

ВЫБОР ЭФФЕКТИВНОЙ КОНФИГУРАЦИИ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ОПТИМИЗАЦИИ

Дягель А.В.

**Научный руководитель доктор техн. наук, профессор Семенкин Е.С.
*Сибирский федеральный университет, Институт математики и
фундаментальной информатики***

Генетические алгоритмы (ГА) в различных формах применяются во многих технических и научных проблемах. Одно из наиболее важных применений генетических алгоритмов – оптимизация функций. ГА обладают важными свойствами, благодаря которым они показывают свое превосходство над традиционными методами оптимизации: работают одновременно с множеством решений, используют только значение целевой функции и не используют какую-либо дополнительную информацию, являются стохастической оптимизационной процедурой с глобальным поведением. В то же время, ГА обладают существенным недостатком: их эффективность существенно зависит от точного выбора конфигурации алгоритма, т.е. типов генетических операторов. На практике данную проблему решают различными способами, среди которых теоретический анализ сходимости алгоритма и разработкам самоконфигурируемых алгоритмов. Однако эти подходы могут быть применены только высококвалифицированными специалистами в области эволюционной оптимизации. Рядовой пользователь, например желающий использовать для решения своей задачи MathLabToolBox, вынужден прибегать к численным экспериментам с различными конфигурациями ГА и последующим статистическим анализом результатов.

В данной работе последний подход демонстрируется с помощью программной системы С.С.Бежитского, в которой реализован стандартный ГА и обеспечена возможность выбора всех основных операторов, таких как селекция(турнирная, ранговая, пропорциональна и элитарная), скрещивание(одноточечное, двухточечное и равномерное) и мутация (низкая, средняя и сильная). Рабочее окно программной системы приведено на рисунке 1.

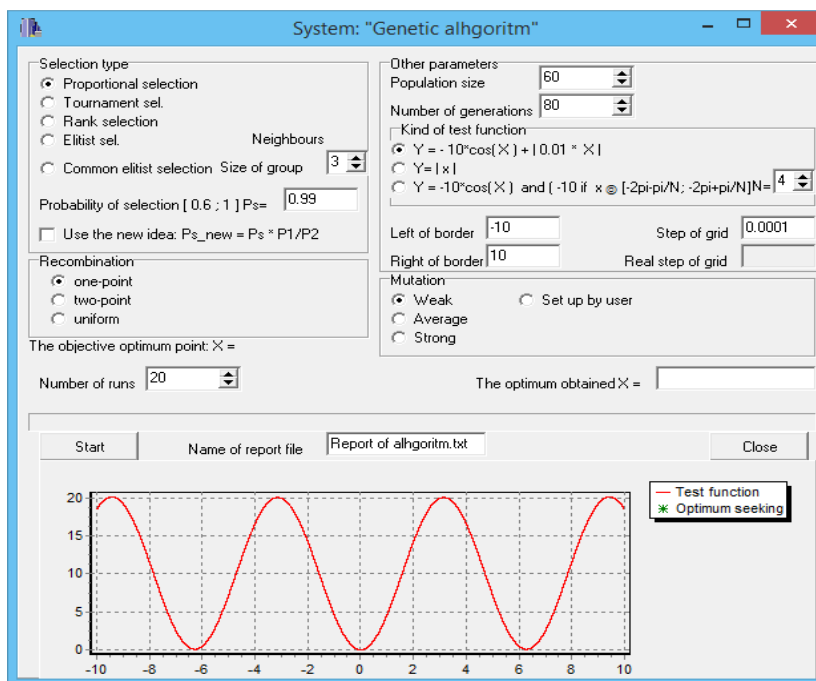


Рисунок 1 - Пример рабочего окна программной системы

Выбор эффективных конфигураций ГА проводился для задач безусловной оптимизации трех целевых функций одной вещественной переменной:

$$1) y_1 = |x|;$$

$$2) y_2 = -10 * \cos x + |0.01 * x|;$$

$$3) y_3 = \begin{cases} -10 * \cos x - 10, & \text{если } x \in \left[-2\pi - \frac{\pi}{4}; -2\pi + \frac{\pi}{4}\right], \\ -10 * \cos x, & \text{если } x \notin \left[-2\pi - \frac{\pi}{4}; -2\pi + \frac{\pi}{4}\right]. \end{cases}$$

Первая функция – одноэкстремальная, т.е. наиболее легкая для оптимизационных процедур. Вторая функция (см. рис. 1) – многоэкстремальная с локальными экстремумами, значения которых незначительно отличаются от глобального экстремума, т.е. наиболее трудная для оптимизации. Третья функция носит промежуточный характер – многоэкстремальная, но с локальными экстремумами, значительно отличающимися от глобального.

Для каждой из этих функций было проверено 45 конфигураций алгоритма – три типа скрещивания, три уровня мутации и пять типов селекции (турнирная селекция с тремя различными размерами турнира – 2, 5 и 9). В каждом эксперименте каждому алгоритму выделялся одинаковый вычислительный ресурс – 4800 вычислений целевой функции (60 индивидов в популяции, 80 поколений). Вероятность скрещивания (0.99) не изменялась, элитизм не использовался. Все результаты усреднялись по 20 запускам.

Ниже в таблицах 1-3 использованы следующие обозначения: 1т – одноточечное скрещивание, 2т – двухточечное скрещивание, «=» - равномерное скрещивание, числовое значение в столбце «Селекция» означает количество индивидов, участвующих в турнире. Значения, выделенные жирным шрифтом, являются наилучшими.

Таблица 1 –Оценки эффективности работы ГА на примере функции₁

Мутация:	Низкая			Средняя			Высокая		
Скрещивание:	1т	2т	=	1т	2т	=	1т	2т	=
Селекция									
Турнирная 2	1 16 [12;23]	1 16 [7;35]	1 14 [7;23]	1 24 [8;52]	1 22 [7;35]	1 26 [11;36]	0,6 35 [11;62]	0.4 40 [5;76]	0.55 34 [10;72]
Турнирная 5	1 8 [3;17]	1 8 [4;14]	1 6 [2;9]	1 8 [3;15]	1 7 [2;13]	1 7 [4;12]	1 14 [6;36]	1 11 [7;29]	1 11 [1;22]
Турнирная 9	1 6 [4;12]	1 6 [2;12]	1 6 [4;10]	1 7 [4;12]	1 6 [2;11]	1 6 [3;8]	1 8 [5;13]	1 8 [5;11]	1 7 [4;10]
Пропорцио- нальная	0.2 30 [14;43]	0.25 26 [4;56]	0.45 38 [10;68]	0.35 44 [15;72]	0.3 43 [18;58]	0.35 35 [6;80]	0.15 60 [44;70]	0.15 37 [11;43]	0.05 51 [51;51]
Ранговая	1 18 [4;33]	1 15 [8;28]	1 14 [7;24]	1 26 [11;53]	1 22 [6;43]	1 23 [6;45]	0.5 40 [20;78]	0.5 38 [11;73]	0.65 36 [18;60]

Таблица 2 - Оценки эффективности работы ГА на примере функции y_2

Мутация:	Низкая			Средняя			Высокая		
Скращивание:	1т	2т	=	1т	2т	=	1т	2т	=
Селекция									
Турнирная 2	0,95 21 [5;35]	0,85 25 [10;59]	0,7 19 [7;36]	0,95 34 [15;61]	0,9 23 [7;66]	0,5 23 [10;32]	0,4 27 [7;50]	0,8 21 [17;43]	0,15 31 [9;76]
Турнирная 5	0,95 14 [2;64]	1 9 [4;34]	0,9 15 [6;51]	1 9 [5;17]	1 9 [5;18]	0,9 9 [3;15]	1 15 [4;35]	1 17 [6;30]	0,85 11 [7;31]
Турнирная 9	1 7 [4;15]	0,95 13 [3;59]	1 11 [4;48]	1 8 [4;14]	1 8 [4;29]	1 7 [4;14]	1 9 [5;16]	1 9 [2;22]	1 7 [5;18]
Пропорцио- нальная	0,2 43 [9;63]	0,15 29 [3;72]	0,1 29 [15;43]	0,25 41 [9;76]	0,15 49 [16;78]	0,2 41 [27;71]	0,15 32 [12;58]	0,05 36 [36;36]	0,1 37 [14;63]
Ранговая	0,9 19 [7;50]	0,95 25 [10;55]	0,65 23 [7;66]	0,9 30 [12;74]	0,85 43 [6;74]	0,45 28 [9;52]	0,2 35 [21;64]	0,3 35 [15;58]	0,2 21 [9;65]

Таблица 3 – Оценки эффективности работы ГА на примере функции y_3

Мутация:	Низкая			Средняя			Высокая		
Скращивание:	1т	2т	=	1т	2т	=	1т	2т	=
Селекция									
Турнирная 2	0,85 27 [10;74]	0,9 15 [7;47]	0,45 19 [8;28]	0,9 20 [6;51]	0,85 16 [11;55]	0,6 11 [8;41]	0,65 23 [5;70]	0,5 21 [7;58]	0,7 29 [15;74]
Турнирная 5	0,9 22 [6;65]	0,9 21 [4;59]	0,6 32 [5;76]	0,9 7 [4;17]	0,85 6 [3;14]	0,65 6 [5;15]	0,95 26 [3;75]	0,85 10 [7;24]	0,85 9 [3;29]
Турнирная 9	0,7 12 [3;57]	0,7 22 [4;43]	0,55 19 [2;33]	0,6 24 [5;56]	0,9 9 [3;39]	0,8 8 [3;27]	0,9 9 [5;18]	0,9 5 [3;22]	0,85 11 [4;33]
Пропорцио- нальная	0,55 23 [5;65]	0,6 29 [4;77]	0,45 17 [8;31]	0,7 30 [6;77]	0,7 23 [5;61]	0,55 17 [7;64]	0,6 37 [4;48]	0,75 19 [6;58]	0,55 27 [4;61]
Ранговая	0,8 8 [2;22]	0,7 26 [6;51]	0,55 18 [6;32]	0,9 35 [7;59]	0,8 26 [10;40]	0,55 23 [9;34]	0,6 33 [4;73]	0,55 38 [3;76]	0,4 28 [10;73]

Полученные в ходе численного эксперимента оценки эффективности различных конфигураций ГА по трем критериям, приведенные в таблицах 1-3, имеют следующий смысл. Главным критерием эффективности является надежность, т.е. доля успешных запусков, когда алгоритм находил глобальный оптимум с заданной точностью (первое число в ячейках таблиц). Если несколько конфигураций демонстрируют одинаковую надежность, то лучший алгоритм определялся по минимуму затрат на поиск, показателем которых является усредненный по успешным запускам номер поколения

(второе число в ячейках таблиц), на котором впервые был найден искомый оптимум (алгоритм не останавливается в этом случае, а продолжает работу до конца, т.к. априорная информация об оптимуме ему не предоставляется). Если и второй критерий оказывался одинаковым, то лучшим алгоритмом признавался тот, у которого был минимальный разброс по номерам поколений первого обнаружения оптимума (в ячейке таблицы указан интервал разброса номеров поколений, полученных в успешных запусках).

В приведенных выше таблицах мы наблюдаем, что для первой функции самая высокая надежность (1), средний номер поколения, на котором впервые встретилось оптимальное решение (6) и минимальный разброс поколений [3;8] достигаются при средней мутации, равномерном скрещивании и турнирной селекции с размером турнира 9. У второй функции самая высокая надёжность (1), средний номер поколения (7) и минимальный разброс поколений [4;14] достигаются при средней мутации, равномерном скрещивании и турнирной селекции с размером турнира 9. У третьей функции максимальная надёжность (0,95), средний номер поколения (26) и разброс поколений [3;75] достигаются при высокой мутации, одноточечном скрещивании и турнирной селекции с размером турнира 5.

Дополнительные наблюдения состоят в том, что на разных задачах лучшие конфигурации существенно отличаются, при этом на всех задачах неудачный выбор конфигурации алгоритма может привести к его неэффективной работе (надежность 0.05, 0.1), поэтому выбор настроек «наугад» неприемлем. В то же время существуют различные удачные конфигурации, дающие очень высокую надежность (1.00, 0.95), хотя и при очень разных затратах на оптимизацию. Надежно установить лучшую конфигурацию можно, но это требует многократного решения одной и той же задачи (в нашем случае каждая задача была решена по 900 раз), что не может быть признано удачным подходом для практических задач, обычно требующих серьезных вычислительных ресурсов на однократное вычисление целевой функции.

Поэтому основным подходом в развитии эволюционных методов оптимизации должно быть внедрение в практику самонастраивающихся алгоритмов, разработанных экспертами.