






Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Системы автоматики, автоматизированное управление и проектирование

УТВЕРЖДАЮ
Заведующий кафедрой
 Ченцов С.В.
« » 2017 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ
ИНФОРМАЦИОННОЕ СОПРОВОЖДЕНИЕ РЕШЕНИЯ ЗАДАЧИ
РАЗМЕЩЕНИЯ

27.04.04 Управление в технических системах

27.04.04.01 Интегрированные системы управления производством

Научный руководитель		<u>16.06.</u> 2017г.	доцент, канд. техн. наук Е.Е. Носкова
Выпускник		<u>16.06.</u> 2017г.	М.А. Карнаухов
Рецензент		<u>16.06.</u> 2017г.	доцент, канд. техн. наук О.Н. Моргунова
Нормоконтролер		<u>16.06.</u> 2017г.	Т.А. Грудинова

Красноярск 2017

РЕФЕРАТ

Магистерская диссертация по теме "Информационное сопровождение решения задачи размещения" содержит 81 страницу текстового документа, 42 иллюстрации, 4 таблицы, 37 формул, 5 приложений, 13 использованных источников.

ЭЛЕКТРОРАДИОЭЛЕМЕНТЫ, МЕТОД ВЕТВЕЙ И ГРАНИЦ, УЧЕТ ТЕПЛОВЫХ МОДЕЛЕЙ ЭЛЕМЕНТОВ, РЕШЕНИЕ ЗАДАЧИ РАЗМЕЩЕНИЯ.

Актуальность исследования. Решение задачи размещения электронных компонентов на ПП как задачи многокритериальной глобальной оптимизации при минимизации суммарной длины соединений и оптимизации тепловых режимов работы печатной платы.

Цель. Целью диссертационной работы модификация метода ветвей и границ, для решения размещения как задачи глобальной оптимизации при минимизации суммарной длины соединений и оптимизации тепловых режимов работы печатной платы.

Задачи. Для достижения указанной цели необходимо решить следующие задачи:

- Проанализировать методы, используемые для решения задачи размещения как задачи глобальной оптимизации;
- Разработать математическую модель задачи размещения с учетом тепловых характеристик компонентов;
- Разработать алгоритм модифицированного метода ветвей и границ;
- Разработать программное обеспечение на основе предложенного алгоритма решения задачи размещения с учетом тепловых моделей элементов и требуемых технологических ограничений;
- Исследовать эффективность предложенного метода и алгоритма с помощью вычислительного эксперимента.

СОДЕРЖАНИЕ

Реферат	2
Введение.....	4
1 Алгоритмы и методы решения задачи размещения	6
1.1 Постановка задачи размещения.....	6
1.2 Задача размещения как задача квадратичного назначения	18
1.2.1 Полный перебор	21
1.2.2 Метод ветвей и границ как сокращение полного перебора.....	22
1.3 Решение задачи размещения с учетом тепловых характеристик с помощью современных САПР.....	30
2 Решение задачи размещения с оптимизацией тепловых режимов	40
2.1 Постановка задачи теплового анализа	40
2.2 Модификация метода ветвей и границ. Учет тепловых режимов	48
3 Технологические аспекты программной реализации.....	51
3.1 Выбор инструментария разработки программы	51
3.2 Функциональное назначение программного продукта	53
3.3 Разработка программного обеспечения	53
3.4 Структура программного продукта.....	55
3.5 Формирование исходных данных.....	58
3.6 Формирование выходных данных	63
3.7 Анализ результата работы модифицированного алгоритма	65
Заключение	76
Список использованных источников	78
Приложение А	80
Приложение Б.....	97
Приложение В.....	98
Приложение Г	101
Приложение Д.....	116

ВВЕДЕНИЕ

При конструкторском проектировании радиоэлектронной аппаратуры (РЭА) решаются задачи, связанные с поиском наилучшего варианта конструкции, удовлетворяющего требованиям технического задания и максимально учитывающего возможности технологической базы производства. Одной из основных задач синтеза конструкций является задача размещения элементов коммутационной схемы на заданном коммутационном поле. Размещение элементов – это задача определения их местоположения на коммутационном поле в конструктивном модуле такого, при котором создаются наилучшие условия для решения последующей задачи трассировки соединений с учетом конструктивно–технологических требований и ограничений. Среди существующих алгоритмов размещения группа последовательных алгоритмов в наибольшей степени имитирует действия инженера проектировщика, рассчитывая при этом локальный критерий оптимальности. Поиск глобального экстремума при решении задачи размещения возможен только при использовании метода полного перебора и его сокращения через применение методов направленного перебора. Одним из основных методов направленного перебора является метод ветвей и границ, применение которого при решении задач размещения электронных компонентов на печатной плате (ПП) позволяет найти глобальный экстремум.

При его реализации можно осуществить комплексирование критерия оптимальности, т.е. решить задачу размещения как задачу многокритериальной глобальной оптимизации.

Актуальность исследования. Решение задачи размещения электронных компонентов на ПП как задачи многокритериальной глобальной оптимизации при минимизации суммарной длины соединений и оптимизации тепловых режимов работы печатной платы.

Цель. Целью диссертационной работы модификация метода ветвей и границ, для решения размещения как задачи глобальной оптимизации при

минимизации суммарной длины соединений и оптимизации тепловых режимов работы печатной платы.

Задачи. Для достижения указанной цели необходимо решить следующие задачи:

- Проанализировать методы, используемые для решения задачи размещения как задачи глобальной оптимизации;
- Разработать математическую модель задачи размещения с учетом тепловых характеристик компонентов;
- Разработать алгоритм модифицированного метода ветвей и границ;
- Разработать программное обеспечение на основе предложенного алгоритма решения задачи размещения с учетом тепловых моделей элементов и требуемых технологических ограничений;
- Исследовать эффективность предложенного метода и алгоритма с помощью вычислительного эксперимента.

Практическая ценность. Разработанная программа, на основе модифицированного метода ветвей и границ, позволит решить задачи размещения и теплового анализа в одной подсистеме, что позволит сэкономить затраченное время инженера–проектировщика на разработку печатной платы.

Методы исследования. В работе были использованы: математический аппарат дискретной математики; модели и методы теории разработки систем автоматизированного проектирования, а именно, методы конструирования, постановка задачи размещения как задача глобальной оптимизации, алгоритмы размещения; теория теплообмена; методология и технология разработки программного обеспечения.

Апробирование предложенных решений проводилось при проектировании заданных печатных узлов. Достоверность результатов учета тепловых характеристик элементов, при решении задачи их размещения на печатной плате, доказана на основе теплового расчета характеристик спроектированных печатных плат, полученных в коммерческих САПР.

1 Алгоритмы и методы решения задачи размещения

1.1 Постановка задачи размещения

Задача размещения элементов является одной из основных задач конструкторского этапа проектирования электронных устройств и состоит в определении оптимального пространственного расположения элементов на коммутационном поле.

Исходной информацией при решении задач размещения являются: данные о конфигурации и размерах коммутационного пространства, определяемые требованиями установки и крепления данной сборочной единицы в аппаратуре; количество и геометрические размеры конструктивных элементов, подлежащих размещению; схема соединений, а также ряд ограничений на взаимное расположение отдельных элементов, учитывающих особенности разрабатываемой конструкции. Задача сводится к отысканию для каждого размещаемого элемента таких позиций, при которых оптимизируется выбранный показатель качества и обеспечиваются наиболее благоприятные условия для последующего электрического монтажа. Особое значение эта задача приобретает при проектировании аппаратуры на печатных платах.

Основная сложность в постановке задач размещения заключается в выборе целевой функции. Связано это с тем, что одной из главных целей размещения является создание наилучших условий для дальнейшей трассировки соединений, что невозможно проверить без проведения самой трассировки. Любые другие способы оценки качества размещения (минимум числа пересечений ребер графа, интерпретирующего электрическую схему соединений, разбиение графа на минимальное число плоских суграфов и т.д.), хотя и позволяют создать благоприятные для трассировки условия, но не гарантируют получение оптимального результата, поскольку печатные проводники представляют собой криволинейные отрезки конечной ширины, конфигурация которых определяется в процессе их построения и зависит от

порядка проведения соединений. Следовательно, если для оценки качества размещения элементов выбрать критерий, непосредственно связанный с получением оптимального рисунка металлизации печатной платы, то конечный результат может быть найден только при совместном решении задач размещения, выбора очередности проведения соединений и трассировки, что практически невозможно вследствие огромных затрат машинного времени.

Поэтому все применяемые в настоящее время алгоритмы размещения используют промежуточные критерии, которые лишь качественно способствуют решению основной задачи: получению оптимальной трассировки соединений. К таким критериям относятся:

- минимум суммарной взвешенной длины соединений;
- минимум числа соединений, длина которых больше заданной;
- минимум числа пересечений проводников;
- максимальное число соединений между элементами, находящимися в соседних позициях либо в позициях, указанных разработчиком;
- максимум числа цепей простой конфигурации.

Наибольшее распространение в алгоритмах размещения получил первый критерий. Это объясняется следующими причинами: уменьшение длин соединений улучшает электрические соединения устройства и упрощает трассировку печатных проводников и снижает трудоемкость изготовления.

В зависимости от конструкции коммутационной платы и способа выполнения соединений, расстояние между позициями установки элементов подсчитывается по одной из формул:

$$d^{(1)}_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad (1.1)$$

$$d^{(2)}_{ij} = |x_i - x_j| + |y_i - y_j|, \quad (1.2)$$

$$d^{(3)}_{ij} = (x_i - x_j)^t + (y_i - y_j)^t. \quad (1.3)$$

где (x_i, y_i) и (x_j, y_j) – координаты i -ой и j -ой позиции коммутационной платы.

Формула (1.1) соответствует проведению проводников по кратчайшему пути между соединяемыми точками.

Формула (1.2) предлагает раскладку проводников параллельным сторонам платы.

Формула (1.3) применяется при наличии особых требований к максимальной длине отдельных соединений (как правило, $t=2$).

При практической реализации алгоритма размещения часто используют представления конструктивных элементов и позиции на коммутационной плате точками, совпадающие с их геометрическими центрами, а все соединения между элементами приводят к попарно взвешенным связям.

При практической реализации алгоритмов размещения часто используют представление конструктивных элементов и позиций на коммутационной плате точками, совпадающими с их геометрическими центрами, а все соединения между элементами приводят к попарно взвешенным связям.

Весовые оценки связей учитывают такие характеристики схемы, как число электрических цепей между элементами, теплонагруженность элементов, распространения сигналов в цепях и т.д.

В общем виде задача размещения конструктивных элементов на коммутационной плате формулируется следующим образом. Коммутационная схема представлена взвешенным графом соединений, описываемым матрицей соединений:

$$R = \| \| r_{ij} \| \|_{n \times n}, \quad (1.4)$$

где r_{ij} – число связей между e_i и e_j элементами, а n – количество элементов. Коммутационное поле описывается матрицей расстояний:

$$D = \| \| d_{ij} \| \|_{m \times m}, \quad (1.5)$$

где d_{ij} – расстояние между i и j позициями, а m – количество позиций коммутационного поля, такое что $m \geq n$, а также задано множество установочных мест (позиций) на коммутационной плате l_1, l_2, \dots, l_m ($m \geq n$).

Если критерием качества размещения является минимум суммарной взвешенной длины соединений, то задача состоит в его минимизации (1.14).

в такой постановке эта задача соответствует одной из типовых задач математического программирования – задаче квадратичного назначения (раздел 1.2).

Обычно поле позиций (коммутационная плата) имеет форму прямоугольника $P_{ab} = a \times b$ с координатами $0 \leq x \leq a$ и $0 \leq y \leq b$. Вся площадь платы разбивается на ряд областей (позиций), число которых должно быть не меньше числа размещаемых элементов. В результате получим фиксированные позиции для установки элементов (рисунок 1).

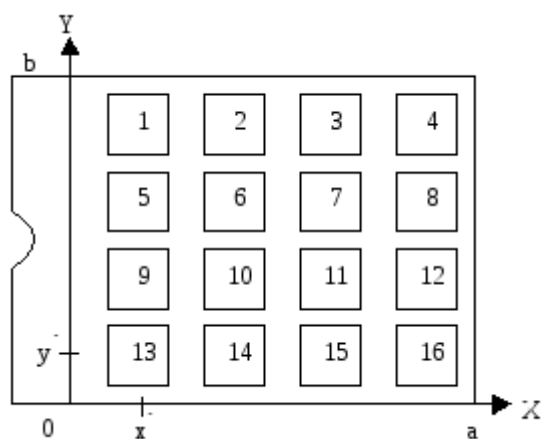


Рисунок 1 – Коммутационное поле

Перед разбиением поверхности коммутационной платы на позиции выделяют области для размещения выводных контактных зон схемы, а также запрещенные области, в которых не должны размещаться элементы схемы.

Будем считать, что соединения исходят из геометрических центров конструктивных элементов. Все конструктивные элементы, подлежащие размещению, можно условно разделить на три группы:

а) нефиксированные элементы, местоположение которых на плате заранее не известно;

б) граничные элементы, к которым относятся элементы, связанные с разъемами, осуществляющие электрическую связь с элементами, расположенными на других коммутационных платах. Т.к. разъемы обычно помещают на внешней стороне коммутационной платы, то эти элементы желательно располагать у границы коммутационного поля;

в) фиксированные элементы, местоположение которых заранее известно (указано разработчиком).

Всю совокупность алгоритмов размещения можно разделить на следующие основные группы, представленные на рисунке 2.

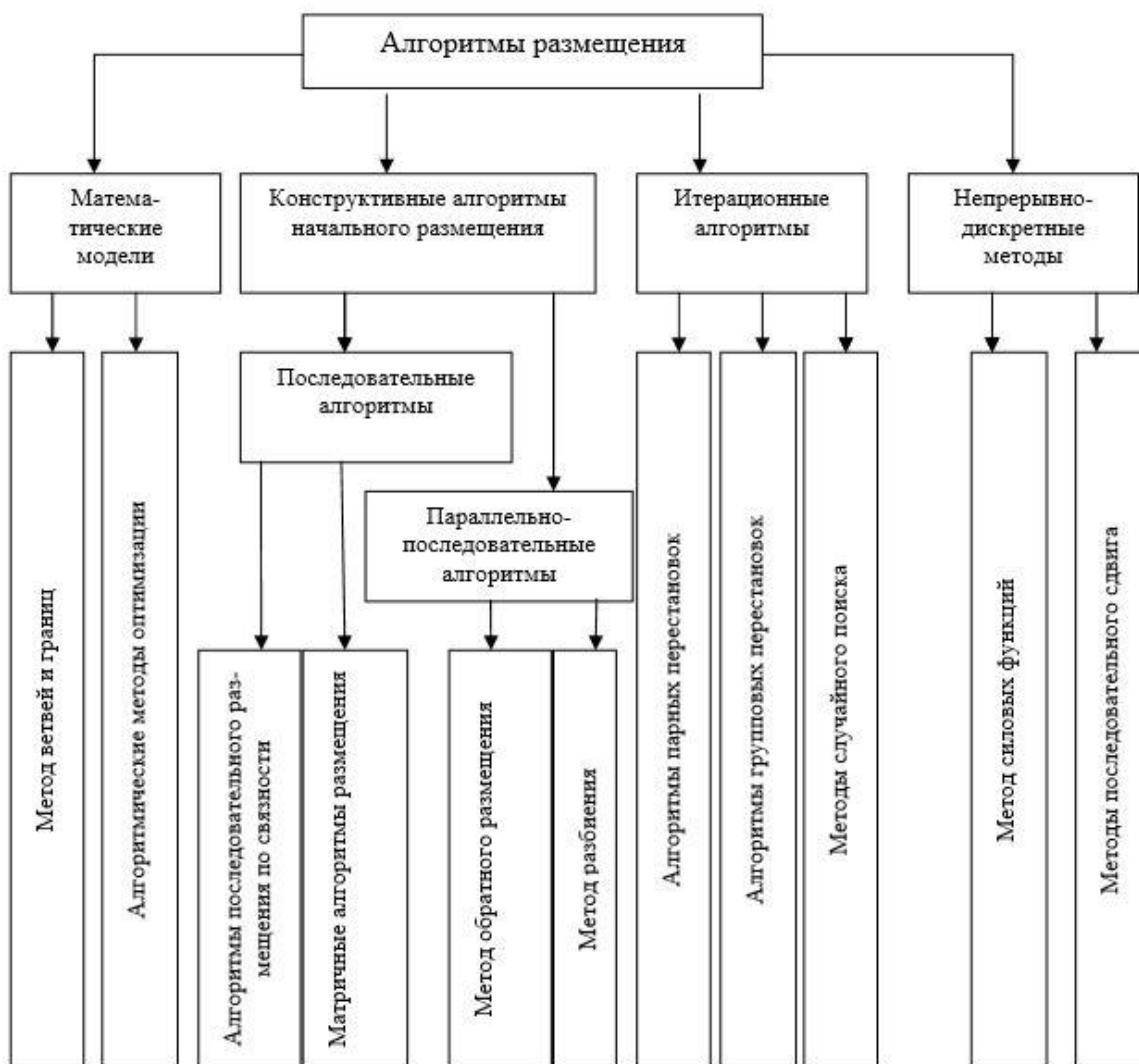


Рисунок 2 – Алгоритмы размещения

Группы конструктивных и итерационных алгоритмов включают приближённые алгоритмы, в основном предназначенные для оптимизации размещения элементов в фиксированном наборе позиций.

В методе обратного размещения конструктивного алгоритма осуществляется предварительная оценка каждого из размещаемых элементов e_1, e_2, \dots, e_n и каждой свободной позиции l_1, l_2, \dots, l_n , после чего все элементы размещаются одновременно.

Пусть дана матрица соединений $R = \|r_{ij}\|_{n \times n}$ и матрица расстояний между позициями $D = \|d_{ij}\|_{n \times n}$.

Для каждого элемента e_i по матрице R найдем суммарное число соединений этого элемента с остальными элементами:

$$r_i = \sum r_{ij}, i = 1, 2, \dots, n, \quad (1.6)$$

Для каждой позиции l_i по матрице D найдем характеристику

$$d_i = \sum d_{ij}, i = 1, 2, \dots, n, \quad (1.7)$$

определяющую суммарное расстояние этой позиции до остальных позиций.

Очевидно, что позиции в центральной части КП имеют меняющуюся характеристику d_i , чем позиции на периферии. Естественно, что центральные позиции наиболее благоприятны для размещения сильно связанных элементов, т. е. элементов с большим значением r_i (1.6). Рассматривая с этой точки зрения выражение для суммарной взвешенной длины соединений и учитывая условия минимальности скалярного произведения $r \times d$, получаем следующий эвристический алгоритм размещения.

- 1) Упорядочить элементы по возрастанию характеристики $r_i : i_1, i_2, \dots, i_n$ ($r_1 \leq r_2 \leq \dots \leq r_n$);
- 2) Упорядочить позиции по убыванию характеристики $d_j : j_1, j_2, \dots, j_n$ ($d_1 \geq d_2 \geq \dots \geq d_n$);
- 3) Определить размещение $p(i_k) = j_k$ $k = 1, 2, \dots, n$.

Описанный метод отличается от большинства алгоритмов размещения чрезвычайной простотой. Он может быть использован даже при ручной методике разработки схем для выбора грубого начального варианта размещения элементов. Опыт использования данного алгоритма показывает, что он, как правило, дает решение несколько уступающее по качеству размещения, выполненному опытным конструктором, в то время как последовательные и матричные алгоритмы размещения дают сокращение длины до 10–20%. [1]

В итерационных алгоритмах используют общие идеи методов последовательных приближений и являются комбинаторными аналогами градиентных методов оптимизации. Для этих алгоритмов необходимо задать

начальный вариант размещения. Итерационные алгоритмы применяются для решения задачи размещения с различными критериями оптимальности $F(p)$: суммарная длина соединений, суммарное число пересечений соединений и т.д.

В любом итерационном алгоритме исследуется некоторое подмножество размещений, в некотором смысле близких к начальному, для выделения в нем размещения с меньшим значением функции-критерия. В большинстве случаев такой процесс приводит к получению локального минимума функции $F(p)$.

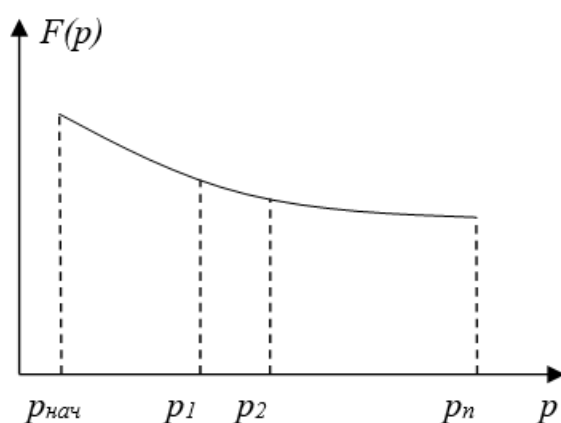


Рисунок 3 – Улучшение размещения

Пусть $F(p)$ – некоторая функция–критерий размещения, а $p_{нач}$ – начальное размещение. Тогда в результате применения итерационного алгоритма размещения получится последовательность размещений $p_{нач}, p_1, p_2, \dots, p_n$, которой соответствует монотонно убывающая последовательность значений $F(p_{нач}) > F(p_1) > F(p_2) > \dots > F(p_n)$ (рисунок 3). Значение $F(p_n)$, вообще говоря, соответствует локальному минимуму функции. Однако практика применения подобных алгоритмов показывает, что получаемые размещения близки к оптимальным. Это, по–видимому, связано с тем, что используемые критерии оптимизации являются относительно пологими функциями, не имеющими «острых» экстремумов.

Характерной чертой итерационных алгоритмов является возможность получения варианта размещения в любой момент итерационного процесса. Поэтому при реализации алгоритмов на ПК, как правило, итерационные

процессы заканчиваются, как только разность значений функции–критерия для двух соседних итераций становится относительно малой:

$$[F(p_k) - F(p_{k-1})] / F(p_k) < \alpha, \quad (1.8)$$

где α – заранее заданное число.

Различные итерационные алгоритмы размещения имеют сходную структуру, содержащую следующие моменты:

- 1) преобразование очередного размещения;
- 2) вычисление функции размещения;
- 3) выбор лучшего размещения;
- 4) переход к следующей итерации и правило остановки.

В качестве начального, может быть взято размещение, полученное одним из конструктивных алгоритмов размещения, с помощью генератора случайных размещений, или заданное инженером–проектировщиком.

В качестве примера можно привести пару методов, которые получили наибольшее распространение, и эффективность которых подтверждается при решении практических задач:

- метод парных перестановок;
- метод групповых перестановок. [1]

Эти алгоритмы требуют существенных затрат машинного времени и используются для получения конечного размещения.

Для непрерывно–дискретных методов размещения, задание фиксированного набора позиции необязательно: размещение элементов осуществляется на непрерывной плоскости. Такое решение представляет наибольший интерес для конструкции, содержащих разногабаритные элементы. Примерами могут служить твердотельные и гибридные интегральные схемы, односторонние печатные платы с навесными элементами и т.п. Вместе с тем, эти методы могут быть использованы и для решения задач размещения, в которых позиции установки элементов фиксированы. В этом

случае решение, полученное непрерывным методом, считается приближением решения соответствующей дискретной задачи.

Процесс размещения элементов, как правило, включает два этапа:

- Решение непрерывной задачи;
- Преобразование этого решения к дискретному множеству позиций.

Большинство способов преобразования непрерывного решения сводится к перемещению элементов из точек, соответствующих полученному решению, в заданные позиции с минимизацией изменения целевой функции.

В методе силовых функции элементы считаются материальными точками, на которые действуют силы притяжения и отталкивания. Сила притяжения между элементами e_i и e_j обычно полагается пропорциональной расстоянию между ними. Силы отталкивания вводятся для предотвращения слияния или перекрытия элементов, а также для получения равномерного их расположения. Далее осуществляется поиск состояния равновесия системы материальных точек, которое и определяет размещение элементов на плоскости.

Сила притяжения вычисляется следующим образом:

$$F_{ij} = r_{ij}f(x_i, y_i, x_j, y_j), \quad (1.9)$$

где $|f(x_i, y_i, x_j, y_j)| = k_f d_{ij}$;

r_{ij} – число соединений между элементами e_i и e_j ;

k_f – коэффициент.

Сила отталкивания между элементами, как правило, считается обратно пропорционально расстоянию между ними:

$$\Phi_{ij} = \varphi(x_i, y_i, x_j, y_j), \quad (1.10)$$

где $\varphi(x_i, y_i, x_j, y_j) = k_\varphi / d_{ij}$;

k_φ – коэффициент.

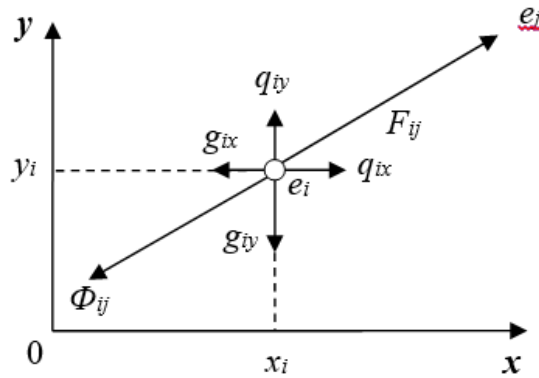


Рисунок 4 – Силы взаимодействия элементов

Статический вариант метода силовых функции предполагает составление уравнения статического равновесия системы материальных точек. Условиями такого равновесия является равенство нулю результирующих сил, действующих на каждую точку системы. На рисунке 4 показаны силы, действующие на элемент e_i . Здесь F_{ij} и Φ_{ij} – соответственно силы притяжения и отталкивания между элементами e_i и e_j ; g_{ix} и q_{ix} – силы притяжения и отталкивания элемента e_i от левой и правой границы коммутационного поля; g_{iy} и q_{iy} – аналогичные силы, действующие в вертикальном направлении.

При определении F_{ij} и Φ_{ij} в соответствии с (1.9) и (1.10) получим для $F_x(i,j)$ и $\varphi_x(i,j)$ следующие выражения:

$$F_x(i, j) = k_f r_{ij}(x_j - x_i), \quad \varphi_x(i, j) = k_f / |x_i - x_j|, \quad (1.11)$$

Аналогичные выражения имеют место для $F_y(i, j)$ и $\varphi_y(i, j)$. Состояние равновесия системы материальных точек наступает, когда

$$R_{ix} = 0, \quad i = 1, 2, \dots, n; \quad R_{iy} = 0, \quad i = 1, 2, \dots, n, \quad (1.12)$$

Подстановка в (1.12) конкретных выражений для сил, действующих на элементы, приводит к алгебраической системе $2n$ уравнений с $2n$

неизвестными, для решения которой могут быть применены известные методы.
[1]

К группе математических моделей относятся, прежде всего, метод ветвей и границ для задачи квадратичного назначения (1.2.2), к которой при определённых упрощениях сводится задача размещения: набор позиций считается фиксированным, элементы рассматриваются как геометрические точки, схема соединений представляется взвешенным графом.

Таблица 1 – Качественные характеристики используемых алгоритмов

Характеристики	Алгоритмы			
	1*	2**	3***	4****
Гарантированный поиск глобального оптимума	+	-	-	-
Гарантированность вычислений	+	-	-	-
Небольшие временные затраты	-	+	-	-
Сужение области поиска	+	-	-	+
Простота реализации	-	+	+	-
* Математические модели задачи размещения. ** Конструктивные алгоритмы начального размещения *** Итерационные алгоритмы улучшения начального варианта размещения **** Непрерывно–дискретные методы размещения				

Как видно из таблицы 1 – достоинством математических моделей является использование глобального поиска оптимального размещения и гарантированное нахождение оптимального решения. Тогда как другие алгоритмы используют локальный поиск оптимального размещения элементов на ПП и не гарантируют нахождение оптимального решения, как правило предлагая решение очень близкое к оптимальному.

1.2 Задача размещения как задача квадратичного назначения

Задача размещения электронных компонентов на печатной плате (ПП) является задачей квадратичного назначения и задачей поиска глобального экстремума.

Пусть даны элементы e_1, e_2, \dots, e_n и для каждой пары элементов заданы весовые коэффициенты $r_{ij}(i, j=1, 2, \dots, n)$, определяющие «степень связи» элементов друг с другом. Таким образом, считаем, что схема задана матрицей соединений (1.4).

Пусть также имеется некоторый фиксированный набор позиций для размещения элементов $l_1, l_2, \dots, l_m (m \geq n)$. Будем полагать, что $m=n$. Если $m > n$, то можно ввести $m-n$ фиктивных элементов, не связанных с остальными. Определим расстояние d_{ij} между парами позиций. В любом случае, если на коммутационном поле фиксированы позиции для размещения элементов, то можно задать матрицу соединений (1.5), в которой элемент d_{ij} равен расстоянию между центрами позиций l_i и l_j . Очевидно, что матрица D – симметрическая с нулевой диагональю ($d_{ii}=0, i=1, 2, \dots, n$).

Произвольное размещение элементов в позициях представляет собой некоторую перестановку $p=p(1), \dots, p(i), \dots, p(n)$, где $p(i)$ задает номер позиции, присвоенный i -му элементу. Таким образом, всего имеется $n!$ различных вариантов размещения элементов.

Рассмотрим задачу минимизации СДС при следующих предположениях. Соединения будем считать условно исходящими из геометрических центров элементов. Кроме того, предполагается совпадение центров элементов и позиций. Как правило, при решении задачи размещения необходимо учитывать предварительное закрепление некоторых элементов в позициях и соединения элементов с внешними выводами. Сопоставляя внешним выводам элемент e_0 и фиксируя расположение элементов, получим упрощенное представление коммутационного поля. Очевидно, что длина соединений между элементами e_i и e_j оценивается величиной $r_{ij}d_{p(i)p(j)}$. Обозначив через L_s множество всех

фиксированных элементов, включая элемент e_0 ; тогда суммарная взвешенная длина соединений элемента e_i с элементами из L_s оценивается по формуле:

$$a_{ip(i)} = \sum_{s \in L_s} r_{ij} d_{p(i)s}, \quad (1.13)$$

где $d_{p(i)s}$ —расстояние между элементом e_i , находящимся в позиции $p(i)$, и элементом e_s .

Учитывая вышесказанное, а также симметричность матриц R и D , выражение для суммарной взвешенной длины соединений (СДС) при произвольном размещении запишется:

$$F(p) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n r_{ij} d_{p(i)p(j)} + \sum_{i=1}^n a_{ip(i)}, \quad (1.14)$$

Таким образом, задача размещения по критерию минимизации СДС состоит в минимизации функционала (1.14) на множестве перестановок p . Данная задача является вариантом общей математической модели, впервые рассмотренной Купмансом и Бекманом и получивший название задачи квадратичного назначения.

В матрице переменных $X = \|x_{ij}\|_{n \times n}$, строки соответствуют элементам, а столбцы – позициям. Переменная $x_{ij} = 1$, если элемент e_i находится в l_j позиции, и $x_{ij} = 0$ в противном случае. Легко установить взаимно–однозначное между матрицей X и некоторой перестановкой элементов в позициях. Поэтому X называют перестановочной матрицей или матрицей назначений. Очевидно, что матрица X содержит в каждом столбце j и в каждом строке i одну единицу:

$$\sum_{i=1}^n x_{ij} = 1, i = 1, 2, \dots, n, \quad \sum_{j=1}^n x_{ij} = 1, j = 1, 2, \dots, n,$$

С учётом введённых обозначений задачу размещения можно записать в форме обобщённой задачи квадратичного назначения: найти матрицу X_0 , для которой

$$F(X_0) = \min\left(\frac{1}{2} \sum_{i,j=1}^n \sum_{s=1}^n r_{ij} d_{ks} x_{ik} x_{js} + \sum_{i,k=1}^n a_{ik} x_{ik}\right), \quad (1.15)$$

Следует, отметить, что линейный член в (1.15) представляет собой функционал обычной (линейной) задачи о назначении. Последняя формулируется следующим образом. Пусть задана некоторая матрица $A = \|a_{ij}\|_{n \times n}$, строки которой соответствуют некоторым объектам, а столбцы местам их назначения. Элемент a_{ij} оценивает некоторую условную стоимость назначения объекта i на место j . Задача состоит в нахождении такого назначения объектов по местам (без повторений), при котором суммарная стоимость всех назначений минимальна. В другой терминологии задача сводится к минимизации

$$\sum_{i=1}^n a_{ip(i)},$$

на множестве перестановок $P = \{p\}$, $A = \|a_{ij}\|_{n \times n}$ – заданная матрица коэффициентов.

К задаче линейного назначения сводится ряд частных задач оптимизации, возникающих при размещении элементов и при распределении их выводов. Из (1.14) следует, что задача размещения несвязанного множества элементов эквивалентна задаче линейного назначения.

Для задачи линейного назначения существуют эффективные точные методы решения. Одним из наиболее распространенных является венгерский алгоритм, позволяющий получить решение для $n > 100$. Задача квадратичного

назначения – существенно сложнее, поскольку «стоимость» назначения каждого элемента зависит от назначения всех связанных с ним элементов.

Для решения задачи квадратичного назначения предложен ряд алгоритмов, основанных на алгоритме полного перебора и методе ветвей и границ.

1.2.1 Полный перебор

Основная идея полного перебора состоит в переборе всех возможных расположений элементов на печатной плате в поисках оптимального. Этот подход можно использовать только тогда, когда общее число элементов ПП, с точки зрения практических вычисления, достаточно мало. Достоинством данного метода является гарантированное нахождение оптимального размещения.

Любая задача из класса NP может быть решена полным перебором. При этом, даже если вычисление целевой функции от каждого конкретного возможного решения задачи может быть осуществлено за полиномиальное время, в зависимости от количества всех возможных решений полный перебор может потребовать экспоненциального времени работы.

Пусть имеется множество элементов $e_1, \dots, e_i, \dots, e_n$ и множество позиций $l_1, \dots, l_i, \dots, l_m$. ($m \geq n$). Если полагать, что $m=n$ то полный перебор будет иметь $n!$ решений. Если же $m > n$, то полный перебор будет иметь $m!$ решений. При среднем количестве элементов на печатной плате – 1500, полный перебор будет иметь $1500!$ решений. На нахождение оптимального решения, в данном случае, могут уйти столетия, что является очевидным недостатком данного метода.

Для описания процесса поиска решений введём дерево решений. Ребра первого яруса дерева пусть соответствуют возможным назначениям элемента e_1 , рёбра второго яруса – возможным вариантам назначения элемента e_2 в свободные позиции и т.д.

Расчёт глобального критерия оптимальности целевой функции осуществляется для всех путей (размещений) полного дерева, что позволяет в конце расчета выбрать оптимальный путь исходящий из начальной вершины.

На рисунке 5 показана дерево полного перебора решений, в котором отмечены возможные назначения.

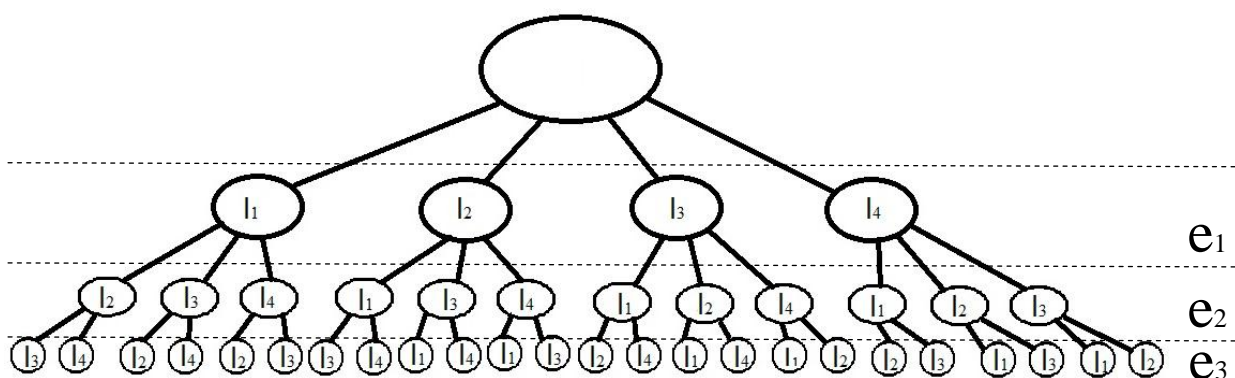


Рисунок 5 – Ветви и графы полного перебора

1.2.2 Метод ветвей и границ как сокращение полного перебора

Метод ветвей и границ является одним из основных методов направленного перебора, применение которого при решении задач размещения электронных компонентов на печатной плате позволяет найти глобальный экстремум. Данный метод представляет особый интерес так как при его модификации можно осуществить комплексирование критерия оптимальности, т.е. решить задачу размещения как задачу многокритериальной глобальной оптимизации учитывая при этом минимизацию суммарной длины соединений и оптимизацию тепловых режимов работы печатной платы.

Основная идея метода ветвей и границ состоит в разбиении всего множества допустимых решений задачи на некоторые подмножества, внутри осуществляется упорядоченный просмотр решений с целью выбора оптимального. Для всех решений, входящие в выделенные подмножества, вычисляется нижняя граница минимального значения целевой функции. Как только нижняя граница становится больше значения целевой функции для

наилучшего из ранее известных решений, подмножество решений, соответствующее этой границе, исключается из исходной области решений. Это обеспечивает сокращение перебора. Процесс поиска, сопровождаемый разбиением поля решений и вычислением нижних границ продолжается до тех пор, пока не будут исключены все решения, кроме оптимального.

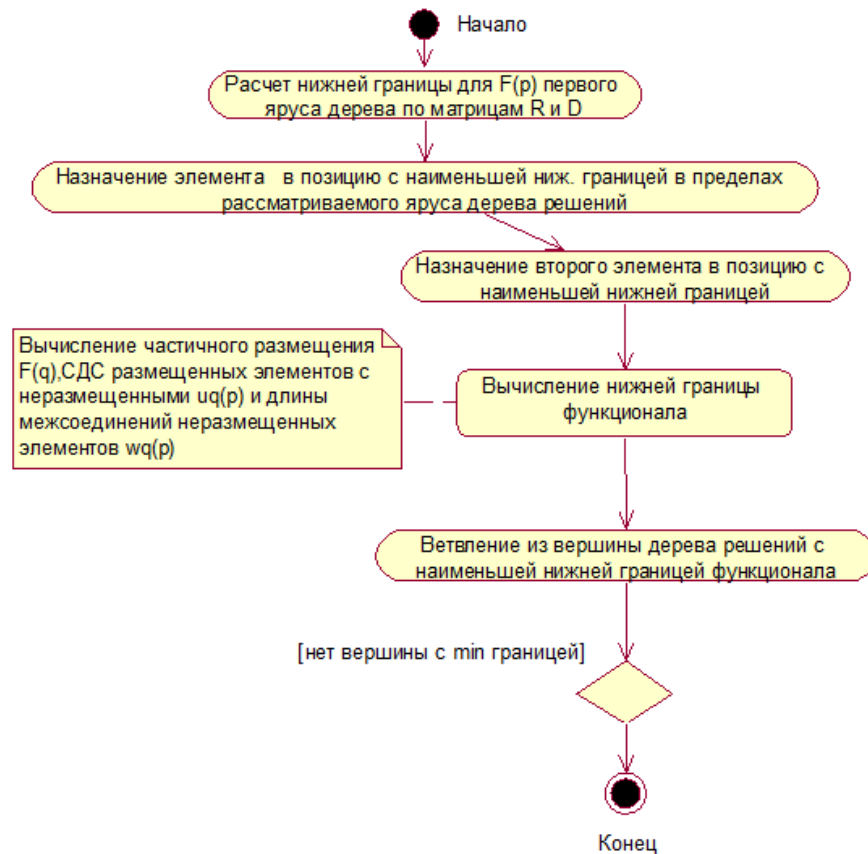


Рисунок 6 – Блок–схема алгоритма метода ветвей и границ

Различия модификации общего метода применительно к задаче размещения (квадратичного назначения) отличаются способами расчёта нижних границ функционала (1.19) и способами разбиения поля решений.

Пусть имеется множество элементов $e_1, \dots, e_i, \dots, e_n$ и множество позиций $l_1, \dots, l_i, \dots, l_n$. Для описания процесса поиска решений введём дерево решений. Ребра первого яруса дерева пусть соответствуют назначениям элемента e_1 , рёбра второго яруса – возможным вариантам назначения элемента e_2 в свободные позиции и т.д.

На рисунке 7 показана часть дерева полного дерева решений, в котором отмечены возможные назначения при условии, что элемент e_1 назначен в позицию l_j .

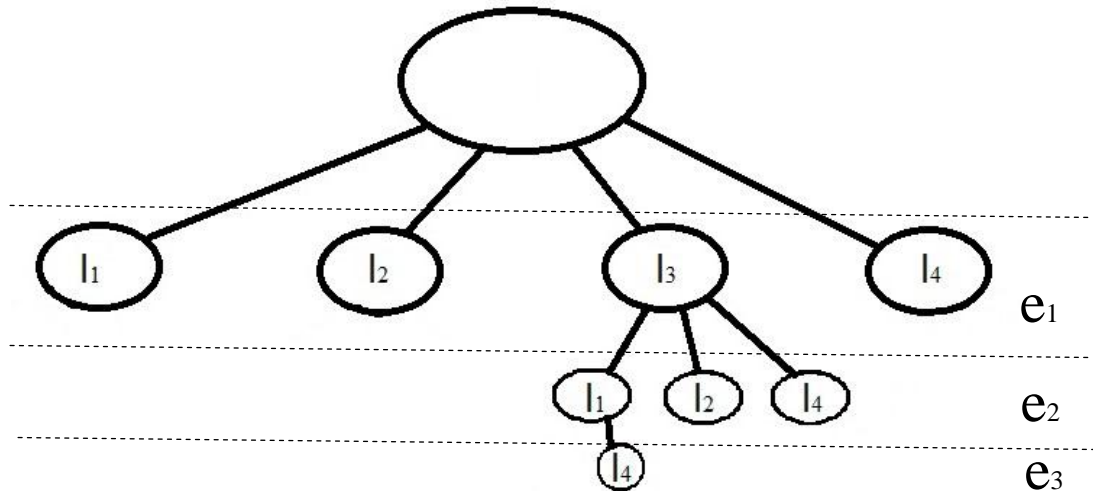


Рисунок 7 – Ветви и графы метода ветвей и границ

Произвольному размещению элементов соответствует в полном дереве некоторый путь, исходящий из начальной вершины. Для каждой вершины дерева можно рассчитать нижнюю границу целевой функции для множества путей (размещений), связанных с этой вершиной. Если эта граница больше значения целевой функции для известного размещения, то дальнейшее продвижение по дереву в данной вершине прекращается, поскольку оно приводит к заведомо не оптимальному решениям.

Частичным размещением q , будем называть назначение элементов множества $E_k = \{e_1, \dots, e_i, \dots, e_k\}$ в позиции множества $L_k = \{l_1, \dots, l_j, \dots, l_k\}$ в соответствии с правилом $j = q(i)$, $k \leq n$. Ясно, что $q \subset p$, где p – некоторое размещение n элементов, в котором k имеют размещение q . Частичному размещению q соответствует в дереве решений путь, проходящий через вершины отдельных назначений.

Если в процессе продвижения по дереву нижние границы не рассчитываются и соответственно не производится усечений дерева, то метод ветвей и границ сводится к полному перебору.

Можно выделить следующие способы отсечения ветвей:

1) Сравнение оценки (нижней – F_H или верхней – F_B границы) со значением целевой функции для уже найденного (опорного) решения $F_{ОП}$. Действительно, если при решении задачи на минимум целевой функции $F_H > F_{ОП}$, то подмножество путей (размещений) не содержит оптимального варианта, и соответствующая вершина дерева решений отсекается. Опорное решение можно получить приближенным алгоритмом заранее либо в ходе решения задачи методами ветвей и границ. При разбиении пространства решений по методу в глубину с возвращением опорное решение получается на более ранних этапах построения дерева решения, чем при методе в ширину.

2) Сравнение двух оценок. Такое отсечение выполняется, если, например, в задаче на минимум целевой функции, для подмножества вариантов соответствующей вершины можно найти оценку снизу F_H и сверху $F_{ОП}$. Тогда, если для некоторого подмножества путей (размещений), окажется что $F_H \geq F_{ОП}$, то ветвление в соответствующей вершине прекращается.

Прекращение ветвления, если соответствующее подмножество не содержит оптимального решения, можно установить, во – первых, «В особых точках» по значениям оценочной функции, например нижней границы, и во – вторых, если известен оптимальный среди вариантов этого подмножества. Чем точнее будет получена оценка, т.е. чем ближе она будет к значению целевой функции для оптимального варианта подмножества, тем меньшее количество вершин дерева решений будет построено и исследовано.

Точность оценки зависит от принципа разбиения множества вариантов на подмножества и виды оценочной функции или способа ее вычисления. Следовательно, если возможно использовать несколько принципов разбиения, необходимо выбирать тот, при котором оценки более точны. Как правило, число отсечений тем больше, чем сильнее отличаются оценки подмножеств.

Таким образом, лучшими являются тот принцип разбиения и такая оценочная функция, при которых разность между оценками подмножеств наибольшая, а сами оценки вычисляются с наибольшей точностью в смысле их

близости к значению целевой функции для оптимального варианта подмножества. Число отсечений зависит и от способа ветвления, хотя никаких точных оценок и рекомендаций узнать нельзя.

Рассмотрим основные способы ветвления.

1) Разбиение множества вариантов на подмножества по методу «в ширину» и выбор вершины ветвления по минимуму (максимуму) оценочной функции. Сначала выполняется разбиение всего множества вариантов, т.е. строятся на следующем от корня уровне все или часть вершин его потомков. Затем на каждом шаге выбирается вершина ветвления по минимуму нижней границы (максимуму верхней для задачи на максимум целевой функции) и разбивается соответствующее ей подмножество вариантов. В ходе ветвления используется, если это возможно, тот или иной способ отсечения ветвей и вершин. Процесс выбора вершин и ветвления повторяется для всех висячих вершин, т.е. еще не вариантов решения задачи.

2) Разбиение множества вариантов по методу «в глубину с возвращением», т. е. последовательное построение ветвей. Строят полностью одну ветвь дерева решений, т.е. находят один вариант решения задачи. Полученное для этого варианта значение целевой функции может использоваться как отсекающая оценка. Далее ветвление выполняется последовательно от построенной ветви, начиная с вершины, предшествовавшей конечной. При этом, если не происходит отсечение, новая ветвь достраивается до конца. Процедура повторяется по отношению к вершинам новой ветви. После построения всех возможных ветвей, проходящих через вершину, сопоставленную множеству путей (размещений), отыскивается следующая вершина и процесс повторяется. При последовательном способе ветвления построение дерева решений может выполняться, начиная с левой ветви – слева направо, или с правой – справа налево. Количество построенных вершин дерева решений зависит от очередности развития ветвей.

3) *Комбинация двух рассмотренных способов.* Простейшая комбинация заключается в следующем. Строится одна ветвь. Далее развитие

дерева решений происходит по методу «в ширину», полученное опорное решение используется для отсечения, выбор вершины ветвления выполняется по минимуму нижней границы в задачах на минимум целевой функции (по максимуму верхней).

В ходе решения, если возможно, уточняется значение отсекающей оценки. Оптимальное решение будет найдено, когда в задаче на минимум целевой функции значение для некоторой конечной вершины меньше нижней границы F_H для всех висячих вершин и меньше значения целевой функции F для всех остальных конечных вершин (в задаче на поиск максимума целевой функции соответственно «больше»).

Рассмотрим способы вычисления нижних границ функционала при частичных размещениях q . Обозначим первый член в выражении $F(p)$ через $w(p)$, а второй через $v(p)$:

$$F(p) = w(p) + v(p) , \quad (1.16)$$

Нижняя граница для $F(p)$ получена суммированием нижних границ для $w(p)$ и $v(p)$. Обозначим их соответственно b_F , b_w и b_v .

Для вычисления b_w воспользуемся следующим свойством: если $r=(r_1, r_2, \dots, r_m)$ и $d=(d_1, \dots, d_2, \dots, d_m)$ – два вектора, то минимум скалярного произведения r и d , т.е. минимум $\sum_{i=1}^m r_i d_{p(i)}$ на множестве всех перестановок p , соответствует расположению составляющих вектора r в возрастающем порядке, а составляющих вектора d в убывающем.

Нижняя граница для $F(p)$

$$b_F = b_w + b_v , \quad (1.17)$$

должна быть соотнесена начальной вершине дерева решений.

Пусть теперь имеется некоторое частичное размещение q , определяющее назначение элементов из множества E_k в позиции из множества L_k . Тогда можно легко рассчитать вклад в общую длину соединений уже размещенных элементов и вычислить новую нижнюю границу соединений.

Запишем (1.14) для $F(p)$ в следующем виде:

$$F(p) = F(q) + \sum_{i \in E'_k} \sum_{s \in E'_k} r_{is} d_{q(i)p(s)} + \sum_{i \in E'_k} \sum_{s \in E'_k} r_{is} d_{q(i)p(s)} + \sum_{i \in E'_k} a_{ip(i)}, \quad (1.18)$$

где E'_k – множество не размещенных элементов для частичного размещения $q \subset p$.

Структуру (1.18) можно записать в компактной форме:

$$F(p) = F(q) + w_q(p) + v_q(p) + u_q(p), \quad (1.19)$$

где $F(q)$ – длина межсоединений размещённых элементов;

$w_q(p)$ и $v_q(p)$ аналогичны членам выражения (1.16) и соответствуют межсоединениям неразмещенных элементов;

$u_q(p)$ представляет собой суммарную длину соединений размещенных элементов с неразмещенными.

Поскольку $F(q)$ полностью определяется частными размещениями q , для вычисления новой нижней границы для $F(p)$ необходимо рассчитать нижние границы для остальных членов выражения (1.19). Границы для $w_q(p)$ и $u_q(p)$ рассчитываются по матрицам R и D , а нижняя граница для $v_q(p)$ определяется решением задачи линейного назначения неразмещенных элементов E'_k в незанятые позиции L'_k .

Выбор направления поиска в дереве решений может быть организован различными способами. Как правило, очередное ветвление производится из вершины дерева с наименьшим значением нижней границы.

Далее поиск продолжается в соответствии с общей схемой метода ветвей и границ. Процесс заканчивается тогда, когда в просмотренной части дерева решений отсутствуют вершины, для которых нижняя граница меньше, чем у наилучшего из известных решений. Это решение может быть либо получено одним из приближенных алгоритмов, и в этом случае доказывается оптимальность этого решения, либо определено в процессе поиска путем завершения некоторого частичного размещения q . В последнем случае в дереве решений определяется путь, соответствующий оптимальному размещению.

Расчет нижних границ при частичных размещениях осуществляется с целью усечения дерева решений. Естественно, что чем точнее эти границы, тем существеннее усечение. В связи с этим укажем более точный способ вычисления нижней границы для $F(p)$.

Пусть r_i – вектор–строка матрицы R без элемента $r_{ii}=0$, а d_j – вектор–строка матрицы D без элемента $d_{jj}=0$. При назначении элемента e_i в позицию l_j минимальная длина его связей с остальными элементами равна минимуму скалярного произведения векторов r_i и d_j . Обозначим его через $r_i \cdot d_j$. образуем матрицу $A' = \|a'_{ij}\|_{n \times n}$, в которой элемент a'_{ij} представляет оценку назначения элемента e_i в позицию l_j :

$$a'_{ij} = r_i \cdot d_j + a_{ij}, \quad (1.20)$$

где a_{ij} – элемент линейной части $v(p)$ функционала (1.14) или (1.15).

Теперь нижняя граница для $F(p)$ может быть определена решением задачи линейного назначения для матрицы A .

Распространение этого способа на вычисление границ при частичном размещении q производится путем образования матрицы $A'_k = \|a'_{ij}\|_{(n-k) \times (n-k)}$ для неразмещенных элементов из множества E'_k .

1.3 Решение задачи размещения с учетом тепловых характеристик с помощью современных САПР

Любая система проектирования печатных плат представляет собой сложный комплекс программ, обеспечивающий сквозной цикл, начиная с прорисовки принципиальной схемы и заканчивая генерацией управляющих файлов для оборудования изготовления фотошаблонов, сверления отверстий, сборки и электроконтроля. Однако не все САПР имеют в своем арсенале подсистему решающую задачу теплового анализа.

Таблица 2 – Коммерческие САПР учитывающие тепловые характеристики

Тип САПР	Фирма – разработчик	Продукты
САПР электронных устройств	Cadence Design Systems, Inc (США)	Sigrity.
	Mentor Graphics, Inc (США)	HyperLynx Thermal; FloTHERM PCB; FloEFD
	ANSYS	ANSYS Icepak

HyperLynx Thermal. Увеличение температуры компонента на каждые 10 °С после 100 °С уменьшает его среднее время наработки на отказ на целых 50% [2]. Для решения этой проблемы в процессе проектирования печатных плат продукт компании *Mentor Graphics HyperLynx Thermal* позволяет анализировать температуру печатной платы, размещённых на ней компонентов, температуры переходов, а также фиксировать превышение допустимых температур компонентов. Возможен анализ распределения тепла на поверхности частично или полностью разведённой печатной платы. Продукт предназначен для предварительного расчёта тепловых характеристик печатной платы в режиме топологической модели.

Импорт данных осуществляется из наиболее распространённых САПР печатных плат. Осуществлена интеграция с *Mentor Graphics Xpedition Enterprise*, что позволяет проектировщику перенести модель и начать тепловой

анализ сразу же после разработки топологии. Так же возможно чтение форматов *ODB++*, *IDF*, *PCB*.

В *HyperLynx Thermal* присутствуют встроенные библиотеки материалов и электронных компонентов с обширным набором параметров. Пользователь может создавать свои библиотеки и дополнять существующие в случае, если этого требует проект, а также редактировать параметры компонентов, размещённых на плате, изменять их расположение и устанавливать элементы охлаждения, например, тепловые трубки, радиаторы или металлизированные контактные площадки.

В качестве исходных данных для компонентов задаются: координаты; геометрические размеры; тип корпуса; количество, размеры и материал выводов; толщина и материал зазора между компонентом и платой; рассеиваемая мощность; тепловое сопротивление переход/корпус; максимальные температуры корпуса и перехода.

Для окружающей среды задаётся температура, давление, гравитация, скорость и направление воздушного потока, параметры корпуса.

Моделирование проводится для основных механизмов теплопередачи: конвекции, теплопроводности, излучения без возможности выбора пользователем отдельных механизмов. Вычисления производятся с использованием разностных схем, рассчитанных с помощью адаптивных локально переизмельчаемых сеток. Это позволяет получить быстрые, но точные результаты. Точность расчётов в пределах +/- 10% [3]

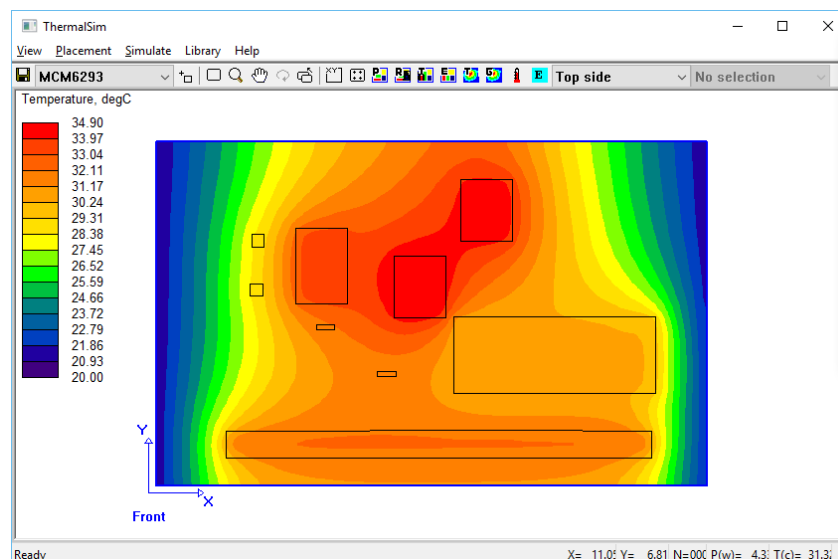


Рисунок 8 – Цветовая карта распределения температур в HyperLynx Thermal

Refdes	Package Temperat...	Junction Temperature	Dissipation
U-001	30,3	44,6	0,475
U4	34,9	34,9	0,546
U2	34,5	34,5	0,546
U1	33,5	33,5	0,667
R2	29	29	0,027
R1	28,9	28,9	0,027
J1	32,3	32,3	2,05
C2	31,6	31,6	0
C1	31,8	31,8	0

Рисунок 9 – Таблица численных значений температур в HyperLynx Thermal

Результаты выводятся в виде цветowych карт (см. рисунок 8) и численных значений температур (см. рисунок 9), что даёт пользователю наглядное представление о возможном перегреве элементов.

FloTHERM PCB. Продукт компании *Mentor Graphics FloTHERM PCB* представляет собой программный пакет из линейки *FloTHERM*, предназначенный для решения задач теплового моделирования на уровне печатной платы с возможностью задания параметров окружающей среды.

Служит в качестве общей платформы между инженерами по электронике и инженерами по механике для быстрой тепловой оценки предлагаемых макетов печатных плат [4].

Импорт исходных данных, содержащих геометрию платы и размещение компонентов осуществляется из наиболее распространённых САПР печатных плат в таких форматах, как *IDF*, *CSV* и т. д.

Встроенные библиотеки содержат разные виды корпусов, математических моделей компонентов (2R, детализированные), материалов, элементов охлаждения от разных производителей. Библиотеку можно дополнять пользовательскими данными. В случае необходимости использования детализированной модели отдельного компонента, не включенного в стандартную библиотеку, можно использовать встроенный модуль *FloTHERM PACK*, который после задания основных параметров электронного компонента, генерирует детальную модель и экспортирует её в *FloTHERM PCB* для дальнейшего расчёта.

Возможна установка и настройка вентиляторов, радиаторов, металлизированных контактных площадок для улучшения отвода тепла от компонентов, температура которых после проведения расчёта превышает допустимую.

В качестве исходных данных для компонента задаются: координаты; геометрические размеры; сторона платы; материал (выбирается из библиотеки готовых или создаётся вручную); температура плавления корпуса (вручную или автоматически); рассеиваемая мощность; тип математической модели (Simple или 2Resistor). Для 2R модели дополнительно задаются сопротивления переход–корпус и переход–плата, а также максимальные температуры корпуса и перехода. Для основных корпусов есть возможность выбрать детальную модель из библиотеки.

Для окружающей среды задаётся тип (принудительный обдув, неподвижный воздух, простой слот, кондуктивное охлаждение, плавильная печь). Для каждого типа окружения есть свой список параметров: температура

окружающего воздуха, скорость и направление потока, направление вектора гравитации, высота над уровнем моря и т. д.

При расчёте используется метод конечных элементов. Сетка генерируется автоматически двумя вариантами: *Thermal Trend Prediction* (меньше ячеек, приблизительный расчёт) и *Accurate Thermal Prediction* (больше ячеек, более точный расчёт).

Возможно моделирование для основных видов теплопередачи: кондукции, конвекции, излучения. Расчёт зависит от выбора типа окружающей среды.

Выходными данными являются тепловые карты (рисунок 10), таблицы температур корпусов и платы. Можно сгенерировать отчёт в формате *.html со всеми заданными параметрами и полученными результатами.

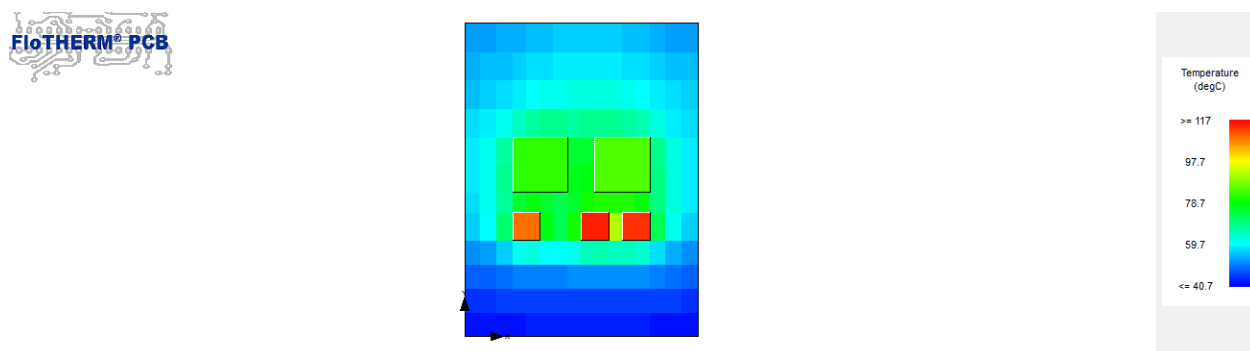


Рисунок 10 – Цветовая карта распределения температур в FloTHERM PCB

Особенностью является то, что данный модуль существует как отдельно *FloTHERM PCB* с возможностью непосредственного расчёта и вывода результатов, так и в форме модуля *FloEDA Bridge*, встроенного во *FloTHERM*, который необходим для создания или импорта геометрии, задания основных параметров модели и передачи во *FloTHERM* для дальнейшего расчёта геометрической модели, например, при помещении в корпус.

ANSYS Icepak. Специализированное программное средство, производства компании ANSYS, предназначенное для разработки электронных устройств и корпусов для них. Обеспечивает надёжную и мощную

вычислительную технологию гидрогазодинамики для анализа и управления температурным режимом электронных устройств [5]. Продукт обладает интерактивным интерфейсом со смарт-объектами для быстрого создания модели устройства. Для ускорения создания модели, можно импортировать данные из наиболее распространённых САПР печатных плат, таких как *Altium*, *Cadence*, *Dassault*, *Mentor Graphics*, *PTC* и *Zuken* в следующих форматах: *CSV*, *IGES*, *STEP*, *TETIN*, *IDF*, *ODB++*.

Программное обеспечение *ANSYS Icepak* содержит множество функций, которые позволяют быстро создавать и моделировать системы охлаждения интегральных схем, печатных плат и устройств в корпусе. Модели создаются простым перетаскиванием иконок предопределённых объектов из встроенных библиотек, содержащих шкафы, вентиляторы, печатные платы, корпуса микросхем, радиаторы и т. д. Такие смарт-объекты несут в себе информацию о геометрии, свойствах материала, параметрах закреплений и граничных условиях, которая необходима для проведения анализа и оптимизации конструкции.

На основе импортированных данных можно извлечь подробные карты теплопроводности для печатных плат и корпусов компонентов. Для плат теплопроводность может быть определена из рисунка трассировки и информации о слоях платы. Это позволяет получить точное представление о теплопроводности в самой плате и повысить точность расчёта.

Для решения применяется метод конечных элементов. Используются передовые алгоритмы для автоматической генерации сеток, которые представляют истинную форму электронных компонентов. Пользователь имеет возможность настраивать плотность сетки, что позволяет подобрать оптимальное время решения с минимальным ущербом точности расчёта.

Решение производится с учётом всех основных механизмов теплообмена: кондукции, конвекции и излучения как для стационарных, так и для переходных режимов. Решатель использует мультисеточную схему для ускорения сходимости при решении задач теплопереноса. Он имеет полную

гибкость сетки и решает даже самые сложные электронные сборки, используя неструктурированные сетки, обеспечивая надёжное и быстрое решение.

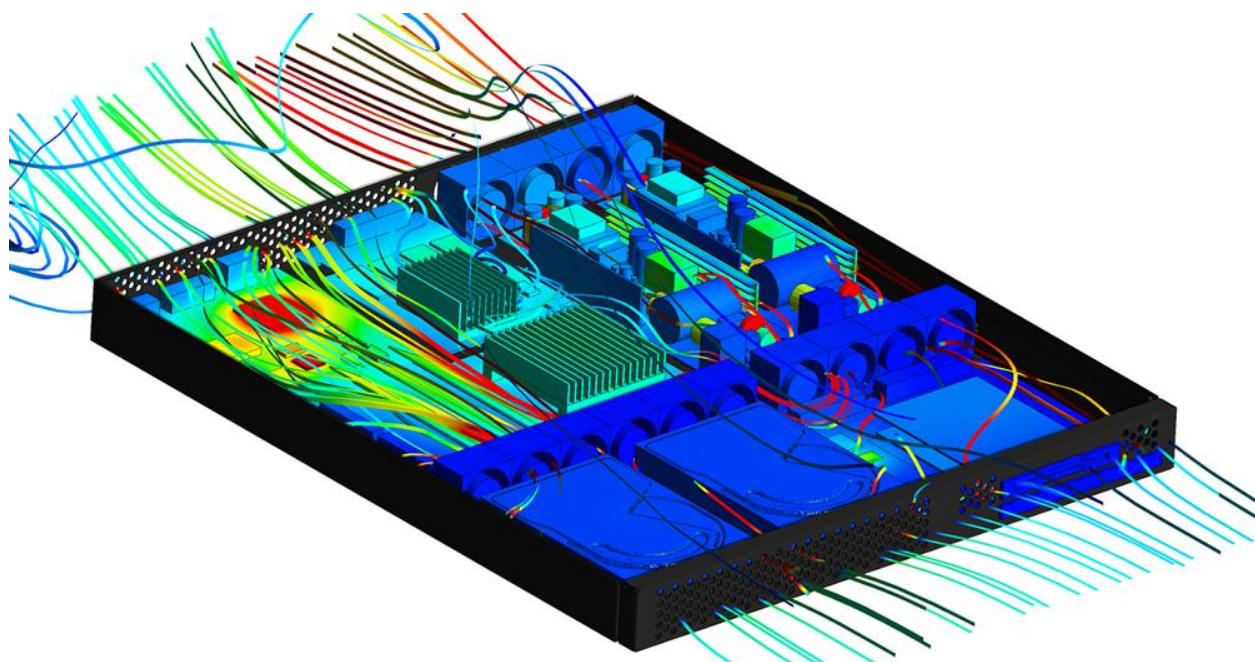


Рисунок 11 – Цветовая карта распределения температур в ANSYS Icepak

Встроен полный набор инструментов постобработки для создания графиков, анимации, отчётов, представляющих результаты моделирования. На рисунке 11 показана карта распределения температур и векторов потоков воздуха в корпусе устройства.

Cadence Sigrity PowerDC. Решение от компании *Cadence* предлагает совместную температурную и электрическую симуляцию. Учитывает не только нагрев шин питания из-за протекающего по ним тока, но и изменение свойств материалов вследствие нагрева. Это позволяет получить достоверную картину тепловых распределений на печатном узле, интегральных схемах и радиаторах, с учетом параметров окружающей среды, скорости обдува и свойств радиатора.

Импорт данных из популярных САПР: *Cadence, Mentor, Altium, PCAD* и в формате *ODB++*

Для проведения расчёта применяется метод конечных элементов.

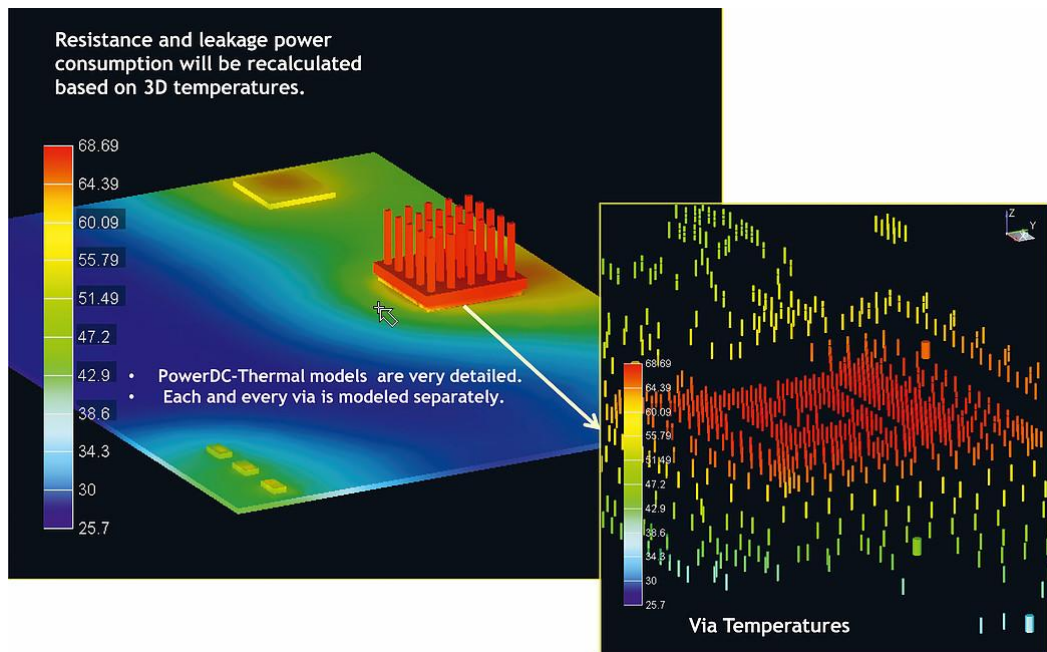


Рисунок 12 – Цветовая карта распределения температур в Sigrity PowerDC

Позволяет моделировать проводники, выводы (в том числе шариковые), переходы, медные и диэлектрические слои, температурные свойства и зависимости материалов, радиаторы, термальные слои, теплостоки [6]. На рисунке 12 показана возможность отображения температур в сквозных переходных отверстиях.

FloEFD от компании *Mentor Graphics* представляет собой полностью встраиваемый в системы CAD пакет для решения задач гидрогазодинамики и теплообмена. Совместно с модулем *FloEFD Electronics Cooling Module* образует специализированный инструмент для расчёта тепловых режимов электронных устройств.

Присутствуют встроенные библиотеки материалов, корпусов, элементов охлаждения. Используется надежный и полностью автоматический генератор расчетных сеток, автоматический контроль сходимости решения.

Особенности: интеграция в основные CAD системы позволяет исключить необходимость переобучения инженера, так как используется привычный ему интерфейс. Так же, *FloEFD* работает непосредственно с моделью системы, в которую интегрирован, что избавляет от необходимости конвертации геометрии в другой формат, иногда приводящей к потере качества модели.

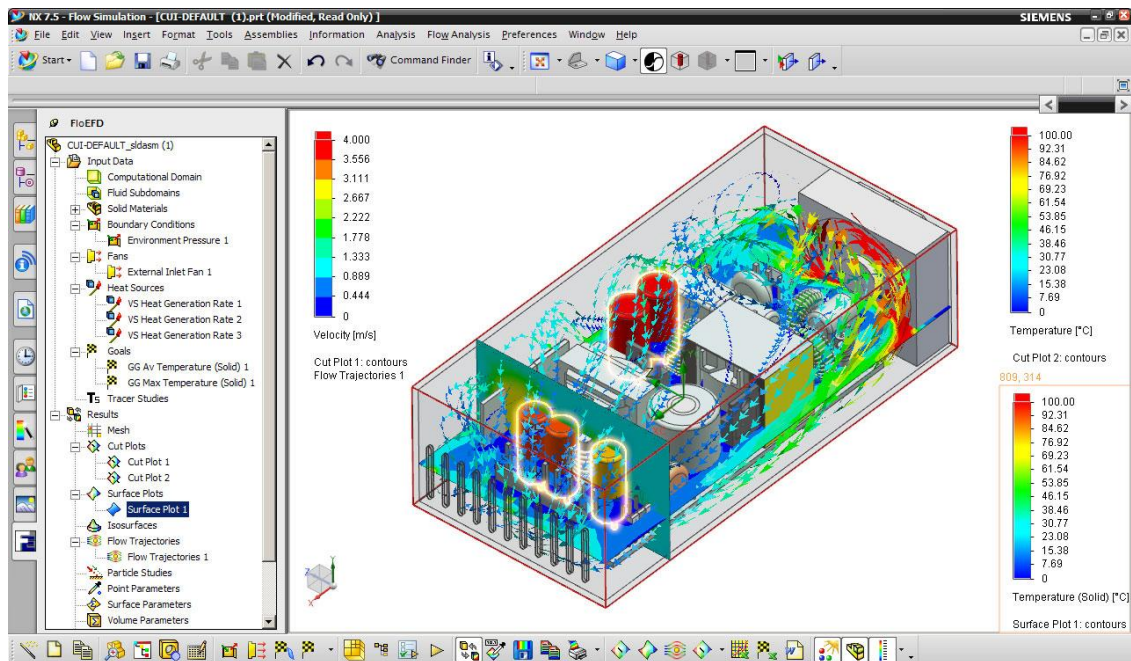


Рисунок 13 – Цветовая карта распределения температур в FloEFD

FloEFD способен рассчитывать постоянный ток в проводниках: джоулево тепло автоматически рассчитывается и включается в расчёты теплопереноса; расчёты электрического потенциала и тока выполняются только в проводящих материалах; электрическое сопротивление материала может быть изотропным, анизотропным или зависящим от температуры [7].

Подсистемы САПР обеспечивают получение законченных проектных решений, которые можно разделить на следующие этапы конструкторского проектирования: Синтез конструкций (компоновка, размещение и трассировка); контроль полученных решений (анализ помехоустойчивости, анализ тепловых характеристик, анализ механических характеристик); Оформление документации конструкторской (КД) и технологической (ТД), включая выпуск носителей информации.

Идеальная САПР предполагает такой порядок работ, когда задача синтеза конструкций и контроля полученных решений работают в одной подсистеме, решая задачу глобальной оптимизации. К сожалению, это не так, и в изложенных выше САПР рассматриваемые нами процессы размещения и теплового анализа выполняются в разных подсистемах, и, в таком случае, не

могут решить задачу оптимального назначения элементов на печатной плате, а значит, что в процессе проектирования печатной платы инженеру–проектировщику приходится поддерживать обратную связь с другими подсистемами, решающими предыдущие этапы конструкторского проектирования, в целях получения приемлемого результата проектирования.

Объединить задачи размещения и теплового анализа в одном цикле решения задач можно созданием промежуточного узкоспециализированного продукта, что сократит время инженера–проектировщика на конструирование печатной платы.

2 Решение задачи размещения с оптимизацией тепловых режимов

2.1 Постановка задачи теплового анализа

Микроминиатюризация конструкций радиоэлектронных устройств приводит к тепловой нагруженности, связанной с перегревом элементов и самого печатного узла. В этом случае печатную плату можно рассматривать в качестве теплового источника, в котором электрическая энергия преобразуется в тепловую. Элементы на печатной плате в нагруженном состоянии, таким образом, представляют источники, стоки и приемники тепловой энергии. По известным данным К.П.Д. преобразования полезного электрического сигнала составляет единицы процента, т. е. РЭС можно рассматривать в качестве нагревателя, имеющего большой К.П.Д.

Увеличение температуры элементов печатной платы связано, с одной стороны, с повышением температуры окружающей среды, а с другой стороны, с выделением тепла за счет рассеиваемой элементами во время их работы мощности. Если в устройствах на электронных лампах и полупроводниках 1 см^2 площади рассеивает в определенных условиях мощность $0,03 - 0,5 \text{ Вт}$, то в современной микроэлектронной аппаратуре это рассеивание обычно достигает $6 - 60 \text{ Вт} / \text{см}^2$. Поэтому размещение элементов должно происходить только с учетом теплового анализа печатной платы.

Для расчета тепловых моделей элементов ПП принимаем, что температура воздуха в каналах, между которыми расположена печатная плата, известна. Элементы ПП обмениваются лучистой энергией с соседними элементами ПП. Охлаждение элементов ПП происходит также путем конвекции в воздушной среде в каналах между платами. Часть теплоты, выделяемой элементом ПП, нагревает печатную плату.

Основные допущения тепловой модели:

– Плата представляет собой анизотропную пластину, тепловое поле которой изменяется по двум координатам, перепадом по толщине пренебрегаем;

– Температура любого элемента ПП есть среднеповерхностная температура его корпуса;

– Тепловой поток от элемента ПП к плате равномерно распределен по площади его крепления;

– Шаг сетки печатной платы по оси x и по оси y равны.

Теплота, выделяемая элементом ПП, уносится воздухом и лучистым потоком, а также передается другим элементам ПП посредством теплопроводности материала печатной платы. Поэтому для нахождения параметров тепловой связи между элементами ПП необходимо учитывать: кондукцию (теплопроводность), конвекцию и лучеиспускание (радиацию).

Для произвольно выбранного элемента ПП можем записать уравнение расчета тепла:

$$P = P_{cond} + P_{luch} + P_{conv} , \quad (2.1)$$

где P_{cond} – количество тепла, переданного через ПП ($Вт$);

P_{conv} – количество тепла, рассеянного при помощи конвекции ($Вт$);

P_{luch} – количество теплоты излучаемым одним элементом ПП по отношению к другому ($Вт$);

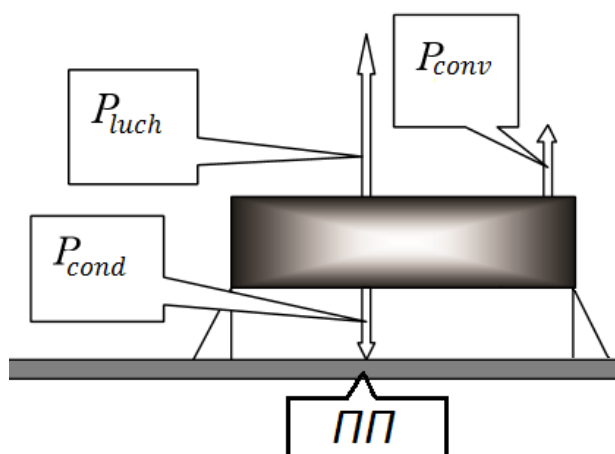


Рисунок 14 –Тепловые потоки элемента печатной платы

Теплопроводность представляет процесс передачи тепловой энергии внутри твердого тела или неподвижной жидкости (газа) от участков с более высокой температурой к участкам с более низкой температурой или же при соприкосновении двух твердых тел с различной температурой.

Процесс теплопроводности связан с понятиями температурного поля и градиента температуры. В общем виде температура θ является функцией координат трехмерного пространства x, y, z и времени t .

$$= f(x, y, z, t), \quad (2.2)$$

Температурное поле можно рассматривать как совокупность значений температуры для всех точек рассматриваемого пространства в определенный промежуток времени.

Для одномерного пространства уравнение (2.2) можно представить в виде:

$$= f(x), \quad (2.3)$$

Геометрическое место точек, имеющих одинаковую температуру, образует изотермическую поверхность. Поскольку в одной и той же точке

пространства не может быть двух различных значений температур, то можно сказать, что изотермические поверхности не пересекаются друг с другом – все они либо кончатся на границе тела, либо замыкаются на себе. Поэтому изменение температуры в теле может иметь место лишь в направлениях, пересекающих изотермические поверхности (рисунок 15).

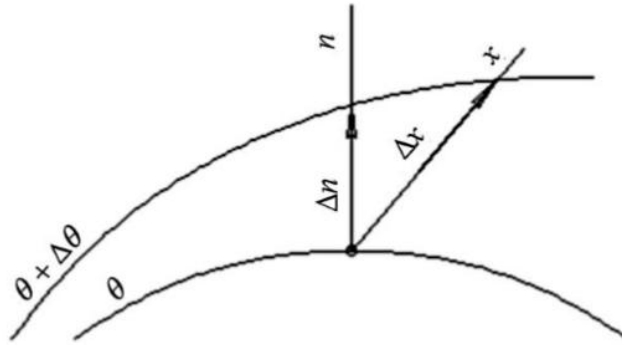


Рисунок 15 – Изотермическая поверхность и температурный градиент

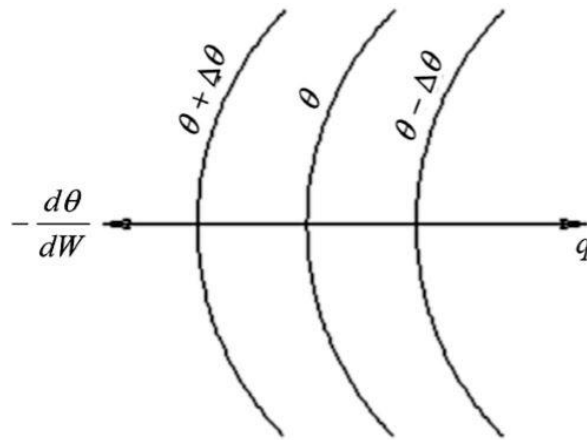


Рисунок 16 – Температурный градиент и удельный тепловой поток

Наибольшее изменение температуры θ имеет место в направлении нормали n к изотермической поверхности.

$$\lim (\Delta\theta / \Delta n)_{\Delta n \rightarrow 0} = \frac{\partial\theta}{\partial n} = grad\theta = \nabla\theta ,$$

(2.4)

где ∇ – математический символ градиента.

Температурный градиент является вектором, имеющим направление по нормали по отношению изотермической поверхности в сторону возрастания температуры. Передача же тепла ведется только в направлении уменьшения значения температуры. Обозначим количество тепла, передающегося через произвольную поверхность в единицу времени, через Q , а удельный тепловой поток – q . Удельным тепловым потоком называют количество тепла (тепловой поток), отнесенное к единице поверхности. Плотность теплового потока совпадает с направлением тепла в данной точке тела и противоположно направлению вектора теплового градиента (см. рисунок 16).

Основной закон теплопроводности носит название закона Фурье:

$$q = -\lambda * grad\theta = -\lambda * \Delta\theta , \quad (2.5)$$

где λ – коэффициент теплопроводности, $Вт/м \times ^\circ C$.

Коэффициент λ характеризует способность данного материала проводить тепло.

$$\lambda = -\frac{|q|}{grad\theta} = \frac{P}{S*t*\Delta\theta/\Delta x} , \quad (2.6)$$

где P – тепловой поток;

S – площадь;

t – время;

$\Delta\theta$ – изменение температуры;

Δx – расстояние между рассматриваемыми изотермическими поверхностями.

Так для сухого воздуха при $\theta = 20^\circ C$ и нормальном давлении $\lambda=0,0276$ $Вт/м \times ^\circ C$, для текстолита $\lambda=0,27$ $Вт/м \times ^\circ C$.

Величина λ зависит от температуры. Для большинства металлов с повышением температуры λ убывает.

Распределение тепла может происходить в стационарном режиме, при котором температурное поле не меняется во времени, и в нестационарном режиме, когда температурное поле зависит от времени.

Для стационарного режима величину теплового потока P , прошедшую по плоской печатной плате можно найти из формулы (2.7).

$$P_{cond} = \lambda \cdot S_p \cdot \Delta\theta / \delta, \quad (2.7)$$

где λ – коэффициент теплопроводности материала ($Вт/м \times ^\circ C$);

S_p – площадь пересечения материалов, через которую проходит тепло ($м^2$);

$\Delta\theta$ – разность температур элементов ПП ($^\circ C$);

δ – расстояние, которое проходит тепло (м), $\delta = h_{н.н.}$

Для определения температурного поля печатной платы применена двухмерная равномерная прямоугольная сетка (рисунок 17).

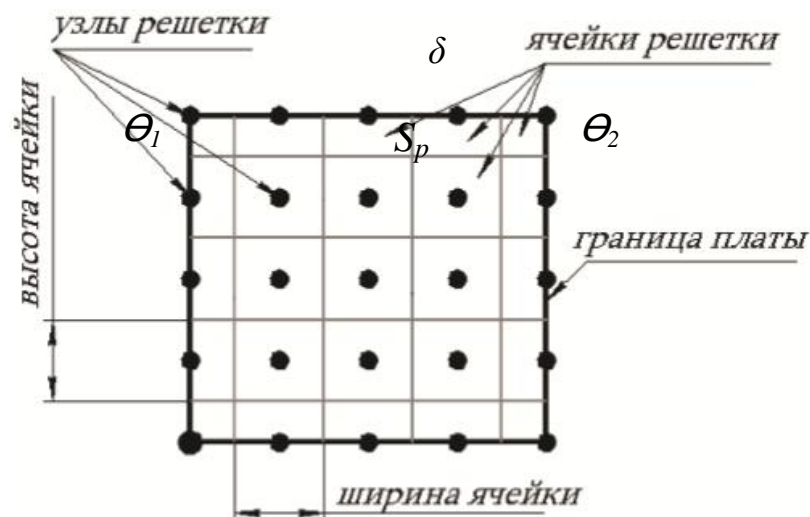


Рисунок 17 – Двухмерная равномерная прямоугольная сетка элементов

Будем считать, что расстояние, которое проходит тепло, исходит из геометрических центров конструктивных элементов.

Конвекция – это передача за счет движения потоков жидкостей или газов. Обычно в печатных платах это передача тепла от ЭРЭ в окружающую среду (как правило, воздух). Существует два варианта конвекции: естественная и принудительная. При естественной конвекции тепло передается за счет уже существующих потоков воздуха, вызванных перепадом температур. В принудительной конвекции движение потоков жидкости или газов создается за счет дополнительных устройств, таких как вентилятор, насос и т.п.

Количество тепла, рассеянного при помощи конвекции в стационарном режиме, можно рассчитать по формуле:

$$P_{conv} = h \cdot S_v \cdot \Delta\theta_o, \quad (2.8)$$

где h – коэффициент теплопередачи ($Bm/m^2 \times ^\circ C$);

S_v – площадь поверхности излучающего элемента (m^2);

$\Delta\theta_o$ – разница между температурой излучающего элемента и температурой окружающей среды ($^\circ C$).

Тепловое излучение – процесс переноса теплоты, обусловленный превращением внутренней энергии вещества в энергию излучения, переносом её в виде электромагнитных волн и поглощением веществом. Определяется только температурой и оптическими свойствами излучающего тела.

По спектру различают монохроматическое и сложное излучения. Если излучение происходит в узком интервале длин волн от λ до $\lambda + d\lambda$, то оно называется монохроматическим и у параметров, характеризующих монохроматическое излучение, ставится индекс λ или ν . Сложным (интегральным) называется излучение во всём диапазоне длин волн $0 \leq \lambda \leq \infty$. Таким образом, в отличие от других механизмов теплообмена, лучистая энергия имеет не только количественную, но и качественную (спектральную) характеристику.

Количество теплоты излучаемым одним телом по отношению к другому, если поверхности теплообмена этих тел являются параллельными и разделенными прозрачной средой, определяется

$$P_{луч} = c_0 \cdot A_{пр} \cdot S_v * \left[\left(\frac{\Delta T}{100} \right)^4 \right] \quad , \quad (2.9)$$

где $A_{пр}$ – приведенная степень черноты тел, участвующих в теплообмене;

c_0 – постоянная Стефана–Больцмана ($5,67 \text{ Вт}/(\text{м}^2 \times \text{°C}^4)$).

В соответствии с этими механизмами передачи теплоты между телами и средой различают три вида тепловых сопротивлений:

Кондуктивное:

$$R_{\text{Т.КД}} = \int_{x_1}^{x_2} \frac{\delta}{\lambda S_p} \quad , \quad (2.10)$$

где x_1 и x_2 – расстояние от начала отсчета изотермических поверхностей;

δ – элемент длины пути теплового потока (м);

S_p – площадь изотермической поверхности в аналитическом выражении.

Конвективное:

$$R_{\text{Т.КВ}} = \frac{1}{S_v \alpha_k} \quad , \quad (2.11)$$

где S_v – площадь поверхности, с которой осуществляется конвекция;

α_k – коэффициент теплоотдачи путём конвекции.

Лучистое:

$$R_{\text{Т.Л}} = \frac{1}{S_v h} \quad , \quad (2.12)$$

где S_v – площадь поверхности тела, излучающего энергию;

h – коэффициент теплоотдачи излучением.

Тепловое сопротивление (R_m) ($^{\circ}C/Wm$) – это термин, используемый для описания способности корпуса микросхемы передавать тепло от кристалла к окружающей среде. Устройства с меньшим тепловым сопротивлением отдают тепло лучше. R_m измеряется в $^{\circ}C/Wm$, и для инженеров является инструментом, связывающим электрическую мощность (рассеиваемую) в ваттах с температурой в градусах Цельсия. [10, 11]

Зная тепловое сопротивление кондуктивное и конвективное, а также рассеиваемую мощность, можно вычислить $\Delta\theta$, используя следующее выражение:

$$\Delta\theta = R_m \cdot P_p \quad , \quad (2.13)$$

где P_p – рассеиваемая мощность элемента ПП (Wm).

Значения рассеиваемой мощности можно взять из паспортных данных элементной базы компонента.

2.2 Модификация метода ветвей и границ. Учет тепловых режимов

В данном разделе объединим учет тепловых моделей элементов и глобальную оптимизацию СДС в одной математической модели метода ветвей и границ (раздел 1.2.2), тем самым сформируем комплексный критерий оптимальности. Для этого необходимо изменить формулу расчета нижней границы $F(p)$ (1.19) метода ветвей и границ, добавив к нему расчет теплового коэффициента:

$$F(p) = F(q) + w_q(q) + v_q(p) + u_q(p) + K_{ij} \quad , \quad (2.14)$$

где K_{ij} – тепловой коэффициент.

Тепловой коэффициент получается сложением всех видов теплообмена элемента p_{ij} с соседними элементами печатной платы. Основная формула для расчета теплового коэффициента элемента ПП выглядит так:

$$K_{ij} = k \times \sum_{i=i-1}^{i=i+1} \sum_{j=j-1}^{j=j+1} Q_{cond} + Q_{luch} + Q_{conv}, \quad (2.15)$$

где k – весовой коэффициент учета тепла при расчете нижней границы (0 – 100%), необходим для регулировки влияния теплового коэффициента на нижнюю границу (2.14) метода ветвей и границ. Задается в интерфейсе пользователя.

Согласно формулам расчета теплопроводности (2.7), конвекции (2.8) и излучения (2.9) получим:

$$K_{ij} = k \times \sum_{i=i-1}^{i=i+1} \sum_{j=j-1}^{j=j+1} \left(\frac{\lambda \cdot S_p \cdot \Delta_3}{\delta} \right) + c_0 \cdot A_{пр} \cdot S_v \cdot \left[\left(\frac{\Delta_3}{100} \right)^4 \right] + (h \cdot S_v \cdot \Delta_0), \quad (2.16)$$

Подставив формулы (2.10) и (2.11) в формулу (2.13) и убрав все константы, мы увидим, что $\Delta\theta$ и P_p имеют прямо пропорциональную зависимость, а это значит, что наших приближенных расчетах в формуле (2.16) можно принять $\Delta\theta \approx P_p$. Приведенную степень черноты тел, участвующих в теплообмене, для упрощения расчетов, возьмем равное единице, что означает что элементы печатной платы абсолютно черные. Так, основная формула теплового коэффициента в стационарном режиме имеет вид:

$$K_{ij} = k \times \sum_{i=i-1}^{i=i+1} \sum_{j=j-1}^{j=j+1} \left(\frac{\lambda \cdot S_p \cdot P_p}{\delta} \right) + c_0 \cdot S_v \cdot \left[\left(\frac{P_p}{100} \right)^4 \right] - (h \cdot S_v \cdot P_p), \quad (2.17)$$

где P_p – тепловая мощность, рассеиваемая элементом печатной платы ($Вт$);

При размещении элемента e подбирается такая позиция p на ПП, при которой критерий оптимальности (2.14) будет минимальным. Элементы в занятых позициях участвуют в расчетах теплового коэффициента при размещении соседних элементов ПП, тем самым нивелируется размещение сильно греющихся элементов в соседних позициях.

3 Технологические аспекты программной реализации

3.1 Выбор инструментария разработки программы

Хотя Windows API (Application Programming Interface, интерфейс программирования приложений) обладает всем необходимым для создания графического интерфейса пользователя, использование этих доступных «инструментов» требует больших затрат времени и практического опыта. Даже библиотеки, призванные облегчить процесс написания программ для ОС Windows, – такие как, например, MFC, не дают процессу создания программ той простоты и легкости как хотелось бы. И самый большой недостаток, связанный с применением таких библиотек, – это платформозависимость. [12]

Qt creator является платформонезависимым и не оставляет без внимания пользователей Microsoft Windows, Mac OS X, Linux, Solaris, AIX, Irix, NetBSD, OpenBSD, HP-UX, FreeBSD и других клонов UNIX с X11 и мобильных устройств базирующихся на IOS, Android, Windows Phone, Windows RT и BlaskBerry. В следствии чего на Qt значительно сокращается время разработки, поскольку не приходится писать код многократно – под каждую платформу, и, что не менее важно, отпадает необходимость знать специфику каждой из платформ, для которой пишется программа. В Qt можно использовать самые передовые инструменты для отладки и совершенствования, улучшения кода программ – например, абсолютно бесплатную интегрированную среду разработки XCode для Mac OS X. Стоит так же сказать, что у Qt также есть модули для удобной работы с XML, базами данных (с интеграцией системой MVC), OpenGL, аудио/видео-работы (Phonon), сетевого программирования, WebKit2. Для большинства задач, которые стоят перед среднестатистическим проектом — этого хватает в 90% случаев, а с модулями редко случаются проблемы.

Для ускорения и упрощения создания пользовательских интерфейсов Qt представляет программу Qt Designer, позволяющую делать это в интерактивном

режиме. Очень сильно повысить скорость создания пользовательских интерфейсов можно так же и при помощи технологии Qt Quick, модули и инструменты которой являются неотъемлемой частью Qt..

Несмотря на то, что библиотека Qt изначально создавалась для языка программирования C++, существуют модули для этой библиотеки во многих языках – например: Qt# на C#, PerlQt на Perl, PyQt на Python, PHP и т. д. Так же программы, реализованные с помощью Qt, могут использовать язык сценариев Qt Script.

Qt прекрасно документирована, благодаря чему с помощью программы Qt Assistant можно всегда подчерпнуть о ней любую интересующую вас информацию. А если и этого окажется не достаточно, то Qt является библиотекой с открытым исходным кодом (Open Source), и всегда можно взглянуть в него и детально разобраться в том, как работает та или иная часть этой библиотеки. Также стоит отметить наличие форумов, таких как <https://forum.qt.io> и <http://www.cyberforum.ru/qt/>, где каждый желающий может получить ответ на волнующий его вопрос. Проверено мной что форумы очень живые и там реально отвечают на вопросы, возникающие, в процессе разработки приложений в Qt creator.

На сегодняшний день Qt – это продукт, широко используемый разработчиками всего мира. Компаний, ориентированных на эту библиотеку, более четырех тысяч. В число активных пользователей Qt входят такие компании, как: Adobe, Amazon, AMD, Bosch, Blackberry, Cannon, Cisco Systems, Disney, Intel, IBM, Panasonic, Pioneer, Philips, Oracle, HP, Goober, Google, NASA, NEC, Neonway, Nokia, Samsung, Siemens, Sony, Xerox, Xilinx, Yamaha и др.

Из вышеизложенного можно сделать вывод, что библиотека Qt creator, как никакое другое решение, подходит для наших задач, а именно, для написания нетривиальной, платформонезависимой программы на ПК.

3.2 Функциональное назначение программного продукта

Алгоритм размещения на основе метода ветвей и границ с учетом тепловых характеристик компонентов, размещаемых на печатной плате, был реализован программно на языке C++ в библиотеке Qt creator.

Основное назначение программы – это автоматизация решения задачи размещения элементов ПП с учетом их тепловых характеристик и требуемых технологических ограничений.

Основными потребителями информации программы являются инженеры–проектировщики. Инженер–проектировщик имеет возможность самостоятельно проводить автоматическое размещение элементов с учетом тепловых характеристик элементов, а также ручное размещение фиксированных элементов, местоположение которых заранее известно.

3.3 Разработка программного обеспечения

Разработка информационных систем включает в себя несколько этапов. Однако всегда начальным этапом создания системы является изучение, анализ и моделирование деятельности заказчика для возможного улучшения и оптимальных методов работы, которые и будут реализованы в создаваемом приложении.

Прежде чем решить эти проблемы и приступить к разработке системы необходимо иметь четкое описание методологии разработки, адаптированной к конкретному проекту. На основе выбранной методологии производится выбор конкретных проектных инструментов и программных средств.

Был использован комбинированный подход к проектированию. Это наиболее популярный на сегодняшний день способ формализации требований к системе и построению ее архитектуры. Его популярность обусловлена сочетанием преимуществ функционального и объектного подходов к проектированию: функциональный подход хорош на этапе выдвижения

требований и описания бизнес–процессов, а объектный – на этапе создания архитектуры системы, достаточно понятной для программиста, и дальнейшей реализации проекта в объектно–ориентированной среде программирования.

Под моделью ПО в общем случае понимается формализованное описание системы ПО на определенном уровне абстракции. Каждая модель определяет конкретный аспект системы, использует набор диаграмм и документов заданного формата, а также отражает точку зрения и является объектом деятельности различных людей с конкретными интересами, ролями или задачами. Графические (визуальные) модели представляют собой средства для визуализации, описания, проектирования и документирования архитектуры системы.

Поскольку сложность систем повышается, важно располагать хорошими методами моделирования. Хотя имеется много других факторов, от которых зависит успех проекта, но наличие строгого стандарта языка моделирования является весьма существенным. Состав моделей, используемых в каждом конкретном проекте, и степень их детальности в общем случае зависят от следующих факторов:

- сложности проектируемой системы;
- необходимой полноты ее описания;
- знаний и навыков участников проекта;
- времени, отведенного на проектирование.

Визуальное моделирование оказало большое влияние на развитие ТС ПО вообще и CASE средств в частности. Понятие CASE (Computer Aided Software Engineering) используется в настоящее время в весьма широком смысле. Первоначальное значение этого понятия, ограниченное только задачами автоматизации разработки ПО, в настоящее время приобрело новый смысл, охватывающий большинство процессов жизненного цикла ПО. CASE технология представляет собой совокупность методов проектирования ПО, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область, анализировать эту модель на всех стадиях

разработки и сопровождения ПО и разрабатывать приложения в соответствии с информационными потребностями пользователей. Большинство существующих CASE средств основано на методах структурного или объектно–ориентированного анализа и проектирования, использующих спецификации в виде диаграмм или текстов для описания внешних требований, связей между моделями системы, динамики поведения системы и архитектуры программных средств.

3.4 Структура программного продукта

Структура программного продукта может быть представлена при помощи UML. UML – это язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML–моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем.

Диаграмма классов является одной из форм статического описания системы с точки зрения ее проектирования, показывая ее структуру. Диаграмма классов не отображает динамическое поведение объектов, изображенных на ней классов. На диаграммах классов показываются классы, интерфейсы и отношения между ними.

Диаграмма классов разработанного программного продукта представлена на рисунке 18.

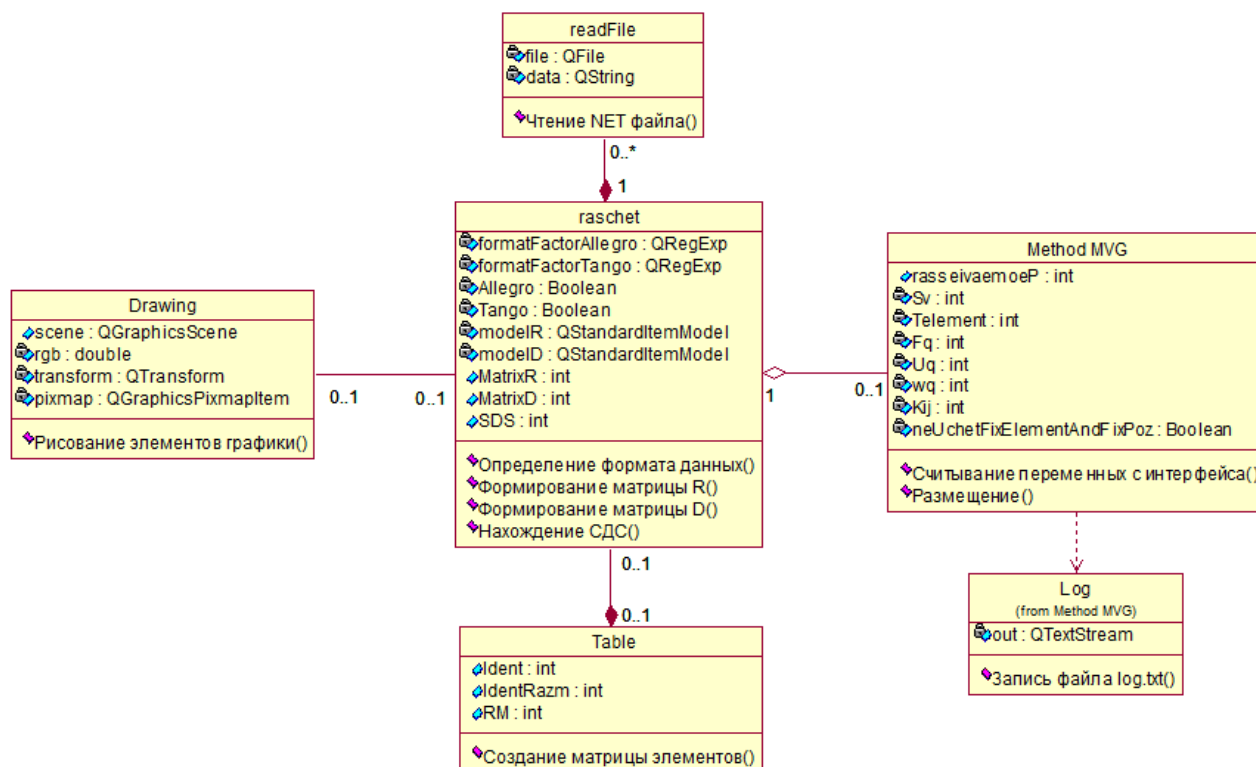


Рисунок 18 – Диаграмма классов

Диаграмма последовательности является одной из разновидности диаграмм взаимодействия и предназначена для моделирования взаимодействия объектов системы во времени, а также обмена сообщениями между ними.

Одним из основных принципов объектно–ориентированного программирования является способ информационного обмена между элементами системы, выражающийся в отправке и получении сообщений друг от друга. Таким образом, основные понятия диаграммы последовательности связаны с понятием «объект» и «сообщение».

На диаграмме последовательности объекты в основном представляют экземпляры класса или сущности, обладающие поведением. В качестве объектов могут выступать пользователи, инициирующие взаимодействие, классы, обладающие поведением в системе или программные компоненты, а иногда и системы в целом. На рисунке 19 представлена диаграмма последовательности демонстрирующая взаимодействие программы и пользователя, а на рисунке 20 взаимодействие входных данных с программными блоками.

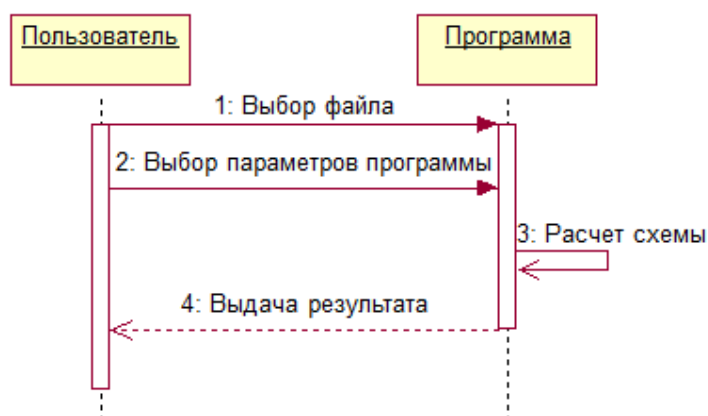


Рисунок 19 – Диаграмма последовательности со стороны пользователя

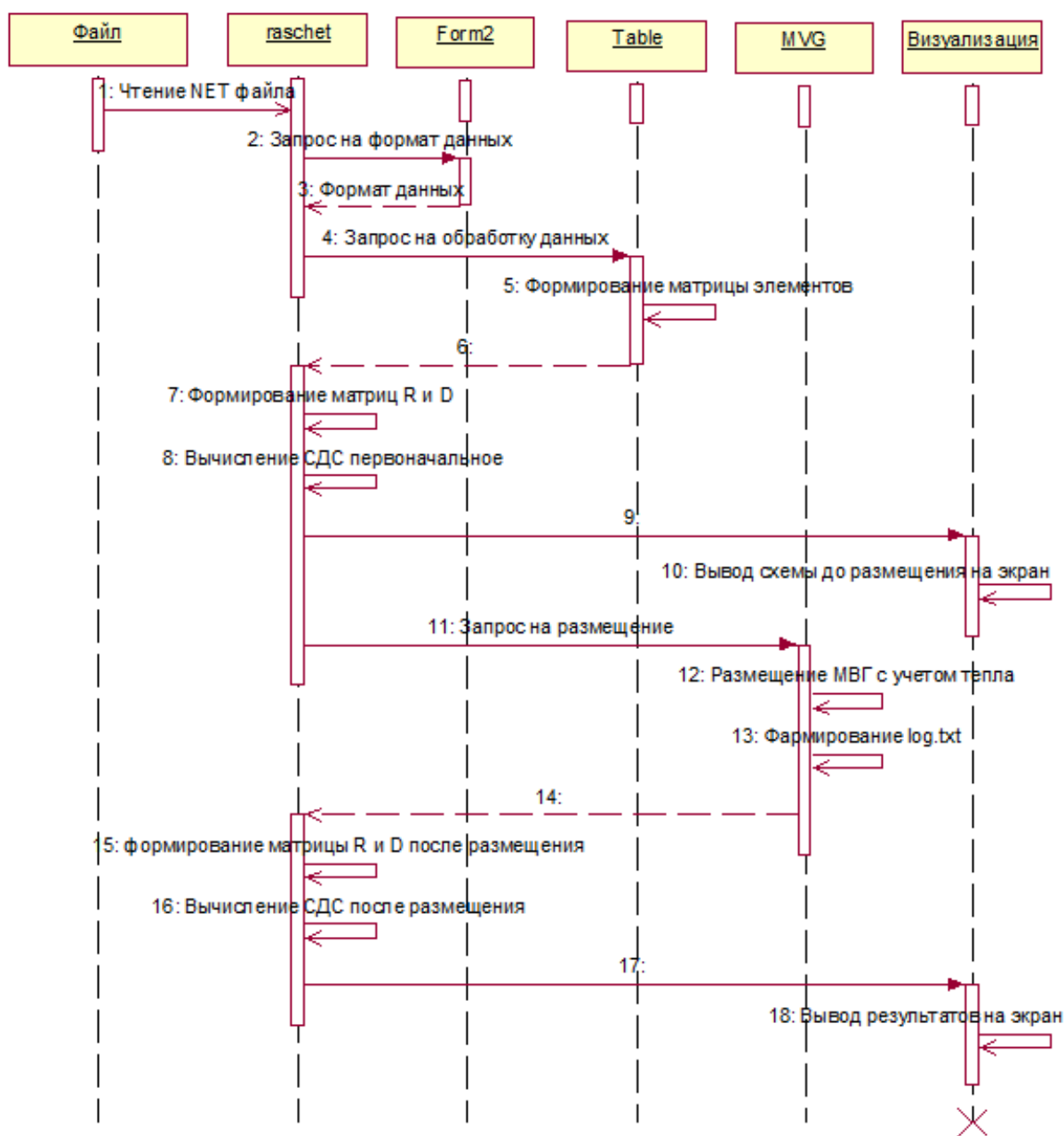


Рисунок 20 – Диаграмма последовательности перехода данных в программе

3.5 Формирование исходных данных

Исходными данными для разработанной программы являются коммутационные схемы в виде файла списка соединений формата .NET, сгенерированные в современных коммерческих САПР печатных плат: OrCAD; MentorGraphics; Altium Designer.

Файл .NET (Netlist file) – это текстовый файл, содержащий обозначения компонентов, псевдонимы узлов к которым подключены компоненты. Файл соединений располагается в директории САПР.

Вы можете использовать функцию импорта файла netlist (файла списков соединений) в представленной программе. Представленная программа поддерживает импорт в несколько популярных форматов, таких как Tango и Allegro.

Для создания файла списка соединений в САПР OrCAD необходимо:

1. Открыть проект и зайти в менеджер проекта;
2. Выделить файл принципиальной схемы, шкала команд станет активной;
3. Переходим по дереву команд Tools → Create Netlist (рисунок 21);
4. Выбираем вкладку Other;
5. В Formatters выбираем «orTango.dll» или «Allegro.dll» и нажимаем «ОК». (рисунок 22).

В дереве менеджера проекта появится файл списка соединений формата – .NET. Найти его можно в корне данного проекта.

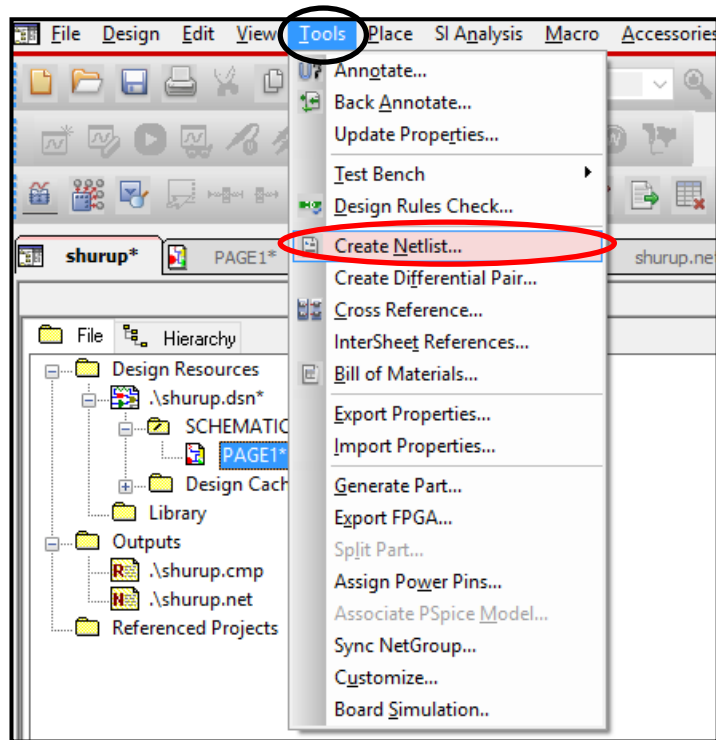


Рисунок 21 – Создание файла списка соединений

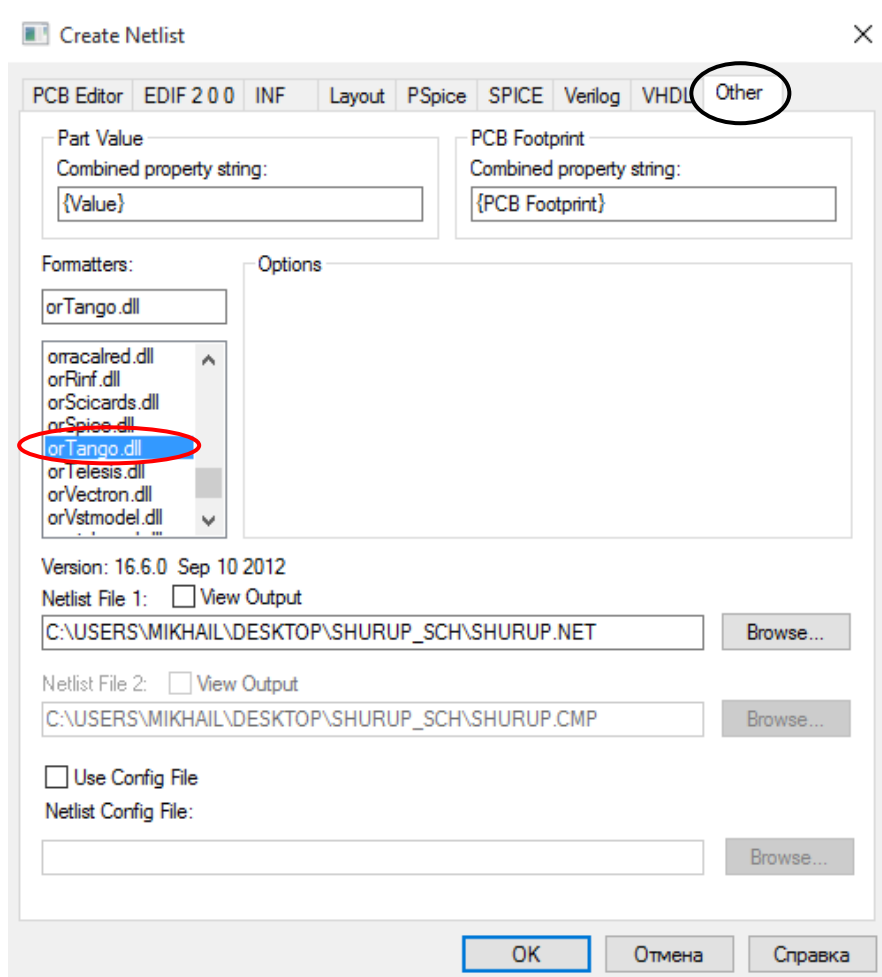


Рисунок 22 – Создание netlist для Tango

Если считанный netlist не имеет структуру формата данных Tango или Allegro, представленная программа выдаст ошибку и предложит открыть другой netlist.

Структура формата данных для Tango и Allegro представлена на рисунках 23 и 24. Ключевыми словами для распознавания программой формата данных NET файлов являются слова в первых строчках, отмеченные желтым цветом – «Revised:» и «\$PACKAGES» для Tango и Allegro соответственно.

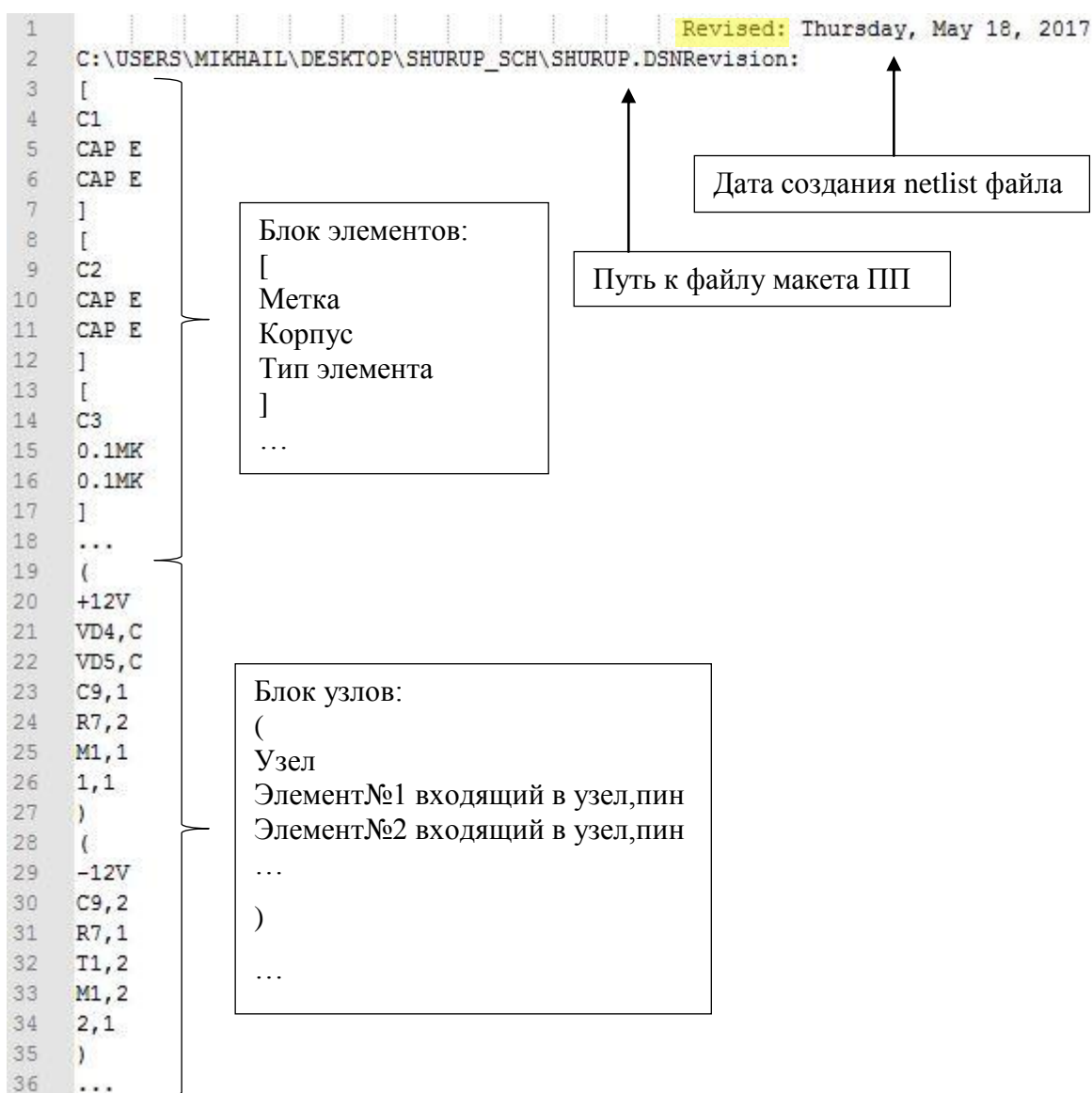


Рисунок 23 – Структура Netlist для Tango

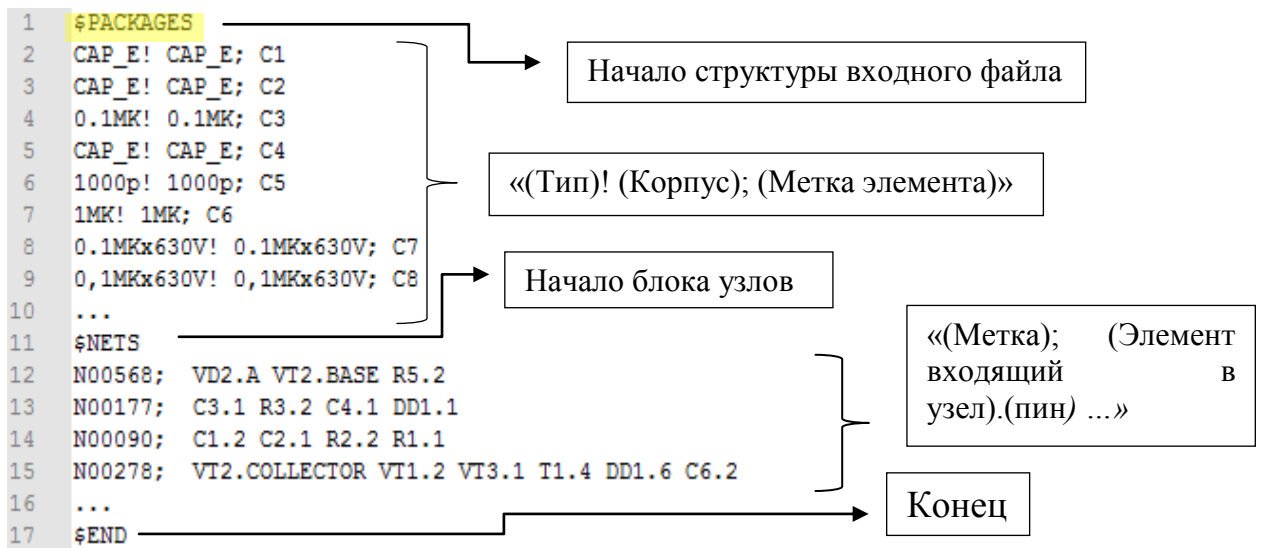


Рисунок 24 – Структура Netlist для Allegro

Полная структура файлов .NET представлена в «приложение Б» данной диссертационной работы.

Теперь рассмотрим процесс импорта netlist–а, созданного программой САПР OrCad, в разработанную программу. Для этого нажимаем кнопку “Файл” в главном меню разработанной программы, затем в “Open Dialog” указываем файл формата .NET, сгенерированный ранее в коммерческих САПР, и нажимаем «Открыть». После открытия ожидаем нескольких секунд, в течении которых программа пытается найти перечисленные в netlist компоненты и корпуса. В окне №3 (рисунок 27) в виде таблицы представится перечень компонентов, с указанием их метки, типа и корпуса.

По данным соединений элементов, взятых из netlist файла, в интерфейсе программы формируются матрица соединений $R=r_{ij}/n \times n$ и матрица расстояний между позициями $D=d_{ij}/n \times n$, как это видно на рисунке 25 и 26. Также наблюдается симметричность матриц относительно центральной диагонали, что свидетельствует о правильности их формирования.

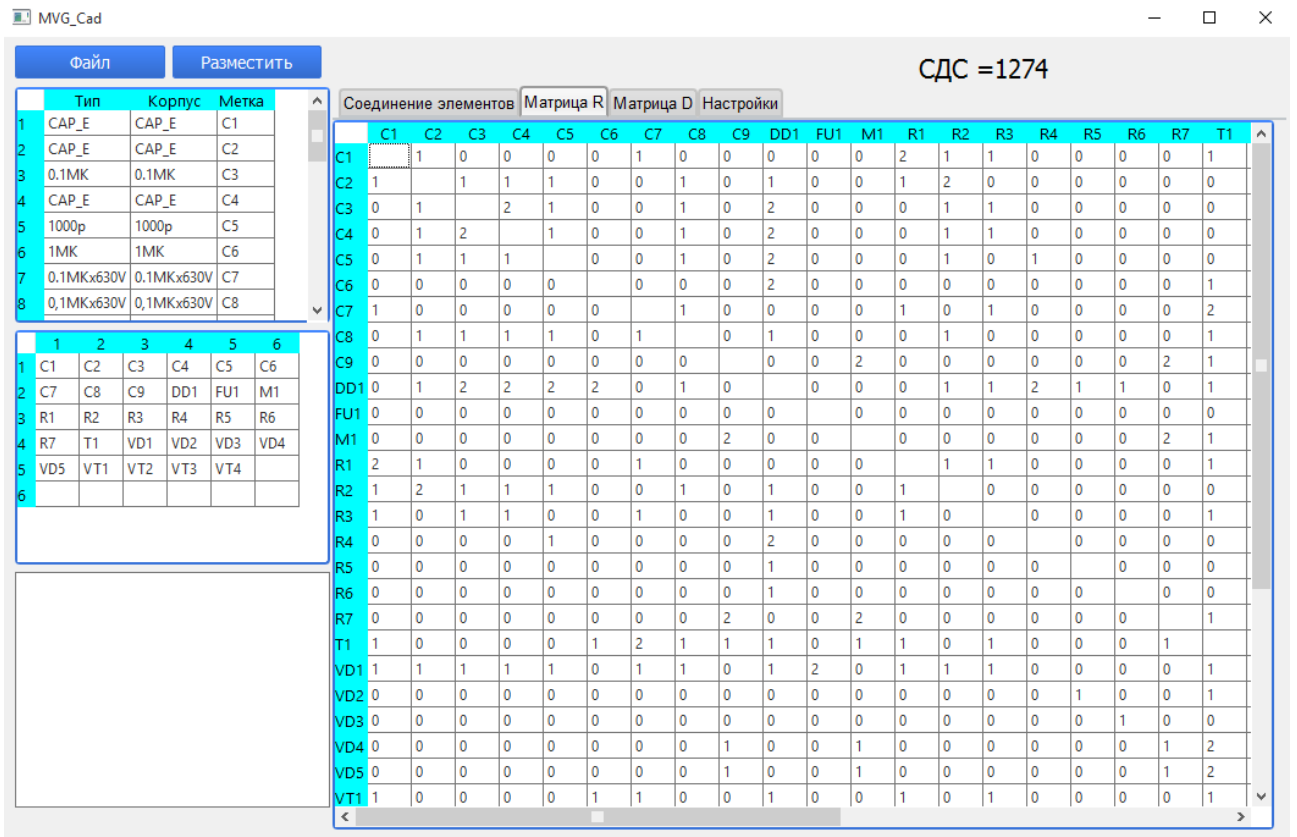


Рисунок 25 – Матрица соединений

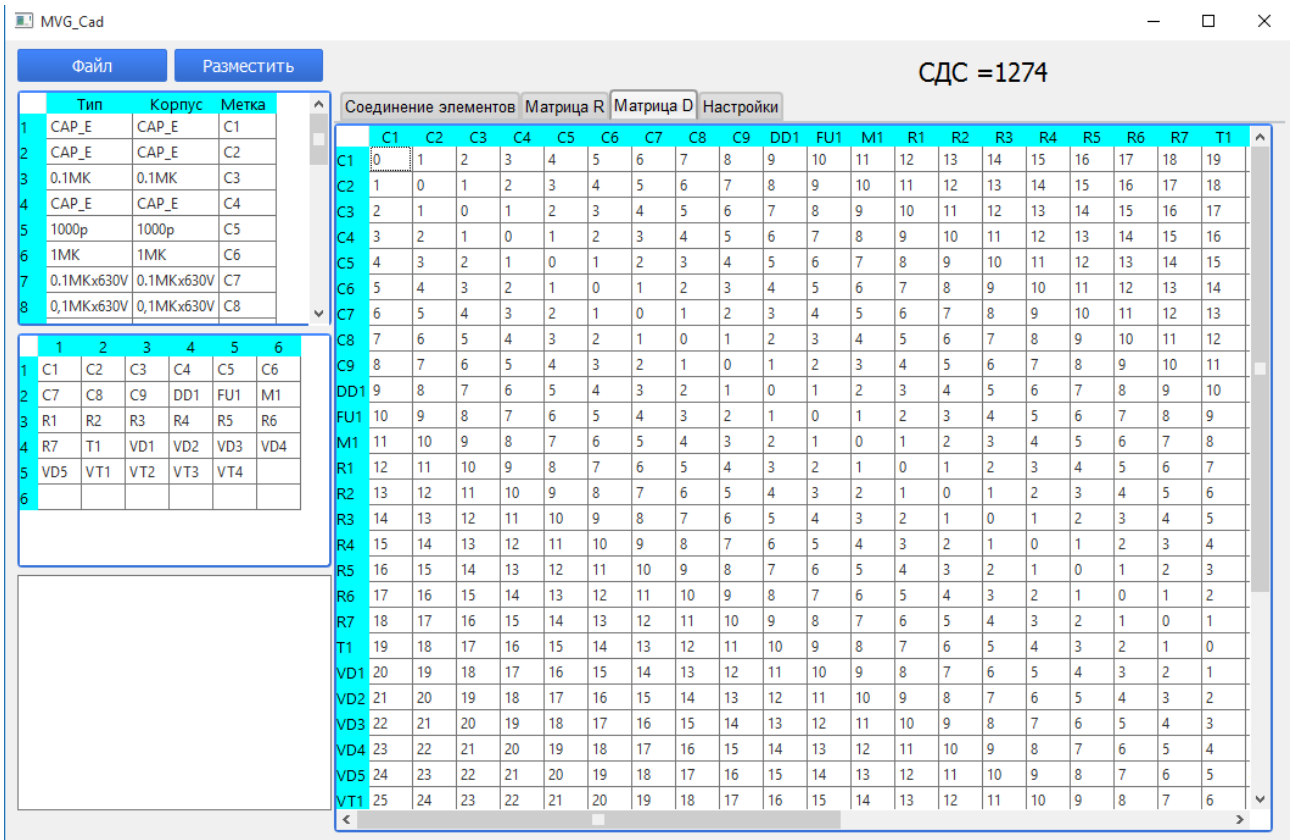


Рисунок 26 – Матрица расстояний между позициями

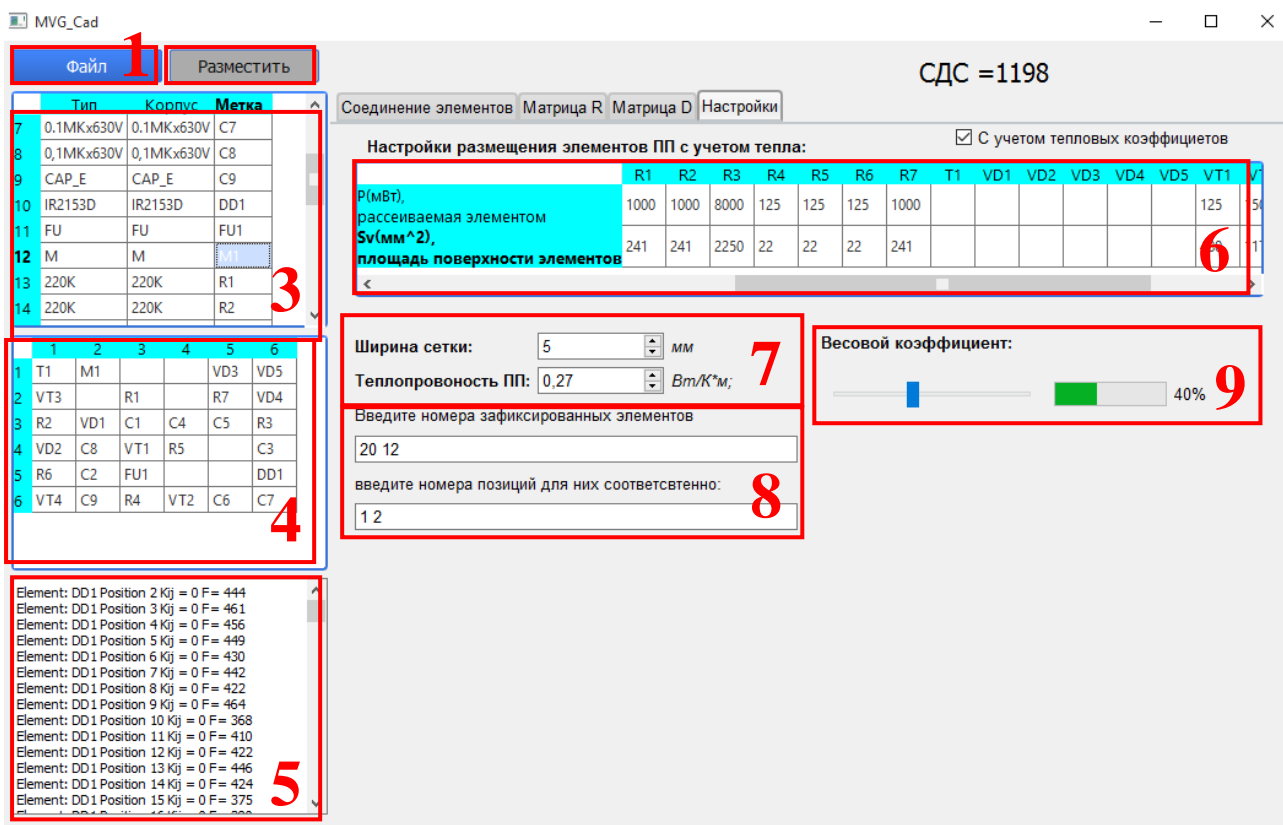


Рисунок 27 – Настройки в интерфейсе пользователя

Для расчета теплового коэффициента нижней границы – K_{ij} (2.17) через программный интерфейс задаются следующие значения во вкладке «Настройки» (см. рисунок 27):

- Рассеиваемая мощность элементов, $mВт$ (№6);
- Площадь поверхности элементов, $мм^2$ (№6);
- Ширина сетки, $мм$ (№7);
- Коэффициент теплопроводности ПП, $Вт/м \times ^\circ C$ (№7);
- Номера зафиксированных элементов и их позиции, (№8);
- Весовой коэффициент, задаваемый ползунком, % (№9).

3.6 Формирование выходных данных

Выходными данными является информация, предоставляемая инженеру–проектировщику программой, как результат срабатывания алгоритма

глобальной оптимизации при минимизации суммарной длины соединений и оптимизации тепловых режимов работы печатной платы.

Файл log. Протокол с записями о событиях в хронологическом порядке. Имеет формат .txt, формируется сразу после срабатывания алгоритма. В нем расписан весь процесс поиска нижней границы для каждого элемента печатной платы и назначения их в позиции, так же демонстрирует формирование матрицы тепловых коэффициентов элементов. На рисунке 28 рассмотрен пример выбора выгодной позиции для последнего размещаемого элемента печатной платы с учетом тепла.

Как видно из примера для элемента «FU1» рассчитывается нижняя граница «F» в каждой свободной позиции. В конце перебора позиций идет назначение элемента в ту позицию, в которой нижняя граница метода была минимальна. Строка назначения имеет вид – «Element: FU1 Advantageous position 30 Fmin= 16170».

После назначения элемента в протоколе log.txt выводится матрица размещенных элементов – «Matrix of placed elements on the scheme...» и матрица тепловых коэффициентов размещенных элементов – «Matrix of thermal coefficients of placed elements...». Данные матрицы отмечены желтым цветом на рисунке 28.

При удачном окончании алгоритма в протокол log.txt записывается строчка – «The process is complete...», которая является последней.


```

...
-----
Element: FU1 Position 18 Kij = 10450 F= 16840
Element: FU1 Position 19 Kij = 10600 F= 17221
Element: FU1 Position 20 Kij = 11115 F= 17975
Element: FU1 Position 24 Kij = 10000 F= 16218
Element: FU1 Position 26 Kij = 11195 F= 17535
Element: FU1 Position 30 Kij = 10000 F= 16170
Element: FU1 Position 31 Kij = 10000 F= 16170
Element: FU1 Position 32 Kij = 10570 F= 16740
Element: FU1 Advantageous position 30 Fmin= 16170
-----
Matrix of placed elements on the scheme...
R4      C6      VT3      C7      C9      R6
VD4     C3      VT4      VD3     R5      VT2
VD5     VT1     T1       C8      C4      C1
*       *       *       DD1     VD1     R2
*       M1     *       C5      C2      R7
FU1     *       *       R3      VD2     R1
-----
Matrix of thermal coefficients of placed elements...
400     70      200     70      70      200
185     70      200     185     400     200
250     200     150     70      70      125
0        0        0        625     150     600
0       10000   0        70      70      150
10      0        0        500     150     500
-----
The process is complete...

```

Рисунок 28 – Пример файла log.txt

Выгодное размещение элементов на коммутационном поле без/с учета тепловых моделей элементов и визуальный нагрев элементов в случае учета тепла (рисунок 33).

Значение СДС. Также выходными данными программы является суммарная взвешенная длина соединений до и после срабатывания алгоритма. Находится в правом верхнем углу программы (рисунок 31).

Именно уменьшение СДС после срабатывания алгоритма говорит нам о улучшении электрических соединения устройства, что упрощает трассировку печатных проводников (следующий этап конструкторского проектирования) и снижает трудоемкость изготовления.

3.7 Анализ результата работы модифицированного алгоритма

Проведем анализ результатов размещения методом ветвей и границ с учетом тепловых моделей элементов.

Для начала определим входные данные. Сгенерируем через OrCAD netlist файл с 29 элементами, представленный в приложении Б, который соответствует принципиальной схеме 1, представленной на рисунке 29.

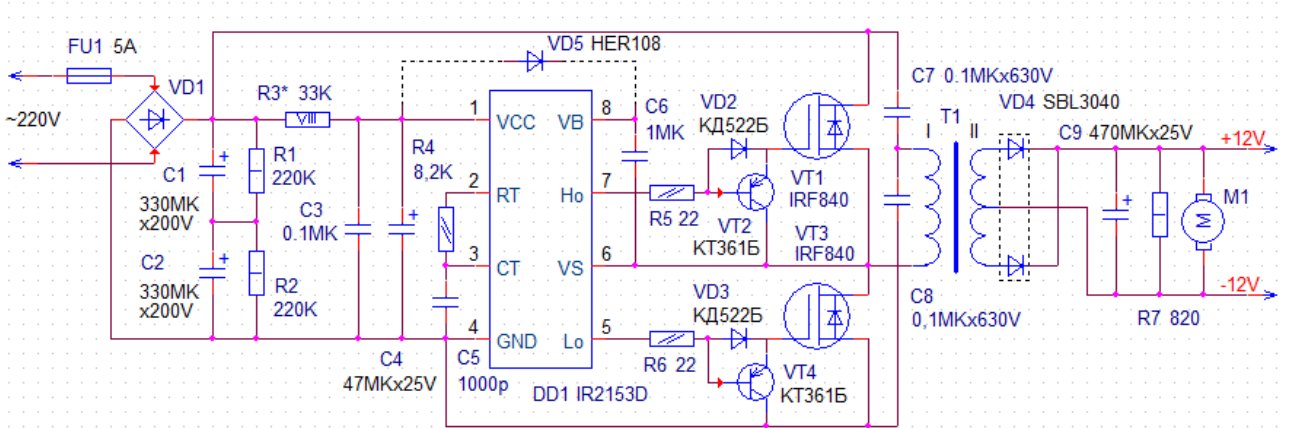


Рисунок 29 – Принципиальная схема 1

На рисунке 30 представлен интерфейс программы, который демонстрирует положение элементов на печатной плате, и начальное СДС не размещенных элементов схемы 1.

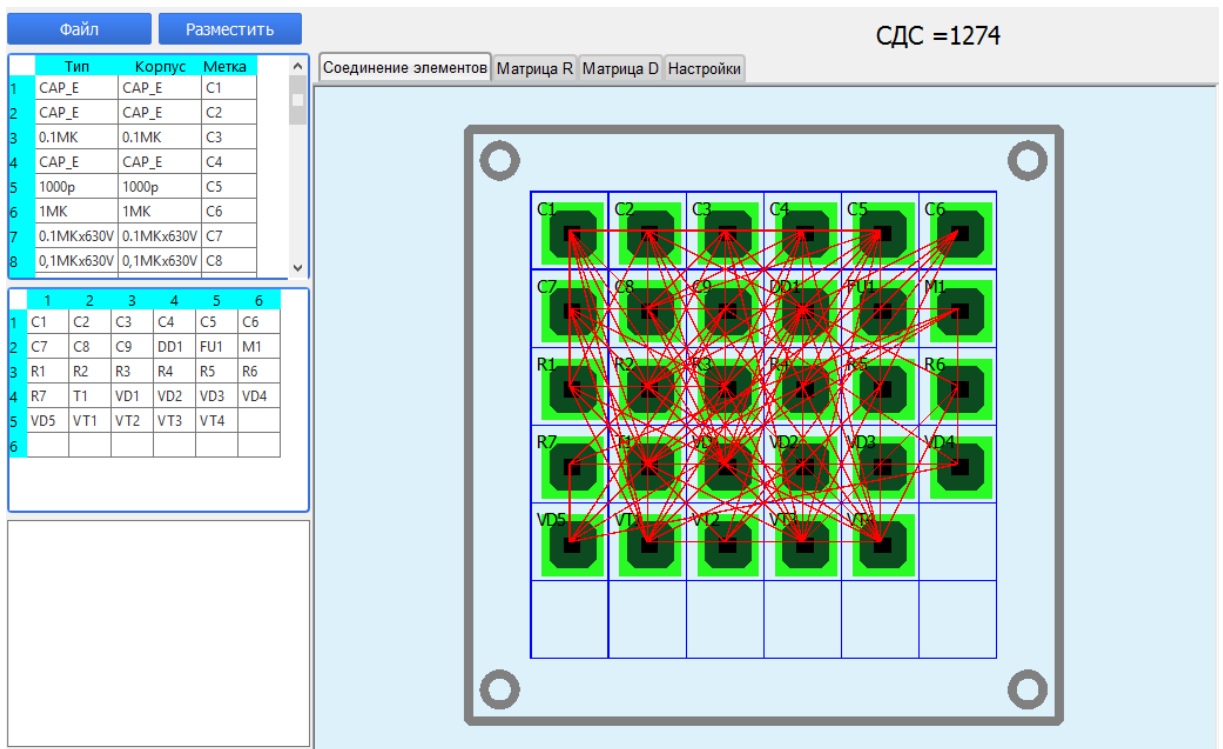


Рисунок 30 – Первоначальное размещение схемы 1

На рисунке 31 мы видим результат размещения первой схемы стандартным методом ветвей и границ без учета тепловых моделей элементов. Уменьшение СДС на 15,4%. Время срабатывания алгоритма размещения 0,64 секунды.

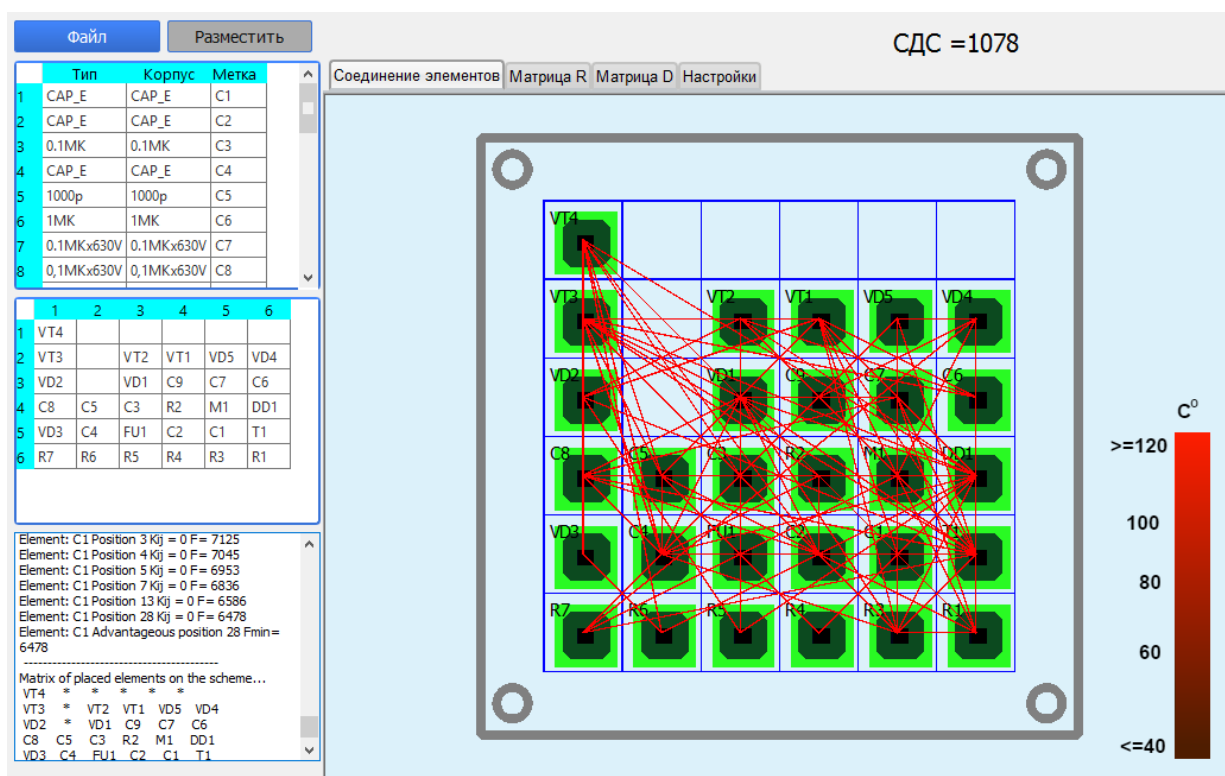


Рисунок 31 – Размещение схемы 1 без учета тепла

Для размещения модифицированным алгоритмом ветвей и границ необходим учет тепловых моделей элементов. Для этого необходимо задать рассеиваемую мощность и площади поверхностей элементов, которые возьмем из паспортных данных элементов.

Таблица 3 – Паспортные данные элементов схемы 1

Элемент	Метки элементов на схеме	Максимально допустимая постоянная рассеиваемая мощность, P_p (Вт) при $T = +25^{\circ}\text{C}$	Площадь поверхности элементов, S_v (мм^2)
IR2153D	DD1,	0,625	45
IRF840	VT1,VT3	0,125	480

КТ361Б	VT2, VT4	0,15	117
220К	R1, R2, R7	1	241
33К	R3	8	2250
8,2К	R4, R5, R6	0,125	22

Элементы печатной платы с малой рассеиваемой мощностью не будем учитывать в расчете. Ширина сетки во всех испытаниях возьмем 5мм, теплопроводность печатной платы возьмем для текстолита равное 0,27 Вт/м×°С, весовой коэффициент 40%.

На рисунке 32 представлено размещение схемы 1 на печатной плате с учетом тепловых моделей элементов, указанных в таблице 3. Уменьшение СДС на 4,9 %, что на 10,5% меньше минимизации СДС при размещении без учета тепла. Объясняется это тем что позиции выбираются не только из соображений минимизации СДС, но и оптимизации тепловых режимов. Время срабатывания алгоритма размещения с учетом тепла 0,79 секунды, что на 0,15 секунды дольше чем размещение без учета тепла.

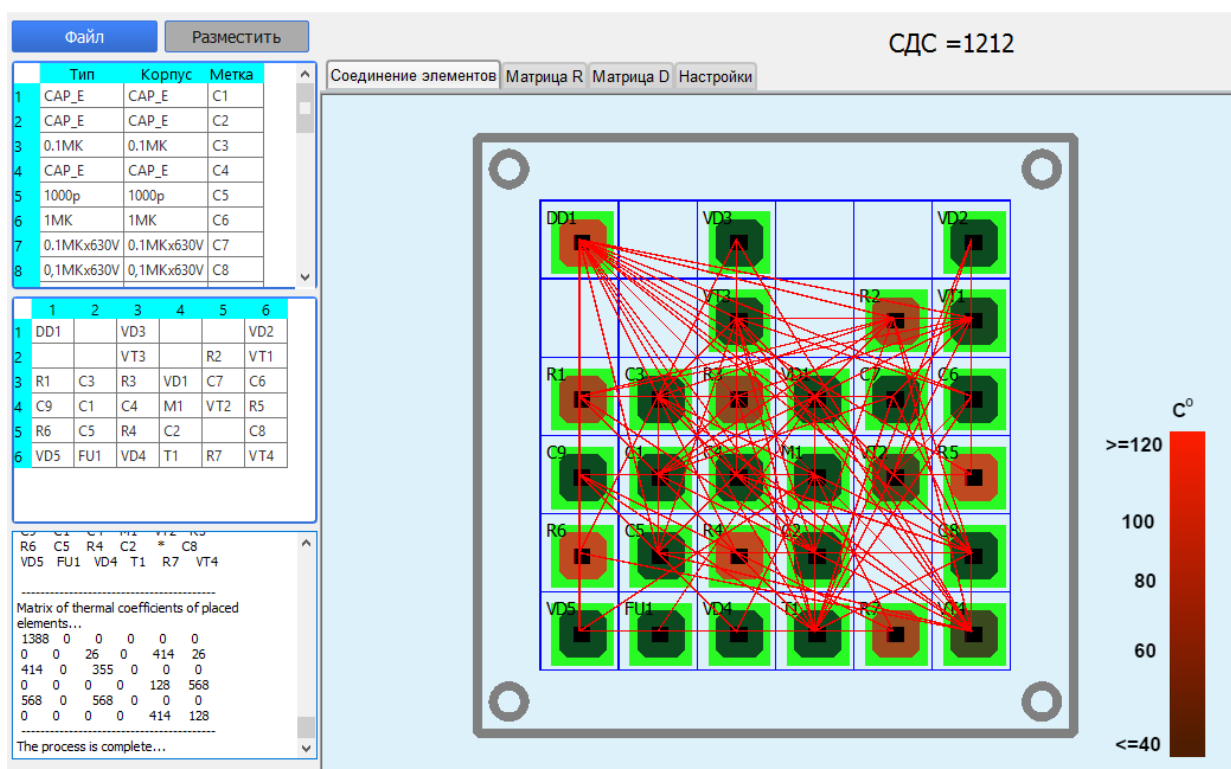


Рисунок 32 – Размещение схемы 1 с учетом тепла

Также рассмотрим размещение модифицированным алгоритмом ветвей и границ при наличии фиксированных элементов, местоположение которых заранее известно (указано разработчиком). Элементы T1 и M1 фиксируем в позициях 1 и 2 печатной платы. Данное размещение представлено на рисунке 33.

В результате размещения СДС уменьшилось на 6%, что на 1,1% лучше, чем в размещении с учетом тепла, но без фиксированных элементов. Объясняется это меньшей оптимизацией тепла вокруг фиксированных элементов, но, по-видимому, лучшим расположением элементов для минимизации СДС.

Протокол с записями о событиях в хронологическом порядке log.txt данного размещения схемы 1 представлен в приложении Г.

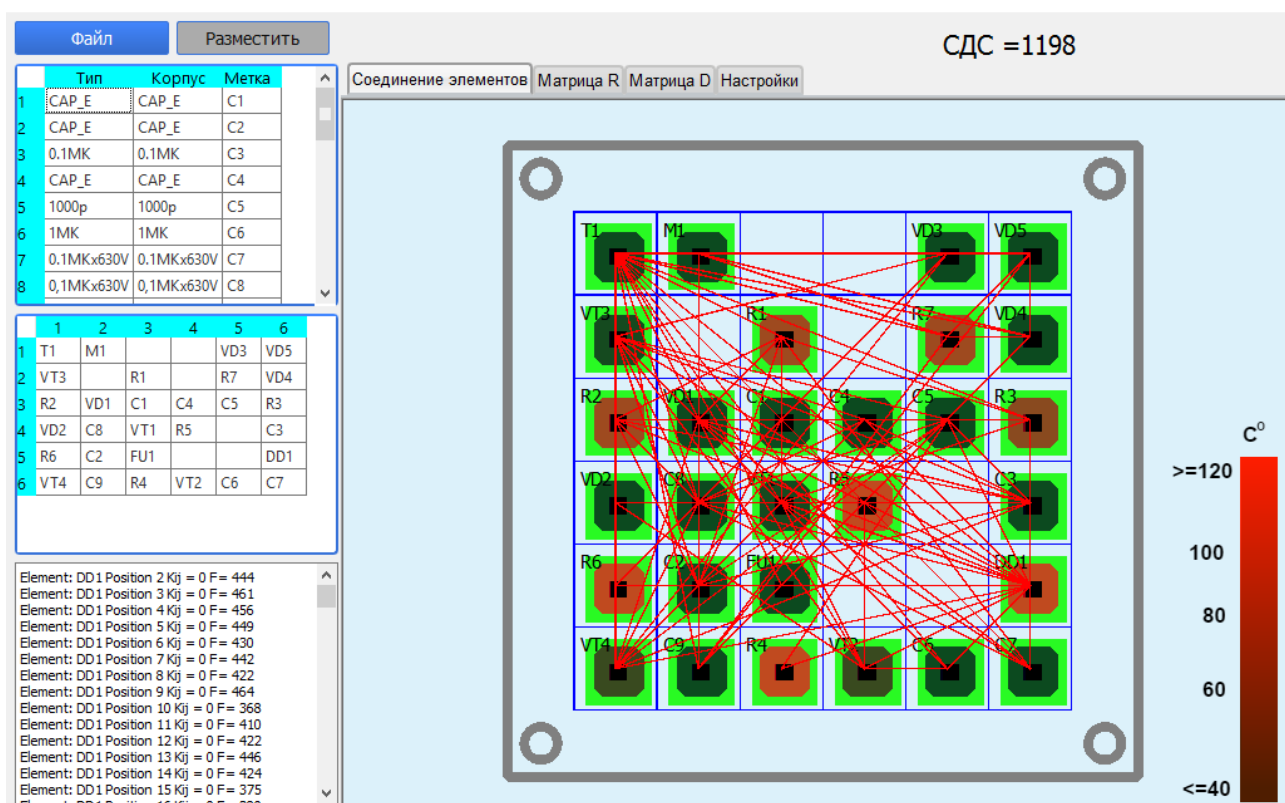


Рисунок 33 – Размещение с фиксированными элементами

Для анализа модифицированного алгоритма в коммерческой САПР были построены тепловые карты данной печатной платы до размещения (рисунок

34), модифицированным алгоритмом ветвей и границ, и после (рисунок 35), с учетом паспортных данных элементов (таблица 3) и размещения (см. рисунок 33).

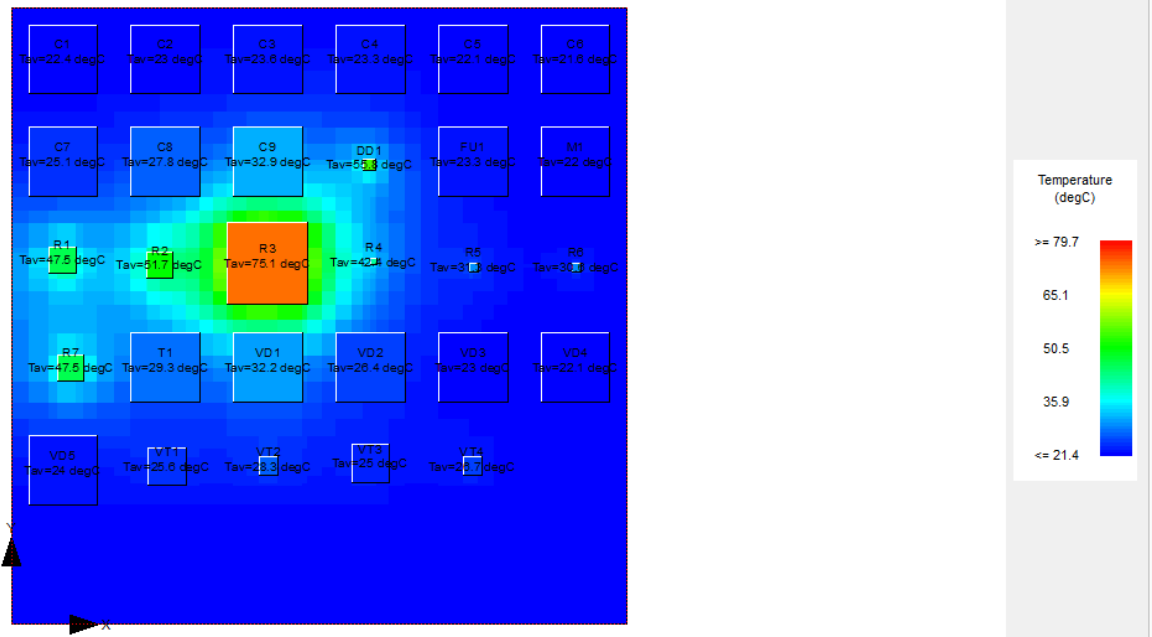


Рисунок 34 – Тепловая карта схемы 1 до размещения

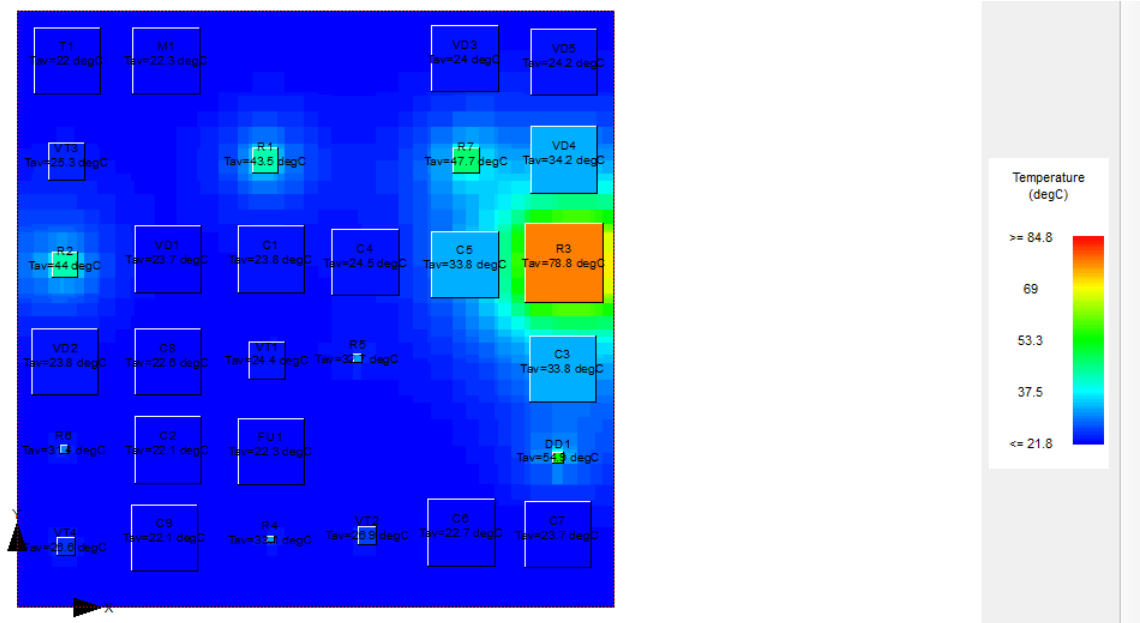


Рисунок 35 – Тепловая карта схемы 1 после размещения модифицированным алгоритмом

Теперь, для анализа программы, сгенерируем через OrCAD netlist файл с 65 элементами, представленным в приложении В, который соответствует принципиальной схеме 2, представленной на рисунке 36.

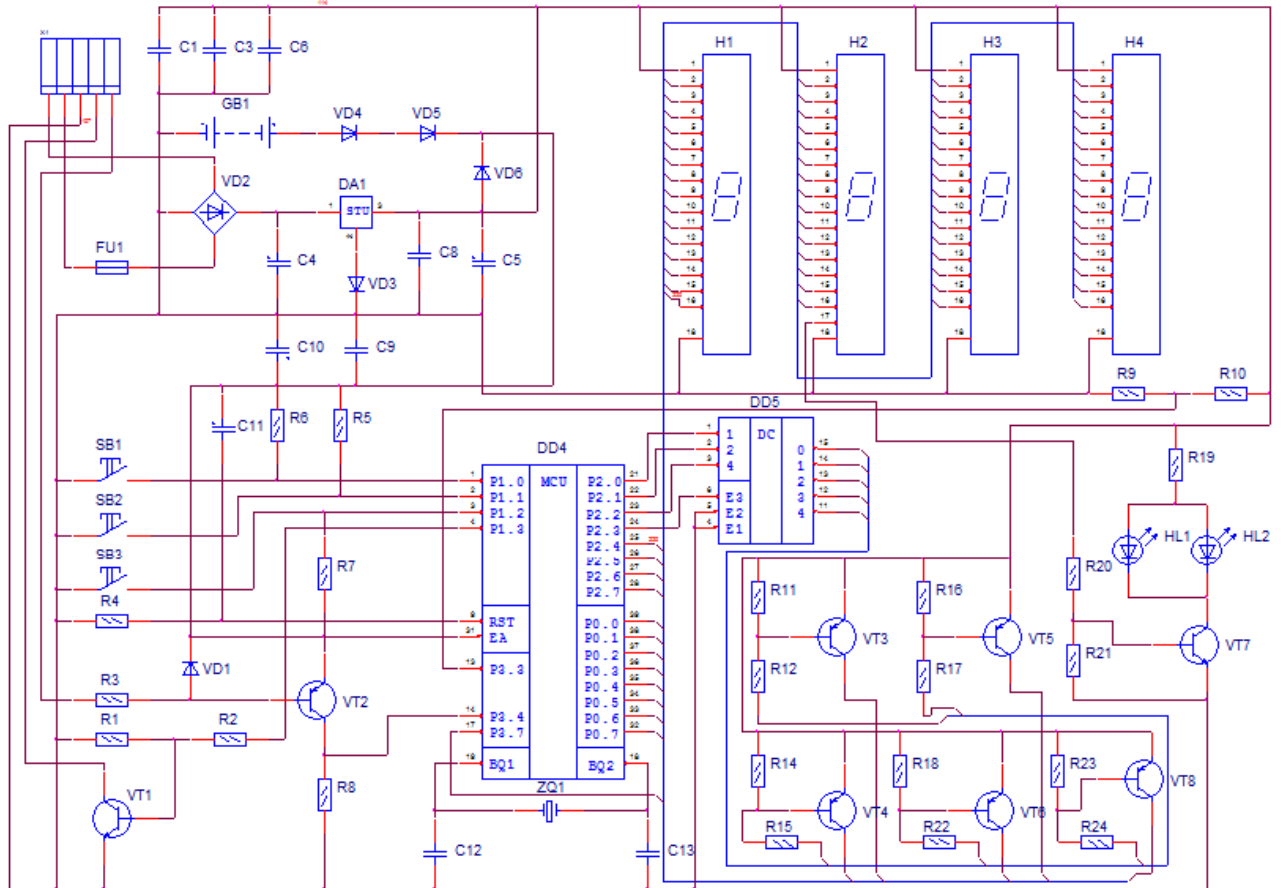


Рисунок 36 – Принципиальная схема 2

На рисунке 37 представлен интерфейс программы, который демонстрирует положение элементов на печатной плате, и начальное СДС до размещения элементов схемы 2.

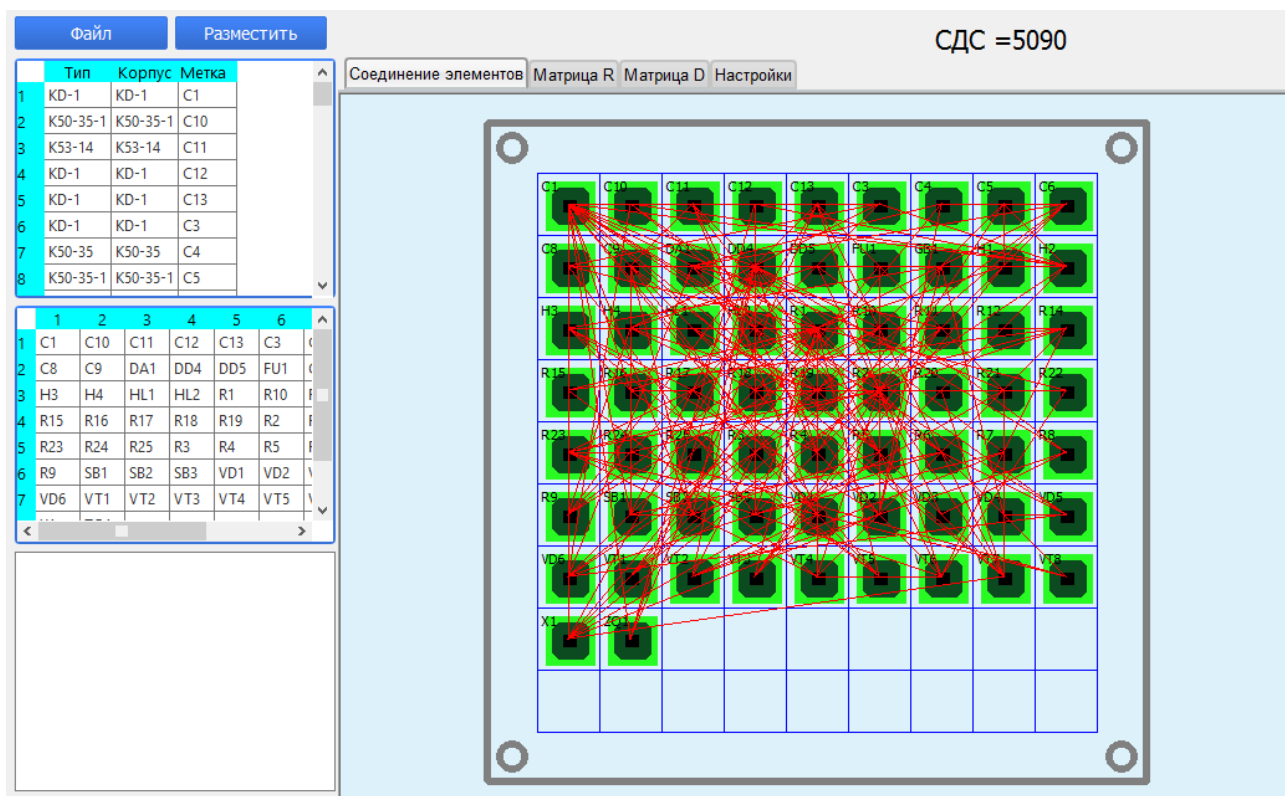


Рисунок 37 – Первоначальное размещение схемы 2

На рисунке 38 мы видим результат размещения первой схемы стандартным методом ветвей и границ без учета тепловых моделей элементов. Уменьшение СДС на 44,3%. Время срабатывания алгоритма размещения 7,12 секунды.

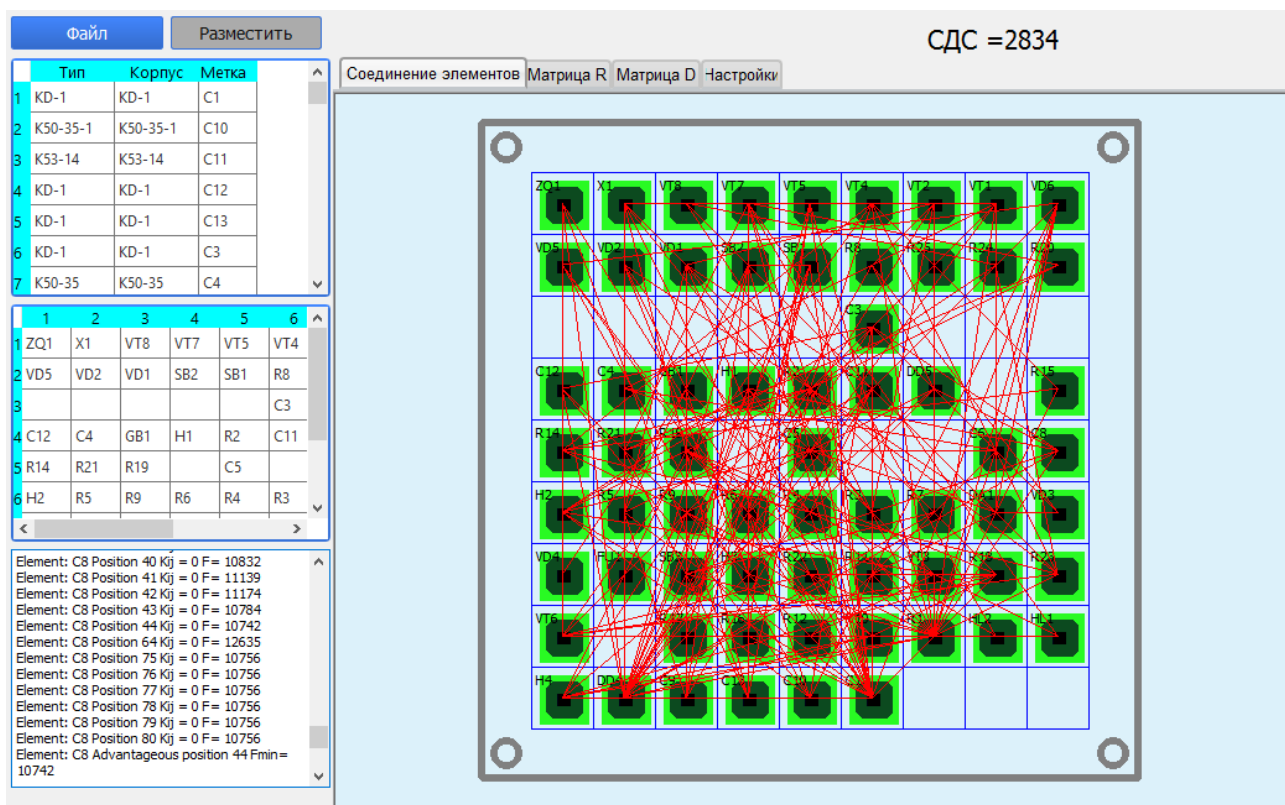


Рисунок 38 – Размещение схемы 2 без учета тепла

Таблица 4 – Паспортные данные элементов схемы 2

Элемент	Метки элементов на схеме	Максимально допустимая постоянная рассеиваемая мощность, $P_p(\text{Вт})$ при $T = +25^\circ\text{C}$	Площадь поверхности элементов, $S_v(\text{мм}^2)$
AT89C51	DD4	0,125	267
K555ID7	DD5	0,051	135
KT3107A	VT2	0,3	123
KT3102A	VT1	0,25	82
MLT-012	Все элементы R	0,125	49
KBL04	VD2	4	1033
KP142EH5A	DA1	10	543
KT973A	VT3, VT4, VT5, VT 6, VT8	8	273
KT503G	VT7	0,35	128
L-813SGD	HL1, HL2	0,075	581

На рисунке 39 представлено размещение схемы 2 на печатной плате с учетом тепловых моделей элементов, указанных в таблице 4. Уменьшение СДС на 35,5%, что на 8,8% меньше минимизации СДС при размещении без учета тепла. Время срабатывания алгоритма размещения модифицированным алгоритмом с учетом тепла 9,3 секунды, что на 2,18 секунды дольше чем размещение стандартным методом ветвей и границ без учета тепла.

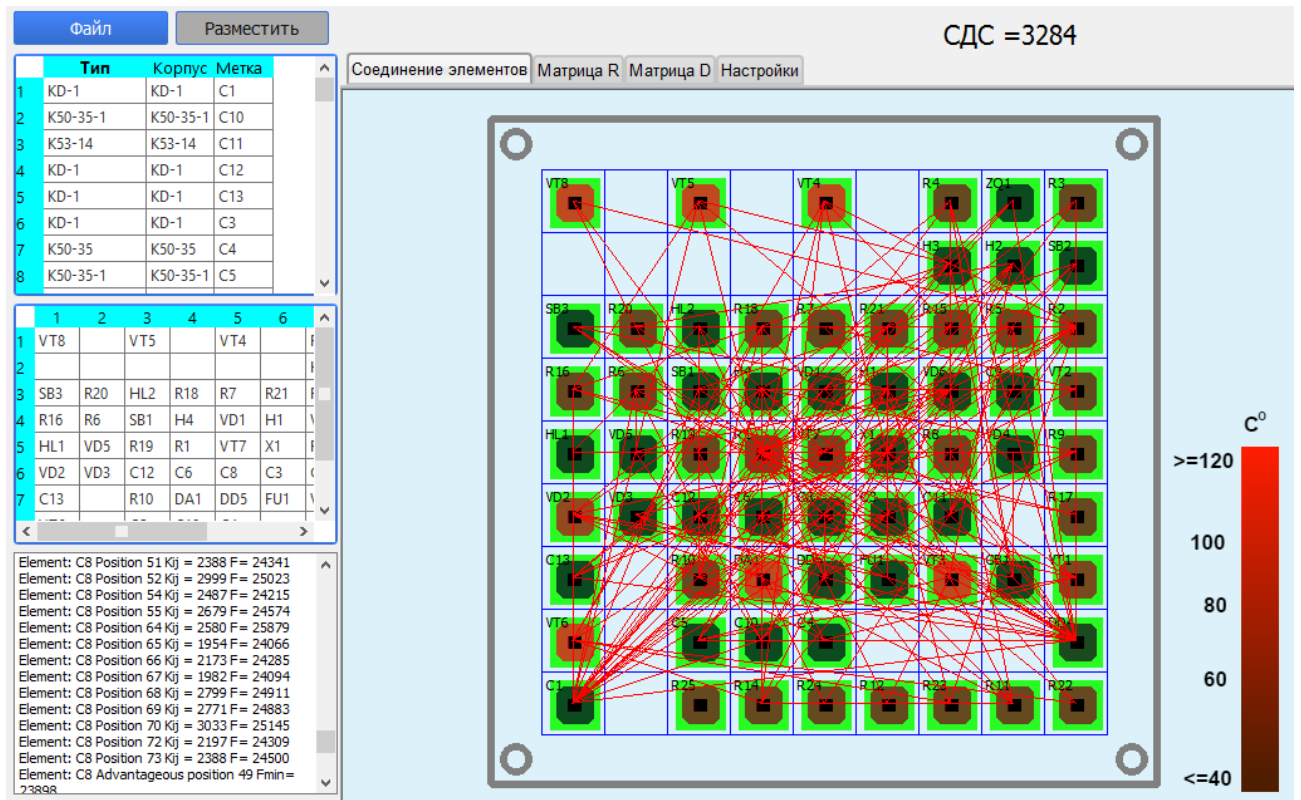


Рисунок 39 – Размещение схемы 2 с учетом тепла

Рассмотрим размещение с учетом тепла при наличии фиксированных элементов, местоположение которых заранее известно (указано разработчиком). Элементы X1, HL1 и HL2 фиксируем в позициях 1, 8 и 9 соответственно. Данное размещение представлено на рисунке 40. В результате размещения СДС уменьшилось на 33,2%, что на 2,3% хуже, чем в размещении с учетом тепла, но без фиксированных элементов.

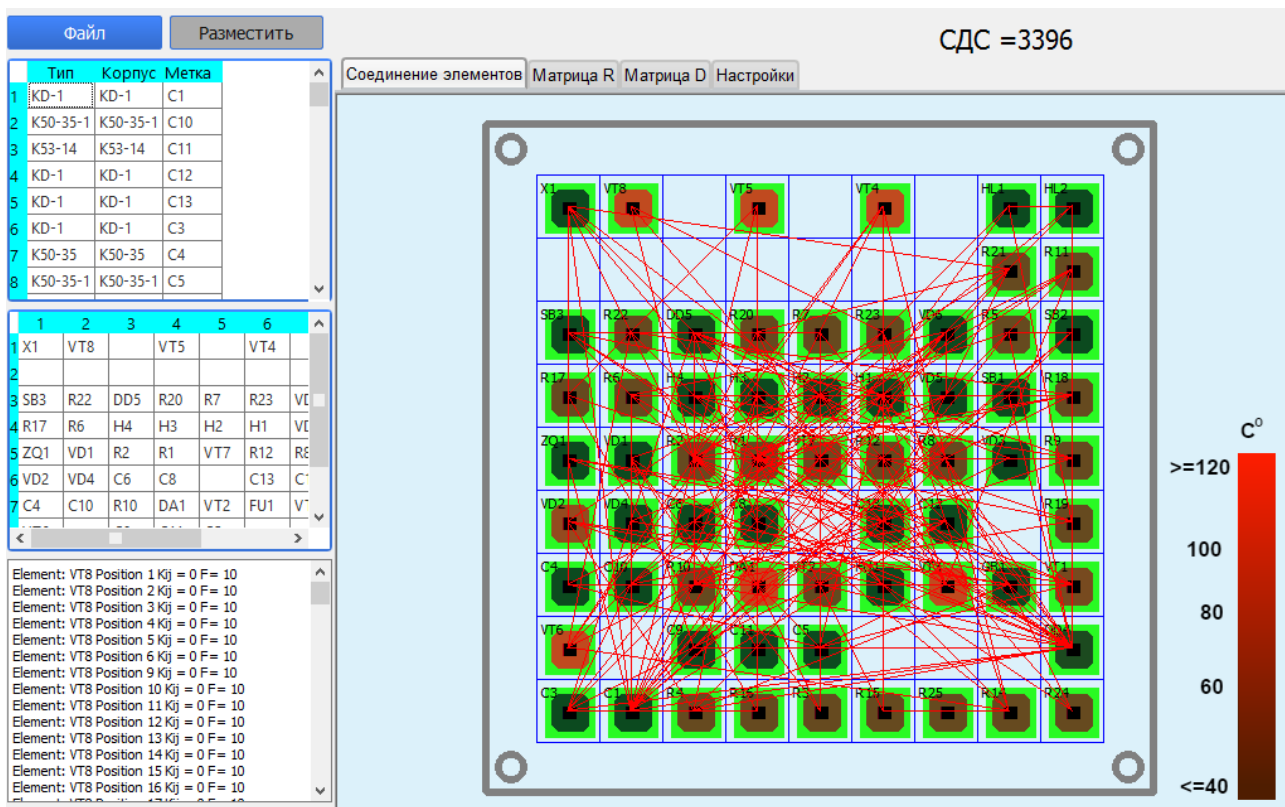


Рисунок 40 – Размещение схемы 2 с фиксированными элементами

Для анализа модифицированного алгоритма в были построены тепловые карты данной печатной платы до размещения (рисунок 41), модифицированным алгоритмом ветвей и границ, и после (рисунок 42), с учетом паспортных данных элементов (таблица 4) и размещения (рисунок 40).

Испытания показали, что программное решение, полученное на основе модифицированного алгоритма ветвей и границ с учетом тепловых характеристик элементов, обеспечивает равномерное размещение теплонагруженных элементов на печатной плате (рисунок 35, 42), при этом обеспечивает минимизацию СДС на 4,9% и 35,5% для первой и второй схемы соответственно.

Также, стоит сказать, что выполнение модифицированного алгоритма занимает на 25 – 30% больше процессорного времени, по сравнению со стандартным алгоритмом размещения.

ЗАКЛЮЧЕНИЕ

В ходе диссертационного проектирования была разработана модификация метода ветвей и границ, для решения задачи размещения как задачи глобальной оптимизации при минимизации суммарной длины соединений и оптимизации тепловых режимов работы печатной платы.

Анализ разработанной программы показал, что модифицированный алгоритм ветвей и границ отлично минимизирует СДС и при этом обеспечивает равномерное размещение теплонагруженных элементов на печатной плате.

В дальнейшем, разработанная программа может быть использована, как вспомогательный инструмент при решении задач размещения и теплового анализа электрорадиоэлементов на ПП, уменьшая время, затрачиваемое инженером–проектировщиком на решение этих задач, тем самым, позволяя уделить больше внимание другим задачам проектирования печатных плат.

СПИСОК СОКРАЩЕНИЙ

РЭА – радиоэлектронная аппаратура;

CAD – computer aided design;

CAE – computer aided engineering;

CAM – computer aided manufacturing;

ПО – программное обеспечение;

ПП – печатная плата;

МВГ – метод ветвей и границ;

РЭА – радиоэлектронная аппаратура;

КП – коммутационное поле;

СДС – суммарная длина соединений;

ЭРЭ – электрорадиоэлемент.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Селютин, В. А. Машинное конструирование электронных устройств / В.А. Селютин. – Москва: Советское радио, 1977. – 384 с.
- 2 Норенков, И. П. Основы теории проектирования САПР: Учебник для вузов по специальности «Вычислительные машины, комплексы, системы и сети» / И. П. Норенков, В. Б. Маничев. – Москва: Высшая школа, 1990. – 335 с.
- 3 Программный продукт HyperLynx Thermal [электронный ресурс] : техническая спецификация. – Режим доступа:
http://s3.mentor.com/public_documents/datasheet/pcb/simulate-thermal-ds.pdf
- 4 FloTHERM PCB [электронный ресурс] : техническая спецификация. – Режим доступа: <https://www.mentor.com/products/mechanical/flotherm/flotherm-pcb/technical-specifications>
- 5 Ansys IcePak [электронный ресурс] : техническая спецификация. – Режим доступа: <http://www.ansys.com/Products/Electronics/ANSYS-Icepak>
- 6 Температурно-электрическая симуляция – Sigriy PowerDC [электронный ресурс]: техническая спецификация – Режим доступа: http://www.cadence.com/rl/Resources/datasheets/Cadence_Sigriy_PowerDC_ds.pdf
- 7 Программный модуль FloEFD [электронный ресурс] : техническая спецификация. – Режим доступа:
http://s3.mentor.com/public_documents/datasheet/products/mechanical/products/floefd-electronics-cooling-module.pdf
- 8 СТО 4.2-07-2014. Система менеджмента и качества. Общие требования к построению, изложению и оформлению документов учебной и научной деятельности. Введ. впервые; дата введения 30.12.2013. Красноярск, 2014. – 60 с.
- 9 Туищев, А. И. Расчет тепловых режимов и характеристик печатных плат радиоэлектронных средств и вычислительной техники при кондуктивном теплообмене / А. И. Туищев, И. О. Губанов. – Тольятти: Вестник НГИЭИ, 2014. – С.84 – 88.

- 10 Дульнев, Г. Н. Методы расчёта теплового режима приборов / Г. Н. Дульнев, В. Г. Парфенов, А. В. Сигалов. – Москва: Радио и связь, 1990. – 306 с.
- 11 Норенков, И. П. Основы теории проектирования САПР: Учебник для вузов по специальности «Вычислительные машины, комплексы, системы и сети» / И. П. Норенков, В. Б. Маничев. – Москва: Высшая школа, 1990. – 335 с.
- 12 Библиотека Qt creator [электронный ресурс] : техническая спецификация. – Режим доступа: <https://www.qt.io/ru/>
- 13 Кознов, Д. В. Основы визуального моделирования / Д. В. Кознов. – Москва: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2008. – 246 с.

ПРИЛОЖЕНИЕ А

Программный код разработанных методов и алгоритмов

A.1 Считывание NET файла

```
void Widget::readFile()
{
    datastr.clear();
    dataCircuit.clear();
    QString str = QFileDialog::getOpenFileName(0, "Open Dialog", "", "*.NET
*.net");
    QFile file(str); // создаем объект класса QFile
    while(QStringList(datastr).isEmpty() != true)
    {
        datastr[del] = "";
        del++;
    }

    if (!file.open(QIODevice::ReadOnly)) // Проверяем, возможно ли открыть наш
    файл для чтения
        return; // если это сделать невозможно, то завершаем функцию
    else
    {
        data = file.readAll(); //считываем все данные с файла в объект data
        ui->pushButton_2->setStyleSheet("QPushButton#pushButton_2 {color: #fff;
border: 1px solid #3873d9; background-color: qlineargradient( x1: 0, y1: 0, x2:
0, y2: 1, stop: 0 #4287ff, stop: 1 #356ccc);}");
        file.close();
        cyclOne =0;
        QApplication::setOverrideCursor(Qt::BusyCursor);
        raschet();
        QApplication::restoreOverrideCursor();
    }
}
```

A.2 Определение формата данных NET файла

```
if(cyclOne==1)
{
    QRegExp formatFactorAllegro("*$PACKAGES\r*");
    formatFactorAllegro.setPatternSyntax(QRegExp::Wildcard);
    QRegExp formatFactorTango("*Revised: *");
    formatFactorTango.setPatternSyntax(QRegExp::Wildcard);

    if(formatFactorAllegro.exactMatch(data))
    {
        dataCircuit = data.split(QRegExp("\n")); //datacir - это строки
связей после nets
        datastr = data.split(QRegExp("[!;]{0,1}\\s+"));
        Allegro = true;
        Tango = false;
        Table();
    }
}
```



```

        while(R != dataCircuit.length()-1)
        {
            if( dataCircuit[R]=="$NETS\r")
            {
                index = R;
            }
            R++;
        }
    }
    else if(formatFactorTango.exactMatch(data))
    {
        dataCircuit = data.split(QRegExp("[\\(\\)]")); //datacir - это строки
связей с 1, а не с 0
        datastr = data.split(QRegExp("[\\n\\r\\r\\n]"));
        index = 0;
        Tango = true;
        Allegro = false;
        Table();
    }
    else
    {
        QMessageBox mb("Упс! Ошибочка=",
            "Этот .Net файл не читается.\n"
            "Хотите открыть другой файл?",
            QMessageBox::Information,
            QMessageBox::Yes | QMessageBox::Default,
            QMessageBox::No,
            QMessageBox::Cancel | QMessageBox::Escape);
        mb.setButtonText(QMessageBox::Yes, "Открыть");
        mb.setButtonText(QMessageBox::No, "Отмена");
        switch(mb.exec()) {
        case QMessageBox::Yes:
            readFile();
            break;
        case QMessageBox::No:
            // exit without saving
            break;
        case QMessageBox::Cancel:
            // don't save and don't exit
            break;
        }
    }
}

```

A.3 Размещение элементов графики

```

QGraphicsScene *scene = new QGraphicsScene(ui->graphicsView);
ui->graphicsView->setScene(scene);
QPen pen(Qt::blue); //Просто выбираем цвет для карандашика
QPen pen2(Qt::red);
QPen pen3(Qt::darkGray, 6);
QPen pen4(Qt::darkGray, 10); //Цвет нагрева элемента
QColor color;

///---последовательное размещение элементов---//
int l =0;
int r =0;
for(int i = 0; i <RM; i++)
{
    for(int j = 0; j<RM; j++)
    {

```

```

        if(IdentMatrix2[i][j] != "")
        {
            l = j*(ui->spinBox->text().toInt())*10+(ui->spinBox-
>text().toInt()*10)/7;
            r = i*(ui->spinBox->text().toInt())*10+(ui->spinBox-
>text().toInt()*10)/7;
            QGraphicsPixmapItem* pixmap = scene->addPixmap(QPixmap(QString(
QCoreApplication::applicationDirPath() + "\\2.bmp")));
            QTransform transform = pixmap->transform();
            transform.translate(l,r);
            pixmap->setTransform(transform);
            //размещение градиента тепла
            double rgb = 0;
            if(cyclOne >1)
            {
                rgb = ceil(rasseivaемоеPMatrix[i][j]/2);
            }
            if(abs(rgb) >255) rgb=255;
            color.setRgb(rgb,0,30,180);
            pen4.setColor(color);
            scene->addRect(l+10,r+10,20,20,pen4);
        }
    }
}
///---размещение на таблице элементов---///
int index23 = 0;
for(int i = 0; i <= RM-1; i++)
{
    for(int j = 0; j<=RM-1; j++)
    {
        int l = j*(ui->spinBox->text().toInt())*10/*+(ui->spinBox-
>text().toInt())/2*/;
        int r = i*(ui->spinBox->text().toInt())*10/*+(ui->spinBox-
>text().toInt())/2*/;
        QGraphicsTextItem* pixmap2 = scene->addText(Ident[index23]);
        QTransform transform2 = pixmap2->transform();
        transform2.translate(l,r);
        pixmap2->setTransform(transform2);
        index23++;
    }
}

///---сетка---///
for(int i = 0; i <= RM; i++)
{
    int t=i*(ui->spinBox->text().toInt())*10;
    scene->addLine(0,0+t,RM*(ui->spinBox->text().toInt()*10),0+t,pen);//x
    scene->addLine(0+t,0,0+t,RM*(ui->spinBox->text().toInt()*10),pen);//y
}
{
    int t=RM*(ui->spinBox->text().toInt())*10;
    scene->addEllipse(-30,-30,20,20,pen3);
    scene->addEllipse(t+10,t+10,20,20,pen3);
    scene->addEllipse(-30,t+10,20,20,pen3);
    scene->addEllipse(t+10,-30,20,20,pen3);

    scene->addRect(-40,-40,t+80,t+80,pen3);
}

```

А.4 Считывание связей с NET файла, соединение элементов на коммутационном поле и вывод матриц R и D

```
///---считывание связей с data, создание матрицы R---///

{
    //создание модели таблицы---//

    QStandardItemModel *modelR = new QStandardItemModel;//модель для матрицы
R
    QStandardItem *itemR;

    QStandardItemModel *modelD = new QStandardItemModel;//модель для матрицы
D
    QStandardItem *itemD;

    //Заголовки столбцов
    QStringList horizontalHeader;
    for(int i=0; i < RM*RM; i++)
    {
        horizontalHeader.append(Ident[i]);
    }
    QStringList verticalHeader;
    for(int i=0; i < (RM)*(RM); i++)
    {
        verticalHeader.append(Ident[i]);
    }

    modelR->setHorizontalHeaderLabels(horizontalHeader);
    modelR->setVerticalHeaderLabels(verticalHeader);

    QString p1 = "*";
    QString p2 = "*";
    QString p3 = "*";
    int close = 0;
    int ff = 0;

    for(int i = 0; i < RM*RM; i++)
    {
        for(int j = 0; j < RM*RM; j++)
        {
            if(i != j && Ident[i] != "" && Ident[j] != "")

            {
                for(int indexizm = index+1; close < 1; indexizm++)
                {
                    //проверка в прямом порядке
                    p1 += Ident[i];
                    p1 += p2;
                    p1 += Ident[j];
                    p1 += p3;
                    QRegExp rx1(p1);
                    rx1.setPatternSyntax(QRegExp::Wildcard);
                    if(rx1.exactMatch(dataCircuit[indexizm]))
                    {
                        ff++;
                    }
                    p1="*";
                    //проверка в обратном порядке
                    p1 += Ident[j];
                }
            }
        }
    }
}
```

```

p1 += p2;
p1 += Ident[i];
p1 += p3;
QRegExp rx2(p1);
rx2.setPatternSyntax(QRegExp::Wildcard);
if(rx2.exactMatch(dataCircuit[indexizm]))
{
    ff++;
}
p1="*";
if(indexizm == dataCircuit.length()-1)
{
    close = 1;
    QString fft = "";
    fft += QString::number(ff);
    itemR = new QStandardItem(fft);
    modelR->setItem(i, j, itemR);
    itemD = new QStandardItem("0");
    modelD->setItem(i, j, itemD);
    MatrixR[i][j]= ff;          //Запоминаем матрицу R

    // соеинение линиями //
    if(ff > 0)
    {
        int chetchik = 0;
        double pp1,pp2,pp3,pp4 = 0;
        for(int x = 0; x< RM; x++)
        {
            for(int y = 0; y< RM; y++)
            {
                if(chetchik == i)
                {
                    pp1=(x)*(ui->spinBox-
>text().toDouble()*10);
                    pp2=(y)*(ui->spinBox-
>text().toDouble()*10);
                }
                if(chetchik == j)
                {
                    pp3=(x)*(ui->spinBox-
>text().toDouble()*10);
                    pp4=(y)*(ui->spinBox-
>text().toDouble()*10);
                }
                chetchik++;
            }
        }
        scene->addLine(pp2+(ui->spinBox-
>text().toInt()*10)/2,pp1+(ui->spinBox->text().toInt()*10)/2,pp4+(ui->spinBox-
>text().toInt()*10)/2,pp3+(ui->spinBox->text().toInt()*10)/2,pen2);
        //построение матрицы D
        {
            QStringList horizontalHeader;
            for(int i=0; i < RM*RM; i++)
            {
                horizontalHeader.append(Ident[i]);
            }
            QStringList verticalHeader;
            for(int i=0; i < (RM)*(RM); i++)
            {
                verticalHeader.append(Ident[i]);
            }
        }
    }
}

```

```

        modelD-
>setHorizontalHeaderLabels(horizontalHeader);
        modelD-
>setVerticalHeaderLabels(verticalHeader);
        int D = 0;

        if(cyclOne>1)
        {
            int xD = (pp1 - pp3)/(ui->spinBox-
>text().toInt()*10);
            int yD = (pp2 - pp4)/(ui->spinBox-
>text().toInt()*10);

            D = abs(xD)+abs(yD);

            QString DD = "";
            DD += QString::number(D);
            itemD = new QStandardItem(DD);
            modelD->setItem(i, j, itemD);
            MatrixD[i][j] = D;//запоминаем матрицу D
    в двумерном массиве

        }
        else
        {

            for(int p = 0; p<RM*RM; p++)
            {
                for(int y = 0; y<RM*RM; y++)
                {
                    D = p - y;
                    D=abs(D);
                    QString DD = "";
                    DD += QString::number(D);
                    itemD = new QStandardItem(DD);
                    modelD->setItem(p, y, itemD);
                    MatrixD[p][y] = D;//запоминаем
    матрицу D в двумерном массиве

                }
            }
        }
    }
}
close =0;
ff=0;
}
}

//скрываем пустые строки пустых позиций
/*for(int i = 0; i < RM*RM; i++)
{
    if(Ident[i] == "")
    {
        ui->tableView_3->hideRow(i);
        ui->tableView_3->hideColumn(i);
        ui->tableView_4->hideRow(i);
        ui->tableView_4->hideColumn(i);
    }
}*/
//выводим матрицу R

```

```

    ui->tableView_3->setModel(modelR);
    ui->tableView_3->resizeRowsToContents();
    ui->tableView_3->resizeColumnsToContents();
    ui->tableView_3-
>setEditTriggers(QAbstractItemView::NoEditTriggers); //делаем таблицу не
редактируемой
    //выводим матрицу D
    ui->tableView_4->setModel(modelD);
    ui->tableView_4->resizeRowsToContents();
    ui->tableView_4->resizeColumnsToContents();
    ui->tableView_4->setEditTriggers(QAbstractItemView::NoEditTriggers);
    // конец создания модели таблицы
}

```

А.5 Нахождение СДС

```

///---Нахождение СДС---///
{
    int SDS = 0;
    for(int i = 0; i<RM*RM; i++)
    {
        for(int j = 0; j<RM*RM; j++)
        {
            SDS = SDS + 0.5 * (MatrixR[i][j]*MatrixD[i][j]);
        }
    }
    QString stringSds = "СДС =";
    stringSds += QString::number(SDS);
    ui->label->setText(stringSds);
    if(cyclOne > 1)
    {
        for(int i = 0; i<1000; i++)
        {
            for(int j = 0; j<1000; j++)
            {
                if(MatrixD[i][j] > 0)
                    MatrixD[i][j] = 0;
                if(MatrixR[i][j]>0)
                    MatrixR[i][j] = 0;
            }
        }
        ui->pushButton_2->setStyleSheet("QPushButton#pushButton_2 {color:
#111; border: 1px solid #3873d9; background: #ababab;}");
    }
}

```

А.6 Отображение таблицы №3 и №4 (рисунок 27) и определение массива элементов – Ident

```

    ui->tableView_5->resizeColumnsToContents(); //динамический размер колонок
в 5-ой таблице

    for(int i = 0; i< RM*RM; i++)
    {
        Ident[i].clear();
    }

```

```

    IdentRazm[i].clear();
}

///---начало записи первой таблицы----///
{
    QStandardItemModel *model = new QStandardItemModel;
    QStandardItem *item;

    //Заголовки столбцов
    QStringList horizontalHeader;
    horizontalHeader.append("Тип");
    horizontalHeader.append("Корпус");
    horizontalHeader.append("Метка ");

    model->setHorizontalHeaderLabels(horizontalHeader);

    if(Allegro == true)
    while(datastr[k] != "$NETS") //отправка в таблицу
    {
        for(int j = 0; j <= 2; j++)
        {
            item = new QStandardItem(datastr[k]);
            model->setItem(value11, j, item);
            k++;
        }
        Ident[value11] += datastr[k-1]; //получаем список элементов в
одномерном массиве до nets
        value11++;
    }
    if(Tango == true)
    {
        k=10;
        QString vremTable[3];
        int iteration = 0;
        while(k < datastr.length())
        {
            iteration = 0;
            if(datastr[k] == "[")
            {
                for(int f = 0; f<=2; f++)
                {
                    vremTable[f].clear();
                }
                while(datastr[k+1] != "]")
                {
                    if(datastr[k+1] != "" && datastr[k+1] != " ")
                    {
                        vremTable[iteration] = datastr[k+1];
                        iteration++;
                    }
                    k++;
                }
                if(vremTable[2] != "" && vremTable[2] != " " &&
vremTable[1] != "" && vremTable[1] != " " && vremTable[0] != "" && vremTable[0]
!= " ")
                {
                    for(int j = 0; j <= 2; j++)
                    {
                        item = new QStandardItem(vremTable[j]);
                        model->setItem(value11, 2-j, item);
                    }
                    Ident[value11] += vremTable[0]; //получаем список
элементов в одномерном массиве до nets
                    value11++;
                }
            }
        }
    }
}

```

```

        }
        k++;
    }
}

ui->tableView->setModel(model);

ui->tableView->resizeRowsToContents();
ui->tableView->resizeColumnsToContents();
ui->tableView-
>setEditTriggers(QAbstractItemView::NoEditTriggers); //делаем таблицу не
редактируемой

}

//////////----начало записи матрицы элементов----/////
{
    //расчитываем размерность матрицы
    {
        RM = ceil(sqrt(value11));
    }

    QStandardItemModel *model_2 = new QStandardItemModel;
    QStandardItem *item_2;

    int I = 0;
    for(int i = 0; i <= RM-1; i++)
    {
        for(int j = 0; j <= RM-1; j++)
        {
            item_2 = new QStandardItem(Ident[I]);
            model_2->setItem(i, j, item_2);
            IdentMatrix2[i][j] = Ident[I]; //запомним в двумерном
массиве

            I++;
        }
    }

    ui->tableView_2->setModel(model_2);

    ui->tableView_2->resizeRowsToContents();
    ui->tableView_2->resizeColumnsToContents();
    ui->tableView_2-
>setEditTriggers(QAbstractItemView::NoEditTriggers); //делаем таблицу не
редактируемой
}

//////////----конец записи таблицы----/////

```

А.7 Метод ветвей и границ с учетом тепловых характеристик и запись файла – log.txt

```

//обнуление массивов тепловых коэффициентов
memset(rasseivaемоеP, 0, 1000*sizeof(int));
memset(rasseivaемоеPArray, 0, 1000*sizeof(int));
memset(IdentVPorydkeVozrast, -1, 1000*sizeof(int));

QApplication::setOverrideCursor(Qt::BusyCursor);

```



```

 QPixmap myPixmapThermo( "Thermo.png" );

 ui->label_7->setPixmap( myPixmapThermo );

 QString strFixedElement;
 QString strFixedPoz;
 neUchetFixElementAndFixPoz = false;
 if(ui->lineEdit_6->text() != "")
 {
     //считываем и разделяем массив не размещаемых элементов
     strFixedElement = ui->lineEdit_6->text();
     strLineFixedElement = strFixedElement.split(QRegExp("[\\s,.]"));

     //считываем и разделяем массив не размещаемых позиций для фиксированных
элементов
     strFixedPoz = ui->lineEdit_7->text();
     strLineFixedPoz = strFixedPoz.split(QRegExp("[\\s,.]"));
 }
 else     neUchetFixElementAndFixPoz = true;
 //считываем тепловые коэф-ы
 for(int i = 0; i<RM*RM; i++)
 {
     rasseivaemoeP[i] = (ui->tableView_5->model()->data(ui->tableView_5-
>model()->index(0, i)).toInt())/((ui->tableView_5->model()->data(ui-
>tableView_5->model()->index(1, i)).toInt()+1);
 }

 if(strLineFixedElement.length() != strLineFixedPoz.length())
 {
     QMessageBox msgBox;
     msgBox.setText("Длина массива фиксированных элементов не равна длине
их позиций. \nЭти данные не будут учтены в расчетах");
     msgBox.exec();
     neUchetFixElementAndFixPoz = true;
 }
 if(neUchetFixElementAndFixPoz == false)
 {
     for(int i =0; i<strLineFixedElement.length(); i++)
     {
         poziciya[strLineFixedPoz[i].toInt()-1]=strLineFixedPoz[i].toInt()-1;
//запоминаем размещенные позиции
         MVGposition[strLineFixedElement[i].toInt()-
1]=strLineFixedPoz[i].toInt()-1; //массив выгодных позиций
         IdentRazm[strLineFixedPoz[i].toInt()-1] =
Ident[strLineFixedElement[i].toInt()-1];
     }
 }
 QFile file1(QString( QApplication::applicationDirPath() + "\\log.txt"));
 file1.open(QIODevice::WriteOnly);
 QTextStream out(&file1);

 int maxRasseivaemoeP = 0; //размещаем теплонагруженные элементы вперед
 bool zanyaty = false;
 for(int n = 0; n < RM*RM; n++)
 {
     for(int y = 0; y < RM*RM; y++)
     {
         zanyaty = false;
         for (int i = 0; i <RM*RM; i++)
         {
             if(IdentVPorydkeVoзраст[i] == y)
             {
                 zanyaty = true;

```

```

        break;
    }
}
if (zanyaty == true) continue;
if (rasseivaemoeP[y] >= maxRasseivaemoeP)
{
    maxRasseivaemoeP = rasseivaemoeP[y];
    IdentVPorydkeVozrast[n] = y;
}
}
maxRasseivaemoeP = 0;
}

for(int ee = 0; ee< RM*RM; ee++)//номер элемента
{
    int e = 0;
    e=IdentVPorydkeVozrast[ee];
    if(e == -1) continue;
    if (Ident[e] == "") continue;
    nextCycle = false;
    if(neUchetFixElementAndFixPoz == false)
    {
        for(int i =0; i<strLineFixedElement.length(); i++)
        {
            if(e == strLineFixedElement[i].toInt()-1) nextCycle = true;
        }
        if(nextCycle == true) continue;
    }
    for(int i = 0; i < RM*RM; i++)
    {
        result[i] = 0; //обнуление нижней границы
    }
    min = 999999999;
    for(int l = 0; l <=(RM*RM)-1; l++)//позиция элемента
    {
        nextCycle = false;
        if(neUchetFixElementAndFixPoz == false)
        {
            for(int i =0; i<strLineFixedPoz.length(); i++)
            {
                if(l == strLineFixedPoz[i].toInt()-1) nextCycle = true;
            }
            if(nextCycle == true) continue;
        }
        /// рассчитываем Uq(p) суммарная длина соединенных с
        неразмещенными
        //получение чисел из матрицы R в порядке возрастания
        int MRUq[h] = {0};
        int lenR = 0;
        for(int j = 0;j<=RM*RM; j++)
        {
            if(j>e)
            {
                {
                    if(MatrixR[e][j] != 0)
                    {
                        MRUq[lenR]=MatrixR[e][j];
                        lenR++;
                    }
                }
            }
        }
        //}
        //размещение цифр в порядке возрастания

```

```

for(int n = 0; n < lenR; n++)
{
    for(int i = 0; i < lenR; ++i)
    {
        if(MRUq[i] > MRUq[i + 1])
        {
            tmp = MRUq[i];
            MRUq[i] = MRUq[i+1];
            MRUq[i+1] = tmp;
        }
    }
}

////получение чисел из матрицы D в порядке убывания
int MDUq[h] = {0};
//int tmp = 0;
int lenD = 0;
for(int i = 0; i<=RM*RM; i++)
{
    for(int j = 0; j<=RM*RM; j++)
    {
        if(((j!=poziciya[j]) || (j !=1)) && ((i == poziciya[j]) ||
(i == 1)))//вычеркиваем строку и столбец размещенного элемента
        {
            if(MatrixR[i][j] != 0)
            {
                MDUq[lenD]=MatrixD[i][j];
                lenD++;
            }
        }
    }
}
//размещение цифр в порядке убывания
for(int n = 0; n < lenD; n++)
{
    for(int i = 0; i < lenD; ++i)
    {
        if(MDUq[i] > MDUq[i - 1])
        {
            tmp = MDUq[i];
            MDUq[i] = MDUq[i-1];
            MDUq[i-1] = tmp;
        }
    }
}
////перемножаем
{
    for(int i = 0; i<lenD;i++)
    {
        Uq[l] = Uq[l]+MRUq[i]*MDUq[i];
    }
}

////расчитываем wq(p) межсоединения не размещенных элементов
//преобразуем матрицу R и размещаем в порядке возрастания
lenR = 0;
int MRwq[h] = {0};
{
    for(int i = 0; i<=RM*RM; i++)
    {
        for (int j = 0; j<=RM*RM; j++)
        {
            if(i>e)//вычеркиваем строки размещенного элемента

```

```

        {
            if(j>i)
            {
                if(MatrixR[i][j] != 0)
                {
                    MRwq[lenR]=MatrixR[i][j];
                    lenR++;
                }
            }
        }
    }
}
//размещение цифр в порядке возрастания
for(int n = 0; n < lenR; n++)
{
    for(int i = 0; i < lenR-1; ++i)
    {
        if(MRwq[i] > MRwq[i + 1])
        {
            tmp = MRwq[i];
            MRwq[i] = MRwq[i+1];
            MRwq[i+1] = tmp;
        }
    }
}

////получение чисел из матрицы D в порядке убывания
int MDwq[h] = {0};
int lenD = 0;
for(int i = 0; i<=RM*RM; i++)
{
    for (int j = 0; j<=RM*RM; j++)
    {
        if((i>e) && (j!=poziciya[j]) && (j!=1))//вычеркиваем
        строку и столбец размещенного элемента
        {
            if(j>i)
            {
                if(MatrixR[i][j] != 0)
                {
                    MDwq[lenD]=MatrixD[i][j];
                    lenD++;
                }
            }
        }
    }
}
//размещение цифр в порядке убывания
for(int n = 0; n < lenD; n++)
{
    for(int i = 0; i < lenD-1; ++i)
    {
        if(MDwq[i] > MDwq[i - 1])
        {
            tmp = MDwq[i];
            MDwq[i] = MDwq[i-1];
            MDwq[i-1] = tmp;
        }
    }
}
////перемножаем
{
    for(int i = 0; i<lenD;i++)

```

```

        {
            wq[l] = (wq[l]+MRwq[i]*MDwq[i]);
        }
    }
}
///---расчитываем расстояние соединенных элементов Fq проверено---
///
Fq = 0;
if(e>1)
{
    for(int i = 1; i < e; i++)
    {
        Fq = Fq + MatrixR[i-1][i] * MatrixD[(MVGposition[i-
1])][(MVGposition[i])];
    }
}
}///цикл перебора позиций i-го l для i-го элемента e закончен

///выбор выгодного элемента на данную позицию с учетом тепловых
характеристик
float Kij = 0;
int zIndex = 0;
int fIndex = 0;
bool closez = false;
{
    for(int i = 0; i<RM*RM; i++)
    {
        {
            closez = false;
            int indexMatrix = 0;
            for(int z =0; z<RM; z++)
            {
                for(int f = 0; f<RM; f++)
                {
                    if (indexMatrix == i)
                    {
                        zIndex =z;
                        fIndex=f;
                        closez=true;
                        break;
                    }
                    indexMatrix++;
                }
                if(closez==true) break;
            }
            for(int x=-1; x<=1; x++)
            {
                for(int y=-1; y<=1; y++)
                {
                    if(zIndex+x >=0 && zIndex+x <=RM && fIndex+y >=0 &&
fIndex+y <= RM)
                    {
                        if(x!=0 || y!=0)
                        {
                            Kij = Kij +
rasseivaemoePMatrix[zIndex+x][fIndex+y];
                        }
                    }
                }
            }
            Kij = Kij*valueProgressBar/100;
            if(ui->checkBox->isChecked() && Kij >= 0 )

```

```

        {
            result[i] = Fq+wq[i]+Uq[i]+Kij;
        }
        else
        {
            result[i]= Fq+wq[i]+Uq[i];
        }
        if(result[i] <0) result[i] = 0;
        Tempkoef[i]=Kij;
        Kij = 0;
    }
}

for(int i = 0; i<RM*RM; i++)
{
    /*for(int k =0; k<strLineFixedElement.length(); k++)
    {
        if(i == strLineFixedElement[k].toInt()-1) nextCycle = true;
    }
    if(nextCycle == true) continue;*/
    if(i == poziciya[i]) continue;
    out <<"Element: " <<Ident[e]<< " Position " << i <<" Kij =
" <<Tempkoef[i]<<" F= " <<result[i]<<"\r" << "\n";
    if(result[i]<min)
    {
        min = result[i];
        poz=i;
    }
}
out <<"Element: " <<Ident[e]<< " Advantageous position " << poz <<"
Fmin= " <<min << "\r" << "\n";
out << " -----" << "\r" << "\n";
poziciya[poz]=poz; //запоминаем размещенные позиции
MVGposition[e]=poz; //массив ВЫГОДНЫХ ПОЗИЦИЙ
///перемещение на ВЫГОДНУЮ ПОЗИЦИЮ
tmt = "";
{
    int period =1;
    QString space = " ";
    int kolSpace =0;
    out << "Matrix of placed elements on the scheme..." << "\r" <<
"\n";

    IdentRazm[MVGposition[e]] = Ident[e];
    for(int h = 0; h < RM*RM; h++)
    {
        if(IdentRazm[h]=="")
        {
            out << " * ";
        }
        else
        {
            if(IdentRazm[h].length() < 5)
            {
                kolSpace = 5 - IdentRazm[h].length();

                for(int i = 0; i<kolSpace; i++)
                {
                    space += " ";
                }
            }
            out << " " <<IdentRazm[h]<<space;
        }
        if(h+1 == RM*period)
        {

```



```

///---Размеще элементов в таблице после срабатывания метода---///
{
    QStandardItemModel *model_3 = new QStandardItemModel;
    QStandardItem *item_3;

    int I = 0;
    for(int i = 0; i <= RM-1; i++)
    {
        for(int j = 0; j <= RM-1; j++)
        {
            item_3 = new QStandardItem(Ident[I]);
            model_3->setItem(i, j, item_3);
            IdentMatrix2[i][j] = Ident[I];    //запомним в двумерном
массиве
            I++;
        }
    }
    ui->tableView_2->setModel(model_3);

    ui->tableView_2->resizeRowsToContents();
    ui->tableView_2->resizeColumnsToContents();
    ui->tableView_2-
>setEditTriggers(QAbstractItemView::NoEditTriggers); //делаем таблицу не
редактируемой
    }
    raschet();
    ui->pushButton_2->setDisabled(1);
}

```


ПРИЛОЖЕНИЕ Б

Входной NET файл для схемы 1 – SHURUP.NET

```
$PACKAGES
CAP_E! CAP_E; C1
CAP_E! CAP_E; C2
0.1МК! 0.1МК; C3
CAP_E! CAP_E; C4
1000р! 1000р; C5
1МК! 1МК; C6
0.1МКx630V! 0.1МКx630V; C7
0,1МКx630V! 0,1МКx630V; C8
CAP_E! CAP_E; C9
IR2153D! IR2153D; DD1
FU! FU; FU1
M! M; M1
220K! 220K; R1
220K! 220K; R2
33K! 33K; R3
8,2K! 8,2K; R4
22! 22; R5
22! 22; R6
820! 820; R7
T! T; T1
MOSTIK! MOSTIK; VD1
VD! VD; VD2
VD! VD; VD3
VD! VD; VD4
VD! VD; VD5
IRF840! IRF840; VT1
KT361! KT361; VT2
IRF840! IRF840; VT3
KT361! KT361; VT4
$NETS
N00568; VD2.A VT2.BASE R5.2
N00177; C3.1 R3.2 C4.1 DD1.1
N00090; C1.2 C2.1 R2.2 R1.1
N00278; VT2.COLLECTOR VT1.2 VT3.1
T1.4 DD1.6 C6.2
N00071; VT1.1 C7.1 C1.1 VD1.3 R1.2 R3.1
+12V; T1.2 C9.2 M1.2 R7.1 M1.1 R7.2
C9.1 VD4.C VD5.C
N00406; VD3.C VT3.3 VT4.EMITTER
N05099; FU1.1 VD1.2
N00364; DD1.7 R5.1
N00335; T1.5 C7.2 C8.1
N01183; FU1.2 VD1.1
N02648; VT4.BASE VD3.A R6.2
N00107; C4.2 VT4.COLLECTOR VT3.2 C8.2
DD1.4 C5.2 VD1.4 R2.1 C3.2 C2.2
N00224; C5.1 R4.1 DD1.3
N00218; DD1.2 R4.2
N00396; VT1.3 VT2.EMITTER VD2.C
N00274; C6.1 DD1.8
N05287; DD1.5 R6.1
N000630; VD5.A T1.3
N000610; VD4.A T1.1
$END
```

ПРИЛОЖЕНИЕ В

Входной NET файл для схемы 2 – 107_allegro.NET

```
$PACKAGES
KD-1! KD-1; C1
K50-35-1! K50-35-1; C10
K53-14! K53-14; C11
KD-1! KD-1; C12
KD-1! KD-1; C13
KD-1! KD-1; C3
K50-35! K50-35; C4
K50-35-1! K50-35-1; C5
KD-1! KD-1; C6
KD-1! KD-1; C8
KD-1! KD-1; C9
KP142EH5A! KP142EH5A; DA1
AT89C51! AT89C51; DD4
K555ID7! K555ID7; DD5
PR-2! PR-2; FU1
476A! 476A; GB1
SA-04! SA-04; H1
SA-04-1! SA-04-1; H2
SA-04! SA-04; H3
SA-04! SA-04; H4
L-813SGD! L-813SGD; HL1
L-813SGD! L-813SGD; HL2
MLT-012! MLT-012; R1
MLT-012! MLT-012; R10
MLT-012! MLT-012; R11
MLT-012! MLT-012; R12
MLT-012! MLT-012; R14
MLT-012! MLT-012; R15
MLT-012! MLT-012; R16
MLT-012! MLT-012; R17
MLT-012! MLT-012; R18
MLT-012! MLT-012; R19
MLT-012! MLT-012; R2
MLT-012! MLT-012; R20
MLT-012! MLT-012; R21
MLT-012! MLT-012; R22
MLT-012! MLT-012; R23
MLT-012! MLT-012; R24
MLT-012! MLT-012; R25
MLT-012! MLT-012; R3
MLT-012! MLT-012; R4
MLT-012! MLT-012; R5
MLT-012! MLT-012; R6
MLT-012! MLT-012; R7
MLT-012! MLT-012; R8
MLT-012! MLT-012; R9
DTS-32! DTS-32; SB1
DTS-32! DTS-32; SB2
DTS-32! DTS-32; SB3
KD521A! KD521A; VD1
KBL04! KBL04; VD2
KD510A! KD510A; VD3
KD510A! KD510A; VD4
KD510A! KD510A; VD5
KD510A! KD510A; VD6
KT3102A! KT3102A; VT1
KT3107A! KT3107A; VT2
KT973A! KT973A; VT3
KT973A! KT973A; VT4
KT973A! KT973A; VT5
KT973A! KT973A; VT6
KT503G! KT503G; VT7
KT973A! KT973A; VT8
SNP-5! SNP-5; X1
HC-49US! HC-49US; ZQ1
$NETS
107; DD4.20 DD5.8 H4.18 R9.1,
    SB2.1 C1.1 SB1.1 C9.2,
    VD3.2 R8.1 DD5.5 DD5.4,
    C13.1 R21.1 VT7.3 X1.3,
    GB1.1 H3.18 R4.1 VD2.1,
    C10.2 C3.1 H2.18 C12.1,
    C8.1 SB3.1 H1.18 R1.1,
    C5.2 C4.2 VT1.3 C6.1
114; DD4.40 DD5.16 C3.2 C8.2,
    C5.1 H1.1 H2.1 VT3.2,
    R18.2 R19.2 R10.2 H4.1,
```

DA1.3 VD6.1 R11.2 C6.2,	N13892; DD4.39
VT5.2 R16.2 H3.1 C1.2,	N13969; DD4.38
VT4.2 R14.2 VT6.2 R23.2,	N14046; DD4.37
VT8.2	N14123; DD4.36
N09911; DD4.24 DD5.6	N14200; DD4.35
N30993; HL1.1 R19.1 HL2.1	N14277; DD4.34
N31023; HL1.2 VT7.2 HL2.2	N14354; DD4.33
N31145; VT7.1 R21.2 R20.1	N14431; DD4.32
N07702; R5.1 SB2.2 DD4.2	N15161; H1.2
N31195; R20.2 H2.17	N15236; H1.3
N06492; DD4.1 R6.1 SB1.2	N15311; H1.4
N07629; R4.2 C11.2 DD4.9	N35263; FU1.1 X1.2
N11471; R9.2 DD4.13 R10.1	N15386; H1.5
N03108; VD2.2 C4.1 DA1.1	N02904; VD2.3 X1.1
N31921; R17.1	N15461; H1.6
N03577; VT1.1 R1.2 R2.1	N36859; DD4.17
N03746; VT1.2 X1.4	N15536; H1.7
N31853; R22.2	N15611; H1.8
N03809; R3.1 X1.5	N25739; DD4.21 DD5.1
N31598; R18.1 R22.1 VT6.1	N25741; DD5.2 DD4.22
N33382; VT6.3	N15686; H1.9
N02863; FU1.2 VD2.4	N25743; DD5.3 DD4.23
N08006; SB3.2 R7.2 DD4.3	N15761; H1.10
N08268; DD4.4 R2.2	N15836; H1.11
N04376; VD4.2 VD5.1	N15911; H1.12
N04305; GB1.2 VD4.1	N15986; H1.13
N04411; VD6.2 VD5.2 C10.1 R6.2,	N16061; H1.14
R5.2 C11.1 C9.1 VD1.2,	N16136; H1.15
VT2.2 DD4.31 R7.1	N17079; H2.16
N03956; R3.2 VD1.1 VT2.1	N17171; H2.15
N017232; DA1.2 VD3.1	N17263; H2.14
N33718; VT8.3	N17355; H2.13
N33835; R24.1	N17447; H2.12
N33860; R23.1 R24.2 VT8.1	N17539; H2.11
N08674; DD4.14 R8.2 VT2.3	N17631; H2.10
N08989; DD4.19 ZQ1.1 C12.2	N17723; H2.9
N09094; DD4.18 ZQ1.2 C13.2	N17815; H2.8
N12613; DD5.14	N17907; H2.7
N12619; DD5.11	N17999; H2.6
N12581; DD5.15	N18091; H2.5
N12617; DD5.12	N18183; H2.4
N12615; DD5.13	N18275; H2.3
N13661; DD4.26	N18367; H2.2
N13738; DD4.27	N19374; H3.2
N13815; DD4.28	N19479; H3.3

N19584; H3.4	N20634; H3.14
N19689; H3.5	N20739; H3.15
N27493; R14.1 VT4.1 R15.2	N20844; H3.16
N26231; R11.1 R12.2 VT3.1	N22164; H4.16
N19794; H3.6	N22283; H4.15
220; DD4.25 H1.16	N22402; H4.14
N19899; H3.7	N22521; H4.13
N20004; H3.8	N22640; H4.12
N28940; VT4.3	N22759; H4.11
N20109; H3.9	N22878; H4.10
N26571; VT3.3	N22997; H4.9
N29493; R16.1 R17.2 VT5.1	N23116; H4.8
N20214; H3.10	N23235; H4.7
N20319; H3.11	N23354; H4.6
N28070; R15.1	N23473; H4.5
N20424; H3.12	N23592; H4.4
N20529; H3.13	N23711; H4.3
N29686; VT5.3	N23830; H4.2
N30021; R12.1	\$END

ПРИЛОЖЕНИЕ Г

Файл log.txt для схемы 1 с учетом тепла и с фиксированными элементами

Element: DD1 Position 2 Kij = 0 F= 444
 Element: DD1 Position 3 Kij = 0 F= 461
 Element: DD1 Position 4 Kij = 0 F= 456
 Element: DD1 Position 5 Kij = 0 F= 449
 Element: DD1 Position 6 Kij = 0 F= 430
 Element: DD1 Position 7 Kij = 0 F= 442
 Element: DD1 Position 8 Kij = 0 F= 422
 Element: DD1 Position 9 Kij = 0 F= 464
 Element: DD1 Position 10 Kij = 0 F= 368
 Element: DD1 Position 11 Kij = 0 F= 410
 Element: DD1 Position 12 Kij = 0 F= 422
 Element: DD1 Position 13 Kij = 0 F= 446
 Element: DD1 Position 14 Kij = 0 F= 424
 Element: DD1 Position 15 Kij = 0 F= 375
 Element: DD1 Position 16 Kij = 0 F= 380
 Element: DD1 Position 17 Kij = 0 F= 382
 Element: DD1 Position 18 Kij = 0 F= 384
 Element: DD1 Position 19 Kij = 0 F= 424
 Element: DD1 Position 20 Kij = 0 F= 443
 Element: DD1 Position 21 Kij = 0 F= 365
 Element: DD1 Position 22 Kij = 0 F= 367
 Element: DD1 Position 23 Kij = 0 F= 373
 Element: DD1 Position 24 Kij = 0 F= 373
 Element: DD1 Position 25 Kij = 0 F= 432
 Element: DD1 Position 26 Kij = 0 F= 399
 Element: DD1 Position 27 Kij = 0 F= 480
 Element: DD1 Position 28 Kij = 0 F= 473
 Element: DD1 Position 29 Kij = 0 F= 358
 Element: DD1 Position 30 Kij = 0 F= 358
 Element: DD1 Position 31 Kij = 0 F= 358
 Element: DD1 Position 32 Kij = 0 F= 358
 Element: DD1 Position 33 Kij = 0 F= 358
 Element: DD1 Position 34 Kij = 0 F= 358
 Element: DD1 Position 35 Kij = 0 F= 358
 Element: DD1 Advantageous position 29
 Fmin= 358

 Matrix of placed elements on the
 scheme...
 T1 M1 * * * *

```

*   *   *   *   *   *
*   *   *   *   *   *
*   *   *   *   *   *
*   *   *   *   *   DD1
*   *   *   *   *   *
  
```

 Matrix of thermal coefficients of placed
 elements...

```

0   0   0   0   0   0
0   0   0   0   0   0
0   0   0   0   0   0
0   0   0   0   0   0
0   0   0   0   0   555
0   0   0   0   0   0
  
```

 Element: R6 Position 2 Kij = 0 F= 595
 Element: R6 Position 3 Kij = 0 F= 611
 Element: R6 Position 4 Kij = 0 F= 604
 Element: R6 Position 5 Kij = 0 F= 593
 Element: R6 Position 6 Kij = 0 F= 570
 Element: R6 Position 7 Kij = 0 F= 584
 Element: R6 Position 8 Kij = 0 F= 562
 Element: R6 Position 9 Kij = 0 F= 604
 Element: R6 Position 10 Kij = 0 F= 508
 Element: R6 Position 11 Kij = 0 F= 550
 Element: R6 Position 12 Kij = 0 F= 562
 Element: R6 Position 13 Kij = 0 F= 586
 Element: R6 Position 14 Kij = 0 F= 564
 Element: R6 Position 15 Kij = 0 F= 515
 Element: R6 Position 16 Kij = 0 F= 520
 Element: R6 Position 17 Kij = 0 F= 522
 Element: R6 Position 18 Kij = 0 F= 524
 Element: R6 Position 19 Kij = 0 F= 562
 Element: R6 Position 20 Kij = 0 F= 582
 Element: R6 Position 21 Kij = 0 F= 501
 Element: R6 Position 22 Kij = 555 F= 1062
 Element: R6 Position 23 Kij = 555 F= 1055
 Element: R6 Position 24 Kij = 0 F= 496
 Element: R6 Position 25 Kij = 0 F= 558
 Element: R6 Position 26 Kij = 0 F= 523

Element: R6 Position 27 Kij = 0 F= 601
 Element: R6 Position 28 Kij = 555 F= 1160
 Element: R6 Position 30 Kij = 0 F= 498
 Element: R6 Position 31 Kij = 0 F= 498
 Element: R6 Position 32 Kij = 0 F= 498
 Element: R6 Position 33 Kij = 0 F= 498
 Element: R6 Position 34 Kij = 555 F= 1053
 Element: R6 Position 35 Kij = 555 F= 1053
 Element: R6 Advantageous position 24
 Fmin= 496

 Matrix of placed elements on the
 scheme...

T1	M1	*	*	*	*
*	*	*	*	*	*
*	*	*	*	*	*
*	*	*	*	*	*
R6	*	*	*	*	DD1
*	*	*	*	*	*

 Matrix of thermal coefficients of placed
 elements...

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
227	0	0	0	0	555
0	0	0	0	0	0

 Element: R5 Position 2 Kij = 0 F= 745
 Element: R5 Position 3 Kij = 0 F= 760
 Element: R5 Position 4 Kij = 0 F= 751
 Element: R5 Position 5 Kij = 0 F= 736
 Element: R5 Position 6 Kij = 0 F= 709
 Element: R5 Position 7 Kij = 0 F= 725
 Element: R5 Position 8 Kij = 0 F= 701
 Element: R5 Position 9 Kij = 0 F= 743
 Element: R5 Position 10 Kij = 0 F= 647
 Element: R5 Position 11 Kij = 0 F= 689
 Element: R5 Position 12 Kij = 0 F= 701
 Element: R5 Position 13 Kij = 0 F= 725
 Element: R5 Position 14 Kij = 0 F= 703
 Element: R5 Position 15 Kij = 0 F= 654
 Element: R5 Position 16 Kij = 0 F= 659
 Element: R5 Position 17 Kij = 0 F= 661

Element: R5 Position 18 Kij = 227 F= 890
 Element: R5 Position 19 Kij = 227 F= 926
 Element: R5 Position 20 Kij = 0 F= 720
 Element: R5 Position 21 Kij = 0 F= 637
 Element: R5 Position 22 Kij = 555 F= 1195
 Element: R5 Position 23 Kij = 555 F= 1183
 Element: R5 Position 25 Kij = 227 F= 912
 Element: R5 Position 26 Kij = 0 F= 648
 Element: R5 Position 27 Kij = 0 F= 723
 Element: R5 Position 28 Kij = 555 F= 1281
 Element: R5 Position 30 Kij = 227 F= 864
 Element: R5 Position 31 Kij = 227 F= 864
 Element: R5 Position 32 Kij = 0 F= 637
 Element: R5 Position 33 Kij = 0 F= 637
 Element: R5 Position 34 Kij = 555 F= 1192
 Element: R5 Position 35 Kij = 555 F= 1192
 Element: R5 Advantageous position 21
 Fmin= 637

 Matrix of placed elements on the
 scheme...

T1	M1	*	*	*	*
*	*	*	*	*	*
*	*	*	*	*	*
*	*	*	R5	*	*
R6	*	*	*	*	DD1
*	*	*	*	*	*

 Matrix of thermal coefficients of placed
 elements...

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	227	0	0
227	0	0	0	0	555
0	0	0	0	0	0

 Element: R4 Position 2 Kij = 0 F= 851
 Element: R4 Position 3 Kij = 0 F= 866
 Element: R4 Position 4 Kij = 0 F= 857
 Element: R4 Position 5 Kij = 0 F= 842
 Element: R4 Position 6 Kij = 0 F= 815
 Element: R4 Position 7 Kij = 0 F= 831
 Element: R4 Position 8 Kij = 0 F= 807
 Element: R4 Position 9 Kij = 0 F= 849

Element: R4 Position 10 Kij = 0 F= 753
 Element: R4 Position 11 Kij = 0 F= 795
 Element: R4 Position 12 Kij = 0 F= 807
 Element: R4 Position 13 Kij = 0 F= 831
 Element: R4 Position 14 Kij = 227 F= 1036
 Element: R4 Position 15 Kij = 227 F= 987
 Element: R4 Position 16 Kij = 227 F= 992
 Element: R4 Position 17 Kij = 0 F= 767
 Element: R4 Position 18 Kij = 227 F= 996
 Element: R4 Position 19 Kij = 227 F= 1030
 Element: R4 Position 20 Kij = 227 F= 1051
 Element: R4 Position 22 Kij = 782 F= 1522
 Element: R4 Position 23 Kij = 555 F= 1278
 Element: R4 Position 25 Kij = 227 F= 1000
 Element: R4 Position 26 Kij = 227 F= 955
 Element: R4 Position 27 Kij = 227 F= 1030
 Element: R4 Position 28 Kij = 782 F= 1585
 Element: R4 Position 30 Kij = 227 F= 970
 Element: R4 Position 31 Kij = 227 F= 970
 Element: R4 Position 32 Kij = 0 F= 743
 Element: R4 Position 33 Kij = 0 F= 743
 Element: R4 Position 34 Kij = 555 F= 1298
 Element: R4 Position 35 Kij = 555 F= 1298
 Element: R4 Advantageous position 32
 Fmin= 743

 Matrix of placed elements on the
 scheme...

T1	M1	*	*	*	*
*	*	*	*	*	*
*	*	*	*	*	*
*	*	*	R5	*	*
R6	*	*	*	*	DD1
*	*	R4	*	*	*

 Matrix of thermal coefficients of placed
 elements...

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	227	0	0
227	0	0	0	0	555
0	0	227	0	0	0

 Element: R7 Position 2 Kij = 0 F= 998

Element: R7 Position 3 Kij = 0 F= 1012
 Element: R7 Position 4 Kij = 0 F= 1001
 Element: R7 Position 5 Kij = 0 F= 982
 Element: R7 Position 6 Kij = 0 F= 950
 Element: R7 Position 7 Kij = 0 F= 969
 Element: R7 Position 8 Kij = 0 F= 939
 Element: R7 Position 9 Kij = 0 F= 984
 Element: R7 Position 10 Kij = 0 F= 883
 Element: R7 Position 11 Kij = 0 F= 925
 Element: R7 Position 12 Kij = 0 F= 937
 Element: R7 Position 13 Kij = 0 F= 962
 Element: R7 Position 14 Kij = 227 F= 1166
 Element: R7 Position 15 Kij = 227 F= 1117
 Element: R7 Position 16 Kij = 227 F= 1122
 Element: R7 Position 17 Kij = 0 F= 897
 Element: R7 Position 18 Kij = 227 F= 1126
 Element: R7 Position 19 Kij = 227 F= 1165
 Element: R7 Position 20 Kij = 227 F= 1185
 Element: R7 Position 22 Kij = 782 F= 1652
 Element: R7 Position 23 Kij = 555 F= 1403
 Element: R7 Position 25 Kij = 454 F= 1350
 Element: R7 Position 26 Kij = 454 F= 1302
 Element: R7 Position 27 Kij = 454 F= 1379
 Element: R7 Position 28 Kij = 782 F= 1719
 Element: R7 Position 30 Kij = 227 F= 1100
 Element: R7 Position 31 Kij = 454 F= 1327
 Element: R7 Position 33 Kij = 227 F= 1100
 Element: R7 Position 34 Kij = 555 F= 1428
 Element: R7 Position 35 Kij = 555 F= 1428
 Element: R7 Advantageous position 10
 Fmin= 883

 Matrix of placed elements on the
 scheme...

T1	M1	*	*	*	*
*	*	*	*	R7	*
*	*	*	*	*	*
*	*	*	R5	*	*
R6	*	*	*	*	DD1
*	*	R4	*	*	*

 Matrix of thermal coefficients of placed
 elements...

0	0	0	0	0	0
0	0	0	0	165	0

```

0 0 0 0 0 0
0 0 0 227 0 0
227 0 0 0 0 555
0 0 227 0 0 0

```

```

-----
Element: R2 Position 2 Kij = 0 F= 1196
Element: R2 Position 3 Kij = 165 F= 1374
Element: R2 Position 4 Kij = 165 F= 1361
Element: R2 Position 5 Kij = 165 F= 1338
Element: R2 Position 6 Kij = 0 F= 1136
Element: R2 Position 7 Kij = 0 F= 1158
Element: R2 Position 8 Kij = 0 F= 1122
Element: R2 Position 9 Kij = 165 F= 1335
Element: R2 Position 11 Kij = 165 F= 1271
Element: R2 Position 12 Kij = 0 F= 1118
Element: R2 Position 13 Kij = 0 F= 1144
Element: R2 Position 14 Kij = 227 F= 1347
Element: R2 Position 15 Kij = 392 F= 1463
Element: R2 Position 16 Kij = 392 F= 1468
Element: R2 Position 17 Kij = 165 F= 1243
Element: R2 Position 18 Kij = 227 F= 1307
Element: R2 Position 19 Kij = 227 F= 1343
Element: R2 Position 20 Kij = 227 F= 1363
Element: R2 Position 22 Kij = 782 F= 1827
Element: R2 Position 23 Kij = 555 F= 1574
Element: R2 Position 25 Kij = 454 F= 1514
Element: R2 Position 26 Kij = 454 F= 1464
Element: R2 Position 27 Kij = 454 F= 1555
Element: R2 Position 28 Kij = 782 F= 1894
Element: R2 Position 30 Kij = 227 F= 1281
Element: R2 Position 31 Kij = 454 F= 1508
Element: R2 Position 33 Kij = 227 F= 1281
Element: R2 Position 34 Kij = 555 F= 1609
Element: R2 Position 35 Kij = 555 F= 1609
Element: R2 Advantageous position 12
Fmin= 1118

```

Matrix of placed elements on the scheme...

```

T1  M1  *  *  *  *
*  *  *  *  R7  *
R2  *  *  *  *  *
*  *  *  R5  *  *
R6  *  *  *  *  DD1
*  *  R4  *  *  *

```

Matrix of thermal coefficients of placed elements...

```

0 0 0 0 0 0
0 0 0 0 165 0
165 0 0 0 0 0
0 0 0 227 0 0
227 0 0 0 0 555
0 0 227 0 0 0

```

```

-----
Element: R1 Position 2 Kij = 0 F= 1467
Element: R1 Position 3 Kij = 165 F= 1643
Element: R1 Position 4 Kij = 165 F= 1627
Element: R1 Position 5 Kij = 165 F= 1604
Element: R1 Position 6 Kij = 165 F= 1551
Element: R1 Position 7 Kij = 165 F= 1581
Element: R1 Position 8 Kij = 0 F= 1370
Element: R1 Position 9 Kij = 165 F= 1592
Element: R1 Position 11 Kij = 165 F= 1514
Element: R1 Position 13 Kij = 165 F= 1555
Element: R1 Position 14 Kij = 227 F= 1591
Element: R1 Position 15 Kij = 392 F= 1704
Element: R1 Position 16 Kij = 392 F= 1708
Element: R1 Position 17 Kij = 165 F= 1484
Element: R1 Position 18 Kij = 392 F= 1712
Element: R1 Position 19 Kij = 392 F= 1750
Element: R1 Position 20 Kij = 227 F= 1606
Element: R1 Position 22 Kij = 782 F= 2057
Element: R1 Position 23 Kij = 555 F= 1808
Element: R1 Position 25 Kij = 454 F= 1752
Element: R1 Position 26 Kij = 454 F= 1693
Element: R1 Position 27 Kij = 454 F= 1798
Element: R1 Position 28 Kij = 782 F= 2135
Element: R1 Position 30 Kij = 227 F= 1517
Element: R1 Position 31 Kij = 454 F= 1744
Element: R1 Position 33 Kij = 227 F= 1517
Element: R1 Position 34 Kij = 555 F= 1845
Element: R1 Position 35 Kij = 555 F= 1845
Element: R1 Advantageous position 8
Fmin= 1370

```

Matrix of placed elements on the scheme...

```

T1  M1  *  *  *  *
*  *  R1  *  R7  *

```



```

R2  *  *  *  *  *
*  *  *  R5  *  *
R6  *  *  *  *  *  DD1
*  *  R4  *  *  *

```

Matrix of thermal coefficients of placed elements...

```

0  0  0  0  0  0
0  0  165  0  165  0
165  0  0  0  0  0
0  0  0  227  0  0
227  0  0  0  0  555
0  0  227  0  0  0

```

Element: R3 Position 2 Kij = 165 F= 1812
Element: R3 Position 3 Kij = 331 F= 1988
Element: R3 Position 4 Kij = 165 F= 1804
Element: R3 Position 5 Kij = 165 F= 1777
Element: R3 Position 6 Kij = 165 F= 1719
Element: R3 Position 7 Kij = 331 F= 1918
Element: R3 Position 9 Kij = 331 F= 1926
Element: R3 Position 11 Kij = 165 F= 1677
Element: R3 Position 13 Kij = 331 F= 1885
Element: R3 Position 14 Kij = 392 F= 1919
Element: R3 Position 15 Kij = 558 F= 2033
Element: R3 Position 16 Kij = 392 F= 1871
Element: R3 Position 17 Kij = 165 F= 1647
Element: R3 Position 18 Kij = 392 F= 1875
Element: R3 Position 19 Kij = 392 F= 1916
Element: R3 Position 20 Kij = 227 F= 1773
Element: R3 Position 22 Kij = 782 F= 2214
Element: R3 Position 23 Kij = 555 F= 1961
Element: R3 Position 25 Kij = 454 F= 1909
Element: R3 Position 26 Kij = 454 F= 1837
Element: R3 Position 27 Kij = 454 F= 1956
Element: R3 Position 28 Kij = 782 F= 2292
Element: R3 Position 30 Kij = 227 F= 1680
Element: R3 Position 31 Kij = 454 F= 1907
Element: R3 Position 33 Kij = 227 F= 1680
Element: R3 Position 34 Kij = 555 F= 2008
Element: R3 Position 35 Kij = 555 F= 2008
Element: R3 Advantageous position 17
Fmin= 1647

Matrix of placed elements on the scheme...

```

T1  M1  *  *  *  *
*  *  R1  *  R7  *
R2  *  *  *  *  R3
*  *  *  R5  *  *
R6  *  *  *  *  DD1
*  *  R4  *  *  *

```

Matrix of thermal coefficients of placed elements...

```

0  0  0  0  0  0
0  0  165  0  165  0
165  0  0  0  0  142
0  0  0  227  0  0
227  0  0  0  0  555
0  0  227  0  0  0

```

Element: VT4 Position 2 Kij = 165 F= 1822
Element: VT4 Position 3 Kij = 331 F= 1998
Element: VT4 Position 4 Kij = 165 F= 1814
Element: VT4 Position 5 Kij = 165 F= 1787
Element: VT4 Position 6 Kij = 165 F= 1729
Element: VT4 Position 7 Kij = 331 F= 1928
Element: VT4 Position 9 Kij = 331 F= 1936
Element: VT4 Position 11 Kij = 307 F= 1829
Element: VT4 Position 13 Kij = 331 F= 1895
Element: VT4 Position 14 Kij = 392 F= 1929
Element: VT4 Position 15 Kij = 558 F= 2043
Element: VT4 Position 16 Kij = 534 F= 2023
Element: VT4 Position 18 Kij = 392 F= 1885
Element: VT4 Position 19 Kij = 392 F= 1926
Element: VT4 Position 20 Kij = 227 F= 1783
Element: VT4 Position 22 Kij = 924 F= 2366
Element: VT4 Position 23 Kij = 697 F= 2113

Element: VT4 Position 25 Kij = 454 F= 1919
 Element: VT4 Position 26 Kij = 454 F= 1847
 Element: VT4 Position 27 Kij = 454 F= 1966
 Element: VT4 Position 28 Kij = 782 F= 2302
 Element: VT4 Position 30 Kij = 227 F= 1690
 Element: VT4 Position 31 Kij = 454 F= 1917
 Element: VT4 Position 33 Kij = 227 F= 1690
 Element: VT4 Position 34 Kij = 555 F= 2018
 Element: VT4 Position 35 Kij = 555 F= 2018
 Element: VT4 Advantageous position 30 Fmin= 1690

 Matrix of placed elements on the scheme...
 T1 M1 * * * *
 * * R1 * R7 *
 R2 * * * * R3
 * * * R5 * *
 R6 * * * * DD1
 VT4 * R4 * * *

 Matrix of thermal coefficients of placed elements...
 0 0 0 0 0 0
 0 0 165 0 165 0
 165 0 0 0 0 142
 0 0 0 227 0 0
 227 0 0 0 0 555
 51 0 227 0 0 0

 Element: VT2 Position 2 Kij = 165 F= 1849
 Element: VT2 Position 3 Kij = 331 F= 2025
 Element: VT2 Position 4 Kij = 165 F= 1840
 Element: VT2 Position 5 Kij = 165 F= 1811
 Element: VT2 Position 6 Kij = 165 F= 1751
 Element: VT2 Position 7 Kij = 331 F= 1951

Element: VT2 Position 9 Kij = 331 F= 1958
 Element: VT2 Position 11 Kij = 307 F= 1851
 Element: VT2 Position 13 Kij = 331 F= 1917
 Element: VT2 Position 14 Kij = 392 F= 1951
 Element: VT2 Position 15 Kij = 558 F= 2065
 Element: VT2 Position 16 Kij = 534 F= 2045
 Element: VT2 Position 18 Kij = 392 F= 1907
 Element: VT2 Position 19 Kij = 392 F= 1948
 Element: VT2 Position 20 Kij = 227 F= 1805
 Element: VT2 Position 22 Kij = 924 F= 2388
 Element: VT2 Position 23 Kij = 697 F= 2135
 Element: VT2 Position 25 Kij = 505 F= 1997
 Element: VT2 Position 26 Kij = 454 F= 1870
 Element: VT2 Position 27 Kij = 454 F= 1993
 Element: VT2 Position 28 Kij = 782 F= 2328
 Element: VT2 Position 31 Kij = 505 F= 1990
 Element: VT2 Position 33 Kij = 227 F= 1712
 Element: VT2 Position 34 Kij = 555 F= 2040
 Element: VT2 Position 35 Kij = 555 F= 2040
 Element: VT2 Advantageous position 33 Fmin= 1712

 Matrix of placed elements on the scheme...
 T1 M1 * * * *
 * * R1 * R7 *
 R2 * * * * R3

```

*   *   *   R5   *   *
R6   *   *   *   *   *   DD1
VT4  *   R4   VT2  *   *

```

Matrix of thermal coefficients of placed elements...

```

0   0   0   0   0   0
0   0   165  0   165  0
165  0   0   0   0   142
0   0   0   227  0   0
227  0   0   0   0   555
51   0   227  51  0   0

```

Element: VT3 Position 2 Kij = 165 F= 1965

Element: VT3 Position 3 Kij = 331 F= 2141

Element: VT3 Position 4 Kij = 165 F= 1954

Element: VT3 Position 5 Kij = 165 F= 1921

Element: VT3 Position 6 Kij = 165 F= 1857

Element: VT3 Position 7 Kij = 331 F= 2059

Element: VT3 Position 9 Kij = 331 F= 2064

Element: VT3 Position 11 Kij = 307 F= 1957

Element: VT3 Position 13 Kij = 331 F= 2023

Element: VT3 Position 14 Kij = 392 F= 2057

Element: VT3 Position 15 Kij = 558 F= 2171

Element: VT3 Position 16 Kij = 534 F= 2151

Element: VT3 Position 18 Kij = 392 F= 2013

Element: VT3 Position 19 Kij = 392 F= 2054

Element: VT3 Position 20 Kij = 227 F= 1911

Element: VT3 Position 22 Kij = 924 F= 2494

Element: VT3 Position 23 Kij = 697 F= 2241

Element: VT3 Position 25 Kij = 505 F= 2113

Element: VT3 Position 26 Kij = 505 F= 2029

Element: VT3 Position 27 Kij = 505 F= 2160

Element: VT3 Position 28 Kij = 833 F= 2497

Element: VT3 Position 31 Kij = 505 F= 2096

Element: VT3 Position 34 Kij = 606 F= 2197

Element: VT3 Position 35 Kij = 555 F= 2146

Element: VT3 Advantageous position 6 Fmin= 1857

Matrix of placed elements on the scheme...

```

T1   M1   *   *   *   *
VT3  *   R1   *   R7   *
R2   *   *   *   *   R3
*   *   *   R5   *   *
R6   *   *   *   *   DD1
VT4  *   R4   VT2  *   *

```

Matrix of thermal coefficients of placed elements...

```

0   0   0   0   0   0
10  0   165  0   165  0
165  0   0   0   0   142
0   0   0   227  0   0
227  0   0   0   0   555
51   0   227  51  0   0

```

Element: VT1 Position 2 Kij = 165 F= 1977

Element: VT1 Position 3 Kij = 331 F= 2151

Element: VT1 Position 4 Kij = 165 F= 1961

Element: VT1 Position 5 Kij = 165 F= 1922

Element: VT1 Position 7 Kij = 341 F= 2067

Element: VT1 Position 9 Kij = 331 F= 2059

Element: VT1 Position 11 Kij = 307 F= 1952

Element: VT1 Position 13 Kij = 341 F= 2028

Element: VT1 Position 14 Kij = 392 F= 2052

Element: VT1 Position 15 Kij = 558 F= 2166

Element: VT1 Position 16 Kij = 534 F= 2146
 Element: VT1 Position 18 Kij = 392 F= 2008
 Element: VT1 Position 19 Kij = 392 F= 2049
 Element: VT1 Position 20 Kij = 227 F= 1908
 Element: VT1 Position 22 Kij = 924 F= 2489
 Element: VT1 Position 23 Kij = 697 F= 2236
 Element: VT1 Position 25 Kij = 505 F= 2115
 Element: VT1 Position 26 Kij = 505 F= 2027
 Element: VT1 Position 27 Kij = 505 F= 2170
 Element: VT1 Position 28 Kij = 833 F= 2508
 Element: VT1 Position 31 Kij = 505 F= 2091
 Element: VT1 Position 34 Kij = 606 F= 2192
 Element: VT1 Position 35 Kij = 555 F= 2141
 Element: VT1 Advantageous position 20 Fmin= 1908

 Matrix of placed elements on the scheme...

T1	M1	*	*	*	*
VT3	*	R1	*	R7	*
R2	*	*	*	*	R3
*	*	VT1	R5	*	*
R6	*	*	*	*	DD1
VT4	*	R4	VT2	*	*

 Matrix of thermal coefficients of placed elements...

0	0	0	0	0	0
10	0	165	0	165	0
165	0	0	0	0	142
0	0	10	227	0	0
227	0	0	0	0	555

51 0 227 51 0 0

 Element: VD5 Position 2 Kij = 165 F= 1984
 Element: VD5 Position 3 Kij = 331 F= 2158
 Element: VD5 Position 4 Kij = 165 F= 1968
 Element: VD5 Position 5 Kij = 165 F= 1929
 Element: VD5 Position 7 Kij = 341 F= 2074
 Element: VD5 Position 9 Kij = 331 F= 2066
 Element: VD5 Position 11 Kij = 307 F= 1959
 Element: VD5 Position 13 Kij = 352 F= 2046
 Element: VD5 Position 14 Kij = 403 F= 2070
 Element: VD5 Position 15 Kij = 568 F= 2183
 Element: VD5 Position 16 Kij = 534 F= 2153
 Element: VD5 Position 18 Kij = 392 F= 2015
 Element: VD5 Position 19 Kij = 403 F= 2067
 Element: VD5 Position 22 Kij = 924 F= 2496
 Element: VD5 Position 23 Kij = 697 F= 2243
 Element: VD5 Position 25 Kij = 516 F= 2133
 Element: VD5 Position 26 Kij = 516 F= 2043
 Element: VD5 Position 27 Kij = 516 F= 2183
 Element: VD5 Position 28 Kij = 833 F= 2513
 Element: VD5 Position 31 Kij = 505 F= 2098
 Element: VD5 Position 34 Kij = 606 F= 2199

Element: VD5 Position 35 Kij = 555 F= 2148

Element: VD5 Advantageous position 5 Fmin= 1929

Matrix of placed elements on the scheme...

T1	M1	*	*	*	VD5
VT3	*	R1	*	R7	*
R2	*	*	*	*	R3
*	*	VT1	R5	*	*
R6	*	*	*	*	DD1
VT4	*	R4	VT2	*	*

Matrix of thermal coefficients of placed elements...

0	0	0	0	0	0
10	0	165	0	165	0
165	0	0	0	0	142
0	0	10	227	0	0
227	0	0	0	0	555
51	0	227	51	0	0

Element: VD4 Position 2 Kij = 165 F= 2016

Element: VD4 Position 3 Kij = 331 F= 2190

Element: VD4 Position 4 Kij = 165 F= 1999

Element: VD4 Position 7 Kij = 341 F= 2102

Element: VD4 Position 9 Kij = 331 F= 2092

Element: VD4 Position 11 Kij = 307 F= 1985

Element: VD4 Position 13 Kij = 352 F= 2072

Element: VD4 Position 14 Kij = 403 F= 2096

Element: VD4 Position 15 Kij = 568 F= 2209

Element: VD4 Position 16 Kij = 534 F= 2179

Element: VD4 Position 18 Kij = 392 F= 2041

Element: VD4 Position 19 Kij = 403 F= 2093

Element: VD4 Position 22 Kij = 924 F= 2522

Element: VD4 Position 23 Kij = 697 F= 2269

Element: VD4 Position 25 Kij = 516 F= 2165

Element: VD4 Position 26 Kij = 516 F= 2069

Element: VD4 Position 27 Kij = 516 F= 2210

Element: VD4 Position 28 Kij = 833 F= 2544

Element: VD4 Position 31 Kij = 505 F= 2124

Element: VD4 Position 34 Kij = 606 F= 2225

Element: VD4 Position 35 Kij = 555 F= 2174

Element: VD4 Advantageous position 11 Fmin= 1985

Matrix of placed elements on the scheme...

T1	M1	*	*	*	VD5
VT3	*	R1	*	R7	VD4
R2	*	*	*	*	R3
*	*	VT1	R5	*	*
R6	*	*	*	*	DD1
VT4	*	R4	VT2	*	*

Matrix of thermal coefficients of placed elements...

0	0	0	0	0	0
10	0	165	0	165	0
165	0	0	0	0	142
0	0	10	227	0	0
227	0	0	0	0	555
51	0	227	51	0	0

Element: VD3 Position 2 Kij = 165 F= 2097

Element: VD3 Position 3 Kij = 331 F= 2269

Element: VD3 Position 4 Kij = 165 F= 2075
 Element: VD3 Position 7 Kij = 341 F= 2169
 Element: VD3 Position 9 Kij = 331 F= 2155
 Element: VD3 Position 13 Kij = 352 F= 2127
 Element: VD3 Position 14 Kij = 403 F= 2149
 Element: VD3 Position 15 Kij = 568 F= 2262
 Element: VD3 Position 16 Kij = 534 F= 2232
 Element: VD3 Position 18 Kij = 392 F= 2094
 Element: VD3 Position 19 Kij = 403 F= 2156
 Element: VD3 Position 22 Kij = 924 F= 2575
 Element: VD3 Position 23 Kij = 697 F= 2324
 Element: VD3 Position 25 Kij = 516 F= 2236
 Element: VD3 Position 26 Kij = 516 F= 2132
 Element: VD3 Position 27 Kij = 516 F= 2287
 Element: VD3 Position 28 Kij = 833 F= 2624
 Element: VD3 Position 31 Kij = 505 F= 2177
 Element: VD3 Position 34 Kij = 606 F= 2278
 Element: VD3 Position 35 Kij = 555 F= 2227
 Element: VD3 Advantageous position 4 Fmin= 2075

 Matrix of placed elements on the scheme...
 T1 M1 * * VD3 VD5
 VT3 * R1 * R7 VD4
 R2 * * * * R3
 * * VT1 R5 * *

R6 * * * * DD1
 VT4 * R4 VT2 * *

 Matrix of thermal coefficients of placed elements...

0	0	0	0	0	0
10	0	165	0	165	0
165	0	0	0	0	142
0	0	10	227	0	0
227	0	0	0	0	555
51	0	227	51	0	0

 Element: VD2 Position 2 Kij = 165 F= 2190
 Element: VD2 Position 3 Kij = 331 F= 2360
 Element: VD2 Position 7 Kij = 341 F= 2248
 Element: VD2 Position 9 Kij = 331 F= 2230
 Element: VD2 Position 13 Kij = 352 F= 2194
 Element: VD2 Position 14 Kij = 403 F= 2214
 Element: VD2 Position 15 Kij = 568 F= 2327
 Element: VD2 Position 16 Kij = 534 F= 2297
 Element: VD2 Position 18 Kij = 392 F= 2159
 Element: VD2 Position 19 Kij = 403 F= 2231
 Element: VD2 Position 22 Kij = 924 F= 2640
 Element: VD2 Position 23 Kij = 697 F= 2391
 Element: VD2 Position 25 Kij = 516 F= 2319
 Element: VD2 Position 26 Kij = 516 F= 2207
 Element: VD2 Position 27 Kij = 516 F= 2370
 Element: VD2 Position 28 Kij = 833 F= 2709

Element: VD2 Position 31 Kij = 505 F= 2242

Element: VD2 Position 34 Kij = 606 F= 2343

Element: VD2 Position 35 Kij = 555 F= 2292

Element: VD2 Advantageous position 18 Fmin= 2159

Matrix of placed elements on the scheme...

T1	M1	*	*	VD3	VD5
VT3	*	R1	*	R7	VD4
R2	*	*	*	*	R3
VD2	*	VT1	R5	*	*
R6	*	*	*	*	DD1
VT4	*	R4	VT2	*	*

Matrix of thermal coefficients of placed elements...

0	0	0	0	0	0
10	0	165	0	165	0
165	0	0	0	0	142
0	0	10	227	0	0
227	0	0	0	0	555
51	0	227	51	0	0

Element: VD1 Position 2 Kij = 165 F= 2287

Element: VD1 Position 3 Kij = 331 F= 2456

Element: VD1 Position 7 Kij = 341 F= 2336

Element: VD1 Position 9 Kij = 331 F= 2314

Element: VD1 Position 13 Kij = 352 F= 2270

Element: VD1 Position 14 Kij = 403 F= 2289

Element: VD1 Position 15 Kij = 568 F= 2401

Element: VD1 Position 16 Kij = 534 F= 2371

Element: VD1 Position 19 Kij = 403 F= 2311

Element: VD1 Position 22 Kij = 924 F= 2714

Element: VD1 Position 23 Kij = 697 F= 2467

Element: VD1 Position 25 Kij = 516 F= 2406

Element: VD1 Position 26 Kij = 516 F= 2284

Element: VD1 Position 27 Kij = 516 F= 2461

Element: VD1 Position 28 Kij = 833 F= 2804

Element: VD1 Position 31 Kij = 505 F= 2316

Element: VD1 Position 34 Kij = 606 F= 2417

Element: VD1 Position 35 Kij = 555 F= 2366

Element: VD1 Advantageous position 13 Fmin= 2270

Matrix of placed elements on the scheme...

T1	M1	*	*	VD3	VD5
VT3	*	R1	*	R7	VD4
R2	VD1	*	*	*	R3
VD2	*	VT1	R5	*	*
R6	*	*	*	*	DD1
VT4	*	R4	VT2	*	*

Matrix of thermal coefficients of placed elements...

0	0	0	0	0	0
10	0	165	0	165	0
165	0	0	0	0	142
0	0	10	227	0	0
227	0	0	0	0	555
51	0	227	51	0	0

Element: FU1 Position 2 Kij = 165 F= 2512

Element: FU1 Position 3 Kij = 331 F= 2681

Element: FU1 Position 7 Kij = 341 F= 2553

Element: FU1 Position 9 Kij = 331 F= 2527

Element: FU1 Position 14 Kij = 403 F= 2500

Element: FU1 Position 15 Kij = 568 F= 2614
 Element: FU1 Position 16 Kij = 534 F= 2584
 Element: FU1 Position 19 Kij = 403 F= 2503
 Element: FU1 Position 22 Kij = 924 F= 2922
 Element: FU1 Position 23 Kij = 697 F= 2659
 Element: FU1 Position 25 Kij = 516 F= 2592
 Element: FU1 Position 26 Kij = 516 F= 2478
 Element: FU1 Position 27 Kij = 516 F= 2649
 Element: FU1 Position 28 Kij = 833 F= 2990
 Element: FU1 Position 31 Kij = 505 F= 2529
 Element: FU1 Position 34 Kij = 606 F= 2630
 Element: FU1 Position 35 Kij = 555 F= 2579
 Element: FU1 Advantageous position 26 Fmin= 2478

 Matrix of placed elements on the scheme...

T1	M1	*	*	VD3	VD5
VT3	*	R1	*	R7	VD4
R2	VD1	*	*	*	R3
VD2	*	VT1	R5	*	*
R6	*	FU1	*	*	DD1
VT4	*	R4	VT2	*	*

 Matrix of thermal coefficients of placed elements...

0	0	0	0	0	0
10	0	165	0	165	0
165	0	0	0	0	142
0	0	10	227	0	0
227	0	0	0	0	555
51	0	227	51	0	0

Element: C9 Position 2 Kij = 165 F= 2892
 Element: C9 Position 3 Kij = 331 F= 3058
 Element: C9 Position 7 Kij = 341 F= 2914
 Element: C9 Position 9 Kij = 331 F= 2896
 Element: C9 Position 14 Kij = 403 F= 2833
 Element: C9 Position 15 Kij = 568 F= 2911
 Element: C9 Position 16 Kij = 534 F= 2891
 Element: C9 Position 19 Kij = 403 F= 2825
 Element: C9 Position 22 Kij = 924 F= 3223
 Element: C9 Position 23 Kij = 697 F= 2967
 Element: C9 Position 25 Kij = 516 F= 2918
 Element: C9 Position 27 Kij = 516 F= 3005
 Element: C9 Position 28 Kij = 833 F= 3343
 Element: C9 Position 31 Kij = 505 F= 2809
 Element: C9 Position 34 Kij = 606 F= 2910
 Element: C9 Position 35 Kij = 555 F= 2859
 Element: C9 Advantageous position 31 Fmin= 2809

 Matrix of placed elements on the scheme...

T1	M1	*	*	VD3	VD5
VT3	*	R1	*	R7	VD4
R2	VD1	*	*	*	R3
VD2	*	VT1	R5	*	*
R6	*	FU1	*	*	DD1
VT4	C9	R4	VT2	*	*

 Matrix of thermal coefficients of placed elements...

0	0	0	0	0	0
10	0	165	0	165	0
165	0	0	0	0	142
0	0	10	227	0	0
227	0	0	0	0	555
51	0	227	51	0	0

 Element: C8 Position 2 Kij = 165 F= 3278
 Element: C8 Position 3 Kij = 331 F= 3441
 Element: C8 Position 7 Kij = 341 F= 3286
 Element: C8 Position 9 Kij = 331 F= 3272
 Element: C8 Position 14 Kij = 403 F= 3180
 Element: C8 Position 15 Kij = 568 F= 3228
 Element: C8 Position 16 Kij = 534 F= 3212
 Element: C8 Position 19 Kij = 403 F= 3147

Element: C8 Position 22 Kij = 924 F= 3540
 Element: C8 Position 23 Kij = 697 F= 3273
 Element: C8 Position 25 Kij = 516 F= 3249
 Element: C8 Position 27 Kij = 516 F= 3362
 Element: C8 Position 28 Kij = 833 F= 3696
 Element: C8 Position 34 Kij = 606 F= 3216
 Element: C8 Position 35 Kij = 555 F= 3165
 Element: C8 Advantageous position 19
 Fmin= 3147

 Matrix of placed elements on the scheme...

T1	M1	*	*	VD3	VD5
VT3	*	R1	*	R7	VD4
R2	VD1	*	*	*	R3
VD2	C8	VT1	R5	*	*
R6	*	FU1	*	*	DD1
VT4	C9	R4	VT2	*	*

 Matrix of thermal coefficients of placed elements...

0	0	0	0	0	0
10	0	165	0	165	0
165	0	0	0	0	142
0	0	10	227	0	0
227	0	0	0	0	555
51	0	227	51	0	0

 Element: C7 Position 2 Kij = 165 F= 3668
 Element: C7 Position 3 Kij = 331 F= 3826
 Element: C7 Position 7 Kij = 341 F= 3652
 Element: C7 Position 9 Kij = 331 F= 3650
 Element: C7 Position 14 Kij = 403 F= 3525
 Element: C7 Position 15 Kij = 568 F= 3527
 Element: C7 Position 16 Kij = 534 F= 3515
 Element: C7 Position 22 Kij = 924 F= 3839
 Element: C7 Position 23 Kij = 697 F= 3563
 Element: C7 Position 25 Kij = 516 F= 3588
 Element: C7 Position 27 Kij = 516 F= 3720
 Element: C7 Position 28 Kij = 833 F= 4048
 Element: C7 Position 34 Kij = 606 F= 3504
 Element: C7 Position 35 Kij = 555 F= 3453
 Element: C7 Advantageous position 35
 Fmin= 3453

Matrix of placed elements on the scheme...

T1	M1	*	*	VD3	VD5
VT3	*	R1	*	R7	VD4
R2	VD1	*	*	*	R3
VD2	C8	VT1	R5	*	*
R6	*	FU1	*	*	DD1
VT4	C9	R4	VT2	*	C7

 Matrix of thermal coefficients of placed elements...

0	0	0	0	0	0
10	0	165	0	165	0
165	0	0	0	0	142
0	0	10	227	0	0
227	0	0	0	0	555
51	0	227	51	0	0

 Element: C6 Position 2 Kij = 165 F= 4076
 Element: C6 Position 3 Kij = 331 F= 4230
 Element: C6 Position 7 Kij = 341 F= 4045
 Element: C6 Position 9 Kij = 331 F= 4048
 Element: C6 Position 14 Kij = 403 F= 3885
 Element: C6 Position 15 Kij = 568 F= 3854
 Element: C6 Position 16 Kij = 534 F= 3846
 Element: C6 Position 22 Kij = 924 F= 4166
 Element: C6 Position 23 Kij = 697 F= 3883
 Element: C6 Position 25 Kij = 516 F= 3924
 Element: C6 Position 27 Kij = 516 F= 4088
 Element: C6 Position 28 Kij = 833 F= 4411
 Element: C6 Position 34 Kij = 606 F= 3820
 Element: C6 Advantageous position 34
 Fmin= 3820

 Matrix of placed elements on the scheme...

T1	M1	*	*	VD3	VD5
VT3	*	R1	*	R7	VD4
R2	VD1	*	*	*	R3
VD2	C8	VT1	R5	*	*
R6	*	FU1	*	*	DD1
VT4	C9	R4	VT2	C6	C7

 Matrix of thermal coefficients of placed elements...

```

0 0 0 0 0 0
10 0 165 0 165 0
165 0 0 0 0 142
0 0 10 227 0 0
227 0 0 0 0 555
51 0 227 51 0 0

```

```

-----
Element: C5 Position 2 Kij = 165 F= 4543
Element: C5 Position 3 Kij = 331 F= 4691
Element: C5 Position 7 Kij = 341 F= 4490
Element: C5 Position 9 Kij = 331 F= 4505
Element: C5 Position 14 Kij = 403 F= 4300
Element: C5 Position 15 Kij = 568 F= 4227
Element: C5 Position 16 Kij = 534 F= 4223
Element: C5 Position 22 Kij = 924 F= 4539
Element: C5 Position 23 Kij = 697 F= 4246
Element: C5 Position 25 Kij = 516 F= 4307
Element: C5 Position 27 Kij = 516 F= 4514
Element: C5 Position 28 Kij = 833 F= 4848
Element: C5 Advantageous position 16
Fmin= 4223

```

```

-----
Matrix of placed elements on the
scheme...
T1 M1 * * VD3 VD5
VT3 * R1 * R7 VD4
R2 VD1 * * C5 R3
VD2 C8 VT1 R5 * *
R6 * FU1 * * DD1
VT4 C9 R4 VT2 C6 C7

```

```

-----
Matrix of thermal coefficients of placed
elements...
0 0 0 0 0 0
10 0 165 0 165 0
165 0 0 0 0 142
0 0 10 227 0 0
227 0 0 0 0 555
51 0 227 51 0 0

```

```

-----
Element: C4 Position 2 Kij = 165 F= 5072
Element: C4 Position 3 Kij = 331 F= 5213
Element: C4 Position 7 Kij = 341 F= 4995
Element: C4 Position 9 Kij = 331 F= 5022
Element: C4 Position 14 Kij = 403 F= 4778

```

```

Element: C4 Position 15 Kij = 568 F= 4648
Element: C4 Position 22 Kij = 924 F= 4971
Element: C4 Position 23 Kij = 697 F= 4668
Element: C4 Position 25 Kij = 516 F= 4750
Element: C4 Position 27 Kij = 516 F= 4986
Element: C4 Position 28 Kij = 833 F= 5333
Element: C4 Advantageous position 15
Fmin= 4648

```

```

-----
Matrix of placed elements on the
scheme...
T1 M1 * * VD3 VD5
VT3 * R1 * R7 VD4
R2 VD1 * C4 C5 R3
VD2 C8 VT1 R5 * *
R6 * FU1 * * DD1
VT4 C9 R4 VT2 C6 C7

```

```

-----
Matrix of thermal coefficients of placed
elements...
0 0 0 0 0 0
10 0 165 0 165 0
165 0 0 0 0 142
0 0 10 227 0 0
227 0 0 0 0 555
51 0 227 51 0 0

```

```

-----
Element: C3 Position 2 Kij = 165 F= 5658
Element: C3 Position 3 Kij = 331 F= 5792
Element: C3 Position 7 Kij = 341 F= 5557
Element: C3 Position 9 Kij = 331 F= 5602
Element: C3 Position 14 Kij = 403 F= 5308
Element: C3 Position 22 Kij = 924 F= 5456
Element: C3 Position 23 Kij = 697 F= 5143
Element: C3 Position 25 Kij = 516 F= 5256
Element: C3 Position 27 Kij = 516 F= 5515
Element: C3 Position 28 Kij = 833 F= 5868
Element: C3 Advantageous position 23
Fmin= 5143

```

```

-----
Matrix of placed elements on the
scheme...
T1 M1 * * VD3 VD5
VT3 * R1 * R7 VD4
R2 VD1 * C4 C5 R3

```

VD2 C8 VT1 R5 * C3
 R6 * FU1 * * DD1
 VT4 C9 R4 VT2 C6 C7

 Matrix of thermal coefficients of placed elements...

0 0 0 0 0 0
 10 0 165 0 165 0
 165 0 0 0 0 142
 0 0 10 227 0 0
 227 0 0 0 0 555
 51 0 227 51 0 0

 Element: C2 Position 2 Kij = 165 F= 6279
 Element: C2 Position 3 Kij = 331 F= 6406
 Element: C2 Position 7 Kij = 341 F= 6151
 Element: C2 Position 9 Kij = 331 F= 6213
 Element: C2 Position 14 Kij = 403 F= 5860
 Element: C2 Position 22 Kij = 924 F= 5981
 Element: C2 Position 25 Kij = 516 F= 5799
 Element: C2 Position 27 Kij = 516 F= 6047
 Element: C2 Position 28 Kij = 833 F= 6408
 Element: C2 Advantageous position 25
 Fmin= 5799

 Matrix of placed elements on the scheme...

T1 M1 * * VD3 VD5
 VT3 * R1 * R7 VD4
 R2 VD1 * C4 C5 R3
 VD2 C8 VT1 R5 * C3
 R6 C2 FU1 * * DD1
 VT4 C9 R4 VT2 C6 C7

 Matrix of thermal coefficients of placed elements...

0 0 0 0 0 0

10 0 165 0 165 0
 165 0 0 0 0 142
 0 0 10 227 0 0
 227 0 0 0 0 555
 51 0 227 51 0 0

 Element: C1 Position 2 Kij = 165 F= 6850
 Element: C1 Position 3 Kij = 331 F= 6969
 Element: C1 Position 7 Kij = 341 F= 6692
 Element: C1 Position 9 Kij = 331 F= 6767
 Element: C1 Position 14 Kij = 403 F= 6375
 Element: C1 Position 22 Kij = 924 F= 6468
 Element: C1 Position 27 Kij = 516 F= 6502
 Element: C1 Position 28 Kij = 833 F= 6867
 Element: C1 Advantageous position 14
 Fmin= 6375

 Matrix of placed elements on the scheme...

T1 M1 * * VD3 VD5
 VT3 * R1 * R7 VD4
 R2 VD1 C1 C4 C5 R3
 VD2 C8 VT1 R5 * C3
 R6 C2 FU1 * * DD1
 VT4 C9 R4 VT2 C6 C7

 Matrix of thermal coefficients of placed elements...

0 0 0 0 0 0
 10 0 165 0 165 0
 165 0 0 0 0 142
 0 0 10 227 0 0
 227 0 0 0 0 555
 51 0 227 51 0 0

 The process is complete...

ПРИЛОЖЕНИЕ Д

Опубликованные материалы

1 XX Всероссийская научно–техническая конференция «Современные проблемы радиоэлектроники». Тема доклада – «Программа решения задачи размещения электронных компонентов с учетом их тепловых характеристик».

Аннотация: представлено решение задачи размещения электронных компонентов на печатной плате как задачи глобальной оптимизации на основе метода ветвей и границ с учетом тепловых характеристик размещаемых компонентов. Программная реализация позволяет скомплексировать критерий оптимальности при минимизации суммарной длины соединений с одновременной оптимизацией теплового режима работы печатного узла.

Авторы: Карнаухов М. А., Наврозов Д. А., Носкова Е. Е. (научный руководитель).

2 Поданы документы в Роспатент на право собственности программным продуктом – «Алгоритм метода ветвей и границ с учетом тепла».