

Министерство образования и науки Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

*Кафедра компьютерной физики и метаматериалов  
на базе Саратовского филиала Института  
радиотехники и электроники  
имени В. А. Котельникова РАН*

**Практические аспекты применения генетических алгоритмов**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ  
студентки 4 курса 431 группы  
направления 03.03.02 «Физика» физического факультета

Тумановой Татьяны Александровны

Научный руководитель

доцент, к.ф.- м.н.,

\_\_\_\_\_  
должность, уч. степень, уч. звание

\_\_\_\_\_  
подпись, дата

А.С. Ремизов

\_\_\_\_\_  
инициалы, фамилия

Зав. кафедрой

д.ф.- м.н., профессор

\_\_\_\_\_  
должность, уч. степень, уч. звание

\_\_\_\_\_  
подпись, дата

В.М. Аникин

\_\_\_\_\_  
инициалы, фамилия

Саратов, 2017

## Введение

**Актуальность работы.** Генетические алгоритмы относятся к классу эвристических алгоритмов и являются достаточно перспективным направлением в области оптимизации и моделирования. Они позволяют найти удовлетворительное решение за приемлемое время вычислений для аналитически неразрешимых или сложнорешаемых проблем.

**Целью** данной квалификационной работы является изучение возможностей генетических алгоритмов в задачах оптимизации. В **задачи** квалификационной работы входит: методическое изложение теоретической базы, необходимой в работе; описание возможностей генетических алгоритмов при решении прикладных задач; апробация генетического алгоритма для задачи оптимизации функции двух переменных.

**Структура и объем работы.** Выпускная квалификационная работа изложена на 47 страницах, состоит из введения, 4 разделов и заключения. Библиографический список включает 11 наименований. Текст содержит 5 таблиц и иллюстрирован 13-ю рисунками.

### Основное содержание работы

В **первой главе** излагаются основные идеи, понятия, терминология и принципы работы генетических алгоритмов (ГА). Также рассмотрено несколько разновидностей - канонический ГА, модель генитор (Genitor), метод прерывистого равновесия, гибридные алгоритмы, СНС (Cross-population selection, Heterogeneous recombination and Cataclysmic mutation), ГА с нефиксированным размером популяции.

**Вторая глава** посвящена вопросам параллельного выполнения генетических алгоритмов, приведены три варианта моделей с естественным параллелизмом - модель "Рабочий и Хозяин", модель диффузии ("островная" модель), и миграционная модель.

В **третьей главе** представлена символьная модель ГА, ее геометрическая интерпретация и теорема шаблонов (The Schema Theorem), приведённая Холландом.

дом. Теорема показывает, как изменяется доля представителей шаблона в популяции, являясь первой попыткой объяснить правильность работы генетических алгоритмов. Следует заметить, что она верна только для классического ГА с его пропорциональным отбором и одноточечным кроссинговером.

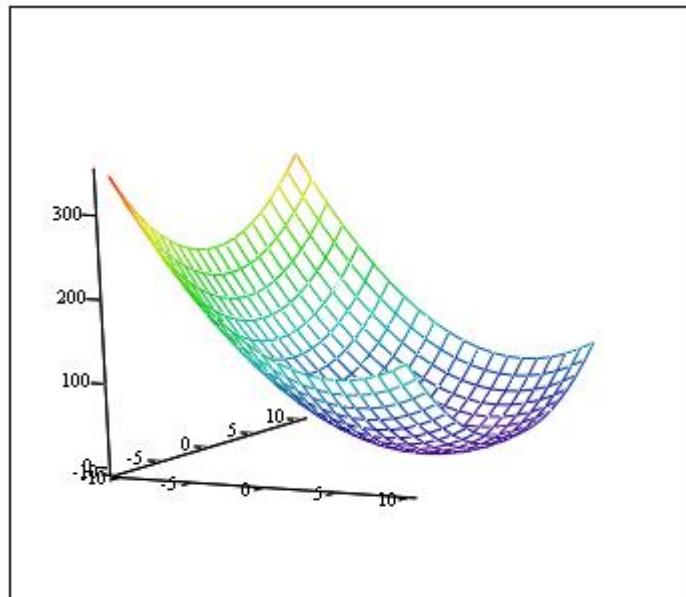
В **третьей главе** рассказывается о вычислительных проблемах при использовании генетических алгоритмов, и некоторых путях их решения. Говорится о проблеме преждевременной сходимости, выборе оптимальных параметров, самоадаптирующихся алгоритмах, рассматриваются достоинства и недостатки.

### Содержание работы

В качестве практического примера в работе анализируется генетический алгоритм для поиска минимума функции двух переменных:

$$f(x, y) = (x - 5)^2 + (y - 1)^2$$

График этой функции представлен на рисунке 1.



f

Рисунок 1.

Оптимизация с помощью генетического алгоритма происходит следующим образом:

1. На первом этапе строится начальная популяция приближений искомого решения, ее можно формировать случайным образом. Применительно к задаче оптимизации функции двух переменных популяция — это набор векторов  $(x_{11}, x_{12}), (x_{21}, x_{22}), \dots, (x_{n1}, x_{n2})$ , где  $n$  — размер популяции.

2. Выбирается пара решений, каждое значение с определенной вероятностью мутирует на какое-либо небольшое значение, и, наконец, формируется новый член популяции — так, что часть его компонент взяты от одного родительского элемента, а часть от другого. Если  $(x_1, x_2)$  и  $(y_1, y_2)$  — родительские элементы, то они продуцируют нового члена популяции вида:  $(x_1 + \Delta_1, y_2 + \Delta_2)$ , где  $\Delta_1, \Delta_2$  — некоторые небольшие случайные значения, характеризующие степень мутации.

3. Предыдущий шаг повторяется некоторое определенное число раз, в результате — популяция увеличивается на данное число новых членов.

4. Далее происходит шаг "естественного" отбора — для каждого члена популяции вычисляется соответствующее значение оптимизируемой функции. Этот массив значений упорядочивается по возрастанию и в следующую популяцию переходит только определенный процент членов, приведших к наименьшим значениям функции.

5. Процедура повторяется с шага 2 и заканчивается, например, по прошествии 100 подобных повторений (эпох).

Этот простейший вариант эволюционного поиска минимума функции двух переменных был реализован на MathCad.

Были выбраны следующие параметры алгоритма:

- размер начальной популяции,  $pSize = 100$ ;
- максимальное количество пар для скрещивания,  $pairsMax = 100$ ;
- доля особей в новой популяции, которые переходят дальше,  $diePart = 0,2$
- критерий остановки - количество шагов,  $nSteps = 200$ ;
- вероятность мутации  $probMut = 0,8$ ;

Начальная популяция была инициализирована случайными числами с равномерным распределением из интервала (0,1) следующим образом:

```
Ppl :=  $\left\{ \begin{array}{l} X \leftarrow \text{runif}(pSize, 0, 1) \\ Y \leftarrow \text{runif}(pSize, 0, 1) \\ \text{for } i \in 0..pSize - 1 \\ \quad P_i \leftarrow (X_i \ Y_i)^T \\ \text{return } P \end{array} \right.$ 
```

В основном алгоритме используется ряд вспомогательных функций (методов). Метод, производящий мутацию одной пары координат:

```
mutate(pair) :=  $\left\{ \begin{array}{l} p \leftarrow \text{runif}(1, 0, 1)_0 \\ \text{pair} + (\text{runif}(2, 0, 1) - 0.5) \cdot 0.05 \text{ if } p < \text{probMut} \\ \text{pair} \text{ otherwise} \end{array} \right.$ 
```

Метод, осуществляющий кроссинговер:

```
crossover(a, b) :=  $\left\{ \begin{array}{l} p \leftarrow \text{runif}(1, 0, 1)_0 \\ (a_0 \ b_1)^T \text{ if } p > 0.5 \\ (b_0 \ a_1)^T \text{ otherwise} \end{array} \right.$ 
```

Метод, который определяет уникальные комбинации пар для скрещивания:

```
cIPairs(pp1) :=  $\left\{ \begin{array}{l} \text{for } i \in 0..length(pp1) - 1 \\ \quad I_i \leftarrow i \\ \text{return submatrix}(FComb(I, 2), 0, 1, 0, \text{pairsMax} - 1) \end{array} \right.$ 
```

в данном методе используется функция  $FComb(v, k)$ , которая по заданному вектору из целых чисел генерирует неповторяющиеся группы значений (в данном случае - пары, т.к. у нас  $k=2$ ), и возвращает их в виде матрицы. Например, комбинации неповторяющихся пар значений из множества 1,2,3:

$$FComb[(1 \ 2 \ 3)^T, 2] = \begin{pmatrix} 1 & 1 & 2 \\ 2 & 3 & 3 \end{pmatrix}$$

Поскольку код этой функции достаточно громоздкий, и не относится непосредственно к тематике генетических алгоритмов, приводить его не будем.

Метод, осуществляющий "естественный отбор" (полученные после мутации и скрещивания особи отбираются для следующего поколения исходя из их пригодности - меньшим значениям целевой функции):

```

reduce(pp1) := | for n ∈ 0..length(pp1) - 1
                |   Xn ← (pp1n)0
                |   Yn ← (pp1n)1
                |   Fn ← f(Xn, Yn)
                |   A ← augment(F, X, Y)
                |   sorted ← csort(A, 0)
                |   for i ∈ 0..ceil(length(pp1)·diePart)
                |     reducedi ← (sortedi,1 sortedi,2)T
- 1 | return reduced

```

И, наконец, метод, осуществляющий эволюционные вычисления:

```

evolve(pp1) := | for n ∈ 0..nSteps - 1
                |   newPpl ← pp1
                |   cIPs ← cIPairs(pp1)
                |   for i ∈ 0..length[(cIPsT)<0>] - 1
                |     | a ← mutate(pp1cIPs0,i)
                |     | b ← mutate(pp1cIPs1,i)
                |     | newItem ← crossover(a, b)
                |     | newPpllength(newPpl) ← newItem
                |   ppl ← reduce(newPpl)
                |   for k ∈ 0..length(ppl) - 1
                |     | Xk ← (pplk)0
                |     | Yk ← (pplk)1
                |     | Fk ← f(Xk, Yk)
                |   A ← augment(F, X, Y)
                |   sorted ← csort(A, 0)
                | return sorted

```

При выбранных параметрах алгоритма, получился следующий результат:

	0	1	2
11	$7.797 \cdot 10^{-8}$	5	1
12	$7.797 \cdot 10^{-8}$	5	1
13	$7.797 \cdot 10^{-8}$	5	1
14	$7.797 \cdot 10^{-8}$	5	1
15	$7.797 \cdot 10^{-8}$	5	1
16	$7.797 \cdot 10^{-8}$	5	1
17	$7.797 \cdot 10^{-8}$	5	1
evolve(Ppl) = 18	$7.797 \cdot 10^{-8}$	5	1
19	$7.797 \cdot 10^{-8}$	5	1
20	$7.797 \cdot 10^{-8}$	5	1
21	$7.797 \cdot 10^{-8}$	5	1
22	$7.797 \cdot 10^{-8}$	5	1
23	$7.797 \cdot 10^{-8}$	5	1
24	$7.797 \cdot 10^{-8}$	5	1
25	$7.797 \cdot 10^{-8}$	5	1
26	$9.768 \cdot 10^{-8}$	5	...

Это последнее поколение в наших вычислениях, как видим, все выжившие особи сошлись в одну точку с координатами (5,1). Это и есть результат вычислений - в данной точке наша функция имеет минимум, а именно, как нетрудно видеть из ее явного вида, в данной точке функция обращается в ноль.

### Заключение

На основании проведенного анализа можно сделать выводы о том, что в настоящее время генетические алгоритмы являются мощным вычислительным средством в разнообразных оптимизационных задачах. ГА отличаются от большинства других оптимизационных и поисковых методов в следующем, они:

- оперируют закодированным множеством параметров, а не с самими параметрами;
- находят популяцию точек, а не отдельную точку;

- используют значение целевой функции, а не ее производную или другие вспомогательные значения;
- применяют вероятностное правило перехода, а не детерминистическое.

Для применения ГА необходимо прежде всего выбрать функцию пригодности (цели) адекватную задаче. Причем целевая функция должна иметь разнообразный рельеф, так как если на поверхности функции есть большие плоские участки, то ГА неэффективен. Это связано с тем, что многие особи в популяции при различии в генотипе не будут отличаться фенотипом, то есть несмотря на то, что особи различны, они имеют одинаковую пригодность, а значит алгоритм не имеет возможности выбрать лучшее решение и направление дальнейшего развития. Эта проблема известна как «проблема поля для гольфа», где все пространство абсолютно одинаково, за исключением лишь одной точки, которая и является оптимальным решением — в этом случае ГА просто остановится или будет блуждать абсолютно случайно. Другое требование к функции пригодности, состоит в требовании минимума вычислительных ресурсов при ее оценке, так как это влияет на скорость алгоритма.



## Список использованных источников

1. *Панченко Т.В.* Генетические алгоритмы: учеб.-метод. пособие. Астрахань: АГУ, 2007. 87 с.;
2. *Гладков Л.А., Курейчик В.В., Курейчик В.М.* Генетические алгоритмы/ Под ред. В.М. Курейчика. 2-е изд., испр. и доп. М.: ФИЗМАТЛИТ, 2006. 320 с.;
3. *Курейчик В. М., Лебедев Б. К., Лебедев О. К.* Поисковая адаптация: теория и практика. М: Физматлит, 2006. С. 272.;
4. *Гладков Л. А., Курейчик В. В, Курейчик В. М. и др.* Биоинспирированные методы в оптимизации: монография. М: Физматлит, 2009. С. 384.;
5. *Рутковская Д., Пилиньский М., Рутковский Л.* Нейронные сети, генетические алгоритмы и нечеткие системы. Sieci neuronowe, algorytmy genetyczne i systemy rozmyte. 2-е изд. — М: Горячая линия-Телеком, 2008. С. 452.;
6. *Скобцов Ю. А.* Основы эволюционных вычислений. Донецк: ДонНТУ, 2008. С. 326.;
7. *Батищев Д.И., Неймарк Е.А., Старостин Н.В.* Применение генетических алгоритмов к решению задач дискретной оптимизации. Нижний Новгород: Нижегородский гос. ун-т им. Н. И. Лобачевского, 2007. 85 с.
8. *Исаев А.* Генетические алгоритмы [Электронный ресурс]. URL: <http://www.algolist.manual.ru>
9. *Исследования по ГА в Мичиганском университете.* [Электронный ресурс]. URL: <http://garage.cse.msu.edu>
10. *Исследования по ГА в университете штата Колорадо.* [Электронный ресурс]. URL: <http://www.cs.colostate.edu>
11. *Работы по генетическим алгоритмам.* СПб : Университет ИТМО, кафедра «Технологии программирования» [Электронный ресурс]. URL: <http://is.ifmo.ru/genalg/>