Transport and Telecommunication, 2007, Volume 8, No 1, 40–52
Transport and Telecommunication Institute, Lomonosov 1, Riga, LV-1019, Latvia

СРАВНИТЕЛЬНЫЙ АНАЛИЗ РЕШЕНИЯ ЗАДАЧ ОПТИМИЗАЦИИ ГЕНЕТИЧЕСКИМИ И ГРАДИЕНТНЫМИ МЕТОДАМИ

Олег Чернышев*, Аркадий Борисов**

**Департамент компьютерных наук, Институт транспорта и связи*
ул. Ломоносова, 1, Рига, LV-1019, Латвия
E-mail: poluno4nik@list.ru

***Кафедра моделирования и имитации, Рижский технический университет*
ул. Калькю, 1, Рига, LV-1658, Латвия
E-mail: Arkadijs.Borisovs@cs.rtu.lv

Данная работа посвящена анализу решения задач оптимизации генетическими и градиентными методами. Для решения задачи сравнительного анализа были разработаны два программных продукта, основной частью которых являются генетический алгоритм для первого и градиентные методы для второго. Данные программные продукты реализованы на языке *Pascal* в среде *Delphi*. Каждый метод был исследован отдельно, а также был произведен сравнительный анализ данных методов.

Ключевые слова: *генетический алгоритм, метод Ньютона, метод наискорейшего спуска, задача оптимизации без ограничений*

1. Введение

Генетические алгоритмы (ГА) представляют собой новое направление в вычислительных методах [3, 4]. В последние годы появилось множество публикаций, посвященных описанию принципов генетических алгоритмов, основанных на концепциях естественного отбора и генетики. И очень часто их возможности демонстрируются на примере решения задач оптимизации и поиска. Традиционно задачи оптимизации решаются градиентными методами. Поэтому сравнение двух подходов представляет научный и практический интерес.

Генетические алгоритмы – поисковые алгоритмы, основанные на механизмах натуральной селекции и натуральной генетики. Они реализуют «выживание сильнейших» среди рассмотренных структур, формируя и изменяя поисковый алгоритм на основе моделирования эволюции поиска. В каждой генерации новое множество искусственных последовательностей создается путем использования части старых и добавления новых частей с «хорошими свойствами». При этом ГА – это не просто случайный поиск, поскольку в них эффективно используется информация, накопленная в процессе эволюции [1, 2].

ГА отличаются от других оптимизационных и поисковых процедур следующим:

- они работают в основном не с переменными задачи, а с закодированным множеством параметров;
- они осуществляют поиск из популяции точек, а не из единственной точки;
- для оценки информации они используют целевую фитнесс-функцию, а не ее различные приращения;
- они используют не детерминированные, а вероятностные правила.

ГА берет множество естественных примеров оптимизационной проблемы и кодирует их как последовательность конечной длины в некотором конечном алфавите.

ГА был впервые описан Д. Голдбергом на основе работ Дж. Холланда [1, 2]. Механизм генетического алгоритма несложен. Он копирует последовательности и переставляет их части. Предварительно генетический алгоритм случайно генерирует популяцию строк-стрингов (хромосом). Затем применяется множество простых операций к начальной популяции и генерируются новые популяции.

Генетический алгоритм включает 3 оператора:

- репродукцию;
- скрещивание;
- мутацию.

Метод наискорейшего спуска на каждой итерации выбирает шаг λ_k из условия минимума функции $f(x)$ в направлении движения, т.е.

$$f(x^{(k-1)} - \lambda_k f'(x^{(k-1)})) = \min_{\lambda>0} \varphi(\lambda), \tag{1}$$

где $\varphi(\lambda) = f(x^{(k-1)} - \lambda f'(x^{(k-1)}))$.

Такой способ выбора λ_k требует меньшего числа итераций, так как он обеспечивает достижение наименьшего значения функции вдоль заданного направления. Однако в этом варианте градиентного метода на каждой итерации требуется решать задачу одномерной минимизации, что приводит к увеличению трудоемкости поиска.

Метод Ньютона предназначен для численного решения задач безусловной минимизации. Метод Ньютона основан на идее замены минимизируемой функции *f(x)* в окрестности точки x^k ее квадратичной аппроксимацией:

$$q(x) = f(x^k) + \nabla f^T(x^k)(x - x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)(x - x^k). \tag{2}$$

Возьмем в качестве точки x^{k+1} точку минимума функции $q(x)$, если такая существует. Это возможно лишь в случае, когда $\nabla^2 f(x^k)$ будет положительно определенной матрицей. Тогда функция $q(x)$ строго выпукла и имеет единственную точку минимума x^{k+1}, определяемую равенством $\nabla q(x^{k+1}) = 0$. Это приводит к линейной системе $\nabla f(x^k) = -\nabla^2 f(x^k)(x^{k+1} - x^k)$, из которой получается итерационная формула:

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k). \tag{3}$$

Формула (3) является методом Ньютона. Данный метод представляет собой метод второго порядка.

Метод Ньютона находит минимум квадратичной функции за один шаг независимо от начальной точки $x^{(0)}$ и степени овражности. Однако сходимость метода Ньютона в случае, когда целевая функция не является квадратичной, существенно зависит от начальной точки $x^{(0)}$. Еще одним недостатком является высокая трудоемкость метода, обусловленная необходимостью вычисления и обращения на каждом шаге матрицы вторых производных минимизируемой функции.

В данной работе для тестирования методов и проведения экспериментов над методами используются 3 тестовые функции:

- функция Де Джонга,
- функция Розенброка,
- функция Растригина.

Функция Де Джонга – самая простая тестовая функция (см. рис. 1). Она также известна как модель сферы. Эта функция непрерывная, выпуклая и унимодальная. Она считается легкой функцией для любого метода оптимизации. Данная функция имеет следующее описание (4):

$$f(x) = \sum_{i=1}^{n} x_i^2. \tag{4}$$

Функция имеет глобальный минимум в точке $x_i = 0$ и $f(x) = 0$ при $i = 1...n$.

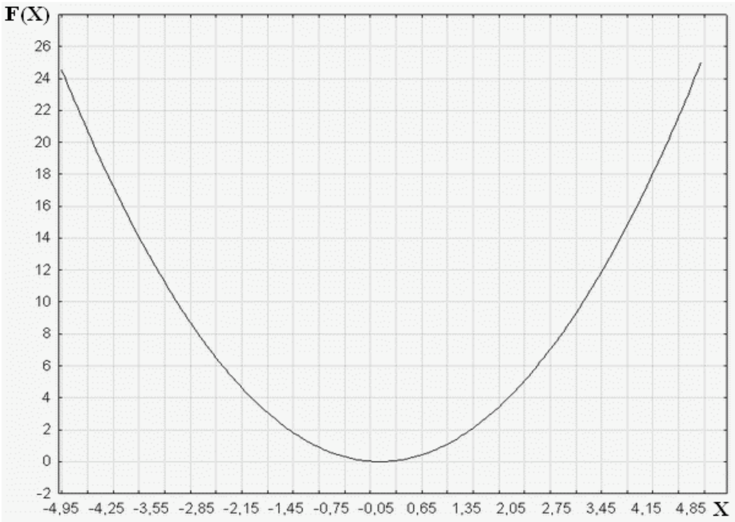

Рис. 1. *Тестовая функция Де Джонга*

Функция Розенброка (см. рис. 2) – классическая проблема оптимизации, также известная как Банановая функция. Функция имеет большое медленно убывающее плато. Найти плато – тривиальная задача, однако конвергенция к глобальному оптимуму трудна, и, следовательно, эта проблема используется для оценки работы алгоритмов оптимизации. Данная функция имеет следующее описание (5):

$$f(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2). \tag{5}$$

Функция имеет глобальный минимум в точке $x_i = 1$ и $f(x) = 0$ при $i = 1...n$.

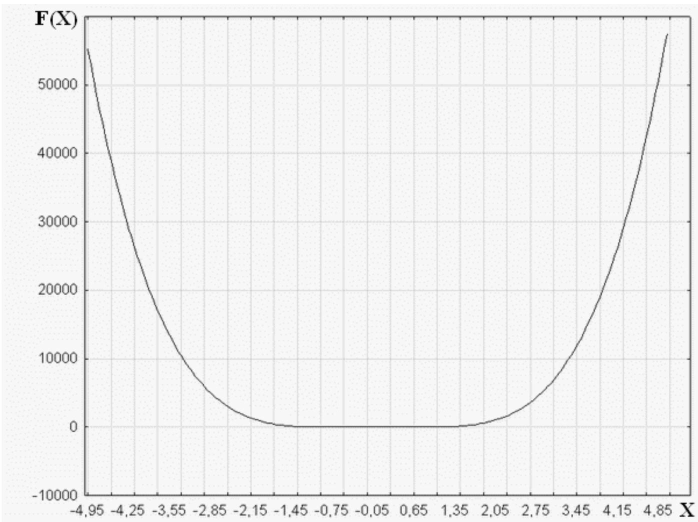

Рис. 2. *Тестовая функция Розенброка*

Функция Растригина (см. рис. 3) основана на функции Де Джонга с дополнением модуля косинуса для получения множества локальных минимумов. Таким образом, данная функция является мультимодальной. Однако расположение минимумов распределяется регулярно. Функция считается сложной для оптимизации из-за сложности рельефа. Данная функция имеет следующее описание (6):

$$f(x) = 10 * n + \sum_{i=1}^{n} (100(x_i^2 - 10 * \cos(2 * \pi * x_i)). \tag{6}$$

Функция имеет глобальный минимум в точке $x_i = 0$ и $f(x) = 0$ при $i = 1...n$.

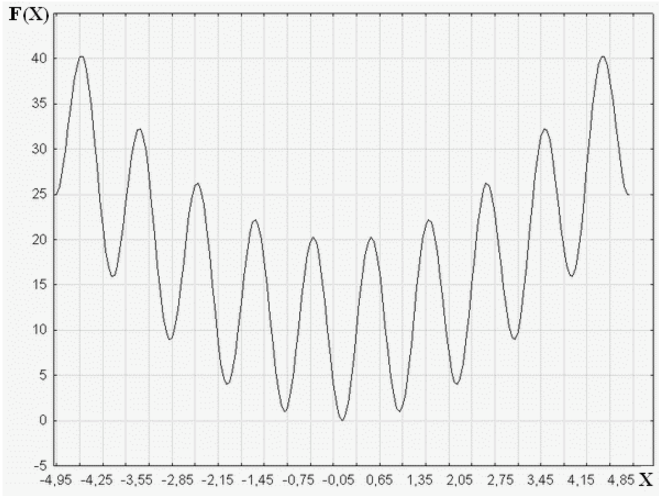

Рис. 3. *Тестовая функция Растригина*

2. Цель исследования

Исследование работы генетического алгоритма и градиентных методов проводилось на основе трех указанных тестовых функций. В процессе исследования был использован программный продукт, разработанный для работы с ГА.

Перечислим основные задачи исследования:

- исследовать работу генетического алгоритма для нахождения оптимума функций,
- исследовать влияние мутации на эффективность работы генетического алгоритма,
- исследовать влияние скрещивания на эффективность работы генетического алгоритма,
- исследовать влияние видов скрещивания на результат работы генетического алгоритма,
- исследовать скорость работы генетического алгоритма в зависимости от набора задаваемых параметров.

Для проведения исследования работы градиентных методов был использован программный продукт, разработанный для работы с градиентными методами. Исследование проводилось по следующим задачам:

- исследовать работу двух градиентных методов – метода наискорейшего спуска и метода Ньютона нахождения оптимума функций,
- сравнить метод наискорейшего спуска и метод Ньютона.

Основываясь на полученных результатах, необходимо было провести сравнение градиентных методов и генетического алгоритма по следующим критериям:

- скорость сходимости,
- точность сходимости.

3. Программная реализация

Программный продукт для работы с генетическим алгоритмом позволяет вводить все необходимые параметры для работы генетического алгоритма, а также выводить результаты работы на каждом шаге.

Пользовательский интерфейс данного программного продукта состоит из трех частей:

- кнопки управления,
- вывода результатов работы генетического алгоритма,
- панели ввода параметров для работы генетического алгоритма.

Для более детального изучения полученных результатов данный программный продукт строит два графика, представленных на рис. 4 и 5.

- График фитнесс-функции (см. рис. 4) показывает значение фитнесс-функций на данном шаге относительно оптимального значения фитнесс-функции, найденного за время поиска оптимума в виде графика. На данном рисунке по горизонтали идут фитнесс-функции, а по вертикали – их значения. Левый столбец показывает оптимальное значение фитнесс-функции, найденное за время поиска оптимума, а остальные показывают значения фитнесс-функции для стринга на данном шаге.

- График сходимости (см. рис. 5) показывает сходимость ГА. Значение сходимости для данного метода определяется совпадением значений фитнесс-функций. Чем больше фитнесс-функций индивидов принимают одинаковые значения, тем ближе к оптимуму. По данному рисунку видно, что по горизонтали располагается номер итерации, а по вертикали – значения сходимости на данной итерации.

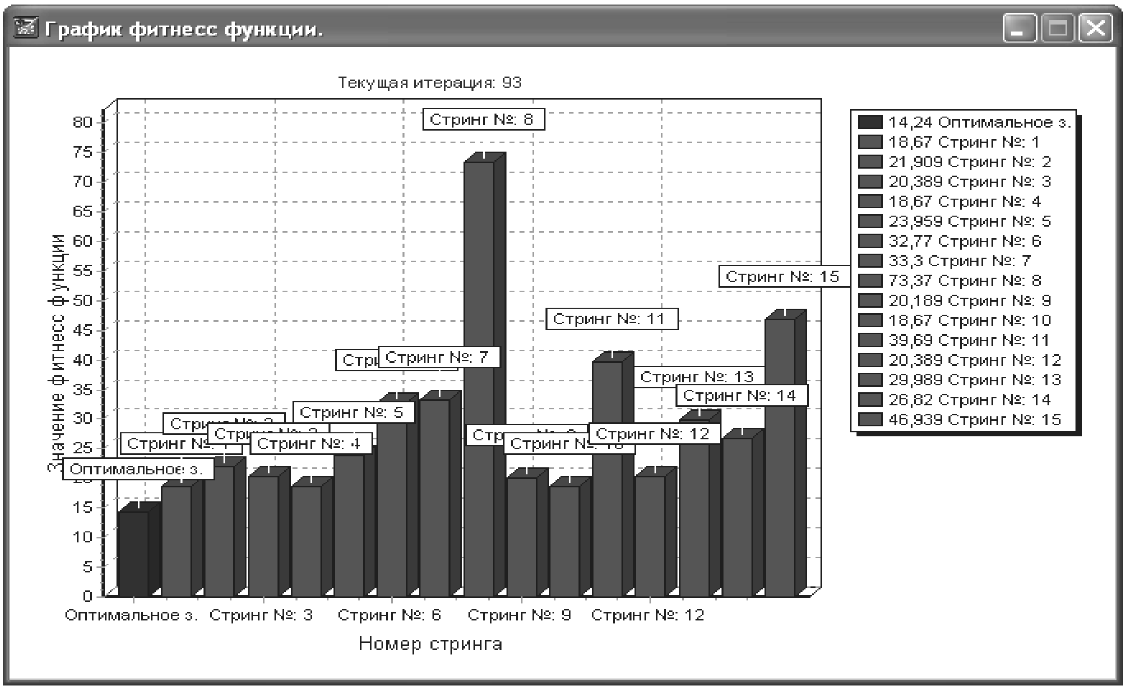

Рис 4. *График фитнесс-функции*

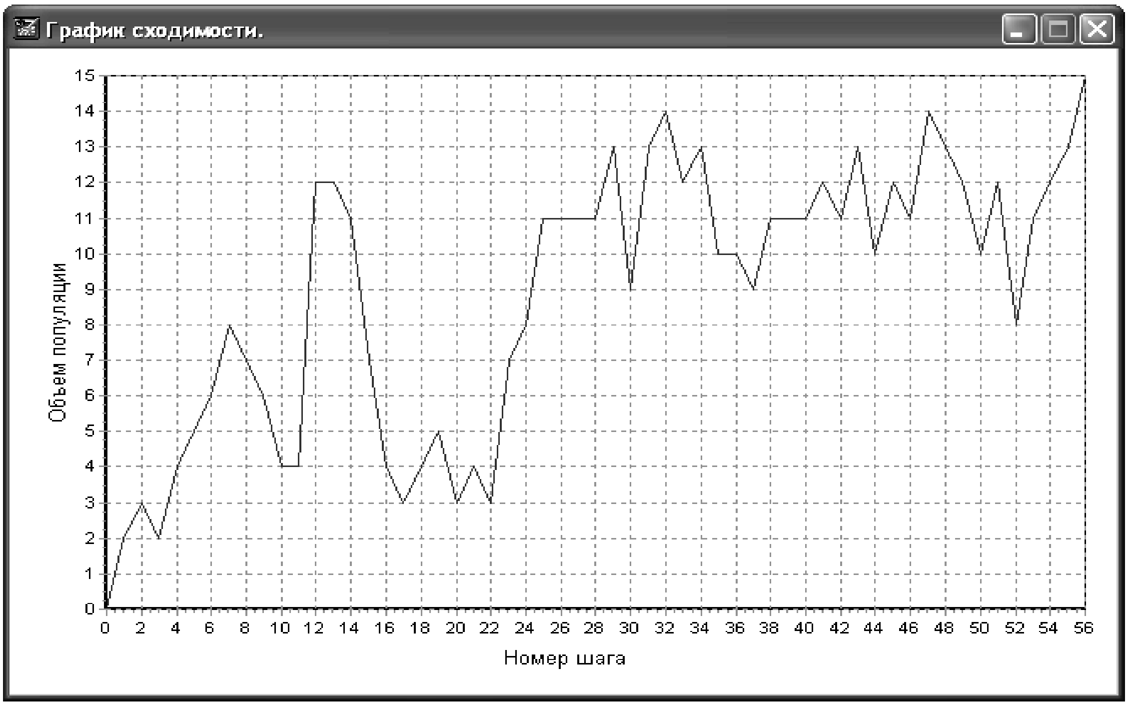

Рис 5. *График сходимости*

Программный продукт для работы с градиентными методами, так же как и программный продукт для работы с генетическим алгоритмом, позволяет вводить все необходимые параметры и выводить результаты работы на каждом шаге.

Пользовательский интерфейс данного программного продукта состоит из трех частей:

- кнопки управления,
- вывода результатов работы градиентных методов,
- панели ввода параметров для работы градиентных методов.

Для более детального изучения полученных результатов данный программный продукт строит график сходимости (см. рис. 6). По данному рисунку видно, что по горизонтали располагается номер итерации, а по вертикали – значения функции на данной итерации.

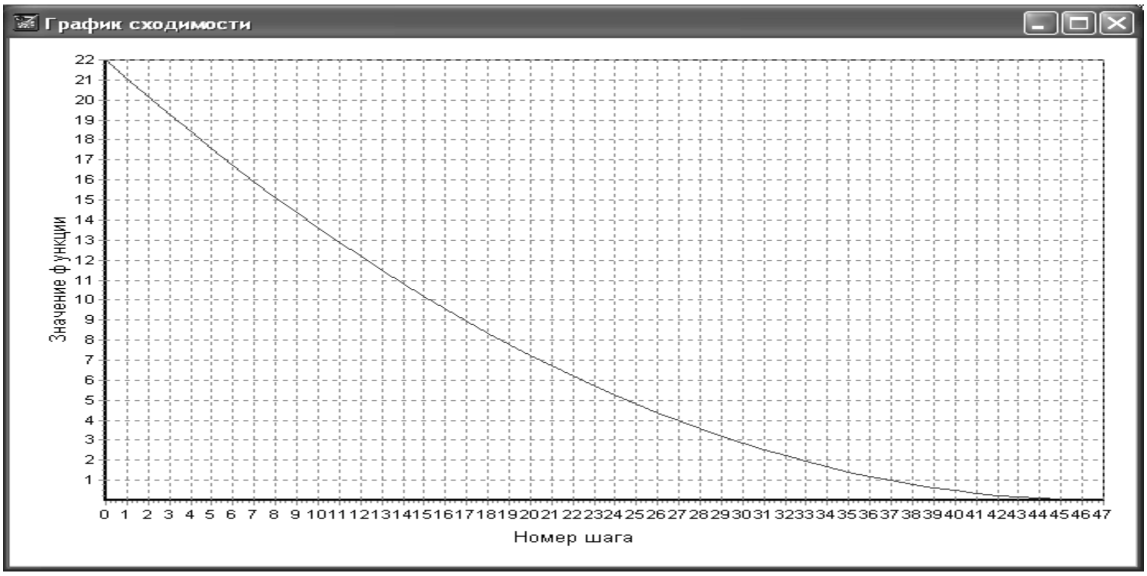

Рис 6. *График сходимости*

4. Полученные результаты

Влияние мутации на результат работы генетического алгоритма

Построим два графика: среднее количество итераций до полной сходимости алгоритма (рис. 7), и средние значения оптимума функции (рис. 8) для каждой вероятности мутации.

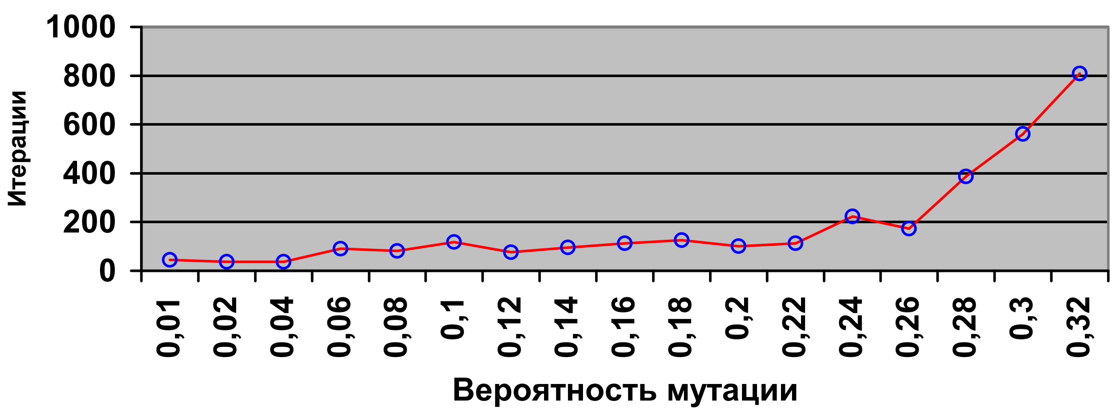

Рис. 7. *Среднее количество итераций до полной сходимости алгоритма*

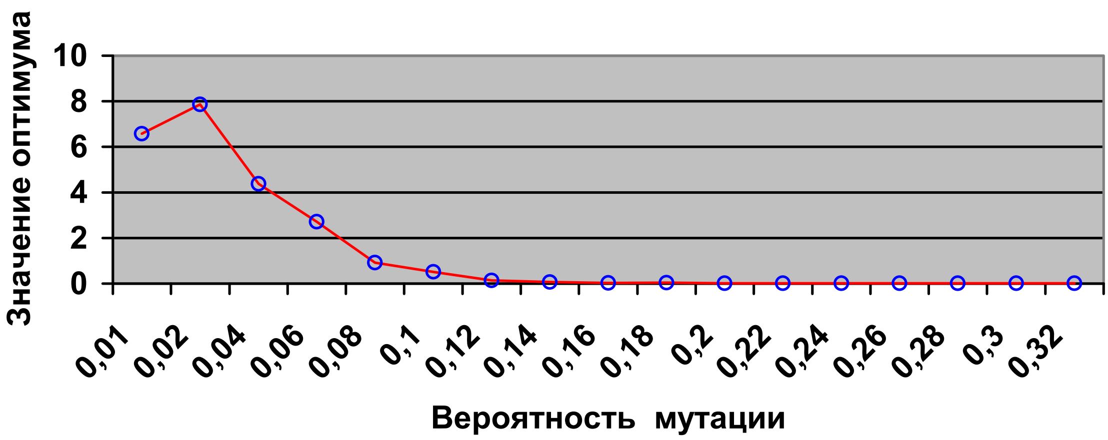

Рис. 8. *Средние значения оптимума функции*

Проанализировав полученные результаты в ходе экспериментов, можно сделать вывод, что наиболее оптимальной вероятностью мутации является вероятность от 0,18 до 0,22.

Влияние скрещивания на результат работы генетического алгоритма

Построим два графика: среднее количество итераций до полной сходимости алгоритма (рис. 9) и средние значения оптимума функции (рис. 10) для каждой вероятности скрещивания.

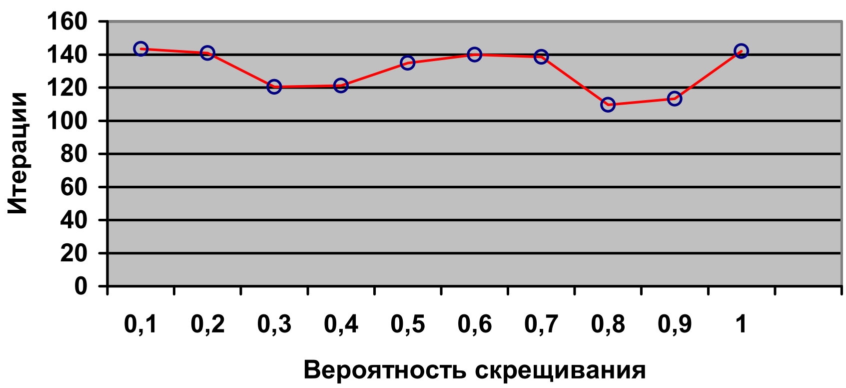

Рис. 9. *Среднее количество итераций до полной сходимости алгоритма*

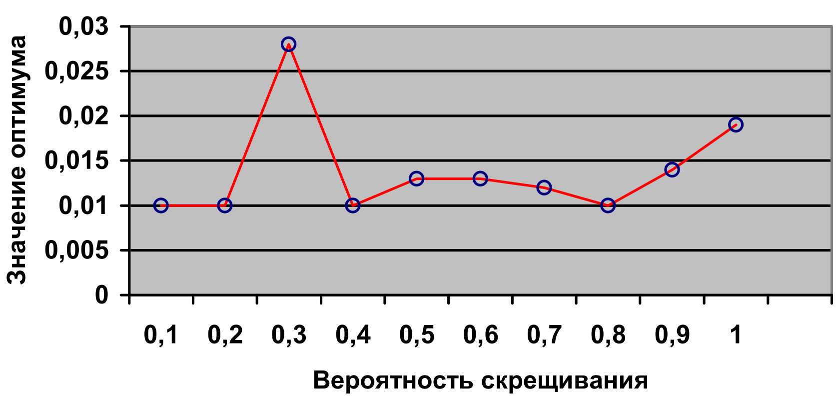

Рис. 10. *Средние значения оптимума функции*

Проанализировав полученные в ходе экспериментов результаты, можно сделать вывод, что наиболее оптимальная вероятность скрещивания составляет 0,8.

Влияние объема популяции на результат работы генетического алгоритма

Построим два графика: среднее количество итераций до полной сходимости алгоритма (рис. 11) и средние значения оптимума функции (рис. 12) для разного объема популяции.

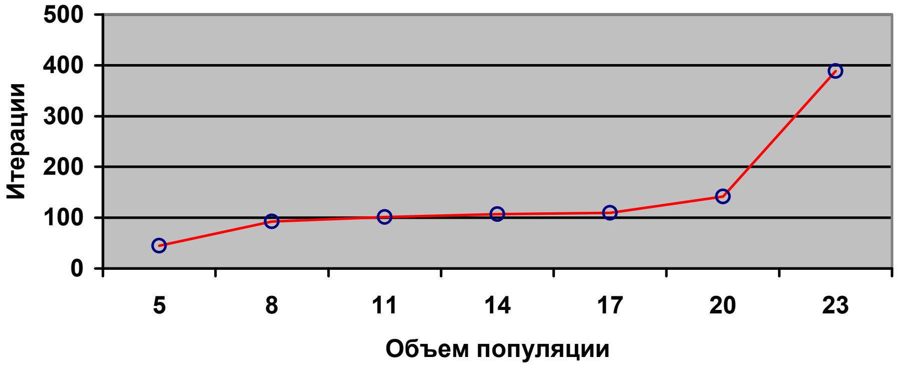

Рис. 11. Среднее количество итераций до полной сходимости алгоритма

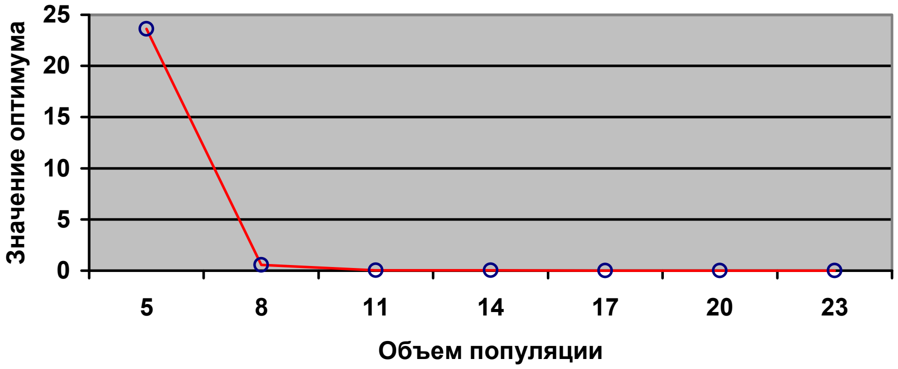

Рис. 12. Средние значения оптимума функции

Проанализировав результаты, полученные в ходе экспериментов, можно сделать вывод, что наиболее оптимальный объем популяции является объем от 15 до 17. Т.е. можно сделать вывод, что для одной переменной необходимы 3 стринга для оптимальной работы генетического алгоритма.

Исследование работы генетического алгоритма для нахождения оптимума функций

Построим два графика: среднее количество итераций (рис. 13) и средние значения оптимума функции (рис. 14) для разного количества переменных.

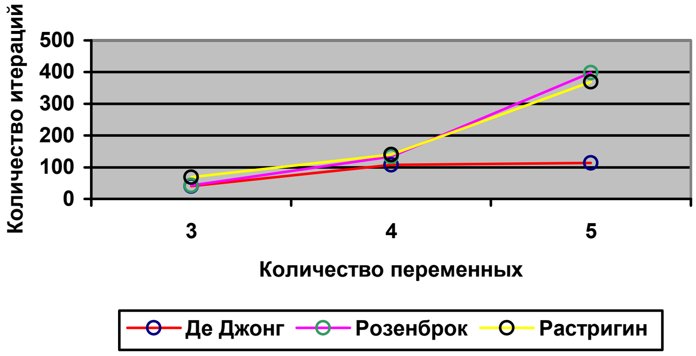

Рис 13. Среднее количество итераций

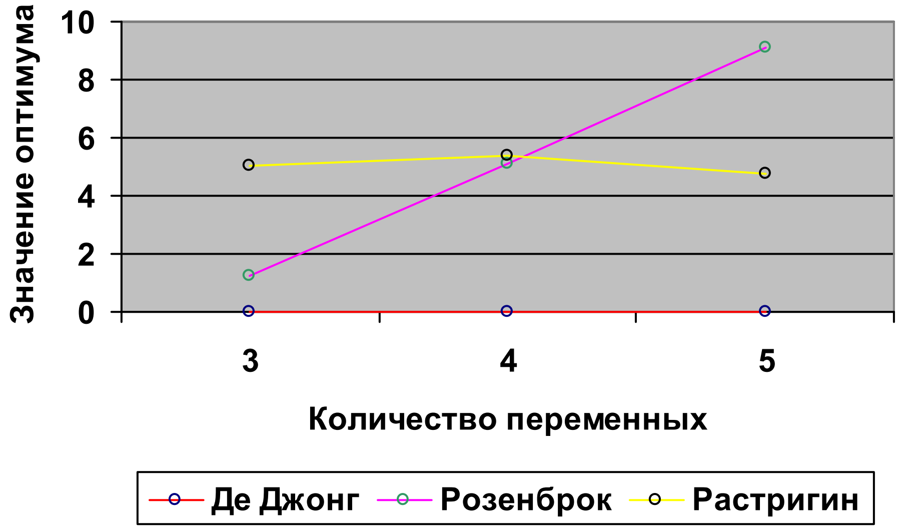

Рис. 14. *Средние значения оптимума функции*

По результатам, полученным в ходе экспериментов, можно сделать вывод, что с увеличением количества переменных увеличивается время поиска оптимума и увеличивается количество итераций. При увеличении количества переменных в среднем разница между полученными значениями оптимумов для каждой функции меняется незначительно.

Сравнение метода наискорейшего спуска и метода Ньютона

Построим графики для наглядного представления полученных результатов и сравнению методов. Для построения графиков будем использовать результаты при начальном приближении 5.

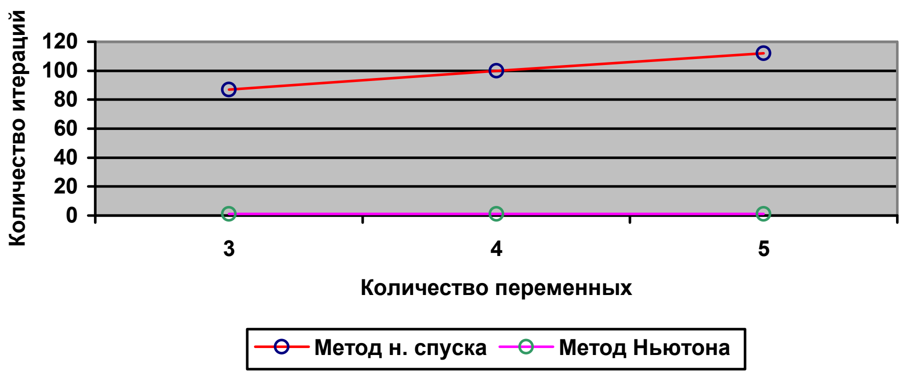

Рис. 15*. Количество итераций для функции Де Джонга до нахождения оптимума*

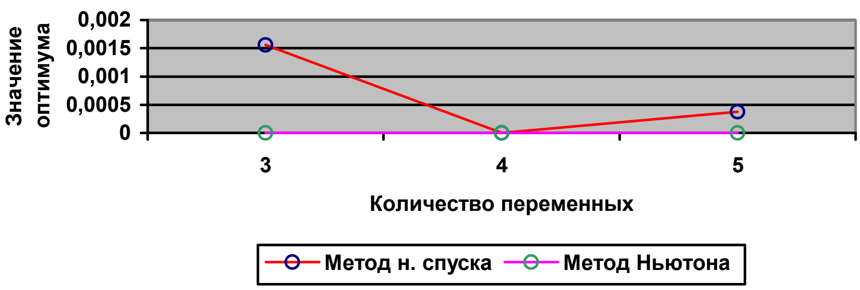

Рис. 16. *Значение оптимума для функции Де Джонга*

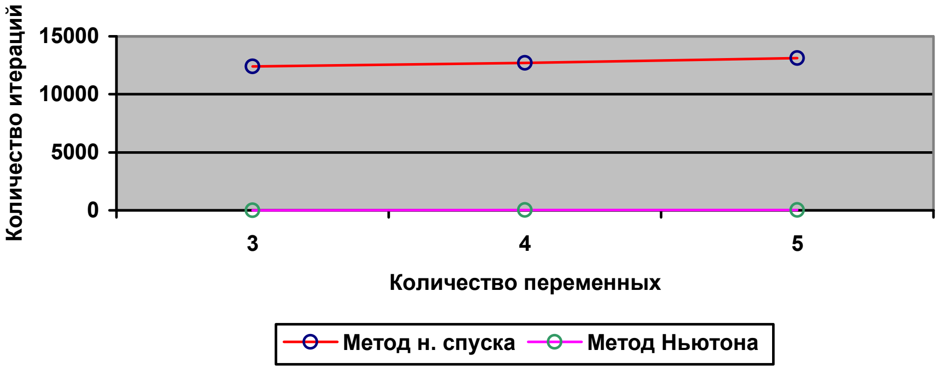

Рис. 17. *Количество итераций для функции Розенброка до нахождения оптимума*

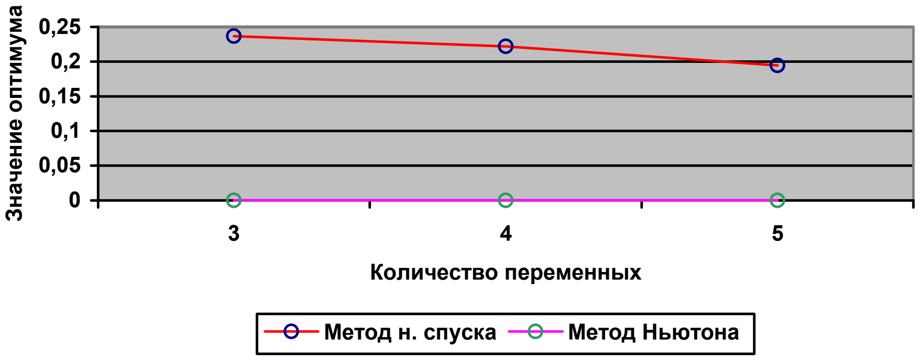

Рис. 18. *Значение оптимума для функции Розенброка*

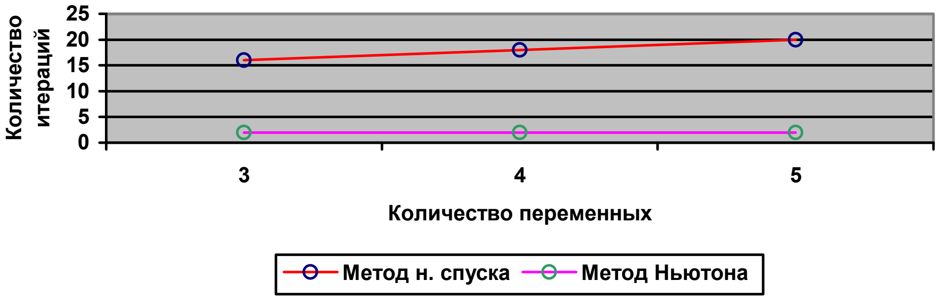

Рис. 19. *Количество итераций для функции Растригина до нахождения оптимума*

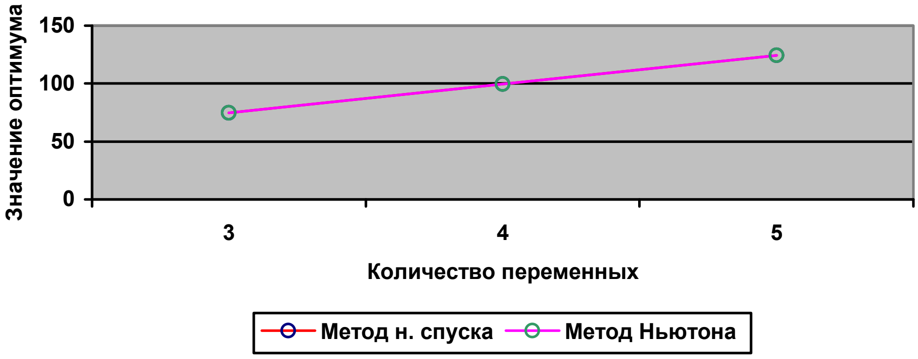

Рис. 20. *Значение оптимума для функции Растригина*

Сравнив два данных градиентных метода, можно сделать вывод, что лучше с поставленной задачей справляется метод Ньютона. Данный метод намного быстрее и дает более точные результаты.

Исследование и сравнение генетического алгоритма и градиентных методов

Построим графики для наглядного представления полученных результатов и сравнения методов. Для построения графиков будем использовать результаты при начальном приближении 5 для градиентных методов.

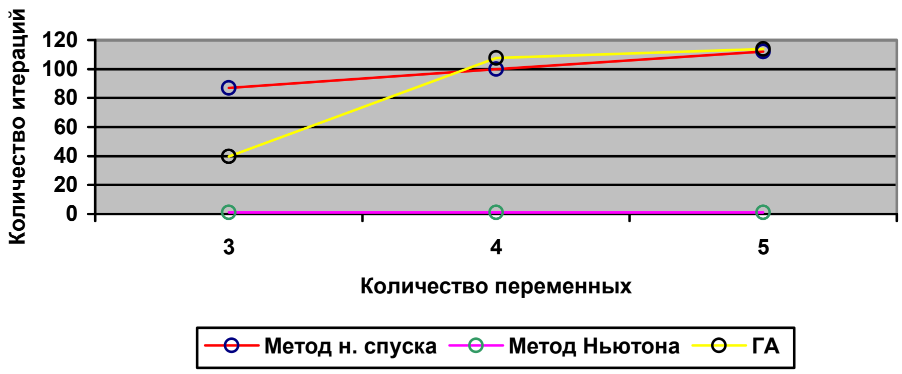

Рис. 21. *Количество итераций для функции Де Джонга до нахождения оптимума*

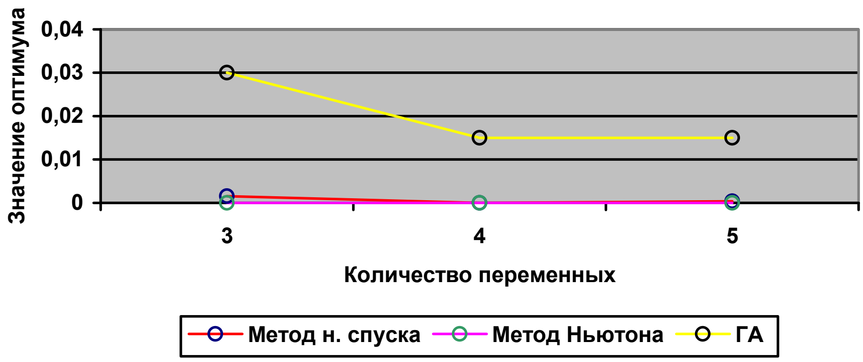

Рис. 22. *Значение оптимума для функции Де Джонга*

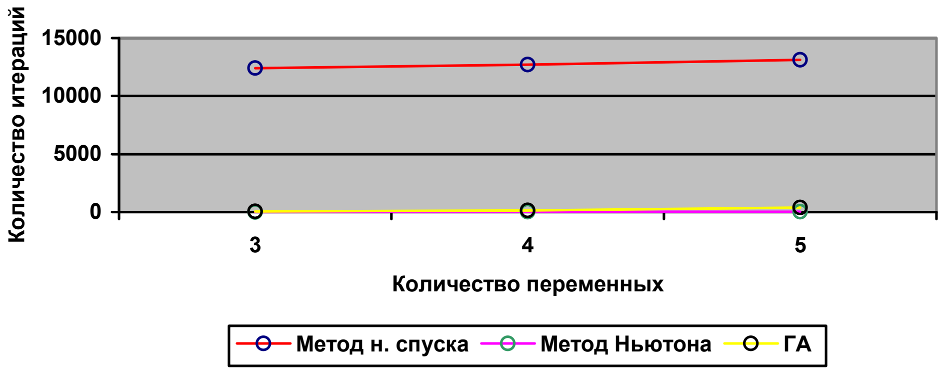

Рис. 23. *Количество итераций для функции Розенброка до нахождения оптимума*

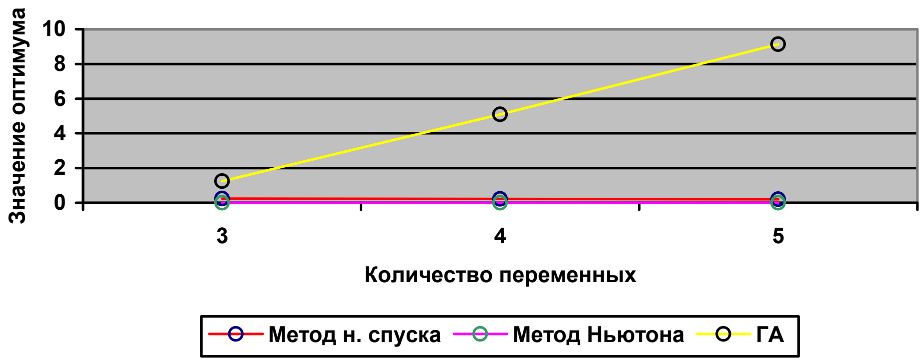

Рис. 24. *Значение оптимума для функции Розенброка*

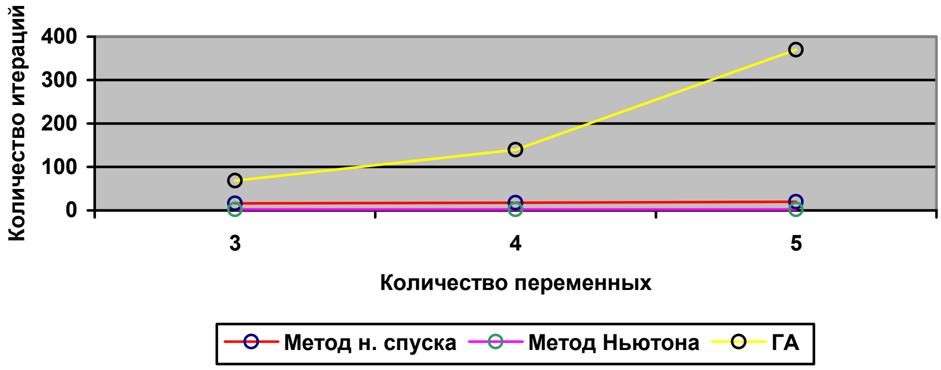

Рис. 25. *Количество итераций для функции Растригина до нахождения оптимума*

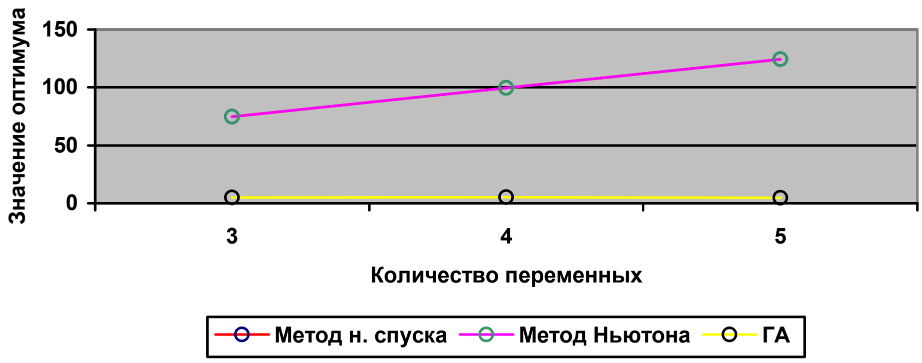

Рис. 26. *Значение оптимума для функции Растригина*

Сравнение работы ГА и градиентных методов по полученным результатам показывает, что ГА является универсальным методом для нахождения оптимума функций любой сложности. Для простых функций ГА работает медленнее градиентных методов, иногда с меньшей точностью, но не намного хуже. Для сложных функций ГА позволяет найти оптимум, когда градиентные методы с этим не справляются. Наиболее оптимальным оказался метод Ньютона для функций Де Джонга и Розенброка. Для функции Растригина оказался оптимальным ГА.

5. Заключение

В статье приведены результаты исследования ГА и градиентных методов, а также сравнение данных методов для нахождения оптимума функций.

При исследовании генетического алгоритма были выбраны оптимальные параметры для нахождения оптимума функции. Оптимальными параметрами являются:

- значения мутации от 0,18 до 0,22,
- значение скрещивания 0,8,
- количество стрингов от 15 до 17 для 5 переменных. Т.е. видно, что для одной переменной необходимы 3 стринга для оптимальной работы генетического алгоритма.

При выборе параметров для работы с ГА нужно быть осторожным, так как при выборе параметра меньше оптимального результат работы алгоритма может быть неточным, но скорость работы алгоритма возрастает. В свою очередь, при выборе параметра больше оптимального скорость работы алгоритма понижается, и алгоритм может не сойтись. Поэтому нужно выбрать оптимальные параметры, при которых работа алгоритма будет наиболее продуктивной.

При исследовании работы ГА для нахождения оптимума функций видно, что генетический алгоритм является универсальным методом для поиска оптимума независимо от сложности функций. Для поиска оптимума простой функции лучше использовать ГА, так как они работают немного быстрее генетического алгоритма. Для поиска оптимума сложной функции лучше использовать генетический алгоритм, так как градиентные методы могут не сойтись из–за особенности функции либо из-за особенности градиентного метода.

Литература

1. Holland, J. H. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
2. Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison–Wesley Pub. Company, 1989.
3. Mitchell, T. M. *Machine Learning.* The McGraw–Hill Companies, Inc, 1997.
4. Джонс М. Т. *Программирование искусственного интеллекта в приложениях*. Москва: ДМК Пресс, 2004.