных систем: Конспект лекций /Е.С. Семёнкин, М.Н. Жукова, В.Г. Жуков, И.А. Панфилов, В.В. Тынченко.- Красноярск: СФУ.- 2007.- 515 С.
217. Hakimi, S.L. Optimum locations of switching centers and the absolute centers and medians of a graph / L.Hakimi. S. // *Operations Research*.- 1964.- Vol. 12, issue 3.- P.450–459.
218. Kariv, O. An Algorithmic Approach to Network Location Problems. II:The P medians / O. Kariv, S.L. Hakimi // *SIAM J. Appl. Math*.- 1979.- Vol. 37.- P.539–560.
219. Resende, M.G.C. Metaheuristic hybridization with Greedy Randomized Adaptive Search Procedures / M.G.C. Resende // *TutORials in Operations Research/ Zhi-Long Chen and S. Raghavan [eds.]*.- INFORMS.- 2008.- P. 295–319.
220. Resende, M.G.C. Scatter search and pathrelinking: Fundamentals, advances, and applications / M.G.C. Resende, C.C. Ribeiro, F.Glover, R. Marti // *Handbook of Metaheuristics [2nd Edition]* / M. Gendreau and J.-Y. Potvin [eds.]-. [s.l.]: Springer.- 2010.- P. 87–107.
221. Rabbani, M. A Novel Approach for Solving a Constrained Location Allocation Problem / M. Rabbani // *International Journal of Industrial Engineering Computations*.- 2013.- Vol. 4, issue 2.- P.203-214.

222. Fathali, J. The p-median and p-center Problems on Bipartite Graphs / J. Fathali, N. J. Rad, S. R. Sherbaf // *Iranian Journal of Mathematical Sciences and Informatics*.- 2014.- Vol. 9, issue 2.- P.37-43.
223. Avella, P. An Aggregation Heuristic for Large Scale p-median Problem / P. Avella, M. Boccia, S. Salerno, I. Vasilyev // *Computers & Operations Research*.- 2012.- Vol. 39, issue 7.- P. 1625–1632, DOI: 10.1016/j.cor.2011.09.016.
224. Avella, P. Computational Study of Large-Scale p-Median Problems / P. Avella, A. Sassano, I. Vasil'ev // *Mathematical Programming*.- 2007.- Vol. 109, issue 1.- P. 89–114, DOI: 10.1007/s10107-005-0700-6.
225. Wesolowsky, G. The Weber problem: History and perspectives / G. Wesolowsky // *Location Science*.- 1993.- No. 1.- P.5-23.
226. Drezner, Z. Trajectory Method for the Optimization of the Multifacility Location Problem with lp Distances / Z. Drezner, Wesolowsky G.O.A // *Management Science*.- 1978.- V.24.- P.1507–1514.
227. Osinuga, I.A. On the Minimum Norm Solution to Weber problem / I.A. Osinuga, O.M. Bamigbola // *SAMSA Conference Proceedings, Windhoek*.- 2007.- P.27-30.
228. Cooper, L. An extension of the generalized Weber problem / L. Cooper // *Journal of Regional Science*.- 1968.- Vol.8, issue 2.- P. 181-197.
229. Kazakovtsev, L.A. Random Constrained Pseudo-Boolean Optimization Algorithm for Multiprocessor Systems and Clusters / L.A. Kazakovtsev // *ICUMT 2012, International Congress on Ultra-Modern Telecommunications*.- S.-Petersburg:IEEE Press.- 2012.- P. 650-656.
230. Казаковцев, Л.А. Параллельный алгоритм случайного поиска с адаптацией для систем с распределенной памятью / Л.А. Казаковцев // *Системы управления и информационные технологии*.- 2012.- № 3(49).- С. 11-15.
231. Dijkstra, E.W. A note on two problems in connexion with graphs / E.W. Dijkstra // *Numerische Mathematik*.- 1959.- V. 1.- P. 269-271.
232. Левитин, А.В. Глава 9. Жадные методы: Алгоритм Дейкстры / А.В. Левитин // *Алгоритмы: введение в разработку и анализ* М.: Вильямс.- 2006.- С. 189-195.
233. Aurenhammer, F. Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure / F. Aurenhammer // *ACM Computing Surveys*.- 1991.- Vol.23(3).- P. 345-405.
234. Антамошкин А.Н. Оптимизация функционалов с булевыми переменными.- Томск: Изд-во Томского университета.- 1987.- 104 С.
235. Antamoshkin, A. *Systems Analysis, Design and Optimization. An Introduction* / A.

- Antamoshkin, H. P. Schwefel, A. Toern, G. Yin, A. Zhilinskas.- Krasnoyarsk:Ofset.- 1993.- 202 P.
236. Antamoshkin, A. Pseudo-Boolean Optimization in Case of an Unconnected Feasible Set / A. Antamoshkin, I. Masich // Models and Algorithms for Global Optimization Optimization and Its Applications.- [s.l.]:Springer Verlag.- 2007.- Vol.4.- P. 111–122.
237. Антамошкин, А.Н. Метод изменяющихся вероятностей / А.Н. Антамошкин, В.Н. Сараев // Проблемы случайного поиска.- Рига: Зинатне.- 1988. — вып. 11.- С. 26–34.
238. Reza, A.W. A Comprehensive Study of Optimization Algorithm for Wireless Coverage in Indoor Area / A.W. Reza, K. Dimiyati, K.A.Noordin, A.S.M.Z. Kausar, Md.S. Sarker // Optimization Letters.- September 2012 [Электронный ресурс] Режим доступа DOI: 10.1007/s11590-012-0543-z, URL: <http://link.springer.com/article/10.1007%2Fs11590-012-0543-z?LI=true> (дата обращения: 10.05.2013).
239. Antamoshkin, A. On Definition of Informative Subsystem of Signs in the Pattern Recognition Problem / A. Antamoshkin, V. Saraev // Computers and Artificial Intelligence.- 1985.- Vol. 4, No.3.- P. 245–252.
240. Antamoshkin, A. Brainwaare for Searchal Pseudoboolean Optimization / A. Antamoshkin // Tenth Prague Conf. on Inf. Theory, Stat. Dec. Functions, Random Processes.- Prague: Academia.- 1986.- P. 231-235.
241. Дегтерев, А.С. Обобщение генетических алгоритмов и алгоритмов схемы МИВЕР / А.С. Дегтерев, Ф.В. Канашкин, А.Д. Сумароков // Исследовано в России.- 2004.- № 7.- С.1658–1665.
242. Казаковцев, Л.А. Параллельная Реализация Метода Изменяющихся Вероятностей / Л.А. Казаковцев, А.А. Ступина // Современные проблемы науки и образования.- 2012.- № 4 [Электронный ресурс] Режим доступа URL: [www.science-education.ru/104-6810](http://www.science-education.ru/104-6810) (дата обращения: 31.10.2012).
243. Трубин, В.А. Эффективный алгоритм для задачи Вебера с прямоугольной метрикой / В.А. Трубин // Кибернетика.- 1978.- № 6, С. 67-70, DOI:10.1007/BF01070282.
244. E. Beasley, J. OR-Library: Distributing Test Problems by Electronic Mail / J. E. Beasley // Journal of the Operational Research Society.- 1990.- Vol. 41, No.11.- P. 1069–1072.
245. Антамошкин, А.Н. Применение метода изменяющихся вероятностей для задач размещения на сети / А.Н.Антамошкин, Л.А.Казаковцев // Вестник СибГАУ.–2014.– №5(57) .–С.95-99.
246. Береснев, В.Л. Алгоритмы решения задачи оптимального выбора динамического ряда изделия / В.Л. Береснев, Г.И. Ибрагимов, Ю.А. Кочетов // Управляемые системы.- Новосибирск: Институт математики СО АН СССР.- 1984.- вып.24.- С. 3-19.



247. Kazakovtsev, L.A. Genetic algorithm with greedy heuristic for capacity planning / L.A. Kazakovtsev, M.N. Gudyma, A.N. Antamoshkin // Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2014 6th International Congress on, 6-8 Oct. 2014, S.-Petersburg: IEEE Press.- P.607-613, DOI:10.1109/ICUMT.2014.7002170.
248. Lucasius, C.B. On K-Medoid Clustering of Large Data Sets with the Aid of a Genetic Algorithm: Background, Feasibility and Comparison / C.B. Lucasius, A.D. Dane, G. Kateman // Analytical Chimica Acta.- 1993.- Vol. 282.- P.647-669.
249. A. Kazakovtsev, L. Genetic Algorithm with Fast Greedy Heuristic for Clustering and Location Problems / L. A. Kazakovtsev, A.N. Antamoshkin. // Informatica.- 2014.- Vol. 38, No. 3.- P.229-240.
250. Казаковцев, Л.А. Детерминированный алгоритм для задачи k-средних и k-медоид / Л.А. Казаковцев // Системы управления и информационные технологии.- 2015.- №1(59).- С. 95-99.
251. Кочетов, Ю.А. Методы локального поиска для дискретных задач размещения Модели и алгоритмы / Ю.А. Кочетов.- Saarbrücken: Lambert Academic Publishing.- 2011.- 259 С.
252. Semenkin, E. Self-configuring Genetic Algorithm with Modified Uniform Crossover Operator / E. Semenkin, M. Semenkina // Advances in Swarm Intelligence. – Lecture Notes in Computer Science.- 2012. Vol. 7331. P. 414-421.
253. Sun, Zh. A Parallel Clustering Method Combined Information Bottleneck Theory and Centroid Based Clustering / Zh. Sun, G. Fox, W. Gu, Zh. Li // The Journal of Supercomputing.- 2014.- Vol. 69, No. 1.- P.452-467, DOI: 10.1007/s11227-014-1174-1.
254. Park, H.-S. A simple and fast algorithm for K-Medoids clustering / H.-S. Park, C.-H. Jun // Expert Systems with Applications.- 2009.- Vol. 36.- P.3336-3341.
255. Kazakovtsev, L.A. Fast Genetic Algorithm with Greedy Heuristic for p-Median and k-Means Problems / L.A. Kazakovtsev, A.A. Stupina // IEEE 2014 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT).- St.-Petersburg:IEEE Press.- 2014.- P.702-706.
256. MacKay, D. Chapter 20. An Example Inference task: Clustering / D. MacKay // Information Theory, Inference and Learning Algorithms. Cambridge.- 2003.- P.284-292.
257. Казаковцев, Л.А. Эволюционный алгоритм с жадной эвристикой для монотонной псевдодобулевой оптимизации / Л.А.Казаковцев // Экономика и менеджмент систем управления. – 2015. – №2.1. – С.145-151.
258. Kellerer, H. Knapsack problems / H. Kellerer, U. Pferschy, D. Pisinger.- 2013.- [s.l.]:Springer.- 548 P.

259. Казаковцев, Л.А. Применение алгоритмов МИВЕР в задачах планирования ассортиментной политики / Л.А. Казаковцев // Вестник НИИ СУВПТ, вып.6: Сб. Научн. трудов / Под общей редакцией профессора Н.В.Василенко. Красноярск: НИИ СУВПТ.- 2001.- С. 42-50.
260. Ступина, А. А. Технология надежностного программирования задач автоматизации управления в технических системах / А.А. Ступина, С.Н. Ежеманская.- Красноярск: СФУ.- 2011.- 164 С.
261. Царев, Р.Ю. Многоатрибутивное формирование гарантоспособных систем управления и обработки информации / Р.Ю. Царев, Д.В. Капулин, Д.В. Машурова, Я.А. Тынченко, Д.Н. Ковтанюк // Вестник СибГАУ.- 2012, № 5(45).- С. 106-110.
262. Ступина, А.А. Развитие методики ускоренной летной квалификации новых технологий на навигационных космических аппаратах / А.А.Ступина, М.И.Мельдер, Л.А.Казаковцев, М.Н.Гудыма.-М.:Роспатент.-2013.- Свидетельство о государственной регистрации программы для ЭВМ № 2014610051 от 09.01.2014.
263. Ступина, А.А. Мультиверсионное программное обеспечение технологии доступа к широкополосным мультимедийным услугам (Multi-versioner 1.0) / А.А.Ступина, Л.А.Казаковцев, М.И.Мельдер.-М.:Роспатент.-2013.- Свидетельство о государственной регистрации программы для ЭВМ № 2014610741 от 16.01.2014.
264. Kochetov, Yu. Formulation space search approach for the teacher/class time tabling problem / Yu. Kochetov, P.Paschenko M. Kononova // Yugoslav Journal of Operations Research.- 2008.- Vol.18, issue 1.- P. 1-11.
265. Семенкин, Е.С. Метод обобщенного адаптивного поиска для синтеза систем управления сложными объектами / Е.С. Семенкин, В.А. Лебедев.- М.: МАКС Пресс.- 2002.- 320 С.
266. Дюбин, Г.Н. Жадные алгоритмы для задачи о ранце: поведение в среднем / Г.Н. Дюбин, А.А. Корбут // Сиб. журн. индустр. матем.- 1999.- Т. 2, № 2.- С. 68–93.
267. Дюбин, Г.Н. Поведение в среднем жадных алгоритмов для минимизационной задачи о ранце – общие распределения коэффициентов / Г.Н. Дюбин, А.А. Корбут // Ж. вычисл. матем. и матем. физ.- 2008.- Т. 48, № 9, С. 1556–1570.
268. Berthold, T. Nonlinear Pseudo-Boolean Optimization: Relaxation or Propagation / T. Berthold, S. Heinz, M.E. Pfetsch // Theory and Applications of Satisfiability Testing - SAT 2009. LNCS. - 2009.- Vol.5584.- P. 441-446, DOI:10.1007/978-3-642-02777-2\_40.
269. Glover, F. Converting the 0-1 Polynomial Programming Problem to a 0-1 Linear Program / F. Glover, E. Woolsey // Operations Research. -1972.- Vol.22, No.1.- P.180-182.
270. Фролов, Е. Оперативное планирование производства / Е. Фролов // Директор

- информационной службы.- 2013, вып.5.- [Электронный ресурс] Режим доступа URL: <http://www.osp.ru/cio/2013/05/13035711/> (дата обращения: 01.10.2013)
271. Фатхутдинов, Р.А. Организация производства, учеб. для студентов вузов / Р.А. Фатхутдинов.- М.: ИНФРА-М.- 2001.- 672 С.
272. Казаковцев, Л.А. Алгоритм календарного планирования / Л.А.Казаковцев, А.Н.Антамошкин // Вестник КрасГАУ.-2015.-№4.-С.215-219.
273. Когаловский, В. Системы планирования производства: отечественные компромиссы развития / В. Когаловский // Директор информационной службы.- 2000.- вып.9.- [Электронный ресурс] URL: <http://www.osp.ru/cio/2000/09/170971/> (дата обращения: 01.10.2013)
274. Масич, И.С. Оптимизация загрузки производственных мощностей литейного производства / И.С. Масич, К.В. Шарыпова // Системы управления и информационные технологии.- 2007.- №3(29).- с.76-80.
275. Еремеев, А.В. О задаче составления расписаний с группировкой машин по технологиям / А.В. Еремеев, Ю.В. Коваленко // Дискретный анализ и исследование операций.- 2011.- Т. 18, № 5.- С. 54-79.
276. Алексеева, Е.В. Построение математических моделей целочисленного линейного программирования. Примеры и задачи: учебное пособие / Е.В.Алексеева.- Новосибирск: НГУ.- 2012, 131 С.
277. Кафаров, В.В. Гибкие автоматизированные производственные системы в химической промышленности. Учебник для ВУЗов / В.В. Кафаров, В.В. Макаров.- М.:Химия.- 1990.- 320 С.
278. Sherali, H.D. A new reformulation-linearization technique for bilinear programming problems / H.D. Sherali, A. Alameddine // Journal of Global Optimization.- 1992.- No.2.- P. 379–410.
279. Adams, W.P. Linearization Strategies for a Class of Zero-One Mixed Integer Programming Problems / W.P. Adams, H.D. Sherali // Operations Research.- 1990.- Vol. 38(2).- P. 217-226.
280. Казаковцев, Л.А. Система поддержки составления графиков загрузки производственных мощностей для непрерывного производства / Л.А.Казаковцев, М.Н.Гудыма, А.А.Ступина, М.И.Мельдер.- М.:Роспатент.- 2013.- Свидетельство о государственной регистрации программы для ЭВМ №2014610048 от 09.01.2014.
281. Balcan, M.-F. Distributed k-means and k-median clustering on general communication topologies / M.-F. Balcan, S. Ehrlich, Y. Liang // Advances in Neural Information Processing Systems.- 2013.- P. 1995-2003.
282. Belacel, N. Fuzzy J-Means: a new heuristic for fuzzy clustering / N. Belacel, P. Hansen, N.

- Mladenovic // Pattern Recognition.- 2002.- Vol.35.- P. 2193–2200.
283. Har-Peled, S. Coresets for k-Means and k-Median Clustering and their Applications / S. Har-Peled, S. Mazumdar // Proc. 36th Annu. ACM Sympos. Theory Comput.- 2003.- P. 291-300.
284. Struyf, A. Clustering in an Object-Oriented Environment / A. Struyf, M. Hubert, P. Rousseeuw // Journal of Statistical Software.- 1997.- issue 1 (4).- P. 1-30.
285. Moreno-Perez, J.A. A Parallel Genetic Algorithm for the Discrete p-Median Problem / J.A. Moreno-Perez, J.L. Roda Garcia, J.M. Moreno-Vega // Studies in Location Analysis.- 1994.- issue 7.- P.131-141.
286. A. Kazakovtsev, L. Wireless Coverage Optimization Based on Data Provided by Built-in Measurement Tools / L. A. Kazakovtsev // WASJ.- 2013.- Special Issue on Techniques and Technologies.- P. 8–15.
287. Масич, И.С. Отбор закономерностей для построения решающего правила в логических алгоритмах распознавания / И.С. Масич, Е.М. Краева // Системы управления и информационные технологии.- 2013, Т.51, вып. 1.1.- С. 170-173.
288. S. Duran, B. Cluster Analysys: a Survey / B. S. Duran, P. L. Odell.- Berlin-Heidelberg-New York:Springer-Verlag.-1977.- 140 P.
289. MacQueen, J.B. Some Methods of Classification and Analysis of Multivariate Observations / J.B. MacQueen // Proceedings of the 5th Berkley Symposium on Mathematical Statistics and Probability.- 1967.- Vol.1.- P.281–297.
290. Rui, X. Survey of Clustering Algorithms / X. Rui, D.Wunsch // Neural Networks, IEEE Transactions on.- 2005.- Vol. 16, issue 3.- P.645-678, doi: 10.1109/TNN.2005.845141.
291. Meira, L.A.A. A Continuous Facility Location Problem and Its Application to a Clustering Problem / L.A.A. Meira, Miyazawa. F.K. // Proceedings of the 2008 ACM symposium on Applied computing (SAC '08).-New York: ACM.- 2008.- P.1826-1831, doi:10.1145/1363686.1364126.
292. Weber, A. Ueber den Standort der Industrien, Erster Teil: Reine Theorie des Standortes/ A. Weber, [Verlag von] J. C. B. Mohr.- Tuebingen: Mohr.- 1922.- 209 P.
293. Vardi, Y. A modified Weiszfeld algorithm for the Fermat-Weber location problem / Y. Vardi, C.-H. Zhang // Mathematical Programming.- Vol. 90, No. 3.- 2001.- P. 559-566, DOI: 10.1007/s101070100222
294. Cabot, A. V. A Network Flow Solution to a Rectilinear Distance Facility Location problem / A. V. Cabot, R. L. Francis, M. A. Stary // American Institute of Industrial Engineers Transactions.- 1970.- Vol. 2.- P. 132-141.
295. Гудыма, М.Н. Алгоритм решения задачи размещения для некоторых специальных метрик

- / М.Н. Гудыма // Системы управления и информационные технологии.- 2013.- № 4(54).- С. 20-23.
296. Kazakovtsev, L.A. Decomposition of the Continuous Weber Problem with French Metro Metric / L.A. Kazakovtsev, P.S. Stanimirovic, M. Ciric // Proceedings of the International Conference on Problems of Modern Agrarian Science Collected scientific works.- Krasnoyarsk: KrasGAU.- 2012.- 5 P. [Электронный ресурс] Режим доступа URL <http://www.kgau.ru/img/konferenc/2012/e2.doc> (дата обращения 19.06.2015).
297. Cooper, L. Location-allocation problem / L. Cooper // Operations Research.- 1963.- Vol. 11.- P. 331-343.
298. Bailey, K. Numerical Taxonomy and Cluster Analysis / K. Bailey // Typologies and Taxonomies / K. Bailey. - [s.l.]: Sage Pubns.- 1994.- 89 P. DOI:10.4135/9781412986397.
299. Tan, P.-N. Cluster Analysis: Basic Concepts and Algorithms / P.-N. Tan, M. Steinbach, V. Kumar // Introduction to Data Mining.- [s.l.]:Addison-Wesley.- 2006.- P. 487–567.
300. Drineas, P. Clustering Large Graphs via the Singular Value Decomposition / P. Drineas, A. Frieze, R. Kannan, S. Vempala, V. Vinay // Machine learning.- 1999.- Vol. 56, No. 1-3.- P. 9-33.
301. Aloise, D. NP-Hardness of Euclidean Sum-of-Squares Clustering / D. Aloise, A. Deshpande, P. Hansen, P. Popat // Machine Learning.- 2009.- Vol. 75.- P. 245-249, DOI:10.1007/s10994-009-5103-0.
302. Bovet, D.P. Introduction to the Theory of Complexity / D.P. Bovet, D.P. Crescenzi.- [s.l.]:PrenticeHall.-69 P., ISBN 0-13-915380-2.
303. Leeuwen, J. van. Algorithms and complexity: Handbook of Theoretical Computer Science /J. van Leeuwen, [ed.].-Vol.A, . Amsterdam: Elsevier.- 1998.- 998 P., ISBN 02627-20140.
304. Turing, A.M. On Computable Numbers, with an Application to the Entscheidungs problem / A.M.Turing //Proceedings of the London Mathematical Society.– 1937.- Vol. 42.- P.230-265.
305. Turing A.M. On Computable Numbers, with an Application to the Entscheidungsproblem: A correction / A.M. Turing //Proceedings of the London Mathematical Society.- 1937.- Vol. 43, issue 6.- P. 544.
306. The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life plus The Secrets of Enigma / J. Copeland [ed.].- Oxford: Clarendon Press (Oxford University Press).- 2004.- 622 P. ISBN 0-19-825079-7.
307. O'Callaghan, L. Streaming-Data Algorithms for High-Quality Clustering / L. O'Callaghan, A. Meyerson, R. Motwani, N. Mishra, S. Guha // Data Engineering: Proceedings 18th International Conference on.- 2002.- P. 685-694, DOI: 10.1109/ICDE.2002.994785.

308. Ackermann, M.R. StreamKM: A Clustering Algorithm for Data Streams / M.R.Ackermann et. al. // J. Exp. Algorithmics.- May 2012.- Vol.17, Article 2.4 [Электронный ресурс] Режим доступа DOI:10.1145/2133803.2184450 (дата обращения 19.06.2015).
309. Mishra, N. Sublinear time approximate clustering / N. Mishra, D. Oblinger, Pitt. L. // 12th SODA.- 2001.- P. 439–447.
310. Czumaj, A. Sublinear Time Approximation for Clustering Via Random Sampling / A. Czumaj, Sohler. C. // Automata, Languages and Programming, Lecture Notes in Computer Science.- Berlin Heidelberg: Springer.- 2004.- Vol. 3142.- P. 396-407, DOI: 10.1007/978-3-540-27836-8 35.
311. Jaiswal, R. A Simple D2-Sampling Based PTAS for k-Means and Other Clustering Problems / R. Jaiswal, A. Kumar, S. Sen // Algorithmica.- 2014.- Vol. 70, issue 1.- P. 22-46, DOI: 10.1007/s00453-013-9833-9.
312. Phoungphol, P. Sample Size Estimation with High Confidence for Large Scale Clustering / P. Phoungphol, Y. Zhang // Proceedings of the 3rd International Conference on Intelligent Computing and Intelligent Systems.- 2011.- [Электронный ресурс] Режим доступа URL <http://www.cs.gsu.edu/phoungphol1/paper/icis2011.pdf> (дата обращения 19.06.2015).
313. Mladenovic, N. The p-median problem: A survey of metaheuristic approaches / N. Mladenovic, J. Brimberg, P. Hansen, . A. Moreno-Perez // European Journal of Operational Research.- 2007.- Vol. 179, issue 3.- P.927–939.
314. Arthur, D. k-Means++:R[14]C The Advantages of Careful Seeding/D. Arthur and S. Vassilvitskii//Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete algorithms [SODA '07].- SIAM.- 2007.- P.1027–1035.
315. Houck, C.R. Comparison of Genetic Algorithms, Random Restart, and Two-Opt Switching for Solving Large Location-Allocation Problems / C. R. Houck, J. A. Joines, G.Kay. M. // Computers and Operations Research.- 1996.- Vol. 23.- P. 587-596.
316. Антамошкин, А.Н. Алгоритм случайного поиска для обобщенной задачи Вебера в дискретных координатах / А.Н. Антамошкин, Л.А. Казаковцев // Информатика и системы управления.- 2013.- № 1(35).- С. 87-98.
317. Казаковцев, Л.А. Параллельный алгоритм для p-медианной задачи / Л.А.Казаковцев, А.Н. Антамошкин, М.Н. Гудыма // Системы управления и информационные технологии.- 2013.- № 52 (2.1).- 2013.- P.124–128.
318. Levanova, T.V. Algorithms of Ant System and simulated annealing for the p-median problem / T.V. Levanova, M.A. Loresh // Automation and remote control. -2004. -Vol.65, issue 30. -P. 431-438.

319. Juel, H. An efficient computational procedure for solving the multifacility rectilinear facilities location problem / H. Juel, R.F. Love // *Oper. Res. Q.*- 1976.- Vol. 27, No. 3.- P.697-703.
320. Muhlenbein, H. The parallel Genetic Algorithm as Function Optimizer / H. Muhlenbein, M. Shomisch, J. Born // *Proceedings of the fourth Conference of Genetic Algorithms.*- San Mateo:Morgan Kaufmann.- 1991.- P. 271-278.
321. Панфилов, И.А. Исследование эффективности работы генетического алгоритма оптимизации с альтернативным представлением решений /И.А. Панфилов, Е.П. Базанова, Е.А.Сопов// *Вестник СибГАУ.*- 2013.- № 4(50).-С.68-71.
322. Jackson,P. Evidence for the SPIN-0 Nature of the Higgs Boson Using Atlant Data/ P.Jackson, N.Soni, W.Edson [et al.] // *PhysicsLetters. Section B: Nuclear, Elementary Particle and High-Energy Physics.*- 2013.- Т. 7266 №1-3.-P.120-144.
323. Kirkconnell, C.S. High Efficiency Digital Cooler Electronics for Aerospace Applications / C.S. Kirkconnell, T.T. Luong, L.S. et al. Shaw // *Proc. SPIE 9070, Infrared Technology and Applications XL.*- Baltimore: SPIE.- 2014.- Article 90702Q.- 15 P. [Электронный ресурс] Режим доступа DOI:10.1117/12.2053075 (дата обращения 01.09.2015).
324. Казаковцев, Л.А. Модификация генетического алгоритма с жадной эвристикой для непрерывных задач размещения и классификации / Л.А. Казаковцев, А.А. Ступина, В.И. Орлов // *Системы управления и информационные технологии.*- 2014.- вып.2(56).- С.35-39.
325. Kazakovtsev, L.A. Modified Genetic Algorithm with Greedy Heuristic for Continuous and Discrete p-Median Problems / L.A. Kazakovtsev, V.I. Orlov, A.A. Stupina, V.L. Kazakovtsev // *Facta Universitatis (Nis) Series Mathematics and Informatics.*- 2015.- Vol. 30, No. 1.- P. 89-106.
326. UCI Machine Learning Repository.-Irvine: University of California.- [Электронный ресурс] Режим доступа URL: <https://archive.ics.uci.edu/ml/datasets.html>, DOI: 10.17616/R3T91Q (дата обращения 13.12.2014).
327. Arthur, D. k-Means Has Polynomial Smoothed Complexity / D. Arthur, B. Manthey, H. Roglin // *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS '09)*, Washington:IEEE Computer Society.- 2009.- P. 405-414, DOI: 10.1109/FOCS.2009.14.
328. Rousseeuw, P. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis / P. Rousseeuw // *Journal of Computational and Applied Mathematics.*- 1987.- Vol. 20.- P. 53-65.
329. Resende, M.G.C. Greedy randomized adaptive search procedures / M.G.C. Resende, C.C. Ribero // *International Series in Operations Research & Management Science.*- 2003.- Vol. 57

- "Handbook on Metaheuristics".- P. 219-249.
330. Zhang, Q. A new and efficient algorithm for spatial clustering / Q. Zhang, I. Couloigner // ICSSA 2005.- Singapore: [s.n].- May 9-12 2005.- P.181-189.
331. Казаковцев, Л.А. Эволюционный алгоритм для задачи k-медоид/ Л.А.Казаковцев // Системы управления и информационные технологии.- 2015.-№2(60) .-С.36-40.
332. Казаковцев, Л.А. Задача классификации электронной компонентной базы / Л.А. Казаковцев, В.И. Орлов, А.А. Ступина, И.С. Масич // Вестник СибГАУ.- 2014.- № 4(56).- С.55-61.
333. Орлов, В.И. О непараметрической диагностике и управлении процессом изготовления электрорадиоизделий / В.И. Орлов, Н.А. Сергеева // Вестник СибГАУ.- 2013.- 2(48).- С.70-75.
334. Kazakovtsev, L.A.Deterministic algorithm with agglomerative heuristic for location problems / L.A.Kazakovtsev, A.A.Stupina // IOP Conf. Series: Materials Science and Engineering.-2015.- Vol.94.-article ID 012016, 9 P. DOI: 10.1088/1757-899X/04/1012016.
335. Казаковцев, Л.А. Метод жадных эвристик для задач размещения / Л.А.Казаковцев, А.Н.Антамошкин // Вестник СибГАУ.-2015.-№2.-С.317-325.
336. Федосов, В.В. Минимально необходимый объем испытаний изделий микроэлектроники на этапе входного контроля / В.В. Федосов, В.И. Орлов // Изв. ВУЗов. Приборостроение.- 2011.-№ 54 (4).- С.68-62.
337. Беляева, Т.П. Достаточность и реализуемость требований к электронной компонентной базе/ Т.П. Беляева // Моделирование систем и процессов.- 2010.- № 3-4.- С. 10-12.
338. Hamiter, L. The History of Space Quality EEE Parts in the United States / L. Hamiter // ESA Electronic Components Conference.- Noordwijk, The Netherlands: ESTEC.- 1990, Nov 12-16.- P. 503-508.
339. MIL-PRF-38535 – Performance Specification: Integrated Circuits (Micricircuit) Manufacturing, General Specifications for.- United States of America: Department of Defence.- 2007.- 188 P.
340. Scott, A.J. The US Semiconductor Industry: a Locations Analysis / A.J.Scott,D.P.Angel// Environment and Planung, series A.- 1987.- Vol.19(7).- P. 875-912.
341. Орлов, В.И. Качество электронной компонентной базы – залог длительной работоспособности космических аппаратов/ В.И. Орлов, В.В. Федосов // Решетневские чтения.- 2013.- Т. 1., № 17.- С. 238-241.
342. Федосов, В.В. Повышение надежности радиоэлектронной аппаратуры космических аппаратов при применении электрорадиоизделий, прошедших дополнительные отбраковочные испытания в специализированных испытательных технических центрах /



- В.В. Федосов, В.Е. Патраев // *Авиакосмическое приборостроение.*– 2006.- № 10.- С.50-55.
343. Калашников, О.А. Методика сертификации электронной компонентной базы бортовой космической аппаратуры по стойкости к дозовому воздействию / О.А.Калашников, А.Ю.Никифоров // *Спецтехника и связь.*- 2011.– № 4-5.– С. 32-38.
344. Калашников, О.А.Функциональный контроль микропроцессоров при проведении радиационных испытаний / О.А.Калашников, П.В. Некрасов., А.А.Демидов // *Приборы и техника эксперимента.*– 2009.– № 2.– 48 С.
345. Патраев, В.Е. Методы обеспечения и оценки надежности космических аппаратов с длительным сроком активного существования: монография / В.Е. Патраев.– Красноярск: Издательство СибГАУ.- 2010.– 136 С.
346. Данилин, Н.С. Диагностика и контроль качества изделий цифровой микроэлектроники / Н.С. Данилин.- М.: Из-востандартов, 1991.- 176 С.
347. Данилин, Н. Проблемы применения современной индустриальной электронной компонентной базы иностранного производства в ракетно-космической технике / Н.Данилин, С.Белослудцев// *Современная электроника.*– 2007.– вып.7.– С. 8-12.
348. Матаре, Г. Электроника дефектов в полупроводниках / Г. Матаре.-М.:Мир.- 1974.
349. Вавилов, В.С. Механизмы образования и миграции дефектов в полупроводниках / В.С.Вавилов, А.Е. Кив, О.Р.Ниязова.- М.: Наука.- 1981.– 368 С.
350. Ooi, M.P.-L. Getting more from the semiconductor test: Data mining with defect-cluster extraction / M.P.-L. Ooi[et al.] // *IEEE Trans.Instrum. Meas.*- 2011.- Vol. 60, No. 10.- P. 3300-3317.
351. РД 11 0682-89. Микросхемы интегральные. Методы неразрушающего контроля диагностических параметров [Введ. с 01.01.1990].– 1989.- С.77.
352. Ooi, M.P.-L. Identifying Systematic Failures on Semiconductor Wafers Using ADCAS /M. P.-L. Ooi[et al.] // *Design &Test,IEEE.*- 2013.- Vol.30 (5).- P. 44-53.
353. Андреев, А.И. Надежность и контроль качества интегральных микросхем (конспект лекций) Часть 2 Контроль качества / А.И. Андреев, И.И.Катеринич, В.Д.Попов –М.: МИФИ.- 2004.–С.120.
354. РД 134-0154-2007. Методические указания. Порядок организации и проведения сертификации изделий иностранной электронной компонентной базы.- М.:ФСС КТ.- 2007.- 26 С.
355. ОСТ В 11 0219. Приборы полупроводниковые. Методы технологических (отбраковочных) испытаний. Спб. :Изд. ОАО РНИИ «Электронстандарт».- 1985.–79 С.
356. ОТ №510-5608-05 Анализ технического состояния и оценка уровня фактической

- надежности и готовности к целевому использованию космических аппаратов народнохозяйственного назначения по результатам изготовления и эксплуатации в 2005 году. Анализ динамики изменения показателей надежности за период с 1994 по 2005 год.- Железногорск: НПО ПМ.- 2005.- 176 С.
357. РД В 22.32.119-89.Методическое пособие по выбору и использованию методов и средств электрофизического диагностирования электрорадиоизделий. М.: Изд-во ЦНИИ 22 МО.- 1989.-210 С.
358. Калашников, О.А. Статистический разброс показателей радиационной стойкости интегральных схем иностранного производства / О.А.Калашников, А.С.Артамонов, Д.В. Бобровский [и др.]// Спецтехника и связь.- 2011.- № 4-5.- С. 45-47.
359. Никифоров, А.Ю. Учет влияния неравномерности ионизации на адекватность лазерного имитационного моделирования объемных ионизационных эффектов в ПП и ИС / А.Ю.Никифоров, П.К.Скоробогатов // Микроэлектроника.– 2008.– Т. 37. № 1.- С. 4-17.
360. Никифоров, А.Ю. Развитие базовой технологии прогнозирования, оценки и контроля радиационной стойкости изделий микроэлектроники / А.Ю.Никифоров, П.К.Скоробогатов, М.Н.Стриханов[и др.] // Известия высших учебных заведений. Электроника.- 2012.- № 5 (97).- С. 18-23.
361. Яненко, А.В. Сравнительный анализ испытаний электронной компонентной базы стойкости к воздействию отдельных ядерных частиц на лазерных имитаторах и ускорителях ионов / А.В.Яненко, А.И. Чумаков, А.А. Печенкин[и др.]// Спецтехника и связь.- 2011.- №4-5.- С. 4-7.
362. ОСТ В 11 0998-99 Микросхемы интегральные. Общие технические условия. М. : Изд-во Минобороны, 1999.– 139 С.
363. Горлов, М. Отбраковочные технологические испытания как средство повышения надежности партий ИС / М. Горлов, Л. Ануфриев, А. Строгонов // ChipNews.- 2001.- №5.- С. 22–26.
364. Анисимов, В.Г. Исследование сложных дефектов упаковки в монокристаллах кремния / В.Г. Анисимов, Л.Н. Данильчук, Ю.А. Дроздов [и др.] // Поверхность. Рентгеновские, синхротронные и нейтронные исследования.- 2004.–№ 11.- С. 74–81.
365. Надежность электрорадиоизделий: Справочник. М.: Изд.-во ЦНИИ 22 МО.- 2006.- 641 С.
366. Урличич, Ю.М. Управление качеством космической радиоэлектронной аппаратуры в условиях глобальной открытой экономики / Ю.М.Урличич.- М.: Макс Пресс.– 2003.- 204 С.
367. Горлов, М.И. Физические основы надежности интегральных схем: учебное пособие /

- М.И.Горлов,С.Ю.Королев.– Воронеж: Издательство Воронежского университета.- 1995.– 200 С.
368. Бутырин, С. А. Моделирование движения деформируемой конструкции спутника SESAT / С. А. Бутырин, С.Е. Сомов // Труды II научной конференции «Проектирование инженерных и научных приложений в среде MATLAB».- г.Санкт-Петербург.- 23-26 октября 2006.- С.498-515.
369. Bartenev, V.A. The Attitude and Orbit Control Subsystems and the Antenna Pointing Performance of the SESAT Spacecraft/V.A.Bartenev, С.P.Titov , V.A.Raevski et al.//4th ESA International Conference on Spacecraft Guidance, Navigation, and Control Systems : and tutorial on modern and robust control, theory, tools and applications.- Noordwijk, The Netherlands: ESTEC.- 2000.- P. 35-48.
370. Kwon, Y. Data mining approaches for modeling complex electronic circuitdesign activities/ Y.Kwon, O.A. Omitaomu, G.-N. Wang // Computer & Industrial Engineering.- 2008.- Vol. 54.- P. 229-241.
371. Khamidullina, N.M. Predictions of integrated circuit serviceability in space radiation fields / N.M. Khamidullina[et al.] // RadiationMeasurements.-1999.- Vol. 30.- P. 633-638.
372. Zhao, X. Defect Pattern Recognition on Nano/Micro Integrated Circuits Wafer / X. Zhao, L. Cui // Proceedings of the 3rdIEEE Int. Conf. on Nano/Micro Engineered and Molecular Systems(Sanya, China,January 6-9, 2008).–Sanya,China:[s.n].- 2008.- P. 519-523.
373. Bechow, L. An Improved Method for Automatic Detection and Location of Defects in Electronic Components Using Scanning Ultrasonic Microscopy / L. Bechow [et al.] // IEEE Transactions on Instrumentation and Measurement.- 2003.- Vol. 52, No. 1.- P.135-142.
374. Харченко, В.С. Анализ структур отказоустойчивых бортовых комплексов при использовании компонент Industry / В.С. Харченко, Ю.Б. Юрченко // Технология и конструирование в электронной аппаратуре.- 2003.- вып. 2.- С. 3-10.
375. Субботин, В. Проблемы обеспечения бортовой космической аппаратуры космических аппаратов электронной компонентной базой / В. Субботин, В. Стешенко // Компоненты и технологии, 2011.- вып.11.- С. 10-12.
376. Pease, R.L. Radiation Testing of Semiconductor Devices for Space Electronics / R.L. Pease, А.Н. Johnston, J.L. Azarevic // Proceedings of the IEEE.– 1998.– vol. 76, No. 11.- P. 1510-1526.
377. Ладыгин, Е.А. Обеспечение надежности электронных компонентов космических аппаратов / Е.А. Ладыгин.- Москва: МИСиС.- 2003.- С.111.
378. Коплярова, Н.В. О непараметрических алгоритмах идентификации нелинейных

- динамических систем / Н.В. Коплярова, Н.А. Сергеева // Вестник СибГАУ.- 2012.- № 5(45).- С.39-45.
379. Ожегин, Ю.А. Направления развития системы управления качеством радиационных испытаний электронной компонентной базы / Ю.А.Ожегин, А.Ю.Никифоров, В.А. Телец [и др.] // Спецтехника и связь.- 2011.- № 4-5.- С. 59-62.
380. Орлов, В.И. Фирменный стиль: надежность и качество / В.И. Орлов, В.В. Федосов // Петербургский журнал электроники.- 2010.- вып. 1(62).- С. 55-64.
381. Орлов, В.И. К вопросу о сертификации ЭРИ ИП / В.И. Орлов, В.В. Федосов // Научно-технический семинар «Обеспечение предприятий радиоэлектронной промышленности надежной электронной компонентной базой. Вопросы импортозамещения».- М.: ЗАО "Тестприбор".- 2014.- [Электронный ресурс] Режим доступа URL [http://www.makd.ru/media/downloads/sections/electro/230714/on\\_the\\_question\\_of\\_certification\\_esi\\_ip.pdf](http://www.makd.ru/media/downloads/sections/electro/230714/on_the_question_of_certification_esi_ip.pdf) (Дата обращения: 03.05.2015)
382. Mitchell T. Machine Learning.- [s.l.]:McGraw-Hill Science/Engineering/Math.- 1997.- 414 P., [http://personal.disco.unimib.it/Vanneschi/McGrawHill\\_-\\_Machine\\_Learning\\_-\\_Tom\\_Mitchell.pdf](http://personal.disco.unimib.it/Vanneschi/McGrawHill_-_Machine_Learning_-_Tom_Mitchell.pdf)
383. Callier, M.F. Linear System Theory / F.F. Callier// New York: Springer-Verlag.- 1991.- 509 P., ISBN 0-387-97573-X.
384. Казаковцев, Л.А. Выбор метрики при классификации электрорадиоизделий по производственным партиям // Л.А.Казаковцев, В.И.Орлов, А.А.Ступина // Программные продукты и системы. – 2015. – №2. –С. 124-129.
385. Максимов, Ю.Я. Алгоритмы решения задач нелинейного программирования / Ю.Я. Максимов, Е.А. Филипповская.- М.: МИФИ.- 1982.- 52 С.
386. Borg, J.F.P. Modern Multidimensional Scaling: Theory and Application / J.F.P. Borg.- [s.l.]:Springer.- 2005.- P. 207-212.
387. Казаковцев, Л.А. Быстрый детерминированный алгоритм для классификации электронной компонентной базы по критерию равнонадежности / Л.А.Казаковцев, И.С.Масич, В.И.Орлов, В.В.Федосов// Системы управления и информационные технологии.– 2015.–№4(62).–С.39-44.
388. Kazakovtsev, L.A. Fast Deterministic Algorithm for EEE Components Classification / L.A.Kazakovtsev, A.N.Antamoshkin, I.S.Masich // IOP Conf. Series: Materials Science and Engineering.–2015.–Vol.94.–article ID 012015, 10 P. DOI: 10.1088/1757-899X/04/1012015.
389. Kazakovtsev, L.A. Classification of Electronic Devices by Production Batches and Quality Classes / L.A.Kazakovtsev, A.A.Stupina, M.V.Karaseva, V.I.Orlov // Facta Universitatis Series

- Mathematics and Informatics.– 2015.– Vol.30, Iss.5.– P.567-581.
390. Davies, D.L. A Cluster Separation Measure / D.L. Davies, D.W. Bouldin // IEEE Transactions on Pattern Analysis and Machine Intelligence. PAMI-1 (2).- 1979.- P.224–227.
391. Thomas, J. New version of Davies-Bouldin index for clustering validation based on cylindrical distance / J.Thomas, M. Santos, M.M. Cofre // V Chilean Workshop on Pattern Recognition (CWPR 2014).- [s.l.:s.n.].- 2014.- [Электронный ресурс] Режим доступа URL [http://www.litrp.cl/ccpr2014/papers/jcc2014\\_submission\\_131.pdf](http://www.litrp.cl/ccpr2014/papers/jcc2014_submission_131.pdf) (дата обращения 01.09.2015).
392. Calinski, R.B. A dendrite method for cluster analysis / R.B. Calinski, J. Harabasz // Communications in Statistics.- 1974.- Vol. 3.- P. 1-27.
393. Krzanowski, W.J. A criterion for determining the number of clusters in a data set / W.J. Krzanowski, Y.T. Lai // Biometrics. -1985.- Vol. 44.-P. 23-34.
394. Hartigan, J.A. Clustering Algorithms / J.A. Hartigan.- New York: Wiley. 1975.- 366 P.
395. Tibshirani, R. Estimating the number of clusters in a data set via the gap statistic / R. Tibshirani, G. Walther, T. Hastie // Journal of the Royal Statistical Society.- Ser. B.- 2001.- Vol. 63.-P. 411–423.
396. Chen, T. Power-efficient hardware architecture of K-means clustering with Bayesian-Information-Criterion processor for multimedia processing applications / T. Chen, C. Sun, H. Su, S. Chien, D. Deguchi, I. Ide, H. Murase // IEEE Journal on Emerging and Selected Topics in Circuits and Systems.- Vol. 1, No. 3.- 2011.-P.357-368.
397. Орлов, А.И. Теория принятия решений / А.И. Орлов.- М.: Издательство «Экзамен».- 2005.-656 С.
398. Жамбю, М. Иерархический кластер-анализ и соответствия / М. Жамбю.- М.: Финансы и статистика.- 1988. —345 С.
399. Орлов, А.И. Устойчивость в социально-экономических моделях/А.И.Орлов.-М.: Наука.- 1979.- 296 С..
400. Kriegel, H.P. Outlier Detection Techniques (Tutorial) / H.P. Kriegel, K. P. Kroeger, Zimek. A. // 13th Pacic Asia Conference on Knowledge Discovery and Data Mining (PAKDD2009).- Bangkok: [s.n.].- 2009.- Vol. 26.- 73 P.- [Электронный ресурс] Режим доступа URL <https://www.siam.org/meetings/sdm10/tutorial3.pdf> (дата обращения 19.06.2015).
401. Орлов, В.И. Система классификации электро-радио изделий по производственным партиям / В.И.Орлов, Л.А.Казаковцев, А.А.Ступина, А.Н.Антамошкин.– М.: Роспатент.– Свидетельство о государственной регистрации программы для ЭВМ №2015612476 от 19.02.2015.

402. Vimal, A. An Experiment with Distance Measures for Clustering / A. Vimal, S. R. Valluri, Karlapalem. K. // International Conference on Management of Data COMAD.-Mumbai, India: [s.n.]- 2008.- P.241-244.
403. Ying, Y. Distance Metric Learning with Eigenvalue Optimization / Y. Ying, P. Li // Journal of Machine Learning Research.- 2012.- Vol. 13.- P.1-26.
404. Osinuga, I.A. A Modeling Framework on Distance Predicting Functions for Location Models in Continuous Space/ I.A.Osinuga, P.S.Stanimirovic, L.A.Kazakovtsev, S.A.Akinleye // Facta Universitatis ser.Math&Inform.-2015.-Vol.30.- iss.4.-P.419-443.
405. Zarinbal, M. Distance functions in location problems / M. Zarinbal // Facility location: Concepts, Models, Algorithms and Case studies / R.Z.Farahani and M.M. Hekmatfar [eds.]- Berlin Heidelberg: Springer-Verlag.-2009.- P. 5-18.
406. Walker, J.H. An empirical study of round and block norms for modeling actual distances, Ph.D. Dissertation/J.H. Walker.- Hamilton, Ontario: McMaster University.- 1991.- 211 P.
407. Alpaydin, E. Parametric distance functions vs. nonparametric neural networks for estimating road travel distances / E. Alpaydin, Altinel I.K, N. Aras // European Journal of Operational Research. -1996.- Vol. 92.- P.230-243.
408. Osinuga, I.A. An overview of norm approximation applications / I.A. Osinuga, Oniyide. O.R. // Far East J. Math. Sciences.- 43(2).- 2010.- P.189-202.
409. Marosevic, T. Data clustering for circle detection / T. Marosevic // Croatian Operational Research Review. -2014.-Vol. 5.- P.24-33.
410. Oniyide, O.R. On the existence of best sample in simple random sampling / O.R. Oniyide, Osinuga. I.A. // J. Math. Assoc. Nigeria.- 33(2B).- 2006.- P.290-294.
411. Stanimirovic, P.S. Discrete location problem on arbitrary surface in  $R^3$  / P.S. Stanimirovic, M. Ciric // Facta Universitatis, (Nis) Ser. Math. Inform. -2010. -Vol. 25.- P.47-56.
412. Revelle, C.S. A bibliography for some fundamental problem categories in discrete location science / H.A. Eiselt, A.S. Daskin, A.S. // Eur. J. Oper. Res.- 2008.- Vol.184.- P. 817-848.
413. Carrizosa, E. Weber problem with alternative transportation System / E. Carrizosa, A.-M. Rodriquez-Chia // European Journal of Operational Research.- 1997.- Vol. 97, No.1.- P.87-93.
414. Cha, S-H. Comprehensive Survey on Distance / Similarity Measures between Probability density functions / S-H. Cha // International Journal of Mathematical Models and Method in Applied Sciences.- 2007.- Vol. 4, No. 1, P.300-307.
415. Chavez, E. Searching in metric spaces / E. Chavez, G. Navarro, R. Baza-Yates, J.L. Marroquin // ACM Computing Surveys. 2001. Vol 33 (3), P.273-321.
416. Dearing, P.M. Planar location problems with Block Distance and Barriers / P.M. Dearing,

- K.Segars Jr.R. Klamroth // *Annals of Operations Research*.- 2005.- Vol. 136.- P.117-143.
417. Drezner, Z. The Central Warehouse Location Problem Revisited / Z. Drezner, C. Scott, Song. J.- S. // *IMA Journal of Management Mathematics*.- Vol. 14, No. 4.- 2003.- P.321-336.
418. Kafer, B. Error bounds for the approximate solution of Restricted planar location problems / B. Kafer, S. Nickel // *European Journal Operational Research*.- Vol. 135.- 2001.- 67-85.
419. Klamroth, K. Planar location problems with line barrier / K. Klamroth // *Optimization*.- Vol. 49.- 2001.- P.517-527.
420. K., Klamroth. Single-facility location problems with barriers / Klamroth. K. // Springer Verlag, Berlin, Heilderberg.- 2002.- 201 P.
421. R., Klein. Voronoi diagrams in the Moscow metric / Klein. R. // *Lecture Notes in Computer Science*.- 1998.- Vol. 344. P.434-444.
422. Kolsters, W.A. Metrics for mining multisets / W.A. Kolsters, J.F.J. Laros // *Research and Development in Intelligent Systems XXIV: Proceedings of AI-2007, the Twenty-seventh SGA International Conference on Innovative Techniques and Applications of Artificial Intelligence / M. Bramer, F. Coenen, M. Petridis [eds.]*.- [s.l.]:Springer.- 2007.- P. 293-303.
423. Kon, M. A single-facility minisum location problem under A-distance / M. Kon, Kushimoto. S. // *Journal Operations Research Society of Japan*.- Vol. 40, No. 1.- 1996.- P.10-20.
424. Pfeiffer, B. A unified model for Weber problem with continuous and network distance / B. Pfeiffer, Klamroth. K. // *Computers and OR*.- Vol. 35, No. 2.- 2008.- P.312-326.
425. Avella, P. Some personal views on the current state and the future of locational analysis / P. Avella et al. // *European Journal of Operational Research*.- 1998.- Vol. 104.- P. 269-287.
426. Campbell, J.F. A hybrid distance approximation for a GIS-based decision support system / J.F. Campbell, A. Labelle, A. Langerin // *Journal Business Logistics*.- 2001.- Vol. 22, No. 2.- P. 165-181.
427. Krarup, J. The impact of distance on location problems / J. Krarup, Pruzan. P.M. // *European Journal Operational Research*.- 1980.- Vol. 4.- P. 256-269.
428. Love, H.F. Mathematical model of road travel distances / H.F. Love, Morris. J.G. // *Management Science*.- 1979.- Vol. 25.- P.130-139.
429. Love, R.F. Facilities Location: models and methods / R.F. Love, J.G. Morris, G.O. Wesolowsky.- Amsterdam: North-Holland.- 2011.- 296 P.
430. Berens, W. The suitability of the weighted - norm in estimating actual road distances / W. Berens // *European Journal of Operational Research*.- 1988.- Vol. 34.- P.39-43.
431. Berens, W. Estimating road distances by Mathematical functions / W. Berens, F.J. Koerling // *European Journal of Operational Research*.- 1985.- Vol. 21.- P.45- 54.

432. Berens, W. On estimating road distances Mathematical functions – a rejoinder / W. Berens, F.J. Koerling // *European Journal of Operational Research*.- 1988.- Vol.36.- P.254-255.
433. Fernandez, J. Estimating actual distances by norm functions: A comparison between the  $p$ -norm and a study about the selection of the data set. / J. Fernandez, P. Fernandez, B. Peleguin // *Computers and OR*.- 2002.- Vol. 29.- P.609-623.
434. Brimberg, J. Estimating distances / J. Brimberg, R.F. Love // *Facility Location: Springer series in Operations Research / Z.Drezner [editor]*.- [s.l.]: Springer-Verlag.- 1995.- P.225-262.
435. Kolesar, P. Determining the relation between the engine travel times and travel distances in New York City / P. Kolesar, W. Walker, J. Hausner // *Operations Research*.- 1975.- Vol. 23.- P.614–627.
436. Uno, T. A location problem with the A-distance in a competitive environment / T. Uno, H. Katagiri, K. Kato // *Proceedings of the International Multi-conference of Engineers and Computer Scientists, 19-21March,2008, Hong-Kong*.- Hong-Kong:[s.n.].- 2008.- P. 19-25.
437. Gonzalez-Brevis, P. Base station location optimization for minimal energy consumption in wireless networks / P. Gonzalez-Brevis, J. Gondzio, F. Yijia, H. V. Poor, J. Thompson, I. Krikidis, Pei-Jung. C. // *Vehicular Technology Conference (VTC Spring): IEEE 73rd, 2011, May 15-18*.- [s.l.]: IEEE Press.- 2011.- P.1–5, doi: 10.1109/VETECS.2011.5956204.
438. G., O.Wesolowsky. Location problems on a sphere / O.Wesolowsky. G. // *Regional Science and Urban Economics*.- 1983.- Vol.12.- P.495-508.
439. Brimberg, J. Properties of distance functions and Minisum location models: Ph.D. Dissertation/ J.Brimberg.- Canada: McMaster University.- 1989.- 287 P., <https://macsphere.mcmaster.ca/bitstream/11375/7417/1/fulltext.pdf>
440. Uster, H. Weighted sum of order  $p$  and minisum location models / H. Uster // Ph.D. Thesis, McMaster University, Canada, 1999.- 210 P., <https://macsphere.mcmaster.ca/bitstream/11375/8414/1/fulltext.pdf>
441. Perreur, J. Central metric and optimal location / J. Perreur, Thisse. J.F. // *J. Regional Science*.- 1974.- Vol. 14.- P.411-421.
442. Francis, R.L. Locational Analysis / R.L. Francis, L.F. McGinnis, J.A. White // *European Journal of Operational Research*.- 1983.- Vol. 12.- P.230-252.
443. Parthasarathy, M. The three-dimensional Fermat- Weber problem with Tchebychev distances / M. Parthasarathy, T. Hale, J. Blackhurst, M. Frank // *Advanced Modelling and Optimization*.- 2006.- Vol. 8, No. 1.- P.65-71.
444. Kazakovtsev, L.A. An Approach to the Multi-facility Weber Problem with Special Metrics / L.A. Kazakovtsev, Stanimirovic. P.S. // *Modelling Symposium (EMS)*, 20-22 Nov.2013,



- European.- Manchester: UkSim.- 2013.- P.119–124, DOI:10.1109/EMS.2013.21.
445. Osinuga, I.A. Planar Weber Location Problem with French Metro Metric / I.A. Osinuga, L.A. Kazakovtsev, P.S. Stanimirovic // Review Bulletin of the Calcutta Mathematical Society.- 2013.- Vol.21, issue 1.- P.7-20.
446. Aykin, T. Multifacility location problems on a sphere / T. Aykin, A.J.B. Babu // International Journal of Mathematics and Mathematical Sciences.- 1987.- Vol. 10, No.3.- P.586-596.
447. Dhar, U.R. A comparative study of three norms for facility location problem on spherical surface / U.R. Dhar, J.R. Rao // New Zealand J. Operations Research.- 1980.- Vol. 8.- P.173-183.
448. Drezner, Z. Facility location on a sphere / Z. Drezner, G.O. Wesolowsky // Journal of Operational Research Society. 1975.- Vol. 29,- P.997-1004.
449. Katz, I.N. Optimum location on a sphere / I.N. Katz, L. Cooper // Computers & Mathematics with Applications.- 1980.- Vol. 6, issue 2.- P. 175-196, DOI:10.1016/0898-1221(80)90027-9.
450. Mwemezi, J.J. Optimal facility location on spherical surfaces: algorithm and Application / J.J. Mwemezi, Y. Huang // New York Science Journal.- 2011.- Vol. 4, issue 7, P.21-28.
451. Plastria, F. Continuous location anno 92: a progress report, Isolde VI survey papers / F. Plastria // Studies in Locational analysis.- 1993.- Vol. 5.- P.85-127.
452. Abdel-Khalel, H. A computer-based Model for optimizing the location of single tower crane in construction sites / H. Abdel-Khalel, K. Shawki, M. Adel // International Journal of Engineering Science and Innovative Technology.- 2013.- Vol.2.- P. 438-446.
453. Warszawski, A. Optimizing the location of facilities on a building sites / A. Warszawski, S. Peer // Journal of the Operational Research Society.- 1973.- Vol. 24.- P.35-44.
454. Alkriz, K. A new model for optimizing the location of cranes and construction facilities using genetic algorithms. In: Khosrowshahi, F. (Ed.) / K. Alkriz, J-C. Mangin // 21st Annual ARCOM Conference, 7-9 September, 2005 (SOAS).- London: University of London, Association of Researchers in Construction Management.- 2005.- Vol.2.- P.981-991.
455. Irizarry, J. Optimizing location of tower cranes on construction sites through GIS and BIM integration / J. Irizarry, E.P. Karan // Journal of Information Technology in Construction (Itcon). 2012. Vol. 17, P.351-366.
456. Kazakovtsev, L.A. Algorithms for location problems based on angular distances / L.A. Kazakovtsev, P.S.Stanimirovis, I.A.Osinuga, M.N. Gudyma, A.N.Antamoshkin // Advances in Operations Research.- 2014.- Vol.2014.- Article ID 701267.- 12 P. [Электронный ресурс] Режим доступа doi:10.1155/2014/701267.
457. Witzgall, C. Optimal location of a Central Facility: Mathematical Models and Concepts:

- National Bureau of Standards Report 8388/C. Witzgall.- Washington D.C.: National Bureau of Standards Report 8388.- 1964.- 95 P.
458. Ward, J.E. A new norm for measuring distance which yields linear location problems / J.E. Ward, R.E. Wendell. // *Operations Research*.- 1980.- Vol. 28, P. 836-844.
459. Ward, J.E. Using block norms for location modeling / J.E. Ward, R.E. Wendell // *Operations Research*.- 1985.- Vol. 33.- P.1074-1090.
460. Fathali, J. Location problems in regions with and Block norms / J. Fathali, Zaferanieh. M. // *Iranian Journal Operations Research*.- 2011.- Vol. 2, No.2.- P.72-87.
461. Thisse, J.F. Some Properties of Location Problems with Block and Round Norms / J.F. Thisse, J.E. Ward, R.E. Wendell // *Operations Research*.- 1984.- Vol. 32.- P. 1309–1327.
462. Hodgson, M.J. The P-Centroid problem on an inclined plane / M.J. Hodgson, R.T. Wong, Honsaker. J. // *Operations Research*.-1987.- Vol. 35.- P. 221-233.
463. Drezner Asymmetric distance location problem / Drezner, G.O. Wesolowsky // *Transportation Science*.- 1989.- Vol. 23, No. 8.- P.201-207.
464. Widmayer, P. On some distance problem in fixed orientations / P. Widmayer, Y.F. Wu, C.K. Wong // *SIAM Journal of Computing*.- 1987.- Vol. 16, No. 8.- P.728-746.
465. Fujita O. Metric based on average distance between sets/O. Fujita // Cornell University -Print Archive.- Ithaca, New York: Cornell University.- 2011.- [Электронный ресурс] Режим доступа arXiv:1103.1049v2 (дата обращения 01.12.2014).
466. Spath H. The minisum location problem for the Jaccard metric / H. Spath // *Operation Research Spektrum*.- 1981.- Vol. 3.- P.91-94.
467. Chierichetti, F. Finding the Jaccard median, Available at F. Chierichetti, R. Kumar, S. Pandey, S. Vassilvtskii // *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*.- [s.l.]: SIAM.- 2010.- P. 293-311.
468. Watson, G.A. An algorithm for the single facility location problem using Jaccard metric / G.A. Watson // *SIAM Journal Scientific and Statistical Computing*.- 1983.- Vol. 4, No. 4.- P.748-756.
469. Kammerdiner, A. On the Hamming distance in Combinatorial optimization problems in hypergraph matchings /A. Kammerdiner, P.A. Krokhmal, P.M. Pardalos // *Optimization Letters*.- 2010.- Vol. 4, issue 4. P. 609-617.
470. Zhang, B. The Center location improvements problem under the Hamming distance / B. Zhang, J. Zhang, Y. He // *Journal of Combinatorial Optimization*.- 2005.- Vol. 9, No. 2.- P. 187-198.
471. Liu, L. Inverse min-max spanning tree problem under the weighted sum-type Hamming distance / L. Liu, E. Yao // *Theoretical Computer Science*.- 2008.- Vol. 396.- P.28-34.
472. Kazakovtsev, L.A. Algorithm for Weber problem with a metric based on Initial Fare / L.A.

- Kazakovtsev, P.S. Stanimirovic // *Journal of Applied Mathematics and Informatics*.- 2015.- Vol. 33, No. 1-2.- P.157-172, DOI: 10.14317/jami.2015.157.
473. Love, R.F. Modelling Inter-City Road Distances by Mathematical Models / R.F. Love, J.G. Morris // *Annals of Operations Research*.- 1972.- Vol. 18.- P.103-112.
474. Love, R.F. A statistical comparison of three goodness-of-fit criteria used in modeling distances / R.F. Love, H. Uster // *Journal of Applied Mathematics and Decision Sciences*.- 2001.- Vol. 5, No. 4.- P.235-251.
475. Afshari, H. Challenges and solutions for location of healthcare facilities / H. Afshari, Q. Peng // *Industrial Engineering and Management*.- 2014.- Vol.3, No. 2.- P.1-12.
476. Basar, A. A taxonomy for emergency service station location problem / A. Basar, B. Catay, T. Unluyuth // *Optimization Letters*.-2012.-Vol. 6, issue 6.-P. 1147-1160, DOI 10.1007/s11590-011-0376-1.
477. Brimberg, J. A survey of solution methods for the Continuous location-allocation problem / J. Brimberg, P. Hansen, N. Mladenovic, S. Salhi // *International Journal of Operations Research*.- 2008.- Vol. 5.- P.1-12.
478. Farahani, R.Z. Multiple criteria location problems: A survey, / R.Z. Farahani, Sefi M. Steadie, N. Asgari, // *Applied Mathematical Modelling*.- 2010.- Vol. 34.- P.1689-1700.
479. Smith, H.K. Locational Analysis: highlights of growth to maturity / H.K. Smith, G. Laporte, P.R. Harper // *Journal of the Operational Research Society*.- 2009.- Vol. 60.- P.140-148.
480. Love, R.F. Solving constrained multifacility location problem involving distances using convex programming / R.F. Love, J.G. Morris // *Operations Research*. 1975. Vol. 23, P.581-587.
481. Singhtaun, C. Comparison of exact algorithms for rectilinear distance single-source capacitated multi-facility Weber problems / C. Singhtaun, P. Charnthikul // *Journal of Computer Science*.- 2010.-Vol. 6, No. 2.- P.112-116.
482. Brimberg, J. A global convergence of a generalized iterative procedure for the minisum location problem with lp-distances / J. Brimberg, Love. R.F. // *Operations Research*.- 1993.- Vol. 41, No. 6.- P.1153-1163.
483. Jiung, J.I. A generalized Weber problem with different gauges for different regions / J.I. Jiung, F. Luan, L.-p. Wung, X.X. Zhu // *Applied Mathematics and Computation*.- 2014.- Vol. 244.- P.244-260.
484. Chen, R. Location problems with costs being sums of powers of Euclidean distances / R. Chen // *Comput. Oper. Res*.- 1984.- Vol. 11.- P.285-294.
485. Minkowski, H. *Gesammelte Abhandlungen, zweiter Band* / H. Minkowski // [s.l.]: Nabu Press.- 2010- 468 S.

486. Chen, R. Noniterative Solution of Some Fermat-Weber Location Problems / R. Chen // *Advances in Operations Research*.- 2011.- Vol. 2011.- Article ID 379505.- 10 P., DOI:10.1155/2011/379505.
487. Durier, R. Geometrical properties of the Fermat-Weber problem / R. Durier, Michelot. C. // *Europ.J. Oper Res.*- 1985.- Vol. 20.- P.332-343.
488. Korn, G.A. *Mathematical handbook for scientists and engineers* / G.A. Korn, T.M. Korn // New York: McGraw-Hill.- 1961.- P.174-175.
489. Marianov, V. *Median Problems in Network* / V. Marianov, D. Serra // *Foundations of Location Theory* / H.A. Eiselt, V. Marianov [eds.]:- [s.l.]: Springer.- 2011.- P. 39-70.
490. Антамошкин, А.Н. Алгоритм для задачи размещения с метрикой Москвы-Карлсруэ / А.Н. Антамошкин, Л.А. Казаковцев // *Системы управления и информационные технологии*.- 2012.- Т.49, № 3.1.- С. 111-115.
491. Казаковцев, Л.А. Свойства простейшей множественной задачи размещения / Л.А.Казаковцев // *Системы управления и информационные технологии*. – 2014. – № 1. – С.65-69.
492. Казаковцев, Л.А. Алгоритм для простейшей множественной задачи размещения / Л.А.Казаковцев // *Системы управления и информационные технологии*. – 2014. – № 1.1. – С.156-160.
493. Korte, B. *Combinatorial optimization: Theory and Algorithms* / B. Korte, J. Vygen // Berlin: Springer.- 2000.- 597 P.
494. Durier, R. On the set of optimal points to the Weber problem: Further results / R. Durier, C. Michelot // *Transportation Science*.- 1994.- Vol. 28.- P.141-149.
495. Казаковцев, Л.А. Алгоритм для задачи размещения с неевклидовой метрикой, основанной на угловом расстоянии / Л.А.Казаковцев // *Фундаментальные исследования*.-2012.-№9, часть 4.-С.918-923.
496. Moon, P. *Circular-Cylinder Coordinates (r, φ, z)* / P. Moon, D.E. Spencer // *Field Theory Handbook, Including Coordinate Systems, Differential Equations, and Their Solutions* [corrected 2nd ed.]- New York City: Springer-Verlag.- 1988.- P. 12-17.
497. Brimberg, J. Extension of the Weiszfeld procedure to a single facility minisum location model with mixed norms / J. Brimberg, R. Love, N. Mladenovic // *Math. Meth. Oper. Res.*- 2009.- Vol. 70.- P.269-283, DOI: 10.1007/s00186-008-0268-z.
498. Love, R.F. A computation procedure for the exact solution of location-allocation problems with rectangular distances / R.F. Love, J.G. Morris // *Naval Res. Logist.*- 1975.- Vol. 22.- P. 441–453.

499. Voevodski, K. Min-sum clustering of porotein sequences with limited distance information / K. Voevodski, M.-F. Balcan, H. Roglin, S.-H. Teng, Y. Xia. // Proceedings of the First International Conference on Similarity-based Pattern Recognition (SIMBAD'11).- Venice, Italy: [s.n].- 2011.- P.192-206.
500. Wright, M.H.. Direct search methods: once scorned, now respectable / M.H. Wright // Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis / D.F. Griffiths, G.A. Watson [eds].- Dundee: [s.n].- 1996.- P.191-208.
501. Powell, M.J.D. An efficient method for finding the minimum of a function of several variables without calculating derivatives / M.J.D. Powell // Computer Journal.- 1964.- Vol. 7, No. 2.- P.155-162.
502. Brimberg, J.. Further notes on convergence of theWeiszfeld algorithm / J. Brimberg // Yugoslav Journal of Operations Research.- 2003.- Vol. 13, No. 2.- P.199-206.
503. Beck, A. Gradient-based algorithms with applications of signal recovery problems / A. Beck, M. Teboulle // Convex Optimization in Signal Processing and Communications / D. Palomar and Y. Eldar [eds].- Cambridge: Cambridge University Press.- 2010.- P.33–88.
504. Drezner, Z. A note on accelerating the Weiszfeld procedure / Z. Drezner // Location Science.- 1995.- Vol. 3, No. 4.- P.275-279.
505. Drezner, Z. On the convergence of the generalized Weiszfeld algorithm / Z. Drezner // Annals of Operation Research.- 2009.- Vol. 167.- P.327-336, DOI:10.1007/s10479-008-0336-z.
506. Drezner, Z. The facility location problem with limited distances / Z. Drezner, A. Mehrez, G.O. Wesolowsky // Transportation Science.- 1991.- Vol. 25.- P.183-187.
507. Fernandes, I.F. On the Weber Facility Location Problem with Limited Distances and Side Constraints / I.F. Fernandes, D. Aloise, D.J. Aloise, P. Hansen, Liberti. L. // Optimization Letters.- 2014.- Vol. 8, No. 2.- 407-424, DOI: 10.1007/s11590-012-0538-9.
508. Kazakovtsev, L.A. Algorithm for constrained Weber problem with feasible region bounded by arcs / L.A. Kazakovtsev // Facta Universitatis, (Nis) Ser. Math. Inform.- 2013.- Vol. 28, issue 3.- P.271-284.
509. Hansen, P. Constrained location and the Weber-Rawls problem / P. Hansen, D. Peeters, Thisse. J.-F. // North-Holland Mathematics Studies.- 1981.- Vol. 59.- P. 147-166.
510. Kuhn, H.W. A note on Fermat's problem / H.W. Kuhn // Mathematical Programming.- 1973.- Vol. 4.- P.98-107.
511. Nocedal, J. Numerical Optimization / J. Nocedal, S. Wright. // Springer Series in Operations Research and Financial Engineering.- 2006.- Vol. XXII.- 636 P.
512. Idrissi, H. A Primal-dual Algorithm for a Constrained Fermat-Weber Problem Involving Mixed

- Norms / H. Idrissi, O. Lefebvre, Michelot. C. // *Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche Opérationnelle.*- 1998.- Vol. 22, No. 4.- P.313-330.
513. Zabudski, G.G. A minimax planar location problem with forbidden zones: its solution algorithm / G.G. Zabudski // *Automation and Remote Control.*- 2004.- Vol. 65, No. 2.- P.241-247 [Translated from *Avtomatika i Telemekhanika*, 2, 2004, P.93-100].
514. Hansen, P. An algorithm for a constrained Weber problem / P. Hansen, D. Peeters, J.-F. Thisse // *Management Science.*- 1982.- Vol. 28, No. 11.- P.1285-1295.
515. Zabudskii, G.G. Model Building and Location Problem Solving in a Plane with Forbidden Gaps / G.G. Zabudskii // *Autom. Remote Control.*- 2006.- Vol. 67, No. 12.- P. 1986-1990.
516. Забудский, Г.Г. Сужение области поиска решения задачи Вебера на плоскости с прямоугольными запрещенными зонами / Г.Г. Забудский, И.В. Амзин // *Автоматика и телемеханика.*- 2012.- №5.- С.71-83.
517. Jones, D.R. Lipschitzian Optimization without the Lipschitz constant / D.R. Jones, C.D. Perttunen, Stuckman. B.E. // *Journal of Optimization Theory and Application*, 79 (1), 1993, P.157-181
518. Finkel, D.E. Convergence Analysis of Sampling Methods for Perturbed Lipschitz Functions / D.E. Finkel, C.T. Kelley // *Pacific Journal of Optimization.*- 2009.- Vol. 5.- P. 339-350.
519. Alzamora, G.S. The Air Transportation Hub-and-Spoke Design Problem. Comparison between a Continuous and Discrete Solution Method / G.S. Alzamora, F.M.P. Raupp, N.D. Pizzolato, A.E. Xavier // *Pesquisa Operacional.*- 2013.- Vol. 33, No. 3.- P. 477-489, DOI: 10.1590/S0101-74382013005000006.
520. Honiden, S. Balancing Graph Voronoi Diagrams / S. Honiden, M.E. Houle, C. Sommer // *Proceedings of the 2009 Sixth International Symposium on Voronoi Diagrams (ISVD '09).*- Copenhagen: AMS.- P.183-191, DOI:10.1109/ISVD.2009.26.
521. L., Kazakovtsev. Algorithm for Approximate Solution of the Generalized Weber Problem with an Arbitrary Metric / Kazakovtsev. L. // *Computer Modeling and Simulation (EMS), 2012 Sixth UKSim//AMSS European Symposium on.-Valetta, Malta: UkSim.*- 2012.- P.109–114, DOI:10.1109/EMS.2012.52.
522. Kazakovtsev, L. Wireless network development as an optimization problem / L.Kazakovtsev // *International conference on problems of modern Agrarian Science: Collected scientific works.*- Krasnoyarsk: KrasGAU [Электронный ресурс] Режим доступа <http://kgau.ru/img/konferenc/2010/e21.doc> (дата обращения 01.06.2015).
523. Kazakovtsev, L. Wireless access points placement as a discrete optimization problem / L. Kazakovtsev // *ICTP Open Access Archive.*- Trieste: ICTP.- 2008 [Электронный ресурс]

- Режим доступа URL <http://eprints.ictp.it/494>, (дата обращения 10.11.2013).
524. Казаковцев, Л.А. Задача выбора оптимального размещения беспроводной сети / Л.А.Казаковцев, М.Н.Гудыма, А.А.Ступина, Ю.И.Кириллов // Современные проблемы науки и образования [электронное издание]. – 2013. – вып.3, Режим доступа URL <http://www.elibrary.ru/download/29613015.pdf>
  525. Hakimi, S.L. Optimum distribution of switching centers in a communication network and some related graph theoretic problems / S.L. Hakimi // Operations Research.- 1965.- Vol. 13, No. 3.- P.462–475.
  526. Mirchandani, P.B. Discrete location theory / P.B. Mirchandani, R. Francis // Wiley-Interscience Series in Discrete Mathematics and Optimization.- New York: John Wiley & Sons, Inc.- 1990.- 555 P.
  527. Daskin, M.S. Network and discrete location: models, algorithms, and applications / M.S. Daskin.- New York: John Wiley & Sons, Inc.- 1995.- 536 P.
  528. Love, R.F. Terminal location problem: a case study supporting the status quo / R.F. Love, W.G. Truscott, J. Walker // J. Opl Res. Soc.- 1985.- Vol. 36.- P.131-136.
  529. Brimberg, J. Properties of ordinary and weighted sums of order  $p$  used for distance estimation / J. Brimberg, R.F. Love // Rechercher Operationnelle.- 1995.- Vol. 29.- P.59-72.
  530. Финкельштейн, Ю.И. Дискретная оптимизация / Ю.И. Финкельштейн и др. // Методы оптимизации в экономико-математическом моделировании.- М.:Наука.- 1991.- С. 350-367.
  531. Казаковцев, Л.А. Системный подход к реализации ассортиментной политики на торговом предприятии / Л.А. Казаковцев, А.В. Статейнова // Вестник НИИ СУВПТ: сб.научн.трудов / Под общей редакцией профессора Н.В.Василенко.- Красноярск: НИИ СУВПТ.- 2001. вып. 6.- С.37-46.
  532. Rappaport T.S. Wireless communications: principles and practice [2nd ed.] / T.S. Rappaport.- IEEE Press.- 2002.- 736 P.
  533. Hu, J. Wireless access point configuration by genetic programming / J. Hu, E. Goodman // Congress on Evolutionary Computation (CEC2004).- 2004.- Vol.1.- P. 1178-1184.
  534. Panyukov, A.V. Polynomial algorithms to finite Veber problem for a tree network / A.V. Panyukov, B.V. Pelzwerger // J. Comput. Appl. Math.- 1991.- Vol. 35.- P. 291-296.
  535. Шангин, Р.Э. Точный алгоритм для решения одного частного случая задачи Вебера в дискретной постановке / Р.Э. Шангин // Прикладная дискретная математика. Приложение.- 2013.- вып. 6, С. 136-137.
  536. Панюков, А.В. Точный алгоритм решения дискретной задачи Вебера для  $k$ -дерева / А.В. Панюков, Р.Э. Шангин // Дискретный анализ и исследование операций.- 2014.- Т. 21, № 3,

- C.64-75.
537. Шангин, Р.Э. О некоторых свойствах N-последовательностьсвязной цепи / Р.Э. Шангин // Вестник ЮУрГУ. Серия Вычислительная математика и информатика.- 2013.- Т. 2, № 1, С.106-113.
  538. Pardalos, P.M. Parallel Search for Combinatorial Optimization: Genetic Algorithms, Simulated Annealing, Tabu Search and GRASP / P.M. Pardalos, L. Pitsoulis, T. Mavridou, M.G.C. Resende // Parallel Algorithms for Irregularly Structured Problems: Lecture Notes in Computer Science.- 1995.- Vol. 980.- P. 317-331.
  539. Pietrosevoli, E. Broadband Link to the Offshore Platform in the Venice Lagoon / E. Pietrosevoli, M. Zennaro, M. Bastianini, T. Minuzzo // Proceedings of ExtremeComm'14.- Galapagos Islands, Ecuador: [s.n.].- 2014. [Электронный ресурс] Режим доступа URL <http://wireless.ictp.it/Papers/ExtremeCom.pdf> (дата обращения 09.07.2015)
  540. Wright, M.H.. Optimization methods for base station placement in wireless applications / M.H.Wright. // Proceedings of IEEE Vehicular Technology Conference.- 1998.- Vol.89.- P. 11513-11517.
  541. Battiti, R. Optimal wireless access point placement for location-dependent services: Technical Report DIT-03-052 / R. Battiti, M. Brunato, A. Delai.- Trento: University of Trento, Informatica e Telecomunicazioni.- 2003 .- 12 P.
  542. Bahl, P. RADAR: An in-building RF-based user location and tracking system. / P. Bahl, V.N. Padmanabhan // Proceedings of IEEE INFOCOM 2000.- 2000.- P.775-784.
  543. Hills, A. Large-scale wireless LAN design / A.Hills // IEEE Comm. Magazine.- 2001.- Vol.39.- P.98-107.
  544. Вишнеvский, В.М. Широкополосные беспроводные сети передачи информации / В.М. Вишнеvский, А.И. Ляхов, С.Л. Портной, И.В. Шахнович.- М.: Техносфера.- 2005.- 592 С.
  545. Foley, J.D. Computer graphics: Principles and practice / J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes // Addison-Wesley Longman Publishing Co.- 1990.- 516 P.
  546. Kazakovtsev, L. Adaptive Model of Static Routing for Traffic Balance between Multiple Telecommunication Channels to Different ISPs / L. Kazakovtsev // ICTP Open Access Archive.- Trieste: ICTP.- 2006.- [Электронный ресурс] Режим доступа URL <http://eprints.ictp.it/415> (дата обращения 25.06.2013).
  547. Казаковцев, Л.А. Система поддержки принятия решений об оптимальном размещении элементов беспроводной сети / Л.А.Казаковцев, М.Н.Гудыма, А.А.Ступина.- М.:Роспатент.-2013.- Свидетельство о государственной регистрации программы для ЭВМ № 2013661595 от 11.12.2013.



## **ПРИЛОЖЕНИЕ А. ИСХОДНЫЕ ДАННЫЕ ЗАДАЧ АВТОМАТИЧЕСКОЙ ГРУППИРОВКИ ЭЛЕКТРОРАДИОИЗДЕЛИЙ**

Исходные данные представляют собой таблицы, в которой приведены результаты тестов ЭРИ. Изделия соответствуют строкам таблицы, тесты (измерения) – столбцам.

На данном этапе для выполнения разработки получены следующие наборы исходных данных.

1. Микросхемы 1526ЛЕ5

619 изделий, 120 измерений, указано, что выборка является смесью 3 партий.

2. Микросхемы ОР297GSZ

700 изделий, 18 измерений, число партий не известно.

3. 1. Полевые транзисторы 2П771А

182 изделия, 18 измерений, указано, что выборка является смесью 2 партий.

3.2. Арсенид-галиевые диоды 3Д713В

258 изделий, 12 измерений, указано, что выборка состоит из изделий одной партии.

3.3. Оптроны транзисторные одноканальные 3ОТ122А

278 изделий, 14 измерений, указано, что выборка является смесью 4 партий.

3.4. Микросхемы на основе БМК Н5503ХМ5-171

166 изделий, 188 измерений, указано, что выборка состоит из изделий одной партии.

4.1. Микросхема Н5503ХМ1-289

229 измерений, выборка содержится в 5 файлах, в каждом из которых примерно по 80 изделий.

4.2. Микросхема 1526ТЛ1

157 измерений, выборка содержится в 13 файлах, в каждом из которых примерно по 100 изделий.

4.3. Диод 2Д522Б

10 измерений, выборка содержится в 10 файлах, в каждом из которых по 150-500 изделий.

Таблица А.1 – Исходные данные для проведения диагностического контроля  
ИМС 1526ЛЕ5, 1526ЛЕ5ММ

Контролируемые параметры, единицы измерения	Тесты	Режим измерения
1 Входной ток низкого уровня, ПЛ(УР), мкА	11-18	U <sub>cc</sub> =11 В, U <sub>IH</sub> =11 В
2 Входной ток высокого уровня, ПН(УР), мкА	19-26	U <sub>cc</sub> =11 В, U <sub>IL</sub> =0 В
3 Максимальное выходное напряжение низкого уровня, UOLmax(УР), В	27-30	U <sub>cc</sub> =5 В, U <sub>IH</sub> =3.5 В
4 Минимальное выходное напряжение высокого уровня, UOHmin(УР), В	31-38	U <sub>cc</sub> =5 В, U <sub>IL</sub> =1.5 В
5 Выходной ток низкого уровня, IOL(УР), мА	39-42	U <sub>cc</sub> =5 В, U <sub>o</sub> =0.4 В
6 Выходной ток высокого уровня, IOH(УР), мА	43-46	U <sub>cc</sub> =5 В, U <sub>o</sub> =2.5 В
7 Ток потребления, I <sub>cc</sub> (УР), мкА	47-55	U <sub>cc</sub> =11 В, U <sub>IH</sub> =11 В
8 Прямая ВАХ верхнего диода, U, В	56,59,62,65,68,71,74,77	И=0.1 мкА
	57,60,63,66,69,72,75,78	И=100 мкА
	58,61,64,67,70,73,76,79	И=1 мА
9 Прямая ВАХ нижнего диода, U, В	80,83,86,89,92,95,98,101	И=0.1 мкА
	81,84,87,90,93,96,99,102	И=100 мкА
	82,85,88,91,94,97,100,103	И=1 мА
10 Прямая ВАХ диода цепи питания, U, В	104	И=0.1 мкА
	105	И=100 мкА
	106	И=10 мА
11 Выходная ВАХ нижнего транзистора, U, В	107,110,113,116	I <sub>o</sub> =0.1 мА
	108,111,114,117	I <sub>o</sub> =0.3 мА
	109,112,115,118	I <sub>o</sub> =1 мА
12 Выходная ВАХ верхнего транзистора, U, В	119,122,125,128	I <sub>o</sub> =0.1 мА
	120,123,126,129	I <sub>o</sub> =0.3 мА
	121,124,127,130	I <sub>o</sub> =1 мА

Результаты измерения для токов представлены в миллиамперах, напряжения - в вольтах.

**ПРИЛОЖЕНИЕ Б. СРАВНЕНИЕ РАБОТЫ РАЗЛИЧНЫХ  
АЛГОРИТМОВ ДЛЯ ЗАДАЧИ АВТОМАТИЧЕСКОЙ ГРУППИРОВКИ  
ЭЛЕКТРОРАДИОИЗДЕЛИЙ**

Таблица Б.1 – Результаты работы алгоритмов для задачи k-средних с различными мерами расстояния и k-медоид при фиксированном ограничении на время счета

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.			
		F(X) среднее	Среднекв. отклонение	min.	max.
<b>Ограничение времени работы алгоритмов – 2с</b>					
1526ТЛ1, 1 партия, N=535, p=10, l <sub>2</sub> <sup>2</sup> , d=167, k-средних	ГАЗЭ+ЛП	68488,16	1604,183	65876,05	70895,75
	ГАЗЭ вещ., $\sigma_e=0.25$	41241,64	274,5124	41053,64	41873,63
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	41320,83	240,2591	41064,81	41726,92
	ГА ФП	41411,09	206,4727	41121,98	41777,8
	ГА классич.	41265,66	253,0916	41068,86	41824,15
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	41705,75	0	41705,75	41705,75
	Детерм. ЖЭ, $\sigma_e=0.001$	41084,52	0	41084,52	41084,52
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	41142,28	164,8013	41059,09	41511,07
	ЖЭ адапт. $\sigma_e=0.001$	41081,39	30,57316	41055,45	41146,67
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	41108,67	30,83444	41060,42	41140,89
	ЖЭ, $\sigma_e=0.25, \beta=1$	41183,32	261,8287	41059,46	41773,39
	ЖЭ, $\sigma_e=0.25, \beta=3$	41095,83	29,15459	41074,2	41143,87
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	41095,39	24,25912	41068,95	41135,84
	ЖЭ, $\sigma_e=0.001, \beta=1$	41080,23	32,08646	41047,54	41140,42
	ЖЭ, $\sigma_e=0.001, \beta=3$	41083,99	22,21444	41056,38	41126,63
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	-	-	-	-
	ЖЛ, $\sigma_e=0.25, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
	ALA мультистарт	41350,63	264,7742	41094,15	41874,77
1526ТЛ1, сборная партия (фактически - 3 партии), N=1234, p=10, l <sub>2</sub> <sup>2</sup> , d=167, k-средних	ГАЗЭ+ЛП	76316,53	3355,313	71935,03	83226,89
	ГАЗЭ вещ., $\sigma_e=0.25$	44026,86	313,0775	43842,59	44532,79
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	44157,72	540,9987	43842,1	45118,74
	ГА ФП	44376,35	484,5951	43842,94	45137,42
	ГА классич.	44243,8	472,6486	43842,94	45009,68
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	45113,56	0	45113,56	45113,56
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	43844,01	1,515657	43841,98	43845,73
	ЖЭ адапт. $\sigma_e=0.001$	44011,36	440,2209	43844,18	45009,68
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	44010,85	440,1839	43842,94	45009,09
	ЖЭ, $\sigma_e=0.25, \beta=1$	43844,57	0,669047	43844,18	43845,73
	ЖЭ, $\sigma_e=0.25, \beta=3$	43844,34	0,995868	43843,52	43845,73
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	43843,93	1,021064	43842,74	43845,73
	ЖЭ, $\sigma_e=0.001, \beta=1$	43845,07	0,828746	43844,18	43845,73
	ЖЭ, $\sigma_e=0.001, \beta=3$	43845,51	0,586012	43844,18	43845,73
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	-	-	-	-
	ЖЛ, $\sigma_e=0.25, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
	ALA мультистарт	44354,11	525,6823	43842,45	45051,41

Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=2c.			
		F(X) среднее	Среднекв. отклонение	min.	max.
2Д522Б, сборная партия (фактически - 5 партий), N=3711, p=10, $l_2^2$ , d=10, k-средних	ГАЗЭ+ЛП	11165,27	623,2074	10011,94	12203,67
	ГАЗЭ вещ., $\sigma_e=0.25$	7719,683	7,299164	7714,506	7735,961
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	7721,796	6,800469	7714,237	7731,85
	ГА ФП	7720,096	8,003918	7714,475	7737,128
	ГА классич.	7721,921	5,393244	7714,406	7725,855
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	7762,998	46,86841	7725,183	7834,725
	ЖЭ адапт. $\sigma_e=0.001$	7780,122	53,99543	7725,855	7834,809
	ЖЭ, $\sigma_e=0.25$ , $\beta=0.5$	7756,036	53,87982	7714,814	7834,629
	ЖЭ, $\sigma_e=0.25$ , $\beta=1$	7757,436	55,55182	7714,798	7868,245
	ЖЭ, $\sigma_e=0.25$ , $\beta=3$	7719,256	5,512919	7714,798	7725,152
	ЖЭ, $\sigma_e=0.001$ , $\beta=0.5$	7742,886	40,53269	7727,212	7834,805
	ЖЭ, $\sigma_e=0.001$ , $\beta=1$	7800,83	67,54767	7714,798	7868,66
	ЖЭ, $\sigma_e=0.001$ , $\beta=3$	7714,921	0,269205	7714,798	7715,525
	ЖЛ, $\sigma_e=0.25$ , $\beta=0.5$	-	-	-	-
	ЖЛ, $\sigma_e=0.25$ , $\beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.25$ , $\beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001$ , $\beta=0.5$	-	-	-	-
	ЖЛ, $\sigma_e=0.001$ , $\beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.001$ , $\beta=3$	-	-	-	-
ALA мультистарт	7722,482	7,548629	7714,805	7735,961	
Н5503ХМ1-289, сборная партия (фактически - 5 партий), N=3711, p=10, $l_2^2$ , d=229, k-средних	ГАЗЭ+ЛП	72033,34	1442,642	69439,39	74435,65
	ГАЗЭ вещ., $\sigma_e=0.25$	43883,34	88,35572	43726,25	44035,96
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	43867,58	99,76776	43747,77	44082,34
	ГА ФП	43838,2	84,94666	43734,08	44018,97
	ГА классич.	43877,05	112,9201	43759,57	44150,53
	ИВС, $\sigma_e=0.25$	-	0	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	43830,25	0	43830,25	43830,25
	Детерм. ЖЭ, $\sigma_e=0.001$	44573,13	0	44573,13	44573,13
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.001$	43733,05	27,72009	43700,33	43769,68
	ЖЭ, $\sigma_e=0.25$ , $\beta=0.5$	43733,95	18,03182	43705,64	43759,95
	ЖЭ, $\sigma_e=0.25$ , $\beta=1$	43739,25	15,33142	43723,28	43762,96
	ЖЭ, $\sigma_e=0.25$ , $\beta=3$	43722,85	24,93519	43702,75	43776,13
	ЖЭ, $\sigma_e=0.001$ , $\beta=0.5$	43738,84	11,434	43720,5	43751,49
	ЖЭ, $\sigma_e=0.001$ , $\beta=1$	43728,01	9,256022	43715,65	43743,63
	ЖЭ, $\sigma_e=0.001$ , $\beta=3$	43716,48	17,1005	43698,56	43737,53
	ЖЛ, $\sigma_e=0.25$ , $\beta=0.5$	-	-	43752,35	-
	ЖЛ, $\sigma_e=0.25$ , $\beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.25$ , $\beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001$ , $\beta=0.5$	-	-	43782,29	-
	ЖЛ, $\sigma_e=0.001$ , $\beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.001$ , $\beta=3$	-	-	-	-
ALA мультистарт	43842,88	105,038	43701,35	44051,44	

Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.			
		F(X) среднее	Среднекв. отклонение	min.	max.
1526ТЛ1, 1 партия, N=535, p=10, l <sub>1</sub> , d=167, k-средних	ГАЗЭ+ЛП	39426,1	461,9336	38756,45	40174,92
	ГАЗЭ вещ., $\sigma_e=0.25$	30547,4	64,25617	30452,76	30635,1
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	30553,8	73,59293	30479,56	30697,15
	ГА ФП	30538,21	49,19797	30477,23	30618,47
	ГА классич.	30544,85	92,25796	30433,9	30701,32
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	30761,52	0	30761,52	30761,52
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	30464,17	16,15407	30431,04	30478,03
	ЖЭ адапт. $\sigma_e=0.001$	30460,48	17,58954	30435,49	30488,31
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	30458,45	36,12319	30426,76	30528,83
	ЖЭ, $\sigma_e=0.25, \beta=1$	30459,17	17,40135	30426,84	30474,85
	ЖЭ, $\sigma_e=0.25, \beta=3$	30481,32	23,79719	30431,61	30506,16
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	30446,15	19,14042	30426,91	30478,37
	ЖЭ, $\sigma_e=0.001, \beta=1$	30459,14	11,96865	30442,87	30478,03
	ЖЭ, $\sigma_e=0.001, \beta=3$	30488	18,55914	30465,23	30507,6
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	-	-	-	-
	ЖЛ, $\sigma_e=0.25, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
	ALA мультистарт	30530,34	71,78038	30464,25	30677,97
1526ТЛ1, сборная партия (фактически - 3 партии), N=1234, p=10, l <sub>1</sub> , d=167, k-средних	ГАЗЭ+ЛП	56705,64	1114,738	55591,2	58398,34
	ГАЗЭ вещ., $\sigma_e=0.25$	41823,32	160,0799	41717,02	42148,15
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	41777,77	102,9083	41717,02	41970,01
	ГА ФП	41754,18	60,07712	41713,21	41848,02
	ГА классич.	41809,23	95,95293	41717,02	41968,13
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	41749,23	58,8598	41714,08	41835,37
	ЖЭ адапт. $\sigma_e=0.001$	41779,22	112,263	41714,08	41994,96
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	41716,67	2,395126	41713,78	41721,03
	ЖЭ, $\sigma_e=0.25, \beta=1$	41738,52	60,65122	41713,21	41875,99
	ЖЭ, $\sigma_e=0.25, \beta=3$	41779,46	78,70181	41717,19	41875,99
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	41716,47	4,289055	41713,51	41725,4
	ЖЭ, $\sigma_e=0.001, \beta=1$	41738,27	58,65238	41713,78	41871,23
	ЖЭ, $\sigma_e=0.001, \beta=3$	41853,45	60,07198	41717,23	41876,52
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	-	-	-	-
	ЖЛ, $\sigma_e=0.25, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
	ALA мультистарт	41815,03	68,374	41714,08	41870,54

Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.			
		F(X) среднее	Среднекв. отклонение	min.	max.
2Д522Б, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>1</sub> , d=10, k- средних	ГАЗЭ+ЛП	14006,32	224,8713	13669,1	14251,31
	ГАЗЭ вещ., $\sigma_e=0.25$	11781,02	6,700716	11774,65	11794,39
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	11785,68	16,93447	11773,35	11822,54
	ГА ФП	11779,97	5,309673	11773,13	11787,61
	ГА классич.	11785,33	8,602068	11775,11	11799,77
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-
	ИВС, $\sigma_e=0.001$				
	ЖЭ адапт. $\sigma_e=0.25$				
	ЖЭ адапт. $\sigma_e=0.001$				
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	11780,27	3,22218	11774,81	11785,18
	ЖЭ, $\sigma_e=0.25, \beta=1$	11780,23	1,792554	11778,19	11783,35
	ЖЭ, $\sigma_e=0.25, \beta=3$	11779,34	1,255175	11777,91	11780,71
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	11779,63	1,216849	11778,17	11781,46
	ЖЭ, $\sigma_e=0.001, \beta=1$	11780,46	2,921288	11777,46	11785,19
	ЖЭ, $\sigma_e=0.001, \beta=3$	11781,55	2,422804	11778,13	11785,3
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	-	-	11779,84	-
	ЖЛ, $\sigma_e=0.25, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=3$				
	ALA мультистарт	11785,25	12,53276	11774,46	11805,64
Н5503ХМ1- 289, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>1</sub> , d=229, k- средних	ГАЗЭ+ЛП	56433,19	310,7503	55964,91	56930,07
	ГАЗЭ вещ., $\sigma_e=0.25$	42820,46	62,67032	42716,81	42892,93
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	42723,53	57,6827	42645	42779,7
	ГА ФП	42750,4	73,52909	42657,88	42845,15
	ГА классич.	42741,02	71,73337	42669,55	42885,49
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	43148,55	0	43148,55	43148,55
	Детерм. ЖЭ, $\sigma_e=0.001$	43131,92	0	43131,92	43131,92
	ИВС, $\sigma_e=0.001$				
	ЖЭ адапт. $\sigma_e=0.25$	42672,65	18,20016	42646,02	42697,11
	ЖЭ адапт. $\sigma_e=0.001$	42659,75	14,14282	42646,59	42679,24
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	42671,75	20,66601	42646,85	42709,48
	ЖЭ, $\sigma_e=0.25, \beta=1$	42679,48	31,42728	42644,62	42724,44
	ЖЭ, $\sigma_e=0.25, \beta=3$	42722,99	55,109	42657,92	42788,76
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	42657,4	15,83775	42630,29	42672,42
	ЖЭ, $\sigma_e=0.001, \beta=1$	42663,74	12,87346	42640,88	42679,47
	ЖЭ, $\sigma_e=0.001, \beta=3$	42692,51	13,17937	42677,97	42714,85
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	-	-	-	-
	ЖЛ, $\sigma_e=0.25, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
	ALA мультистарт				
	42820,89	91,68919	42690,43	42922,61	

Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.			
		F(X) среднее	Среднекв. отклонение	min.	max.
1526ТЛ1, 1 партия, N=535, p=10, l <sub>2</sub> <sup>2</sup> , d=167, k-медоид	ГАЗЭ+ЛП	54969,26	1114,044	54046,86	57067,41
	ГАЗЭ вещ., $\sigma_e=0.25$	53615,28	36,36333	53570,61	53678,42
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	53619,84	72,19232	53570,61	53778,22
	ГА ФП	54128,54	295,9456	53710,42	54591,59
	ГА классич.	54420,59	356,5645	53726,05	54910,83
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	54865,33	0	54865,33	54865,33
	Детерм. ЖЭ, $\sigma_e=0.001$	55076,02	0	55076,02	55076,02
	ИВС, $\sigma_e=0.001$				
	ЖЭ адапт. $\sigma_e=0.25$	53714,68	89,37873	53593,29	53835
	ЖЭ адапт. $\sigma_e=0.001$	53648,14	52,38842	53593,29	53712,76
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	53839,63	103,2514	53738,58	53985,74
	ЖЭ, $\sigma_e=0.25, \beta=1$	53714	92,57771	53593,29	53835
	ЖЭ, $\sigma_e=0.25, \beta=3$	53661,56	85,75532	53593,29	53783,69
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	53783,28	91,50668	53656,1	53911,48
	ЖЭ, $\sigma_e=0.001, \beta=1$	53715,08	58,20574	53605,15	53778,22
	ЖЭ, $\sigma_e=0.001, \beta=3$	53732,91	81,2352	53605,15	53818,03
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	54151,74	315,3198	53698,56	54554,07
	ЖЛ, $\sigma_e=0.25, \beta=1$	54047,26	81,49744	53984,75	54187,47
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	54326,26	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	54041,07	229,4315	53833,68	54510,72
	ЖЛ, $\sigma_e=0.001, \beta=1$	54087,81	200,3797	53833,68	54372,43
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
ALA мультистарт	54373,68	508,1843	53726,72	55149,23	
1526ТЛ1, сборная партия (фактически - 3 партии), N=1234, p=10, l <sub>2</sub> <sup>2</sup> , d=167, k- медоид	ГАЗЭ+ЛП	76527,12	1685,474	73905,2	78963,79
	ГАЗЭ вещ., $\sigma_e=0.25$	56408,83	894,6523	55361,75	57569,93
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	55946,77	504,985	55361,75	56763,11
	ГА ФП	56118,31	857,9773	55378,21	57504,53
	ГА классич.	56224,74	591,7999	55361,75	57117,48
	ИВС, $\sigma_e=0.25$				
	Детерм. ЖЭ, $\sigma_e=0.25$	57131	0	57131	57131
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-
	ИВС, $\sigma_e=0.001$				
	ЖЭ адапт. $\sigma_e=0.25$	56000,9	7,077099	55998,22	56016,95
	ЖЭ адапт. $\sigma_e=0.001$	55907,3	240,5632	55361,75	55998,22
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	55735,23	328,8059	55361,75	55998,22
	ЖЭ, $\sigma_e=0.25, \beta=1$	55907,3	240,5632	55361,75	55998,22
	ЖЭ, $\sigma_e=0.25, \beta=3$	55816,37	310,5658	55361,75	55998,22
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	56014,43	22,35868	55998,22	56045,59
	ЖЭ, $\sigma_e=0.001, \beta=1$	55695,29	304,7995	55361,75	55998,22
	ЖЭ, $\sigma_e=0.001, \beta=3$	55998,22	0	55998,22	55998,22
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	57572,96	1680,355	56016,95	59925,95
	ЖЛ, $\sigma_e=0.25, \beta=1$	56503,21	1062,831	55998,22	58841,54
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	56228,68	760,2916	55361,75	57840,88
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	55998,22	-
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
ALA мультистарт	56036,16	441,9047	55378,21	56764,13	

Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.			
		F(X) среднее	Среднекв. отклонение	min.	max.
2Д522Б, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>2</sub> <sup>2</sup> , d=10, k-медоид	ГАЗЭ+ЛП	10731,22	375,0125	10183,18	11229,31
	ГАЗЭ вещ., $\sigma_e=0.25$	8256,999	102,2087	8096,167	8364,051
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	8376,217	150,1142	8265,063	8629,745
	ГА ФП	8371,359	168,7122	8168,053	8576,806
	ГА классич.	8348,236	106,8432	8234,429	8490,581
	ИВС, $\sigma_e=0.25$				
	Детерм. ЖЭ, $\sigma_e=0.25$	-	0	-	-
	Детерм. ЖЭ, $\sigma_e=0.001$	0	0	-	0
	ИВС, $\sigma_e=0.001$				
	ЖЭ адапт. $\sigma_e=0.25$	8951,09	59,43573	8900,839	9062,453
	ЖЭ адапт. $\sigma_e=0.001$	8958,052	57,63062	8885,045	9054,963
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	8909,335	40,67501	8865,373	8972,085
	ЖЭ, $\sigma_e=0.25, \beta=1$	8703,506	415,8929	8089,399	9025,67
	ЖЭ, $\sigma_e=0.25, \beta=3$	8919,944	32,91986	8887,94	8982,219
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	8932,346	39,87828	8864,237	8995,414
	ЖЭ, $\sigma_e=0.001, \beta=1$	8922,497	16,10003	8909,529	8945,816
	ЖЭ, $\sigma_e=0.001, \beta=3$	8947,776	34,73965	8908,563	9001,961
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	-	-	-	-
	ЖЛ, $\sigma_e=0.25, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
	ALA мультистарт	8280,247	145,7731	8114,867	8567,841
Н5503ХМ1-289, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>2</sub> <sup>2</sup> , d=229, k-медоид	ГАЗЭ+ЛП	57926,07	733,5031	57302,73	59247,57
	ГАЗЭ вещ., $\sigma_e=0.25$	57334,92	85,1819	57302,73	57528,1
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	57354,39	71,98364	57302,73	57460,27
	ГА ФП	57588,7	149,7942	57380,19	57746,45
	ГА классич.	58183,5	174,3709	58011,15	58398,39
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	61479,75	0	61479,75	61479,75
	Детерм. ЖЭ, $\sigma_e=0.001$	58622,54	0	58622,54	58622,54
	ИВС, $\sigma_e=0.001$				
	ЖЭ адапт. $\sigma_e=0.25$	57635,82	176,726	57460,27	57962,49
	ЖЭ адапт. $\sigma_e=0.001$	57491,82	143,7374	57380,19	57765
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	57593,26	252,2448	57380,19	58077,84
	ЖЭ, $\sigma_e=0.25, \beta=1$	57487,54	146,1221	57302,73	57689,55
	ЖЭ, $\sigma_e=0.25, \beta=3$	57570,41	124,4886	57460,27	57767,02
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	57617,35	114,3752	57456,66	57824,2
	ЖЭ, $\sigma_e=0.001, \beta=1$	57480,44	106,1851	57380,19	57659,68
	ЖЭ, $\sigma_e=0.001, \beta=3$	57518,45	123,0822	57380,19	57767,02
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	57911,11	367,2494	57432,32	58443,52
	ЖЛ, $\sigma_e=0.25, \beta=1$	57761,95	205,8463	57613,73	58074,47
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	57845,37	284,6465	57614,95	58414,73
	ЖЛ, $\sigma_e=0.001, \beta=1$	57914,11	444,3286	57302,73	58483,2
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
	ALA мультистарт	57901,33	286,103	57533,7	58425,25



Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=15с.			
		F(X) среднее	Среднекв. отклонение	min.	max.
<b>Б) Ограничение времени работы алгоритмов – 15 с.</b>					
1526ТЛ1, 1 партия, N=535, p=10, l <sub>2</sub> <sup>2</sup> , d=167, k-средних	ГАЗЭ+ЛП	41684,19	416,9586	41054,63	42403,72
	ГАЗЭ вещ., $\sigma_e=0.25$	41066,84	11,99397	41053,64	41090,82
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	41054,67	7,07508	41044,79	41068,16
	ГА ФП	41061,68	17,97617	41041,38	41090,21
	ГА классич.	41068,26	20,11588	41039,14	41117,3
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	41705,75	0	41705,75	41705,75
	Детерм. ЖЭ, $\sigma_e=0.001$	41084,52	0	41084,52	41084,52
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	41066,49	7,693992	41058,66	41077,37
	ЖЭ адапт. $\sigma_e=0.001$	41061,31	8,049168	41054,05	41072,95
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	41063,77	12,77174	41042,95	41082,91
	ЖЭ, $\sigma_e=0.25, \beta=1$	41066,91	10,08632	41057,74	41080,72
	ЖЭ, $\sigma_e=0.25, \beta=3$	41066,61	10,88411	41055,99	41088,56
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	41060,33	8,509522	41046,24	41072,99
	ЖЭ, $\sigma_e=0.001, \beta=1$	41056,69	5,45379	41047,54	41062,45
	ЖЭ, $\sigma_e=0.001, \beta=3$	41066,21	9,837841	41056,38	41080,95
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	41296,57	373,3102	41050,65	41903,65
	ЖЛ, $\sigma_e=0.25, \beta=1$	41381,6	360,8979	41070,97	41769,85
	ЖЛ, $\sigma_e=0.25, \beta=3$	41494,85	396,6371	41040,26	41880,61
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	41580,7	347,7389	41066,16	41802,75
	ЖЛ, $\sigma_e=0.001, \beta=1$	41585,46	350,4148	41056,59	41828,74
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
ALA мультистарт	41090,92	53,83281	41050,21	41240,7	
1526ТЛ1, сборная партия (фактически - 3 партии), N=1234, p=10, l <sub>2</sub> <sup>2</sup> , d=167, k-средних	ГАЗЭ+ЛП	58743,05	15974,95	43842,1	79104,14
	ГАЗЭ вещ., $\sigma_e=0.25$	43843,33	0,876148	43841,98	43844,18
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	43842,18	0,504141	43841,98	43843,69
	ГА ФП	43948,56	351,8414	43842,1	45009,41
	ГА классич.	43843,31	0,951543	43842,1	43845,33
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	45113,56	0	45113,56	45113,56
	Детерм. ЖЭ, $\sigma_e=0.001$	45021,21	0	45021,21	45021,21
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	43843,32	0,917788	43841,98	43844,18
	ЖЭ адапт. $\sigma_e=0.001$	43844,18	0	43844,18	43844,18
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	43843,41	0,852229	43841,98	43844,18
	ЖЭ, $\sigma_e=0.25, \beta=1$	43843,64	0,829708	43842,1	43844,18
	ЖЭ, $\sigma_e=0.25, \beta=3$	43843,52	0	43843,52	43843,52
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	43843,57	0,765762	43842,74	43844,18
	ЖЭ, $\sigma_e=0.001, \beta=1$	43843,98	0,54327	43842,74	43844,18
	ЖЭ, $\sigma_e=0.001, \beta=3$	43845,29	0,756538	43844,18	43845,73
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	44177	569,3111	43842,74	45011,09
	ЖЛ, $\sigma_e=0.25, \beta=1$	44177,74	571,1261	43842,74	45017,86
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	44176,42	568,8243	43842,74	45009,09
	ЖЛ, $\sigma_e=0.001, \beta=1$	44601,07	553,3409	43842,94	45016,94
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
ALA мультистарт	43842,67	0,290075	43841,98	43842,94	

Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=15с.			
		F(X) среднее	Среднекв. отклонение	min.	max.
2Д522Б, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>2</sub> <sup>2</sup> , d=10, k-средних	ГАЗЭ+ЛП	7956,146	369,0633	7714,129	8629,784
	ГАЗЭ вещ., $\sigma_e=0.25$	7714,732	0,112576	7714,506	7714,819
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	7714,561	0,234459	7714,219	7714,8
	ГА ФП	7714,379	0,152647	7714,141	7714,635
	ГА классич.	7714,522	0,342544	7714,259	7715,512
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	7902,211	0	7902,211	7902,211
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	7721,178	6,007967	7714,793	7727,582
	ЖЭ адапт. $\sigma_e=0.001$	7721,576	5,694643	7715,481	7727,212
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	7721,909	6,662947	7714,814	7728,255
	ЖЭ, $\sigma_e=0.25, \beta=1$	7718,066	5,572742	7714,773	7727,212
	ЖЭ, $\sigma_e=0.25, \beta=3$	7714,739	0,121612	7714,476	7714,819
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	7719,96	6,491701	7714,669	7727,55
	ЖЭ, $\sigma_e=0.001, \beta=1$	7721,881	6,409195	7714,798	7727,57
	ЖЭ, $\sigma_e=0.001, \beta=3$	7714,795	0,009477	7714,773	7714,8
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	7932,099	366,4525	7714,394	8468,545
	ЖЛ, $\sigma_e=0.25, \beta=1$	7935,872	364,1233	7714,298	8468,994
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	7824,218	284,204	7714,476	8468,635
	ЖЛ, $\sigma_e=0.001, \beta=1$	7827,088	282,9837	7714,285	8468,509
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
	ALA мультистарт	7714,405	0,160155	7714,141	7714,707
Н5503ХМ1-289, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>2</sub> <sup>2</sup> , d=229, k-средних	ГАЗЭ+ЛП	46393,51	8660,852	43732,03	72506,64
	ГАЗЭ вещ., $\sigma_e=0.25$	43694,18	5,717544	43684,03	43703,36
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	43691,44	4,225457	43686,89	43698,08
	ГА ФП	43731,84	16,07238	43711,85	43757,24
	ГА классич.	43744,84	12,53196	43727,93	43773,75
	ИВС, $\sigma_e=0.25$	-	0	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	43830,25	0	43830,25	43830,25
	Детерм. ЖЭ, $\sigma_e=0.001$	44573,13	0	44573,13	44573,13
	ИВС, $\sigma_e=0.001$	0	0	-	0
	ЖЭ адапт. $\sigma_e=0.25$	0	0	-	0
	ЖЭ адапт. $\sigma_e=0.001$	43705,72	9,256906	43698,4	43723,6
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	43710,76	12,59137	43697,64	43727,29
	ЖЭ, $\sigma_e=0.25, \beta=1$	43703,36	8,239231	43693,41	43714,14
	ЖЭ, $\sigma_e=0.25, \beta=3$	43703,4	13,76373	43688,91	43726,61
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	43716,59	15,23396	43701,11	43742,16
	ЖЭ, $\sigma_e=0.001, \beta=1$	43697,28	11,96721	43684,54	43718,39
	ЖЭ, $\sigma_e=0.001, \beta=3$	43705,43	9,925234	43697,63	43724,32
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	43750,14	26,27248	43713,68	43784,36
	ЖЛ, $\sigma_e=0.25, \beta=1$	43754,71	49,47832	43702,32	43854,43
	ЖЛ, $\sigma_e=0.25, \beta=3$	43774,7	44,11336	43724,47	43860,09
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	43732,75	15,62881	43707,93	43752,95
	ЖЛ, $\sigma_e=0.001, \beta=1$	43740,79	24,58402	43710,97	43782,53
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
	ALA мультистарт	43758,97	24,05854	43701,35	43799,11

Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=15с.			
		F(X) среднее	Среднекв. отклонение	min.	max.
1526ТЛ1, 1 партия, N=535, p=10, l <sub>1</sub> , d=167, k-средних	ГАЗЭ+ЛП	39426,1	461,9336	38756,45	40174,92
	ГАЗЭ вещ., $\sigma_e=0.25$	30439,18	9,487806	30426,55	30450,33
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	30425,5	2,7013	30423,18	30430,74
	ГА ФП	30449,84	22,63496	30424,99	30482,56
	ГА классич.	30456,41	33,20892	30425,51	30527,87
	ИВС, $\sigma_e=0.25$	-	0	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	30761,52	0	30761,52	30761,52
	Детерм. ЖЭ, $\sigma_e=0.001$	30539,46	0	30539,46	30539,46
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	30437,19	11,89176	30423,86	30457,07
	ЖЭ адапт. $\sigma_e=0.001$	30443,2	10,1058	30435,49	30465,55
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	30428,86	2,56671	30426,39	30433,39
	ЖЭ, $\sigma_e=0.25, \beta=1$	30438,99	12,92284	30425,7	30461,02
	ЖЭ, $\sigma_e=0.25, \beta=3$	30460,68	13,49443	30431,61	30472,16
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	30431,92	10,91483	30426,26	30456,43
	ЖЭ, $\sigma_e=0.001, \beta=1$	30430,34	5,385294	30426,13	30439,31
	ЖЭ, $\sigma_e=0.001, \beta=3$	30467,6	5,132229	30460,9	30473,7
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	30451,16	24,85679	30423,6	30491,87
	ЖЛ, $\sigma_e=0.25, \beta=1$	30466,59	27,74429	30447,46	30525,19
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	30454,48	25,0127	30424,15	30490,92
	ЖЛ, $\sigma_e=0.001, \beta=1$	30485,06	25,32765	30458,46	30533,19
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
	ALA мультистарт	30474,88	26,87421	30435,49	30503,62
1526ТЛ1, сборная партия (фактически - 3 партии), N=1234, p=10, l <sub>1</sub> , d=167, k-средних	ГАЗЭ+ЛП	56705,64	1114,738	55591,2	58398,34
	ГАЗЭ вещ., $\sigma_e=0.25$	41714,27	1,17955	41713,51	41716,89
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	41714,29	1,258834	41713,51	41717,02
	ГА ФП	41714,43	1,26466	41713,21	41716,72
	ГА классич.	41733,79	50,3852	41713,78	41848,02
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	42136,63	0	42136,63	42136,63
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	41714,29	1,094582	41713,51	41716,72
	ЖЭ адапт. $\sigma_e=0.001$	41714,12	1,25648	41713,21	41716,89
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	41713,96	0,445546	41713,51	41714,84
	ЖЭ, $\sigma_e=0.25, \beta=1$	41714,1	1,177763	41713,21	41716,72
	ЖЭ, $\sigma_e=0.25, \beta=3$	41717,07	0,149181	41716,89	41717,23
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	41713,74	0,247726	41713,51	41714,08
	ЖЭ, $\sigma_e=0.001, \beta=1$	41713,76	0,339144	41713,21	41714,08
	ЖЭ, $\sigma_e=0.001, \beta=3$	41739,76	59,94053	41716,73	41875,69
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	41714,85	1,510622	41713,32	41717,02
	ЖЛ, $\sigma_e=0.25, \beta=1$	41714,99	1,492229	41713,51	41718,15
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	41714,75	1,355851	41713,78	41716,72
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
	ALA мультистарт	41714,81	1,880815	41713,51	41717,85

Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=15с.			
		F(X) среднее	Среднекв. отклонение	min.	max.
2Д522Б, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>1</sub> , d=10, k-средних	ГАЗЭ+ЛП	12422,33	1097,55	11774,81	14241,62
	ГАЗЭ вещ., $\sigma_e=0.25$	11775,99	1,711883	11773,38	11777,91
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	11776	1,790302	11773,35	11778,18
	ГА ФП	11775,16	2,325585	11773,13	11778,48
	ГА классич.	11775,69	2,243109	11773,17	11779,16
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	11794.91965	0	11794.919653	11794.919653
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-
	ИВС, $\sigma_e=0.001$				
	ЖЭ адапт. $\sigma_e=0.25$				
	ЖЭ адапт. $\sigma_e=0.001$				
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	11776,16	1,79084	11774,42	11778,39
	ЖЭ, $\sigma_e=0.25, \beta=1$	11777,55	1,112884	11775,1	11778,33
	ЖЭ, $\sigma_e=0.25, \beta=3$	11777,98	0,136718	11777,86	11778,19
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	11776,33	2,152755	11772,96	11778,36
	ЖЭ, $\sigma_e=0.001, \beta=1$	11778,2	0,743284	11777,46	11779,77
	ЖЭ, $\sigma_e=0.001, \beta=3$	11778,16	0,154197	11777,91	11778,39
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	11776,52	2,144031	11773,35	11778,32
	ЖЛ, $\sigma_e=0.25, \beta=1$	11778,82	4,957253	11773,35	11788,86
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	11778,2	3,032819	11775,09	11784,17
	ЖЛ, $\sigma_e=0.001, \beta=1$	11786,3	17,46161	11773,99	11823,18
	ЖЛ, $\sigma_e=0.001, \beta=3$				
	АЛА мультистарт	11775,5	1,559571	11774,1	11778,23
Н5503ХМ1-289, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>1</sub> , d=229, k-средних	ГАЗЭ+ЛП	56433,19	310,7503	55964,91	56930,07
	ГАЗЭ вещ., $\sigma_e=0.25$	42661,69	16,92555	42643,34	42684,1
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	42650,48	5,292793	42644,17	42658,31
	ГА ФП	42655,73	20,93271	42629,43	42692,42
	ГА классич.	42677,76	29,95391	42636,15	42722,47
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	43148,55	0	43148,55	43148,55
	Детерм. ЖЭ, $\sigma_e=0.001$	43131,92	0	43131,92	43131,92
	ИВС, $\sigma_e=0.001$				
	ЖЭ адапт. $\sigma_e=0.25$	42648,29	11,40647	42634,01	42667,82
	ЖЭ адапт. $\sigma_e=0.001$	42651,81	7,328311	42646,59	42667,57
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	42646,45	6,669056	42637,22	42658,52
	ЖЭ, $\sigma_e=0.25, \beta=1$	42652,5	7,02551	42644,62	42662,23
	ЖЭ, $\sigma_e=0.25, \beta=3$	42657,59	10,77978	42644,47	42670,67
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	42640,46	11,01157	42625,72	42653,97
	ЖЭ, $\sigma_e=0.001, \beta=1$	42638,27	8,611374	42628,02	42651,13
	ЖЭ, $\sigma_e=0.001, \beta=3$	42658,58	11,73806	42646,36	42678,43
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	42697,49	51,95109	42643,07	42799,3
	ЖЛ, $\sigma_e=0.25, \beta=1$	42694,55	37,76707	42649,21	42769,96
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	42671,92	52,52753	42639,49	42787,16
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
	АЛА мультистарт	42672,65	33,02221	42640,64	42733,87

Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=15с.			
		F(X) среднее	Среднекв. отклонение	min.	max.
1526ТЛ1, 1 партия, N=535, p=10, l <sub>2</sub> <sup>2</sup> , d=167, k-медоид	ГАЗЭ+ЛП	53774,54	201,3956	53570,61	54093,07
	ГАЗЭ вещ., $\sigma_e=0.25$	53570,61	7,86E-12	53570,61	53570,61
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	53570,61	7,86E-12	53570,61	53570,61
	ГА ФП	53672,84	99,0096	53582,47	53778,22
	ГА классич.	53801,94	153,6583	53638,52	54034,2
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	54865,33	0	54865,33	54865,33
	Детерм. ЖЭ, $\sigma_e=0.001$	55076,02	0	55076,02	55076,02
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	53590,2	14,39388	53570,61	53605,15
	ЖЭ адапт. $\sigma_e=0.001$	53602,81	18,12927	53593,29	53638,52
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	53644,1	60,02887	53570,61	53712,76
	ЖЭ, $\sigma_e=0.25, \beta=1$	53595,04	25,83398	53570,61	53628,33
	ЖЭ, $\sigma_e=0.25, \beta=3$	53595,87	9,554007	53582,47	53605,15
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	53696,14	99,43524	53582,47	53833,68
	ЖЭ, $\sigma_e=0.001, \beta=1$	53590,2	10,84168	53570,61	53605,15
	ЖЭ, $\sigma_e=0.001, \beta=3$	53595,66	5,302403	53593,29	53605,15
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	53765,27	49,36295	53698,56	53846,86
	ЖЛ, $\sigma_e=0.25, \beta=1$	53709,16	138,9541	53570,61	53902,04
	ЖЛ, $\sigma_e=0.25, \beta=3$	53801,15	150,1069	53605,15	54022,32
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	53840,4	133,5393	53593,29	53995,45
	ЖЛ, $\sigma_e=0.001, \beta=1$	53728,42	46,75348	53678,42	53778,22
	ЖЛ, $\sigma_e=0.001, \beta=3$	53820,45	166,7971	53605,15	54022,32
	ALA мультистарт	53907,5	228,145	53582,47	54198,91
1526ТЛ1, сборная партия (фактически - 3 партии), N=1234, p=10, l <sub>2</sub> <sup>2</sup> , d=167, k-медоид	ГАЗЭ+ЛП	55814,76	333,9305	55361,75	56118,47
	ГАЗЭ вещ., $\sigma_e=0.25$	55361,75	7,86E-12	55361,75	55361,75
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	55361,75	7,86E-12	55361,75	55361,75
	ГА ФП	55455,03	239,605	55361,75	55998,22
	ГА классич.	55574,27	271,3057	55361,75	56083,57
	ИВС, $\sigma_e=0.25$				
	Детерм. ЖЭ, $\sigma_e=0.25$	57131	0	57131	57131
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-
	ИВС, $\sigma_e=0.001$				
	ЖЭ адапт. $\sigma_e=0.25$	55452,68	240,5632	55361,75	55998,22
	ЖЭ адапт. $\sigma_e=0.001$	55553,38	304,9125	55361,75	55998,22
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	55474,58	232,9591	55361,75	55998,22
	ЖЭ, $\sigma_e=0.25, \beta=1$	55408,9	97,95398	55361,75	55623,38
	ЖЭ, $\sigma_e=0.25, \beta=3$	55482,01	230,1871	55361,75	55998,22
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	55474,58	232,9591	55361,75	55998,22
	ЖЭ, $\sigma_e=0.001, \beta=1$	55364,1	6,220381	55361,75	55378,21
	ЖЭ, $\sigma_e=0.001, \beta=3$	55644,3	331,9451	55361,75	55998,22
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	55915	236,8664	55378,21	56016,95
	ЖЛ, $\sigma_e=0.25, \beta=1$	55998,22	0	55998,22	55998,22
	ЖЛ, $\sigma_e=0.25, \beta=3$	56602,87	1444,802	55998,22	59877,14
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	55826,15	294,5315	55361,75	55998,22
	ЖЛ, $\sigma_e=0.001, \beta=1$	56014,11	27,54303	55998,22	56062,05
	ЖЛ, $\sigma_e=0.001, \beta=3$	56004,99	17,90477	55998,22	56045,59
	ALA мультистарт				
	55417,44	125,8347	55361,75	55702,19	

Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=15с.			
		F(X) среднее	Среднекв. отклонение	min.	max.
2Д522Б, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>2</sub> <sup>2</sup> , d=10, k-медоид	ГАЗЭ+ЛП	10440,75	767,7918	8907,068	11229,31
	ГАЗЭ вещ., $\sigma_e=0.25$	8167,633	82,3343	8082,939	8308,032
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	8189,273	55,18503	8093,754	8241,281
	ГА ФП	8191,589	70,2997	8108,876	8295,172
	ГА классич.	8190,107	64,89582	8102,107	8270,578
	ИВС, $\sigma_e=0.25$				
	Детерм. ЖЭ, $\sigma_e=0.25$	8502,325	0	8502,325	8502,325
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-
	ИВС, $\sigma_e=0.001$				
	ЖЭ адапт. $\sigma_e=0.25$	8810,581	223,8693	8304,179	8923,612
	ЖЭ адапт. $\sigma_e=0.001$	8820,961	233,7664	8292,287	8931,26
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	8803,182	212,5272	8322,47	8905,751
	ЖЭ, $\sigma_e=0.25, \beta=1$	8581,264	387,9482	8089,399	8901,708
	ЖЭ, $\sigma_e=0.25, \beta=3$	8709,716	310,8033	8209,548	8906,228
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	8859,684	61,46769	8729,611	8917,038
	ЖЭ, $\sigma_e=0.001, \beta=1$	8804,552	218,9172	8309,637	8915,399
	ЖЭ, $\sigma_e=0.001, \beta=3$	8803,738	269,746	8192,491	8920,117
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	8922,186	43,29741	8874,693	8968,821
	ЖЛ, $\sigma_e=0.25, \beta=1$	8907,388	39,65478	8851,685	8969,009
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	8907,813	75,25842	8863,319	9077,517
	ЖЛ, $\sigma_e=0.001, \beta=1$	8956,733	76,08909	8877,522	9102,696
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
	ALA мультистарт	8163,18	61,34988	8094,976	8242,068
Н5503ХМ1-289, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>2</sub> <sup>2</sup> , d=229, k-медоид	ГАЗЭ+ЛП	57423,46	191,8602	57302,73	57791,99
	ГАЗЭ вещ., $\sigma_e=0.25$	57302,73	0	57302,73	57302,73
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	57302,73	0	57302,73	57302,73
	ГА ФП	57364,62	62,87532	57327,81	57456,66
	ГА классич.	57584,58	111,3013	57405,27	57737,97
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	61479,75	0	61479,75	61479,75
	Детерм. ЖЭ, $\sigma_e=0.001$	58622,54	0	58622,54	58622,54
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	57339,88	60,25596	57302,73	57460,27
	ЖЭ адапт. $\sigma_e=0.001$	57328,44	36,51752	57302,73	57380,19
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	57373,08	55,80955	57302,73	57460,27
	ЖЭ, $\sigma_e=0.25, \beta=1$	57302,73	0	57302,73	57302,73
	ЖЭ, $\sigma_e=0.25, \beta=3$	57335,78	60,61827	57302,73	57456,66
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	57388,22	88,38207	57302,73	57508,62
	ЖЭ, $\sigma_e=0.001, \beta=1$	57324,86	37,79812	57302,73	57380,19
	ЖЭ, $\sigma_e=0.001, \beta=3$	57362,01	61,54943	57302,73	57460,27
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	57559,59	181,311	57302,73	57755,52
	ЖЛ, $\sigma_e=0.25, \beta=1$	57436,47	136,5429	57302,73	57643,37
	ЖЛ, $\sigma_e=0.25, \beta=3$	57640,46	268,6301	57302,73	58145,91
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	57532	87,83554	57380,19	57615,6
	ЖЛ, $\sigma_e=0.001, \beta=1$	57439,54	89,0829	57302,73	57528,1
	ЖЛ, $\sigma_e=0.001, \beta=3$	57806,14	362,7907	57302,73	58297,19
	ALA мультистарт	57656,71	64,00913	57533,7	57738,7

Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=60с.			
		F(X) среднее	Среднекв. отклон.	min.	max.
В) Ограничение времени работы алгоритмов – 60 с.					
1526ТЛ1, 1 партия, N=535, p=10, l <sub>2</sub> <sup>2</sup> , d=167, k-средних	ГАЗЭ+ЛП	41132,3	216,5863	41049,53	41783,6
	ГАЗЭ вещ., $\sigma_e=0.25$	41053,11	4,998814	41043,74	41059,12
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	41040,03	0,99638	41038,53	41041,21
	ГА ФП	41046,22	4,862504	41039,28	41056,11
	ГА классич.	41045,72	5,457131	41039,14	41056,79
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	41705,75	0	41705,75	41705,75
	Детерм. ЖЭ, $\sigma_e=0.001$	41084,52	0	41084,52	41084,52
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	41054,23	1,953287	41051,59	41056,84
	ЖЭ адапт. $\sigma_e=0.001$	41051,57	3,141884	41046,24	41055,4
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	41053,22	7,767535	41042,95	41066,98
	ЖЭ, $\sigma_e=0.25, \beta=1$	41050,04	3,865743	41045,81	41055,33
	ЖЭ, $\sigma_e=0.25, \beta=3$	41054,11	4,208448	41048,16	41058,96
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	41051,49	3,709106	41046,24	41057,25
	ЖЭ, $\sigma_e=0.001, \beta=1$	41053,46	5,969522	41042,59	41061,47
	ЖЭ, $\sigma_e=0.001, \beta=3$	41057,94	7,114785	41048,16	41071,73
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	41161,57	266,7337	41043,74	41765,48
	ЖЛ, $\sigma_e=0.25, \beta=1$	41059,72	12,16037	41046,24	41082,3
	ЖЛ, $\sigma_e=0.25, \beta=3$	41056,05	9,623252	41040,26	41068,86
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	41059,28	8,792072	41044,24	41067,7
	ЖЛ, $\sigma_e=0.001, \beta=1$	41118,14	155,3008	41050,76	41469,71
	ЖЛ, $\sigma_e=0.001, \beta=3$	41083,04	7,282893	41070,64	41092,38
АЛА мультистарт	41055,59	5,315283	41047,34	41062,33	
1526ТЛ1, сборная партия (фактически - 3 партии), N=1234, p=10, l <sub>2</sub> <sup>2</sup> , d=167, k-средних	ГАЗЭ+ЛП	43843,72	1,319931	43842,1	43845,73
	ГАЗЭ вещ., $\sigma_e=0.25$	43842,6	0,676282	43841,98	43844,18
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	43841,98	1,53E-11	43841,98	43841,98
	ГА ФП	43842,1	0,094546	43841,98	43842,34
	ГА классич.	43842,44	0,234982	43842,1	43842,88
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	45113,56	0	45113,56	45113,56
	Детерм. ЖЭ, $\sigma_e=0.001$	45021,21	0	45021,21	45021,21
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	43842,4	0,450815	43841,98	43842,88
	ЖЭ адапт. $\sigma_e=0.001$	43843,92	0,536677	43842,75	43844,18
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	43842,21	0,290342	43841,98	43842,74
	ЖЭ, $\sigma_e=0.25, \beta=1$	43842,49	0,659644	43841,98	43843,78
	ЖЭ, $\sigma_e=0.25, \beta=3$	43843,32	0,345237	43842,75	43843,52
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	43842,12	0,218044	43841,98	43842,59
	ЖЭ, $\sigma_e=0.001, \beta=1$	43843,32	0,9767	43842,1	43844,18
	ЖЭ, $\sigma_e=0.001, \beta=3$	43844,18	0	43844,18	43844,18
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	43843,09	0,39873	43842,74	43843,52
	ЖЛ, $\sigma_e=0.25, \beta=1$	43842,58	0,531936	43841,98	43843,52
	ЖЛ, $\sigma_e=0.25, \beta=3$	43843,29	0,970042	43842,74	43845,4
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	43842,51	0,46305	43841,98	43842,94
	ЖЛ, $\sigma_e=0.001, \beta=1$	43842,92	0,583907	43842,34	43844,18
	ЖЛ, $\sigma_e=0.001, \beta=3$	44191,73	596,6553	43842,74	45118,74
АЛА мультистарт	43842,36	0,316511	43841,98	43842,74	

Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=60с.			
		F(X) среднее	Среднекв. отклон.	min.	max.
2Д522Б, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>2</sub> <sup>2</sup> , d=10, k-средних	ГАЗЭ+ЛП	7714,614	0,383694	7714,129	7715,503
	ГАЗЭ вещ., $\sigma_e=0.25$	7714,66	0,19542	7714,156	7714,773
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	7714,287	0,089993	7714,156	7714,406
	ГА ФП	7714,221	0,061262	7714,141	7714,298
	ГА классич.	7714,211	0,067842	7714,138	7714,303
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	7902,211	0	7902,211	7902,211
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	7714,608	0,248137	7714,239	7714,811
	ЖЭ адапт. $\sigma_e=0.001$	7717,91	5,318579	7714,781	7725,765
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	7714,645	0,252164	7714,219	7714,828
	ЖЭ, $\sigma_e=0.25, \beta=1$	7714,614	0,226605	7714,26	7714,796
	ЖЭ, $\sigma_e=0.25, \beta=3$	7714,293	0,085118	7714,209	7714,476
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	7716,233	3,951935	7714,343	7725,183
	ЖЭ, $\sigma_e=0.001, \beta=1$	7714,901	0,291889	7714,773	7715,562
	ЖЭ, $\sigma_e=0.001, \beta=3$	7714,732	0,084969	7714,547	7714,773
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	7716,284	5,069544	7714,26	7727,779
	ЖЛ, $\sigma_e=0.25, \beta=1$	7716,016	4,241316	7714,298	7725,632
	ЖЛ, $\sigma_e=0.25, \beta=3$	7716,49	4,89817	7714,287	7727,557
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	7714,364	0,131661	7714,144	7714,505
	ЖЛ, $\sigma_e=0.001, \beta=1$	7715,996	4,250555	7714,285	7725,632
	ЖЛ, $\sigma_e=0.001, \beta=3$	7722,03	7,241906	7714,493	7731,482
	ALA мультистарт	7714,265	0,098253	7714,141	7714,477
Н5503ХМ1-289, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>2</sub> <sup>2</sup> , d=229, k-средних	ГАЗЭ+ЛП	43739,69	20,31075	43702,28	43766,87
	ГАЗЭ вещ., $\sigma_e=0.25$	43687,01	4,596077	43678,79	43693,63
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	43680,82	3,302637	43675,79	43686,87
	ГА ФП	43716,26	8,402585	43708,14	43736,26
	ГА классич.	43715,8	6,166048	43703,31	43724,42
	ИВС, $\sigma_e=0.25$	-	0	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	43830,25	0	43830,25	43830,25
	Детерм. ЖЭ, $\sigma_e=0.001$	44573,13	0	44573,13	44573,13
	ИВС, $\sigma_e=0.001$	0	0	-	0
	ЖЭ адапт. $\sigma_e=0.25$	0	0	-	0
	ЖЭ адапт. $\sigma_e=0.001$	43691,02	3,087926	43684,45	43693,51
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	43699,21	6,032778	43692,04	43711,26
	ЖЭ, $\sigma_e=0.25, \beta=1$	43691,72	6,898894	43684,45	43703,25
	ЖЭ, $\sigma_e=0.25, \beta=3$	43688,81	6,230507	43680,28	43700,12
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	43704,39	7,696556	43694,11	43719,47
	ЖЭ, $\sigma_e=0.001, \beta=1$	43691,67	7,368125	43684,19	43703,13
	ЖЭ, $\sigma_e=0.001, \beta=3$	43686,3	2,600117	43683,36	43690,45
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	43717,25	11,08307	43705,63	43733,45
	ЖЛ, $\sigma_e=0.25, \beta=1$	43714,63	12,79796	43702,32	43734,84
	ЖЛ, $\sigma_e=0.25, \beta=3$	43720,1	17,90737	43692,5	43738,93
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	43720,43	10,20816	43707,93	43740,98
	ЖЛ, $\sigma_e=0.001, \beta=1$	43713,57	11,27041	43695,14	43727,59
	ЖЛ, $\sigma_e=0.001, \beta=3$	43723,19	20,38476	43703,31	43760,96
	ALA мультистарт	43735,46	18,04498	43701,35	43753,06



Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=60с.			
		F(X) среднее	Среднекв. отклон.	min.	max.
1526ТЛ1, 1 партия, N=535, p=10, l <sub>1</sub> , d=167, k-средних	ГАЗЭ+ЛП	30459,98	30,23103	30426,05	30513,34
	ГАЗЭ вещ., $\sigma_e=0.25$	30427,59	5,199556	30423,04	30438,74
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	30423,32	0,317576	30423,02	30423,89
	ГА ФП	30425,07	2,49062	30423,33	30430,18
	ГА классич.	30425,7	0,845359	30424,43	30426,64
	ИВС, $\sigma_e=0.25$	-	0	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	30761,52	0	30761,52	30761,52
	Детерм. ЖЭ, $\sigma_e=0.001$	30539,46	0	30539,46	30539,46
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	30426,63	2,469775	30423,86	30431,04
	ЖЭ адапт. $\sigma_e=0.001$	30427,86	2,537132	30425,7	30431,04
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	30424,94	2,161574	30422,88	30429,08
	ЖЭ, $\sigma_e=0.25, \beta=1$	30426,38	1,45165	30424,99	30429,37
	ЖЭ, $\sigma_e=0.25, \beta=3$	30438,62	9,387275	30423,89	30452,69
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	30424,88	1,508445	30422,88	30427,4
	ЖЭ, $\sigma_e=0.001, \beta=1$	30425,18	1,701892	30422,88	30427,4
	ЖЭ, $\sigma_e=0.001, \beta=3$	30461,27	5,462344	30452,62	30467,04
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	30425,61	1,650209	30423,59	30428,31
	ЖЛ, $\sigma_e=0.25, \beta=1$	30432,17	8,663448	30425,96	30449,77
	ЖЛ, $\sigma_e=0.25, \beta=3$	30446,15	15,80213	30424,28	30465,18
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	30431,21	7,601373	30423,45	30441,21
	ЖЛ, $\sigma_e=0.001, \beta=1$	30439,71	15,15822	30423,45	30462,81
	ЖЛ, $\sigma_e=0.001, \beta=3$	30489,96	12,48601	30465,59	30502,56
	АЛА мультистарт	30436,18	13,65297	30423,59	30457,99
1526ТЛ1, сборная партия (фактически - 3 партии), N=1234, p=10, l <sub>1</sub> , d=167, k-средних	ГАЗЭ+ЛП	41714,56	1,177396	41713,78	41717,02
	ГАЗЭ вещ., $\sigma_e=0.25$	41713,44	0,160787	41713,21	41713,62
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	41713,21	7,86E-12	41713,21	41713,21
	ГА ФП	41713,64	0,273881	41713,21	41714,08
	ГА классич.	41714,2	1,219102	41713,21	41716,89
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	42136,63	0	42136,63	42136,63
	Детерм. ЖЭ, $\sigma_e=0.001$	41714,08	0	41714,08	41714,08
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	41713,44	0,204253	41713,21	41713,78
	ЖЭ адапт. $\sigma_e=0.001$	41713,43	0,26076	41713,21	41713,78
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	41713,37	0,172237	41713,21	41713,62
	ЖЭ, $\sigma_e=0.25, \beta=1$	41713,44	0,228408	41713,21	41713,78
	ЖЭ, $\sigma_e=0.25, \beta=3$	41714,71	1,607466	41713,21	41717,03
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	41713,35	0,150488	41713,21	41713,51
	ЖЭ, $\sigma_e=0.001, \beta=1$	41713,47	0,222609	41713,21	41713,78
	ЖЭ, $\sigma_e=0.001, \beta=3$	41716,95	0,220612	41716,73	41717,23
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	41713,41	0,163338	41713,21	41713,62
	ЖЛ, $\sigma_e=0.25, \beta=1$	41713,76	0,176412	41713,51	41714,08
	ЖЛ, $\sigma_e=0.25, \beta=3$	41714,52	1,074095	41713,78	41716,72
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	41713,55	0,322869	41713,21	41714,08
	ЖЛ, $\sigma_e=0.001, \beta=1$	41713,97	0,485732	41713,51	41714,8
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-
	АЛА мультистарт	41713,58	0,101557	41713,51	41713,78

Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=60с.			
		F(X) среднее	Среднекв. отклон.	min.	max.
2Д522Б, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>1</sub> , d=10, k-средних	ГАЗЭ+ЛП	11777,4	4,648725	11773,94	11787,29
	ГАЗЭ вещ., $\sigma_e=0.25$	11775,98	1,699985	11773,38	11777,85
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	11774,78	1,495346	11773,35	11777,82
	ГА ФП	11773,03	0,14706	11772,83	11773,2
	ГА классич.	11773,05	0,154192	11772,86	11773,2
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	11794.91965	0	11794,91965	11794.91965
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-
	ИВС, $\sigma_e=0.001$				
	ЖЭ адапт. $\sigma_e=0.25$				
	ЖЭ адапт. $\sigma_e=0.001$				
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	11774,31	0,829105	11773,11	11775,48
	ЖЭ, $\sigma_e=0.25, \beta=1$	11774,94	2,097868	11773,2	11777,92
	ЖЭ, $\sigma_e=0.25, \beta=3$	11777,89	0,024153	11777,86	11777,91
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	11774,92	1,894164	11772,96	11777,91
	ЖЭ, $\sigma_e=0.001, \beta=1$	11777,38	0,965782	11775,26	11777,91
	ЖЭ, $\sigma_e=0.001, \beta=3$	11777,91	0,021108	11777,87	11777,95
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	11774	0,934401	11772,86	11775,08
	ЖЛ, $\sigma_e=0.25, \beta=1$	11775,67	2,45675	11772,9	11778,25
	ЖЛ, $\sigma_e=0.25, \beta=3$	11779,35	1,944551	11777,91	11782,54
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	11774,03	1,241051	11772,9	11776,3
	ЖЛ, $\sigma_e=0.001, \beta=1$	11777,72	1,957866	11773,99	11780,61
	ЖЛ, $\sigma_e=0.001, \beta=3$				
ALA мультистарт	11773,92	0,524973	11773,18	11774,44	
Н5503ХМ1-289, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>1</sub> , d=229, k-средних	ГАЗЭ+ЛП	42702,94	56,21193	42646,88	42797,03
	ГАЗЭ вещ., $\sigma_e=0.25$	42643,56	4,579624	42634,54	42648,58
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	42630,52	5,241642	42625,22	42638,59
	ГА ФП	42638,23	4,22367	42629,43	42642,28
	ГА классич.	42639,88	6,369143	42632,83	42651,31
	ИВС, $\sigma_e=0.25$	-	0	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	43148,55	0	43148,55	43148,55
	Детерм. ЖЭ, $\sigma_e=0.001$	43131,92	0	43131,92	43131,92
	ИВС, $\sigma_e=0.001$				
	ЖЭ адапт. $\sigma_e=0.25$	42638,12	5,672711	42631,57	42646,24
	ЖЭ адапт. $\sigma_e=0.001$	42636,75	7,926397	42626,11	42644,62
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	42634,21	6,829447	42627,86	42645,01
	ЖЭ, $\sigma_e=0.25, \beta=1$	42639,77	6,897585	42626,69	42646,39
	ЖЭ, $\sigma_e=0.25, \beta=3$	42645,96	4,227195	42639,99	42651,98
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	42633,72	4,828461	42625,72	42639,82
	ЖЭ, $\sigma_e=0.001, \beta=1$	42635,3	8,30144	42625,92	42645,39
	ЖЭ, $\sigma_e=0.001, \beta=3$	42650,84	6,229508	42643,81	42663,26
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	42636,22	6,214561	42624,4	42643,07
	ЖЛ, $\sigma_e=0.25, \beta=1$	42642,75	5,326172	42637,67	42653,39
	ЖЛ, $\sigma_e=0.25, \beta=3$	42672,35	24,84948	42627,95	42705,28
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	42639,7	8,019028	42628,19	42652,02
	ЖЛ, $\sigma_e=0.001, \beta=1$	42650,56	6,692517	42645,15	42662,54
	ЖЛ, $\sigma_e=0.001, \beta=3$	42673,57	23,35594	42645,3	42712,29
ALA мультистарт	42643,47	7,54821	42633,29	42654,58	

Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=60с.				
		F(X) среднее	Среднекв. отклон.	min.	max.	
1526ТЛ1, 1 партия, N=535, p=10, l <sub>2</sub> <sup>2</sup> , d=167, k-медоид	ГАЗЭ+ЛП	53578,78	10,81479	53570,61	53593,29	
	ГАЗЭ вещ., $\sigma_e=0.25$	53570,61	7,86E-12	53570,61	53570,61	
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	53570,61	7,86E-12	53570,61	53570,61	
	ГА ФП	53661,13	98,65555	53570,61	53766,37	
	ГА классич.	53625,54	55,00193	53570,61	53720,51	
	ИВС, $\sigma_e=0.25$	-	-	-	-	
	Детерм. ЖЭ, $\sigma_e=0.25$	54865,33	0	54865,33	54865,33	
	Детерм. ЖЭ, $\sigma_e=0.001$	55076,02	0	55076,02	55076,02	
	ИВС, $\sigma_e=0.001$					
	ЖЭ адапт. $\sigma_e=0.25$	53578,78	10,81479	53570,61	53593,29	
	ЖЭ адапт. $\sigma_e=0.001$	53585,9	9,178974	53570,61	53593,29	
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	53578,93	8,674131	53570,61	53593,29	
	ЖЭ, $\sigma_e=0.25, \beta=1$	53578,93	8,674131	53570,61	53593,29	
	ЖЭ, $\sigma_e=0.25, \beta=3$	53591,13	4,841108	53582,47	53593,29	
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	53588,63	23,36143	53570,61	53638,52	
	ЖЭ, $\sigma_e=0.001, \beta=1$	53577,24	9,019553	53570,61	53593,29	
	ЖЭ, $\sigma_e=0.001, \beta=3$	53588,96	5,929123	53582,47	53593,29	
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	53645,44	78,62517	53570,61	53766,37	
	ЖЛ, $\sigma_e=0.25, \beta=1$	53584,42	9,430773	53570,61	53593,29	
	ЖЛ, $\sigma_e=0.25, \beta=3$	53595,66	5,302403	53593,29	53605,15	
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	53702,78	64,17542	53593,29	53766,37	
	ЖЛ, $\sigma_e=0.001, \beta=1$	53655,72	83,90697	53593,29	53778,22	
	ЖЛ, $\sigma_e=0.001, \beta=3$	53646,11	105,022	53593,29	53833,68	
	ALA мультистарт	53595,35	19,27045	53570,61	53628,33	
	1526ТЛ1, сборная партия (фактически - 3 партии), N=1234, p=10, l <sub>2</sub> <sup>2</sup> , d=167, k-медоид	ГАЗЭ+ЛП	55364,1	6,220381	55361,75	55378,21
		ГАЗЭ вещ., $\sigma_e=0.25$	55361,75	7,86E-12	55361,75	55361,75
ГАЗЭ вещ.частичн., $\sigma_e=0.25$		55361,75	7,86E-12	55361,75	55361,75	
ГА ФП		55452,68	240,5632	55361,75	55998,22	
ГА классич.		55364,1	6,220381	55361,75	55378,21	
ИВС, $\sigma_e=0.25$						
Детерм. ЖЭ, $\sigma_e=0.25$		57131	0	57131	57131	
Детерм. ЖЭ, $\sigma_e=0.001$		55998,22	0	55998,22	55998,22	
ИВС, $\sigma_e=0.001$						
ЖЭ адапт. $\sigma_e=0.25$		55361,75	7,86E-12	55361,75	55361,75	
ЖЭ адапт. $\sigma_e=0.001$		55381,31	33,39536	55361,75	55430,19	
ЖЭ, $\sigma_e=0.25, \beta=0.5$		55361,75	7,86E-12	55361,75	55361,75	
ЖЭ, $\sigma_e=0.25, \beta=1$		55361,75	7,86E-12	55361,75	55361,75	
ЖЭ, $\sigma_e=0.25, \beta=3$		55371,53	25,86794	55361,75	55430,19	
ЖЭ, $\sigma_e=0.001, \beta=0.5$		55366,45	8,030477	55361,75	55378,21	
ЖЭ, $\sigma_e=0.001, \beta=1$		55361,75	7,86E-12	55361,75	55361,75	
ЖЭ, $\sigma_e=0.001, \beta=3$		55371,53	25,86794	55361,75	55430,19	
ЖЛ, $\sigma_e=0.25, \beta=0.5$		55604,47	294,1579	55361,75	55998,22	
ЖЛ, $\sigma_e=0.25, \beta=1$		55455,03	239,605	55361,75	55998,22	
ЖЛ, $\sigma_e=0.25, \beta=3$		55907,3	240,5632	55361,75	55998,22	
ЖЛ, $\sigma_e=0.001, \beta=0.5$		55548,22	241,5122	55361,75	55998,22	
ЖЛ, $\sigma_e=0.001, \beta=1$		55634,52	340,2077	55361,75	55998,22	
ЖЛ, $\sigma_e=0.001, \beta=3$		55907,3	240,5632	55361,75	55998,22	
ALA мультистарт		55364,1	6,220381	55361,75	55378,21	

Продолжение таблицы Б.1

Набор данных, параметры, тип задачи	Метод, его параметры	t=60с.			
		F(X) среднее	Среднекв. отклон.	min.	max.
2Д522Б, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>2</sub> <sup>2</sup> , d=10, k-медоид	ГАЗЭ+ЛП	8102,733	8,671943	8090,421	8114,912
	ГАЗЭ вещ., $\sigma_e=0.25$	8084,229	1,97671	8082,939	8087,725
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	8084,445	2,240546	8083,133	8087,725
	ГА ФП	8111,75	18,3986	8093,521	8138,151
	ГА классич.	8121,171	39,30574	8089,399	8196,989
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	8502,325	0	8502,325	8502,325
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-
	ИВС, $\sigma_e=0.001$	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	8655,48	307,0964	8174,467	8884,082
	ЖЭ адапт. $\sigma_e=0.001$	8506,858	360,3087	8115,938	8893,057
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	8366,626	226,4394	8171,803	8856,277
	ЖЭ, $\sigma_e=0.25, \beta=1$	8289,035	275,1345	8089,399	8878,553
	ЖЭ, $\sigma_e=0.25, \beta=3$	8297,277	266,3781	8128,292	8884,082
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	8491,114	357,8105	8133,989	8884,843
	ЖЭ, $\sigma_e=0.001, \beta=1$	8328,085	258,5847	8112,93	8879,729
	ЖЭ, $\sigma_e=0.001, \beta=3$	8438,484	323,0206	8172,142	8900,823
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	8867,787	9,383919	8851,685	8879,4
	ЖЛ, $\sigma_e=0.25, \beta=1$	8866,943	8,345135	8851,685	8874,693
	ЖЛ, $\sigma_e=0.25, \beta=3$	8865,437	8,239141	8854,192	8877,522
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	8863,903	8,853733	8851,75	8876,907
	ЖЛ, $\sigma_e=0.001, \beta=1$	8872,765	8,189288	8861,673	8884,754
	ЖЛ, $\sigma_e=0.001, \beta=3$	8911,931	4,170648	8908,024	8916,323
ALA мультистарт	8127,961	37,36276	8093,428	8181,986	
Н5503ХМ1-289, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>2</sub> <sup>2</sup> , d=229, k-медоид	ГАЗЭ+ЛП	57302,73	0	57302,73	57302,73
	ГАЗЭ вещ., $\sigma_e=0.25$	57302,73	0	57302,73	57302,73
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	57302,73	0	57302,73	57302,73
	ГА ФП	57364,62	62,87532	57327,81	57456,66
	ГА классич.	57417,45	75,23637	57302,73	57487,68
	ИВС, $\sigma_e=0.25$	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	61479,75	0	61479,75	61479,75
	Детерм. ЖЭ, $\sigma_e=0.001$	58622,54	0	58622,54	58622,54
	ИВС, $\sigma_e=0.001$				
	ЖЭ адапт. $\sigma_e=0.25$	57302,73	0	57302,73	57302,73
	ЖЭ адапт. $\sigma_e=0.001$	57302,73	0	57302,73	57302,73
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	57335,61	41,24227	57302,73	57405,27
	ЖЭ, $\sigma_e=0.25, \beta=1$	57302,73	0	57302,73	57302,73
	ЖЭ, $\sigma_e=0.25, \beta=3$	57302,73	0	57302,73	57302,73
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	57302,73	0	57302,73	57302,73
	ЖЭ, $\sigma_e=0.001, \beta=1$	57302,73	0	57302,73	57302,73
	ЖЭ, $\sigma_e=0.001, \beta=3$	57302,73	0	57302,73	57302,73
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	57367,38	101,2834	57302,73	57575,3
	ЖЛ, $\sigma_e=0.25, \beta=1$	57302,73	0	57302,73	57302,73
	ЖЛ, $\sigma_e=0.25, \beta=3$	57435,21	94,53926	57302,73	57533,7
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	57324,86	37,79812	57302,73	57380,19
	ЖЛ, $\sigma_e=0.001, \beta=1$	57328,3	57,36895	57302,73	57456,66
	ЖЛ, $\sigma_e=0.001, \beta=3$	57410,23	122,4256	57302,73	57611,16
ALA мультистарт	57445,45	89,70995	57302,73	57533,7	

Таблица Б.2. Сравнение работы алгоритмов для задачи k-средних и задачи k-медоид. Динамика изменения достигнутого значения целевой функции и его среднеквадратичного отклонения в зависимости от времени счета

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.		t=5с.		t=15с.		t=30с.		t=60с.		
		F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	
1526ТЛ1, 1 партия, N=535, p=10, l <sub>2</sub> <sup>2</sup> , d=167, k-средних	ГАЗЭ+ЛП	68488,16	1604,183	68488,16	1604,183	41684,19	416,9586	41482,05	369,9859	41132,3	216,5863	
	ГАЗЭ вещ., $\sigma_e=0.25$	41241,64	274,5124	41074,62	12,53702	41066,84	11,99397	41061,83	13,541	41053,11	4,998814	
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	41320,83	240,2591	41069,7	15,93404	41054,67	7,07508	41045,13	3,563143	41040,03	0,99638	
	ГА ФП	41411,09	206,4727	41084,39	29,67328	41061,68	17,97617	41053,5	8,360914	41046,22	4,862504	
	ГА классич.	41265,66	253,0916	41094,8	36,43024	41068,26	20,11588	41053,91	10,8688	41045,72	5,457131	
	IBC, $\sigma_e=0.25$	-	-	-	-	-	-	-	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	41705,75	0	41705,75	0	41705,75	0	41705,75	0	41705,75	0	0
	Детерм. ЖЭ, $\sigma_e=0.001$	41084,52	0	41084,52	0	41084,52	0	41084,52	0	41084,52	0	0
	IBC, $\sigma_e=0.001$	-	-	-	-	-	-	-	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	41142,28	164,8013	41074,3	15,21633	41066,49	7,693992	41061,44	7,245317	41054,23	1,953287	
	ЖЭ адапт. $\sigma_e=0.001$	41081,39	30,57316	41064,84	10,56466	41061,31	8,049168	41059,42	6,011516	41051,57	3,141884	
	ЖЭ, $\sigma_e=0.25$ , $\beta=0.5$	41108,67	30,83444	41069,38	19,4833	41063,77	12,77174	41060,34	11,51536	41053,22	7,767535	
	ЖЭ, $\sigma_e=0.25$ , $\beta=1$	41183,32	261,8287	41073,81	15,7471	41066,91	10,08632	41056,27	6,744525	41050,04	3,865743	
	ЖЭ, $\sigma_e=0.25$ , $\beta=3$	41095,83	29,15459	41069,1	11,99877	41066,61	10,88411	41061,14	5,809579	41054,11	4,208448	
	ЖЭ, $\sigma_e=0.001$ , $\beta=0.5$	41095,39	24,25912	41067,32	7,361063	41060,33	8,509522	41056,57	6,382639	41051,49	3,709106	
	ЖЭ, $\sigma_e=0.001$ , $\beta=1$	41080,23	32,08646	41062,23	11,45886	41056,69	5,45379	41056,34	5,399925	41053,46	5,969522	
	ЖЭ, $\sigma_e=0.001$ , $\beta=3$	41083,99	22,21444	41068,54	10,04252	41066,21	9,837841	41064,39	8,93707	41057,94	7,114785	
	ЖЛ, $\sigma_e=0.25$ , $\beta=0.5$	-	-	41337,52	374,6178	41296,57	373,3102	41271,93	341,1663	41161,57	266,7337	
	ЖЛ, $\sigma_e=0.25$ , $\beta=1$	-	-	41714,9	273,3539	41381,6	360,8979	41273,15	338,8815	41059,72	12,16037	
	ЖЛ, $\sigma_e=0.25$ , $\beta=3$	-	-	-	-	41494,85	396,6371	41165,62	271,4705	41056,05	9,623252	
	ЖЛ, $\sigma_e=0.001$ , $\beta=0.5$	-	-	41790,47	11,97312	41580,7	347,7389	41172,76	269,1565	41059,28	8,792072	
	ЖЛ, $\sigma_e=0.001$ , $\beta=1$	-	-	41715,02	278,6977	41585,46	350,4148	41127,9	151,2948	41118,14	155,3008	
	ЖЛ, $\sigma_e=0.001$ , $\beta=3$	-	-	-	-	-	-	41485,9	369,6709	41083,04	7,282893	
ALA мультисарт	41350,63	264,7742	41144,29	95,99996	41090,92	53,83281	41063,31	10,8855	41055,59	5,315283		

## Продолжение таблицы Б.2

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.		t=5с.		t=15с.		t=30с.		t=60с.		
		F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	
1526ТЛ1, сборная партия (фактически - 3 партии), N=1234, p=10, l <sub>2</sub> <sup>2</sup> , d=167, k-средних	ГАЗЭ+ЛП	76316,53	3355,313	76316,53	3355,313	58743,05	15974,95	44307,77	653,5897	43843,72	1,319931	
	ГАЗЭ вещ., $\sigma_e=0.25$	44026,86	313,0775	43843,54	0,813161	43843,33	0,876148	43842,99	0,852933	43842,6	0,676282	
	ГАЗЭ вещ.частичн., $\sigma_e=0.25$	44157,72	540,9987	43842,56	0,723467	43842,18	0,504141	43842,03	0,143369	43841,98	1,53E-11	
	ГА ФП	44376,35	484,5951	44011,34	389,312	43948,56	351,8414	43948,32	351,9233	43842,1	0,094546	
	ГА классич.	44243,8	472,6486	43843,56	0,895351	43843,31	0,951543	43842,89	0,892791	43842,44	0,234982	
	IBC, $\sigma_e=0.25$	-	-	-	-	-	-	-	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	45113,56	0	45113,56	0	45113,56	0	45113,56	0	45113,56	0	0
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-	45021,21	0	45021,21	0	45021,21	0	0
	IBC, $\sigma_e=0.001$	-	-	-	-	-	-	-	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	43844,01	1,515657	43843,73	1,312602	43843,32	0,917788	43842,58	0,710623	43842,4	0,450815	
	ЖЭ адапт. $\sigma_e=0.001$	44011,36	440,2209	43844,4	0,586012	43844,18	0	43843,92	0,536677	43843,92	0,536677	
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	44010,85	440,1839	43843,91	0,49502	43843,41	0,852229	43842,81	0,71908	43842,21	0,290342	
	ЖЭ, $\sigma_e=0.25, \beta=1$	43844,57	0,669047	43844,18	0	43843,64	0,829708	43843,12	1,017727	43842,49	0,659644	
	ЖЭ, $\sigma_e=0.25, \beta=3$	43844,34	0,995868	43843,83	0,836655	43843,52	0	43843,52	0	43843,32	0,345237	
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	43843,93	1,021064	43843,71	0,676477	43843,57	0,765762	43842,84	0,957183	43842,12	0,218044	
	ЖЭ, $\sigma_e=0.001, \beta=1$	43845,07	0,828746	43843,98	0,54327	43843,98	0,54327	43843,68	0,877163	43843,32	0,9767	
	ЖЭ, $\sigma_e=0.001, \beta=3$	43845,51	0,586012	43845,29	0,756538	43845,29	0,756538	43844,84	0,828746	43844,18	0	
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	-	-	44344,52	624,6575	44177	569,3111	44010,19	440,6952	43843,09	0,39873	
	ЖЛ, $\sigma_e=0.25, \beta=1$	-	-	44681,54	572,3603	44177,74	571,1261	44009,73	440,9407	43842,58	0,531936	
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-	-	-	44347,05	627,76	43843,29	0,970042	
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	-	-	44514,41	627,9872	44176,42	568,8243	43843,29	0,636109	43842,51	0,46305	
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-	44601,07	553,3409	44100	465,1867	43842,92	0,583907	
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-	-	-	-	-	44191,73	596,6553	
ALA мультистарт	44354,11	525,6823	43901,63	194,5553	43842,67	0,290075	43842,55	0,355262	43842,36	0,316511		

## Продолжение таблицы Б.2

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.		t=5с.		t=15с.		t=30с.		t=60с.		
		F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	
2Д522Б, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>2</sub> <sup>2</sup> , d=10, k-средних	ГАЗЭ+ЛП	11165,27	623,2074	10131,58	1618,673	7956,146	369,0633	7722,434	7,577611	7714,614	0,383694	
	ГАЗЭ вещ., $\sigma_e=0.25$	7719,683	7,299164	7714,765	0,149207	7714,732	0,112576	7714,72	0,104638	7714,66	0,19542	
	ГАЗЭ вещ. Частичн., $\sigma_e=0.25$	7721,796	6,800469	7714,726	0,215801	7714,561	0,234459	7714,419	0,194071	7714,287	0,089993	
	ГА ФП	7720,096	8,003918	7714,502	0,178475	7714,379	0,152647	7714,246	0,071818	7714,221	0,061262	
	ГА классич.	7721,921	5,393244	7716,618	4,521679	7714,522	0,342544	7714,374	0,120112	7714,211	0,067842	
	ИВС, $\sigma_e=0.25$	-	-	-	-	-	-	-	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	-	-	7902,211	0	7902,211	0	7902,211	0	7902,211	0	0
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-	-	-	-	-	-	-	-
	ИВС, $\sigma_e=0.001$	-	-	-	-	-	-	-	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	7762,998	46,86841	7721,784	6,491085	7721,178	6,007967	7718,061	5,611629	7714,608	0,248137	
	ЖЭ адапт. $\sigma_e=0.001$	7780,122	53,99543	7723,71	8,460405	7721,576	5,694643	7719,787	5,554729	7717,91	5,318579	
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	7756,036	53,87982	7729,757	19,78226	7721,909	6,662947	7718,306	5,765014	7714,645	0,252164	
	ЖЭ, $\sigma_e=0.25, \beta=1$	7757,436	55,55182	7719,889	6,365362	7718,066	5,572742	7714,799	0,023635	7714,614	0,226605	
	ЖЭ, $\sigma_e=0.25, \beta=3$	7719,256	5,512919	7714,796	0,168723	7714,739	0,121612	7714,399	0,20994	7714,293	0,085118	
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	7742,886	40,53269	7725,353	4,744094	7719,96	6,491701	7716,393	4,181275	7716,233	3,951935	
	ЖЭ, $\sigma_e=0.001, \beta=1$	7800,83	67,54767	7739,597	42,45355	7721,881	6,409195	7716,727	4,630514	7714,901	0,291889	
	ЖЭ, $\sigma_e=0.001, \beta=3$	7714,921	0,269205	7714,799	0,001252	7714,795	0,009477	7714,784	0,013137	7714,732	0,084969	
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	-	-	8255,306	364,9584	7932,099	366,4525	7734,569	46,83972	7716,284	5,069544	
	ЖЛ, $\sigma_e=0.25, \beta=1$	-	-	8158,159	411,0694	7935,872	364,1233	7718,753	7,360263	7716,016	4,241316	
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-	-	-	7724,865	21,60835	7716,49	4,89817	
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	-	-	7825,887	283,4822	7824,218	284,204	7716,421	5,040154	7714,364	0,131661	
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-	7827,088	282,9837	7827,086	282,9842	7715,996	4,250555	
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-	-	-	-	-	7722,03	7,241906	
ALA мультистарт	7722,482	7,548629	7716,632	4,32703	7714,405	0,160155	7714,31	0,097528	7714,265	0,098253		

## Продолжение таблицы Б.2

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.		t=5с.		t=15с.		t=30с.		t=60с.	
		F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$
Н5503ХМ1-289, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>2</sub> <sup>2</sup> , d=229, k-средних	ГАЗЭ+ЛП	72033,34	1442,642	72033,34	1442,642	46393,51	8660,852	43758,29	28,56965	43739,69	20,31075
	ГАЗЭ вещ., $\sigma_e=0.25$	43883,34	88,35572	43706,99	12,08807	43694,18	5,717544	43691,15	5,762717	43687,01	4,596077
	ГАЗЭ вещ. Частичн., $\sigma_e=0.25$	43867,58	99,76776	43712,14	18,51849	43691,44	4,225457	43685,71	3,562377	43680,82	3,302637
	ГА ФП	43838,2	84,94666	43754	28,86743	43731,84	16,07238	43722,74	12,58556	43716,26	8,402585
	ГА классич.	43877,05	112,9201	43784,47	56,4261	43744,84	12,53196	43728,64	10,18989	43715,8	6,166048
	ИВС, $\sigma_e=0.25$	-	0	-	0	-	0	-	0	-	0
	Детерм. ЖЭ, $\sigma_e=0.25$	43830,25	0	43830,25	0	43830,25	0	43830,25	0	43830,25	0
	Детерм. ЖЭ, $\sigma_e=0.001$	44573,13	0	44573,13	0	44573,13	0	44573,13	0	44573,13	0
	ИВС, $\sigma_e=0.001$	0	0	0	0	0	0	0	0	0	0
	ЖЭ адапт. $\sigma_e=0.25$	0	0	0	0	0	0	0	0	0	0
	ЖЭ адапт. $\sigma_e=0.001$	43733,05	27,72009	43715,36	12,12266	43705,72	9,256906	43698,54	5,004222	43691,02	3,087926
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	43733,95	18,03182	43721,26	16,70822	43710,76	12,59137	43710,02	11,8529	43699,21	6,032778
	ЖЭ, $\sigma_e=0.25, \beta=1$	43739,25	15,33142	43718,92	12,05306	43703,36	8,239231	43699,21	5,778718	43691,72	6,898894
	ЖЭ, $\sigma_e=0.25, \beta=3$	43722,85	24,93519	43709,43	21,23598	43703,4	13,76373	43693,46	13,06455	43688,81	6,230507
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	43738,84	11,434	43726,17	13,03848	43716,59	15,23396	43709,26	11,63041	43704,39	7,696556
	ЖЭ, $\sigma_e=0.001, \beta=1$	43728,01	9,256022	43707,14	17,85928	43697,28	11,96721	43693,66	7,68997	43691,67	7,368125
	ЖЭ, $\sigma_e=0.001, \beta=3$	43716,48	17,1005	43712,58	15,20587	43705,43	9,925234	43695,45	7,485388	43686,3	2,600117
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	-	-	43816,48	103,3727	43750,14	26,27248	43729,8	15,16301	43717,25	11,08307
	ЖЛ, $\sigma_e=0.25, \beta=1$	-	-	43770,91	58,70159	43754,71	49,47832	43724,98	18,70859	43714,63	12,79796
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-	43774,7	44,11336	43740,7	29,84639	43720,1	17,90737
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	-	-	43747,86	30,17527	43732,75	15,62881	43727,88	15,20107	43720,43	10,20816
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-	43740,79	24,58402	43723,49	13,12896	43713,57	11,27041
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-	-	-	-	-	43723,19	20,38476
	ALA мультистарт	43842,88	105,038	43761,34	25,6081	43758,97	24,05854	43753,94	20,40622	43735,46	18,04498



## Продолжение таблицы Б.2

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.		t=5с.		t=15с.		t=30с.		t=60с.		
		F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	
1526ТЛ1, 1 партия, N=535, p=10, l <sub>1</sub> , d=167, k-средних	ГАЗЭ+ЛП	39426,1	461,9336	39426,1	461,9336	39426,1	461,9336	30567,93	263,6622	30459,98	30,23103	
	ГАЗЭ вещ., $\sigma_e=0.25$	30547,4	64,25617	30490,1	30,35079	30439,18	9,487806	30435,61	9,155767	30427,59	5,199556	
	ГАЗЭ вещ. Частичн., $\sigma_e=0.25$	30553,8	73,59293	30473,14	38,45799	30425,5	2,7013	30424,07	0,744733	30423,32	0,317576	
	ГА ФП	30538,21	49,19797	30471,13	29,0227	30449,84	22,63496	30435,95	18,09385	30425,07	2,49062	
	ГА классич.	30544,85	92,25796	30495,21	47,40195	30456,41	33,20892	30432,78	8,3324	30425,7	0,845359	
	ИВС, $\sigma_e=0.25$	-	-	-	0	-	0	-	0	-	0	
	Детерм. ЖЭ, $\sigma_e=0.25$	30761,52	0	30761,52	0	30761,52	0	30761,52	0	30761,52	0	
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	30539,46	0	30539,46	0	30539,46	0	30539,46	0	
	ИВС, $\sigma_e=0.001$	-	-	-	-	-	-	-	-	-	-	
	ЖЭ адапт. $\sigma_e=0.25$	30464,17	16,15407	30441,48	16,73587	30437,19	11,89176	30430,03	6,687699	30426,63	2,469775	
	ЖЭ адапт. $\sigma_e=0.001$	30460,48	17,58954	30448,82	13,06918	30443,2	10,1058	30435,9	3,470962	30427,86	2,537132	
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	30458,45	36,12319	30435,9	10,60432	30428,86	2,56671	30427,51	1,563706	30424,94	2,161574	
	ЖЭ, $\sigma_e=0.25, \beta=1$	30459,17	17,40135	30444,98	14,64654	30438,99	12,92284	30428,11	4,111265	30426,38	1,45165	
	ЖЭ, $\sigma_e=0.25, \beta=3$	30481,32	23,79719	30463,76	16,96401	30460,68	13,49443	30451,91	14,68099	30438,62	9,387275	
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	30446,15	19,14042	30436,38	11,38039	30431,92	10,91483	30428,05	3,927215	30424,88	1,508445	
	ЖЭ, $\sigma_e=0.001, \beta=1$	30459,14	11,96865	30444,78	5,601953	30430,34	5,385294	30428,18	5,074603	30425,18	1,701892	
	ЖЭ, $\sigma_e=0.001, \beta=3$	30488	18,55914	30474,49	16,29802	30467,6	5,132229	30464,13	3,241274	30461,27	5,462344	
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	-	-	30477,03	40,98803	30451,16	24,85679	30434,77	12,11978	30425,61	1,650209	
	ЖЛ, $\sigma_e=0.25, \beta=1$	-	-	-	-	30466,59	27,74429	30449,19	25,3325	30432,17	8,663448	
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-	-	-	30467,47	21,51794	30446,15	15,80213	
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	-	-	30533,74	107,4726	30454,48	25,0127	30438,62	17,3223	30431,21	7,601373	
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-	30485,06	25,32765	30468,63	16,01791	30439,71	15,15822	
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-	-	-	-	-	30489,96	12,48601	
	ALA мультистарт											
			30530,34	71,78038	30485,43	27,57333	30474,88	26,87421	30465,11	24,50061	30436,18	13,65297

## Продолжение таблицы Б.2

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.		t=5с.		t=15с.		t=30с.		t=60с.	
		F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$
1526ТЛ1, сборная партия (фактически - 3 партии), N=1234, p=10, l <sub>1</sub> , d=167, k-средних	ГАЗЭ+ЛП	56705,64	1114,738	56705,64	1114,738	56705,64	1114,738	52698,18	7556,402	41714,56	1,177396
	ГАЗЭ вещ., $\sigma_e=0.25$	41823,32	160,0799	41755,52	66,84594	41714,27	1,17955	41713,67	0,221179	41713,44	0,160787
	ГАЗЭ вещ. Частичн., $\sigma_e=0.25$	41777,77	102,9083	41714,78	1,602802	41714,29	1,258834	41713,48	0,206343	41713,21	7,86E-12
	ГА ФП	41754,18	60,07712	41750,4	60,42679	41714,43	1,26466	41714	0,678241	41713,64	0,273881
	ГА классич.	41809,23	95,95293	41755,64	69,26373	41733,79	50,3852	41733,69	50,42924	41714,2	1,219102
	IBC, $\sigma_e=0.25$	-	-	-	-	-	-	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	-	-	42136,63	0	42136,63	0	42136,63	0	42136,63	0
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-	-	-	41714,08	0	41714,08	0
	IBC, $\sigma_e=0.001$	-	-	-	-	-	-	-	-	-	-
	ЖЭ адапт. $\sigma_e=0.25$	41749,23	58,8598	41714,91	1,471741	41714,29	1,094582	41713,67	0,221179	41713,44	0,204253
	ЖЭ адапт. $\sigma_e=0.001$	41779,22	112,263	41714,67	1,597583	41714,12	1,25648	41713,64	0,304708	41713,43	0,26076
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	41716,67	2,395126	41714,4	1,136804	41713,96	0,445546	41713,61	0,090248	41713,37	0,172237
	ЖЭ, $\sigma_e=0.25, \beta=1$	41738,52	60,65122	41714,14	1,169793	41714,1	1,177763	41713,61	0,315126	41713,44	0,228408
	ЖЭ, $\sigma_e=0.25, \beta=3$	41779,46	78,70181	41734,04	44,54828	41717,07	0,149181	41716,05	1,66966	41714,71	1,607466
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	41716,47	4,289055	41714,25	1,244154	41713,74	0,247726	41713,56	0,238458	41713,35	0,150488
	ЖЭ, $\sigma_e=0.001, \beta=1$	41738,27	58,65238	41713,8	0,3202	41713,76	0,339144	41713,68	0,318839	41713,47	0,222609
	ЖЭ, $\sigma_e=0.001, \beta=3$	41853,45	60,07198	41785,23	84,8869	41739,76	59,94053	41717,06	0,175529	41716,95	0,220612
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	-	-	41969,76	672,0724	41714,85	1,510622	41713,76	0,460264	41713,41	0,163338
	ЖЛ, $\sigma_e=0.25, \beta=1$	-	-	-	-	41714,99	1,492229	41714,09	0,492839	41713,76	0,176412
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-	-	-	-	-	41714,52	1,074095
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	-	-	42270,75	952,0199	41714,75	1,355851	41713,86	0,214737	41713,55	0,322869
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-	-	-	41773,71	155,3782	41713,97	0,485732
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-	-	-	-	-	-	-
ALA мультистарт											
		41815,03	68,374	41756,61	72,2261	41714,81	1,880815	41713,74	0,247726	41713,58	0,101557

## Продолжение таблицы Б.2

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.		t=5с.		t=15с.		t=30с.		t=60с.		
		F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	
2Д522Б, сборная партия (фактически - 5 партий), N=3711, p=10, l, d=10, k-средних	ГАЗЭ+ЛП	14006,32	224,8713	14006,32	224,8713	12422,33	1097,55	11782,36	9,058855	11777,4	4,648725	
	ГАЗЭ вещ., $\sigma_e=0.25$	11781,02	6,700716	11776,33	2,287349	11775,99	1,711883	11775,99	1,711883	11775,98	1,699985	
	ГАЗЭ вещ. Частичн., $\sigma_e=0.25$	11785,68	16,93447	11776,68	2,387856	11776	1,790302	11775,73	1,954829	11774,78	1,495346	
	ГА ФП	11779,97	5,309673	11776,06	2,649258	11775,16	2,325585	11773,29	0,509125	11773,03	0,14706	
	ГА классич.	11785,33	8,602068	11778,15	5,760881	11775,69	2,243109	11774,36	1,546701	11773,05	0,154192	
	ИВС, $\sigma_e=0.25$	-	-	-	-	-	-	-	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	-	-	-	-	11794.9196533	0	11794.9196533	0	11794.9196533	0	0
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-	-	-	-	-	-	-	-
	ИВС, $\sigma_e=0.001$											
	ЖЭ адапт. $\sigma_e=0.25$											
	ЖЭ адапт. $\sigma_e=0.001$											
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	11780,27	3,22218	11777,37	1,916893	11776,16	1,79084	11775,31	1,337668	11774,31	0,829105	
	ЖЭ, $\sigma_e=0.25, \beta=1$	11780,23	1,792554	11778,92	0,675255	11777,55	1,112884	11777,49	1,071773	11774,94	2,097868	
	ЖЭ, $\sigma_e=0.25, \beta=3$	11779,34	1,255175	11778,14	0,179245	11777,98	0,136718	11777,92	0,029182	11777,89	0,024153	
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	11779,63	1,216849	11777,98	1,225369	11776,33	2,152755	11775,52	1,751415	11774,92	1,894164	
	ЖЭ, $\sigma_e=0.001, \beta=1$	11780,46	2,921288	11778,59	1,133542	11778,2	0,743284	11777,73	1,208519	11777,38	0,965782	
	ЖЭ, $\sigma_e=0.001, \beta=3$	11781,55	2,422804	11779,32	1,361796	11778,16	0,154197	11777,96	0,080861	11777,91	0,021108	
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	-	-	11778,94	2,407646	11776,52	2,144031	11775,38	1,691617	11774	0,934401	
	ЖЛ, $\sigma_e=0.25, \beta=1$	-	-	-	-	11778,82	4,957253	11777,45	2,374853	11775,67	2,45675	
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-	-	-	11809,01	53,3231	11779,35	1,944551	
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	-	-	11782,62	4,280145	11778,2	3,032819	11775,32	1,774088	11774,03	1,241051	
ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-	11786,3	17,46161	11778,97	3,462207	11777,72	1,957866		
ЖЛ, $\sigma_e=0.001, \beta=3$												
ALA мультистарт												
		11785,25	12,53276	11777,14	2,982665	11775,5	1,559571	11774,05	0,576843	11773,92	0,524973	

## Продолжение таблицы Б.2

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.		t=5с.		t=15с.		t=30с.		t=60с.	
		F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$
Н5503ХМ1-289, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>1</sub> , d=229, k-средних	ГАЗЭ+ЛП	56433,19	310,7503	56433,19	310,7503	56433,19	310,7503	42757,78	98,55463	42702,94	56,21193
	ГАЗЭ вещ., $\sigma_e=0.25$	42820,46	62,67032	42690,69	36,28064	42661,69	16,92555	42648,35	3,394538	42643,56	4,579624
	ГАЗЭ вещ. Частичн., $\sigma_e=0.25$	42723,53	57,6827	42678,74	35,6881	42650,48	5,292793	42639,36	5,225814	42630,52	5,241642
	ГА ФП	42750,4	73,52909	42690,78	34,29012	42655,73	20,93271	42646,12	10,75228	42638,23	4,22367
	ГА классич.	42741,02	71,73337	42697,84	40,31416	42677,76	29,95391	42650,77	18,02367	42639,88	6,369143
	IBC, $\sigma_e=0.25$	-	-	-	-	-	-	-	0	-	0
	Детерм. ЖЭ, $\sigma_e=0.25$	43148,55	0	43148,55	0	43148,55	0	43148,55	0	43148,55	0
	Детерм. ЖЭ, $\sigma_e=0.001$	43131,92	0	43131,92	0	43131,92	0	43131,92	0	43131,92	0
	IBC, $\sigma_e=0.001$										
	ЖЭ адапт. $\sigma_e=0.25$	42672,65	18,20016	42657,49	13,41301	42648,29	11,40647	42642,48	5,883006	42638,12	5,672711
	ЖЭ адапт. $\sigma_e=0.001$	42659,75	14,14282	42654,68	9,097027	42651,81	7,328311	42642,05	9,543589	42636,75	7,926397
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	42671,75	20,66601	42650,13	8,574639	42646,45	6,669056	42638,33	4,817996	42634,21	6,829447
	ЖЭ, $\sigma_e=0.25, \beta=1$	42679,48	31,42728	42658,15	8,797811	42652,5	7,02551	42646,71	3,704245	42639,77	6,897585
	ЖЭ, $\sigma_e=0.25, \beta=3$	42722,99	55,109	42664,97	15,78238	42657,59	10,77978	42650,87	5,228868	42645,96	4,227195
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	42657,4	15,83775	42644,71	13,82459	42640,46	11,01157	42638,29	8,48082	42633,72	4,828461
	ЖЭ, $\sigma_e=0.001, \beta=1$	42663,74	12,87346	42648,2	12,77229	42638,27	8,611374	42636,12	9,638739	42635,3	8,30144
	ЖЭ, $\sigma_e=0.001, \beta=3$	42692,51	13,17937	42667,31	24,90645	42658,58	11,73806	42652,78	6,981912	42650,84	6,229508
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	-	-	-	-	42697,49	51,95109	42653,51	19,13367	42636,22	6,214561
	ЖЛ, $\sigma_e=0.25, \beta=1$	-	-	-	-	42694,55	37,76707	42664,11	15,43933	42642,75	5,326172
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-	-	-	42683,65	23,31677	42672,35	24,84948
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	-	-	42734,18	104,792	42671,92	52,52753	42648,08	9,423085	42639,7	8,019028
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-	-	-	42662,64	22,02625	42650,56	6,692517
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-	-	-	-	-	42673,57	23,35594
ALA мультистарт											
	42820,89	91,68919	42699,97	52,59251	42672,65	33,02221	42658,56	15,82734	42643,47	7,54821	

## Продолжение таблицы Б.2

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.		t=5с.		t=15с.		t=30с.		t=60с.		
		F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	
1526ТЛ1, 1 партия, N=535, p=10, $l_2^2$ , d=167, k-медоид	ГАЗЭ+ЛП	54969,26	1114,044	53885,23	218,2658	53774,54	201,3956	53629,72	53,24109	53578,78	10,81479	
	ГАЗЭ вещ., $\sigma_e=0.25$	53615,28	36,36333	53572,3	4,481348	53570,61	7,86E-12	53570,61	7,86E-12	53570,61	7,86E-12	
	ГАЗЭ вещ. Частичн., $\sigma_e=0.25$	53619,84	72,19232	53570,61	7,86E-12	53570,61	7,86E-12	53570,61	7,86E-12	53570,61	7,86E-12	
	ГА ФП	54128,54	295,9456	53738,26	166,2791	53672,84	99,0096	53671,15	100,4516	53661,13	98,65555	
	ГА классич.	54420,59	356,5645	54113,45	246,7429	53801,94	153,6583	53663,1	53,51714	53625,54	55,00193	
	IBC, $\sigma_e=0.25$	-	-	-	-	-	-	-	-	-	-	
	Детерм. ЖЭ, $\sigma_e=0.25$	54865,33	0	54865,33	0	54865,33	0	54865,33	0	54865,33	0	
	Детерм. ЖЭ, $\sigma_e=0.001$	55076,02	0	55076,02	0	55076,02	0	55076,02	0	55076,02	0	
	IBC, $\sigma_e=0.001$											
	ЖЭ адапт. $\sigma_e=0.25$	53714,68	89,37873	53618,87	56,54767	53590,2	14,39388	53585,26	10,76316	53578,78	10,81479	
	ЖЭ адапт. $\sigma_e=0.001$	53648,14	52,38842	53622,67	41,62851	53602,81	18,12927	53587,71	9,428892	53585,9	9,178974	
	ЖЭ, $\sigma_e=0.25$ , $\beta=0.5$	53839,63	103,2514	53677,73	75,06722	53644,1	60,02887	53621,15	50,73896	53578,93	8,674131	
	ЖЭ, $\sigma_e=0.25$ , $\beta=1$	53714	92,57771	53628,33	62,33571	53595,04	25,83398	53583,72	13,90193	53578,93	8,674131	
	ЖЭ, $\sigma_e=0.25$ , $\beta=3$	53661,56	85,75532	53600,4	6,49409	53595,87	9,554007	53591,13	4,841108	53591,13	4,841108	
	ЖЭ, $\sigma_e=0.001$ , $\beta=0.5$	53783,28	91,50668	53745,51	112,9452	53696,14	99,43524	53624,61	41,51884	53588,63	23,36143	
	ЖЭ, $\sigma_e=0.001$ , $\beta=1$	53715,08	58,20574	53628,71	60,20785	53590,2	10,84168	53582,02	11,34247	53577,24	9,019553	
	ЖЭ, $\sigma_e=0.001$ , $\beta=3$	53732,91	81,2352	53612,69	37,10085	53595,66	5,302403	53591,13	4,841108	53588,96	5,929123	
	ЖЛ, $\sigma_e=0.25$ , $\beta=0.5$	54151,74	315,3198	53857,2	146,4575	53765,27	49,36295	53754,18	49,30623	53645,44	78,62517	
	ЖЛ, $\sigma_e=0.25$ , $\beta=1$	54047,26	81,49744	53812,66	168,3665	53709,16	138,9541	53625,28	65,34173	53584,42	9,430773	
	ЖЛ, $\sigma_e=0.25$ , $\beta=3$	-	-	53983,76	232,603	53801,15	150,1069	53728,8	122,1594	53595,66	5,302403	
	ЖЛ, $\sigma_e=0.001$ , $\beta=0.5$	54041,07	229,4315	53872,89	156,0863	53840,4	133,5393	53776,36	137,5815	53702,78	64,17542	
	ЖЛ, $\sigma_e=0.001$ , $\beta=1$	54087,81	200,3797	53859,93	156,6871	53728,42	46,75348	53714,5	37,70021	53655,72	83,90697	
	ЖЛ, $\sigma_e=0.001$ , $\beta=3$	-	-	-	-	53820,45	166,7971	53732,46	185,2586	53646,11	105,022	
	ALA мультистарт											
			54373,68	508,1843	54118,58	412,19	53907,5	228,145	53769,22	189,6081	53595,35	19,27045

## Продолжение таблицы Б.2

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.		t=5с.		t=15с.		t=30с.		t=60с.	
		F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$
1526ТЛ1, сборная партия (фактически - 3 партии), N=1234, p=10, l <sub>2</sub> <sup>2</sup> , d=167, к-медоид	ГАЗЭ+ЛП	76527,12	1685,474	55972,87	276,81	55814,76	333,9305	55455,03	239,605	55364,1	6,220381
	ГАЗЭ вещ., $\sigma_e=0.25$	56408,83	894,6523	55361,75	7,86E-12	55361,75	7,86E-12	55361,75	7,86E-12	55361,75	7,86E-12
	ГАЗЭ вещ. Частичн., $\sigma_e=0.25$	55946,77	504,985	55361,75	7,86E-12	55361,75	7,86E-12	55361,75	7,86E-12	55361,75	7,86E-12
	ГА ФП	56118,31	857,9773	55486,11	285,3518	55455,03	239,605	55452,68	240,5632	55452,68	240,5632
	ГА классич.	56224,74	591,7999	56022,18	499,1721	55574,27	271,3057	55422,39	125,5121	55364,1	6,220381
	IBC, $\sigma_e=0.25$										
	Детерм. ЖЭ, $\sigma_e=0.25$	57131	0	57131	0	57131	0	57131	0	57131	0
	Детерм. ЖЭ, $\sigma_e=0.001$	-	-	-	-	-	-	55998,22	0	55998,22	0
	IBC, $\sigma_e=0.001$										
	ЖЭ адапт. $\sigma_e=0.25$	56000,9	7,077099	55634,52	340,2077	55452,68	240,5632	55361,75	7,86E-12	55361,75	7,86E-12
	ЖЭ адапт. $\sigma_e=0.001$	55907,3	240,5632	55553,38	304,9125	55553,38	304,9125	55553,38	304,9125	55381,31	33,39536
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	55735,23	328,8059	55520,97	242,7013	55474,58	232,9591	55398,77	97,94235	55361,75	7,86E-12
	ЖЭ, $\sigma_e=0.25, \beta=1$	55907,3	240,5632	55681,68	308,7475	55408,9	97,95398	55361,75	7,86E-12	55361,75	7,86E-12
	ЖЭ, $\sigma_e=0.25, \beta=3$	55816,37	310,5658	55725,45	340,2077	55482,01	230,1871	55391,08	36,58279	55371,53	25,86794
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	56014,43	22,35868	55558,08	301,5521	55474,58	232,9591	55386,09	42,674	55366,45	8,030477
	ЖЭ, $\sigma_e=0.001, \beta=1$	55695,29	304,7995	55545,95	309,0181	55364,1	6,220381	55361,75	7,86E-12	55361,75	7,86E-12
	ЖЭ, $\sigma_e=0.001, \beta=3$	55998,22	0	55998,22	0	55644,3	331,9451	55381,31	33,39536	55371,53	25,86794
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	57572,96	1680,355	56048,46	40,74849	55915	236,8664	55909,65	234,3428	55604,47	294,1579
	ЖЛ, $\sigma_e=0.25, \beta=1$	56503,21	1062,831	56015,02	44,43257	55998,22	0	55998,22	0	55455,03	239,605
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	57494,62	1797,133	56602,87	1444,802	55931,68	254,5158	55907,3	240,5632
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	56228,68	760,2916	55926,26	251,1209	55826,15	294,5315	55772,24	292,3121	55548,22	241,5122
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	56014,11	27,54303	56014,11	27,54303	55998,22	0	55634,52	340,2077
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-	56004,99	17,90477	55998,22	0	55907,3	240,5632
ALA мультистарт											
	56036,16	441,9047	55481,18	183,4018	55417,44	125,8347	55364,1	6,220381	55364,1	6,220381	

## Продолжение таблицы Б.2

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.		t=5с.		t=15с.		t=30с.		t=60с.	
		F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$
2Д522Б, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>2</sub> <sup>2</sup> , d=10, k-медоид	ГАЗЭ+ЛП	10731,22	375,0125	10731,22	375,0125	10440,75	767,7918	8596,216	454,026	8102,733	8,671943
	ГАЗЭ вещ., $\sigma_e=0.25$	8256,999	102,2087	8188,861	91,45949	8167,633	82,3343	8091,133	8,459622	8084,229	1,97671
	ГАЗЭ вещ. Частичн., $\sigma_e=0.25$	8376,217	150,1142	8208,961	74,07244	8189,273	55,18503	8096,164	6,921479	8084,445	2,240546
	ГА ФП	8371,359	168,7122	8225,626	57,33345	8191,589	70,2997	8127,447	29,86082	8111,75	18,3986
	ГА классич.	8348,236	106,8432	8227,147	87,80622	8190,107	64,89582	8175,579	63,0802	8121,171	39,30574
	IBC, $\sigma_e=0.25$										
	Детерм. ЖЭ, $\sigma_e=0.25$	-	0	8502,325	0	8502,325	0	8502,325	0	8502,325	0
	Детерм. ЖЭ, $\sigma_e=0.001$	0	0	-	-	-	-	-	-	-	-
	IBC, $\sigma_e=0.001$										
	ЖЭ адапт. $\sigma_e=0.25$	8951,09	59,43573	8910,165	14,32979	8810,581	223,8693	8807,893	222,6802	8655,48	307,0964
	ЖЭ адапт. $\sigma_e=0.001$	8958,052	57,63062	8914,291	26,92739	8820,961	233,7664	8804,395	226,0986	8506,858	360,3087
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	8909,335	40,67501	8829,419	150,3201	8803,182	212,5272	8487,655	266,3856	8366,626	226,4394
	ЖЭ, $\sigma_e=0.25, \beta=1$	8703,506	415,8929	8592,428	398,1549	8581,264	387,9482	8373,47	352,2425	8289,035	275,1345
	ЖЭ, $\sigma_e=0.25, \beta=3$	8919,944	32,91986	8904,779	19,59236	8709,716	310,8033	8684,15	348,8279	8297,277	266,3781
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	8932,346	39,87828	8897,701	20,2436	8859,684	61,46769	8822,751	128,9407	8491,114	357,8105
	ЖЭ, $\sigma_e=0.001, \beta=1$	8922,497	16,10003	8824,161	227,1098	8804,552	218,9172	8532,493	332,2132	8328,085	258,5847
	ЖЭ, $\sigma_e=0.001, \beta=3$	8947,776	34,73965	8922,147	15,01756	8803,738	269,746	8707,658	335,3176	8438,484	323,0206
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	-	-	9018,434	118,2808	8922,186	43,29741	8895,079	28,75195	8867,787	9,383919
	ЖЛ, $\sigma_e=0.25, \beta=1$	-	-	-	-	8907,388	39,65478	8874,924	18,21427	8866,943	8,345135
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	-	-	-	-	8894,767	39,63402	8865,437	8,239141
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	-	-	9072,871	216,096	8907,813	75,25842	8901,695	62,98011	8863,903	8,853733
	ЖЛ, $\sigma_e=0.001, \beta=1$	-	-	-	-	8956,733	76,08909	8894,266	22,4853	8872,765	8,189288
ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-	-	-	-	-	8911,931	4,170648	
ALA мультистарт											
		8280,247	145,7731	8212,829	61,75955	8163,18	61,34988	8155,124	52,29932	8127,961	37,36276

## Продолжение таблицы Б.2

Набор данных, параметры, тип задачи	Метод, его параметры	t=2с.		t=5с.		t=15с.		t=30с.		t=60с.	
		F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$	F(X) среднее	$\sigma$
Н5503ХМ1-289, сборная партия (фактически - 5 партий), N=3711, p=10, l <sub>2</sub> <sup>2</sup> , d=229, k-медоид	ГАЗЭ+ЛП	57926,07	733,5031	57534,53	209,4023	57423,46	191,8602	57333,26	80,78986	57302,73	0
	ГАЗЭ вещ., $\sigma_e=0.25$	57334,92	85,1819	57302,73	0	57302,73	0	57302,73	0	57302,73	0
	ГАЗЭ вещ. Частичн., $\sigma_e=0.25$	57354,39	71,98364	57302,73	0	57302,73	0	57302,73	0	57302,73	0
	ГА ФП	57588,7	149,7942	57471,72	89,8293	57364,62	62,87532	57364,62	62,87532	57364,62	62,87532
	ГА классич.	58183,5	174,3709	57873,18	321,3957	57584,58	111,3013	57460,85	68,72579	57417,45	75,23637
	IBC, $\sigma_e=0.25$	-	-	-	-	-	-	-	-	-	-
	Детерм. ЖЭ, $\sigma_e=0.25$	61479,75	0	61479,75	0	61479,75	0	61479,75	0	61479,75	0
	Детерм. ЖЭ, $\sigma_e=0.001$	58622,54	0	58622,54	0	58622,54	0	58622,54	0	58622,54	0
	IBC, $\sigma_e=0.001$										
	ЖЭ адапт. $\sigma_e=0.25$	57635,82	176,726	57365,97	78,31532	57339,88	60,25596	57309,89	12,23773	57302,73	0
	ЖЭ адапт. $\sigma_e=0.001$	57491,82	143,7374	57382,63	84,09124	57328,44	36,51752	57306,31	9,479308	57302,73	0
	ЖЭ, $\sigma_e=0.25, \beta=0.5$	57593,26	252,2448	57395,07	53,75815	57373,08	55,80955	57354,02	61,13789	57335,61	41,24227
	ЖЭ, $\sigma_e=0.25, \beta=1$	57487,54	146,1221	57331,88	56,2784	57302,73	0	57302,73	0	57302,73	0
	ЖЭ, $\sigma_e=0.25, \beta=3$	57570,41	124,4886	57402,89	116,7363	57335,78	60,61827	57302,73	0	57302,73	0
	ЖЭ, $\sigma_e=0.001, \beta=0.5$	57617,35	114,3752	57493,81	96,54849	57388,22	88,38207	57313,79	29,27829	57302,73	0
	ЖЭ, $\sigma_e=0.001, \beta=1$	57480,44	106,1851	57357,92	58,27641	57324,86	37,79812	57302,73	0	57302,73	0
	ЖЭ, $\sigma_e=0.001, \beta=3$	57518,45	123,0822	57421,79	70,25814	57362,01	61,54943	57302,73	0	57302,73	0
	ЖЛ, $\sigma_e=0.25, \beta=0.5$	57911,11	367,2494	57725,61	244,0485	57559,59	181,311	57461,18	134,3793	57367,38	101,2834
	ЖЛ, $\sigma_e=0.25, \beta=1$	57761,95	205,8463	57588,15	71,60977	57436,47	136,5429	57376,8	93,96487	57302,73	0
	ЖЛ, $\sigma_e=0.25, \beta=3$	-	-	58072,81	646,8255	57640,46	268,6301	57564,75	176,6332	57435,21	94,53926
	ЖЛ, $\sigma_e=0.001, \beta=0.5$	57845,37	284,6465	57667,12	291,224	57532	87,83554	57406,15	121,4107	57324,86	37,79812
	ЖЛ, $\sigma_e=0.001, \beta=1$	57914,11	444,3286	57641,11	306,9669	57439,54	89,0829	57350,29	73,2403	57328,3	57,36895
	ЖЛ, $\sigma_e=0.001, \beta=3$	-	-	-	-	57806,14	362,7907	57714,88	305,3301	57410,23	122,4256
ALA мультистарт	57901,33	286,103	57735,16	190,3727	57656,71	64,00913	57527,66	142,3037	57445,45	89,70995	



## ПРИЛОЖЕНИЕ В. ОПИСАНИЕ ПРОГРАММЫ КЛАССИФИКАЦИИ ЭЛЕМЕНТНОЙ БАЗЫ КОСМИЧЕСКОГО ПРИМЕНЕНИЯ

Программа представляет собой набор скриптов на языке Python 2. Главным (запускаемым) скриптом является файл `clust_mds.py`. Также в состав системы входят файлы `weisz.py` и `weisznoweight.py`.

Реализована возможность запустить программу с помощью файла `itc.bat` (необходимо присутствие каталога «`dist`»), без необходимости иметь установленную на ПЭВМ программную систему Python.

Программа обрабатывает один входной файл данных, задаваемый первым параметром командной строки. Если параметры командной строки не указаны, данные считываются из файла `data.txt`.

Данные во входном файле данных должны быть записаны в следующем формате:

- Файл содержит  $N$  текстовых строк, где  $N$  – количество векторов данных (количество радиоэлектронных изделий изделий в партии).

- Каждая строка содержит  $D$  числовых значений, разделенных пробелами (или точкой с запятой). Здесь  $D$  – количество измерений (изменяемых параметров электрорадиоизделий).

Исполняемый файл имеет три параметра:

1. файл данных (по умолчанию `data.txt`)
2. шаблон имени выходных файлов (по умолчанию `data.txt`)
3. режим.

Установка режима определяет используемую метрику (метрика городских кварталов  $L_1$  или квадратичная евклидова  $L_2^2$ ), способ нормировки исходных данных (по среднеквадратическому отклонению или равномерная (0-1)) и способ вычисления центра группы (`k-means` или `k-medoids`).

Предусмотрено 8 режимов, кодируемых цифрой: 1 - метрика  $L_1$ , `k-means`, нормировка по среднеквадр.отклонению; 2 - мера  $L_2^2$ , `k-means`, нормировка по среднеквадр.отклонению; 3 - метрика  $L_1$ , `k-medoids`, нормировка по среднеквадр.отклонению; 4 - мера  $L_2^2$ , `k-medoids`, нормировка по среднеквадр.отклонению; 5 -  $L_1$ , `k-means`, нормировка 0-1; 6 -  $L_2^2$ , `k-means`, нормировка 0-1; 7 -  $L_1$ , `k-medoids`, нормировка 0-1; 8 -  $L_2^2$ , `k-medoids`, нормировка 0-1.

По умолчанию используется режим 1.

Запуск программы производится командой «*its*<имя входного файла><имя (префикс) выходных файлов><режим>».

Результаты выводятся в несколько выходных текстовых файлов. Путь и начало имени (префикс) этих файлов задается вторым параметром командной строки. Если параметр не задан, префикс совпадает с именем входного файла данных. Далее данный префикс выходных файлов будем обозначать символом «\*».

Программа выводит следующие выходные файлы:

- Файл \*.res. Содержит значения целевой функции (суммарное расстояние от центров кластеров до векторов данных в нормированном квадратичном евклидовом пространстве) для различного числа кластеров. Каждая строка содержит два числа: число кластеров и соответствующее значение целевой функции.

- Файл \*.npos. Файл координат векторов данных в условном двумерном пространстве для визуализации. Поскольку векторы данных, как правило, имеют большое число измерений, и для визуализации требуется их отображение в двумерную координатную систему. В программе используется алгоритм MDS (multidimensional scaling). Если число векторов данных не превышает 500, для визуализации используются все векторы данных. В противном случае случайным образом выбираются 500 векторов данных для визуализации. В данном файле каждому вектору данных соответствует текстовая строка, содержащая два числа – координаты отображения вектора данных. Если для вектора данных визуализация не предусмотрена, ему соответствует строка «*infinf*».

- Файлы \*.*k*.centers. Здесь *k* – число кластеров. Данные файлы создаются для количества кластеров в диапазоне 1..9. Содержит координаты центров каждого кластера. Каждая строка содержит в начале целое число – номер кластера (нумерация кластеров начинается с 0) и *D* вещественных чисел, разделенных пробелами – координаты центра кластера.

- Файлы \*.*k*.clust. Здесь *k* – число кластеров. Данные файлы создаются для количества кластеров в диапазоне 1..9. Содержит соответствие номеров векторов данных номерам кластеров (вся нумерация начинается с 0). Каждая строка содержит два целых числа – номер вектора данных и номер кластера.

- Файлы \*.*k*.pos. Здесь *k* – число кластеров. Данные файлы создаются для количества кластеров в диапазоне 1..9. Содержит координаты векторов данных, выбранных для визуализации, в условном двумерном пространстве, и соответствующие номера кластеров. Каждая строка содержит 3 числа, разделенные пробелами: номер вектора данных (целое число, нумерация начинается с 0), две координаты (вещественные числа) и номер кластера (целое - нумерация ведется с нуля). Данный файл содержит не более 500 строк, поскольку для визуализации выбираются не более 500 векторов данных.

**ПРИЛОЖЕНИЕ Г. АКТЫ ОБ ИСПОЛЬЗОВАНИИ РЕЗУЛЬТАТОВ****АКТ**

*о внедрении  
результатов исследований Казаковцева Л.А.*

Настоящим актом подтверждается, что результаты исследований докторанта кафедры системного анализа и исследования операций Сибирского государственного аэрокосмического университета им. академика М.Ф.Решетнева Казаковцева Льва Александровича в области методов теории оптимального размещения внедрены и успешно используются в деятельности нашей организации. В частности, метод «оптимального размещения на сети» используется при построении математических моделей для решения задач оптимального размещения пунктов экстренного реагирования МЧС на территории Красноярского края.

Начальник Сибирского филиала  
ФГБУ ВНИИ ГОЧС (ФЦ) МЧС России



Н.В. Трофимова

24.05.2013.



Общество с ограниченной ответственностью  
«Инновационные строительные технологии»  
ИНН 2462218686 КПП 246201001 ОГРН 1112468068107

Юридический адрес: 660013, г. Красноярск, ул. Энергетиков, 73а, стр.16, пом.2  
Почтовый адрес: 660027, г. Красноярск, а/я 4710  
Телефон, факс: (391) 205-28-50, 205-28-54  
E-mail: [ist.met@mail.ru](mailto:ist.met@mail.ru)

**АКТ**  
о внедрении результатов диссертационного исследования  
Казаковцева Л.А.

Настоящим актом подтверждается, что в составе автоматизированной системы управления технологическими процессами ООО «Инновационные строительные технологии» была разработана и внедрена в эксплуатацию подсистема составления производственного расписания изготовления полипропиленовых изделий с повышенными эксплуатационными характеристиками для предприятий металлургической, химической и нефтехимической промышленности. При разработке данной подсистемы был использован генетический алгоритм с жадной эвристикой, разработанный докторантом ФГБОУ ВО «СибГАУ» Казаковцевым Львом Александровичем для решения задач оптимальной загрузки производственных мощностей производства в рамках диссертационного исследования. Внедрение данной подсистемы, позволяющей составлять оптимальный план загрузки производственных мощностей полипропиленового производства с минимальным количеством переключений типов выпускаемой продукции, позволило сократить затраты, связанные с простоем производственного оборудования и повысить качество выпускаемой продукции. Кроме того, внедрение подсистемы позволило сократить сроки изготовления продукции и перейти по некоторым видам продукции к системе производства «точно в срок».

Исполнительный директор



В. Э. Деменков

11.03.2015

АКТ  
о внедрении результатов диссертационного исследования  
Казаковцева Л.А.

Настоящим актом подтверждается, что система автоматизированного формирования и контроля специальных партий электрорадиоизделий космического применения, использующая алгоритмы автоматической группировки данных повышенной точности, в частности – эволюционный алгоритм с вещественным алфавитом и адаптивный алгоритм автоматической группировки на основе метода изменяющихся вероятностей, и созданный с их применением метод автоматической классификации электрорадиоизделий по классам качества и производственным партиям на основе данных входного контроля и разрушающего физического анализа, внедрена и успешно используется в деятельности ОАО «Испытательный технический центр – НПО ПМ».

Внедрение данной системы, разработанной в рамках диссертационного исследования докторанта ФГБОУ ВПО СибГАУ Казаковцева Льва Александровича, позволило регламентировать и автоматизировать процесс отбора электрорадиоизделий для формирования специальных партий изделий, предназначенных для аэрокосмического приборостроения, в результате чего появилась возможность реализации технологического процесса комплектования узлов космических аппаратов продукцией нового класса, позволяющей заменять продукцию иностранного производства класса SPACE.

Директор ОАО «Испытательный  
технический центр – НПО ПМ»



В.И.Орлов  
28.10.2015