

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики
Факультет информационных технологий и программирования
Кафедра «Компьютерные Технологии»

А. В. Демин

Отчет по лабораторной работе
«Построение управляющих автоматов с помощью генетических алгоритмов»

Вариант №4

Санкт-Петербург

2011

Содержание

Введение	2
1 Постановка задачи	2
1.1 Задача об умном муравье	2
2 Алгоритм	3
2.1 Представление особи	3
2.2 Эволюционная стратегия	3
2.3 Функция приспособленности	3
2.4 Операторы мутации	3
2.4.1 p -оператор	4
2.4.2 k -оператор	4
3 Результаты работы	4
3.1 Исследование p -оператора	4
3.2 Исследование k -оператора	5
3.3 Сравнение операторов	7
Заключение	8
Источники	8
Приложение. Построенный автомат	9

Введение

В лабораторной работе проводится сравнение двух операторов мутации при использовании эволюционной стратегии [1]. В качестве примера используется задача об умном муравье.

При выполнении лабораторной работы использовался фреймворк для работы с эволюционными и генетическими алгоритмами Watchmaker [3]. Был написан модуль на языке программирования Java, с помощью которого изучалось поведение эволюционной стратегии на заданных операторах мутации.

1 Постановка задачи

Цель данной лабораторной работы — сравнить эффективность следующих операторов мутации при работе с эволюционной стратегией:

- оператор, с вероятностью p мутирующий каждый ген хромосомы¹;
- оператор, мутирующий случайно выбранные k генов хромосомы².

Сравнение производится на примере задачи об умном муравье, для решения которой с помощью (1+5)-эволюционной стратегии строится автомат Мура [2].

1.1 Задача об умном муравье

Дано поле размером 32×32 клетки, расположенное на поверхности тора. На 89-ти фиксированных клетках содержится по одному яблоку. Схема поля приводится на рис. 1. Муравей начинает движение из клетки *Start* и за один ход может выполнить одно из следующих действий:

- повернуть налево;
- повернуть направо;
- сделать шаг вперед и, если в новой клетке есть яблоко, съесть его;
- ничего не делать.

Кроме того, муравей умеет определять, содержится ли в клетке перед ним яблоко. Всего можно сделать 200 шагов. Требуется построить автомат с конечным числом состояний, управляющий муравьем, который за минимальное число шагов съест максимальное число яблок.

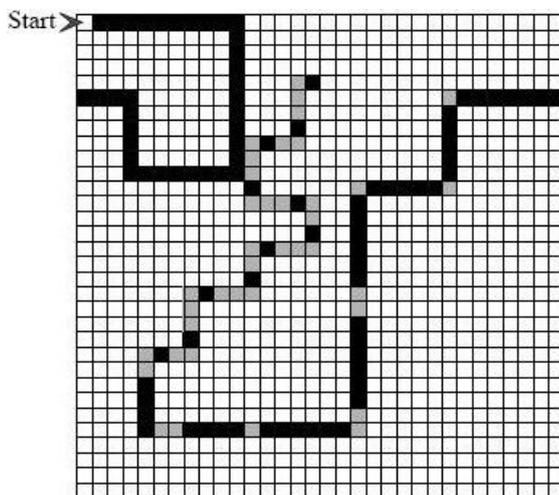


Рис. 1. Игровое поле

¹Далее — p -оператор

²Далее — k -оператор

2 Алгоритм

2.1 Представление особи

Особью эволюционной стратегии для решения данной задачи является конечный автомат Мура — детерминированный конечный автомат, каждому состоянию которого сопоставлено некоторое выходное действие. Различают два вида автоматов Мура: которые формируют выходные действия на основе текущего состояния и которые предварительно обновляют свое состояние, а затем формируют выходное действие. В данной лабораторной работе используется второй вид автоматов Мура с 20 состояниями (*states*).

В задаче об умном муравье два входных воздействия (*signals*): наличие и отсутствие яблока в клетке перед муравьем. Выходных действий четыре: повернуться налево, повернуться направо, сделать шаг вперед, ничего не делать.

Автомат хранится в виде полной таблицы переходов из каждого состояния при каждом из входных воздействий. В каждой ячейке таблицы хранится номер состояния, в которое осуществляется переход. Отдельно хранится таблица выходных действий, сопоставляющая каждому состоянию номер действия, которое необходимо совершить при переходе в данное состояние.

2.2 Эволюционная стратегия

В работе использовалась (1+5)-эволюционная стратегия, которая заключается в следующем. В начале строится первое поколение, содержащее в данном случае одну особь. Далее выполняется итеративный процесс, при котором каждое новое поколение строится на основе предыдущего. Каждое поколение содержит одну особь. При переходе от старого поколения к новому из особи старого поколения строится пять новых особей с помощью оператора мутации. После этого из полученных шести особей в новое поколение переходит одна особь с наибольшим значением функции приспособленности.

Такой алгоритм позволяет более точно определить эффективность используемого оператора мутации, так как исключается случайное воздействие оператора скрещивания.

2.3 Функция приспособленности

Функция приспособленности вычисляется по формуле:

$$fitness = apples + \frac{200 - steps}{200},$$

где *apples* — количество съеденных яблок, *steps* — номер шага, на котором было съедено последнее яблоко. Таким образом, муравей, съедающий большее количество яблок, имеет большую функцию приспособленности, а среди муравьев, съедающих одинаковое количество яблок, наибольшая функция приспособленности будет у муравья, который съедает яблоки за меньшее число шагов.

2.4 Операторы мутации

Изменяемыми параметрами (генами) особи являются:

- конечное состояние каждого перехода (*signals * states* значений);
- действие в каждом состоянии (*states* значений);
- номер стартового состояния (одно значение).

Таким образом, общее количество генов равно:

$$n = signals * states + states + 1 = 61$$

2.4.1 p -оператор

Данный оператор мутирует каждый ген с вероятностью $p \in (0, 1]$. В качестве реализации события, происходящего с заданной вероятностью, использовался класс Probability [4].

2.4.2 k -оператор

Данный оператор мутирует случайно выбранные k ($1 \leq k \leq n$) генов.

3 Результаты работы

Для сравнения p - и k -операторов мутации были проведены дополнительные исследования с целью нахождения значений p и k соответственно, при которых эволюционная стратегия работает наиболее эффективно. Затем для полученных значений p и k были произведены более подробные исследования.

3.1 Исследование p -оператора

Для каждого значения $p = 0.1, 0.2, \dots, 1.0$ было произведено по 100 запусков, в каждом из которых строилось по 10000 поколений, затем значения функций приспособленности усреднялись по запускам для каждого поколения. Результат представлен на графике 1.

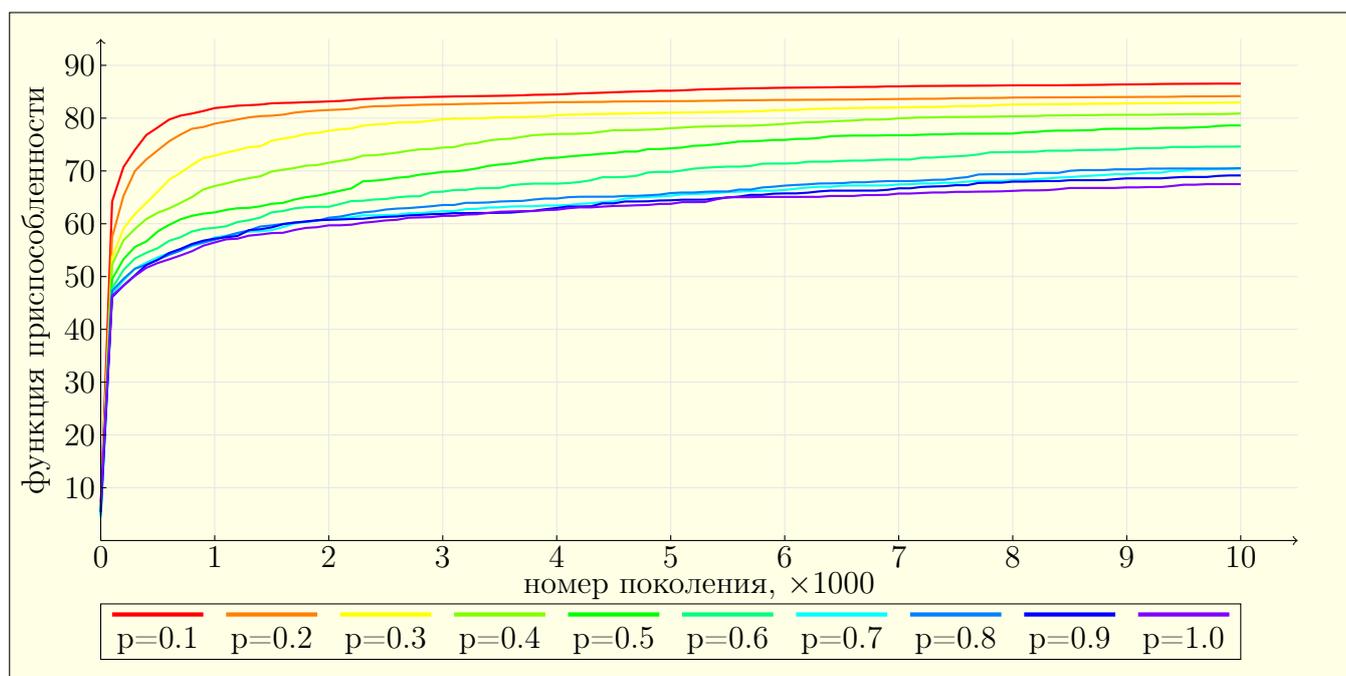


График 1. Зависимость функции приспособленности от номера поколения при использовании p -оператора

Таким образом, p -оператор работает наиболее эффективно при вероятности мутации близкой к 0.1. Для уточнения этих результатов было произведено по 500 запусков, в каждом из которых строилось по 10000 поколений при $p = 0.01, 0.02, \dots, 0.19$. Результат представлен на графике 2.

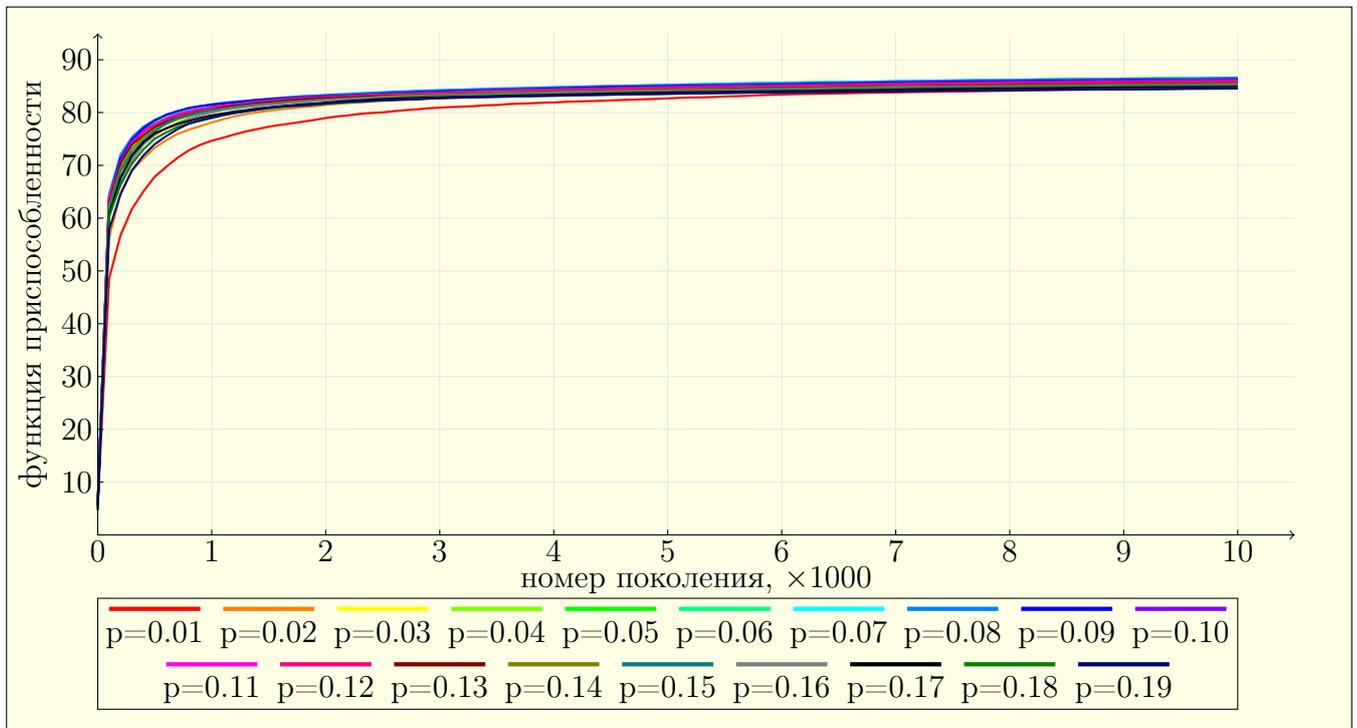


График 2. Зависимость функции приспособленности от номера поколения при использовании p -оператора

Итоговый результат предварительного исследования p -оператора представлен на графике 3.

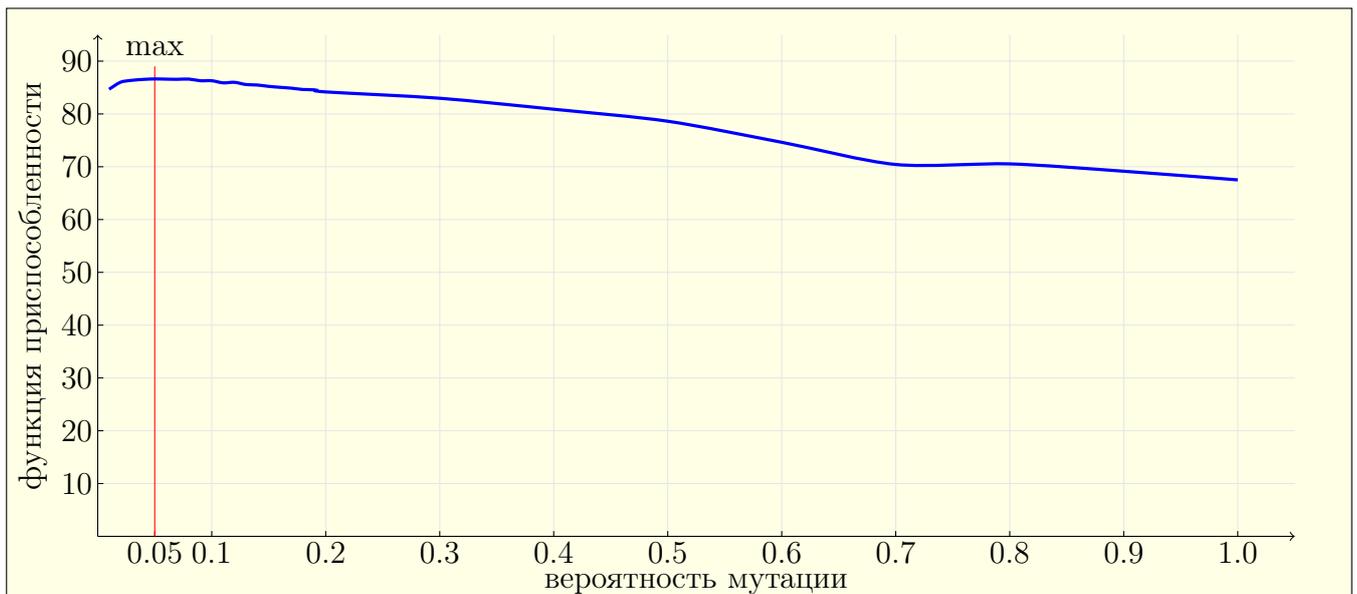


График 3. Зависимость функции приспособленности последнего поколения от вероятности мутации при использовании p -оператора

На основании полученных результатов для более подробного исследования оператора было взято значение вероятности мутации $p = 0.05$.

3.2 Исследование k -оператора

Аналогично исследованию p -оператора, было произведено по 100 запусков, в каждом из которых строилось по 10000 поколений для каждого значения $k = 6, 12, \dots, 60$. После этого значения функций приспособленности усреднялись по запускам для каждого поколения. Результаты вычислений представлены на графике 4.

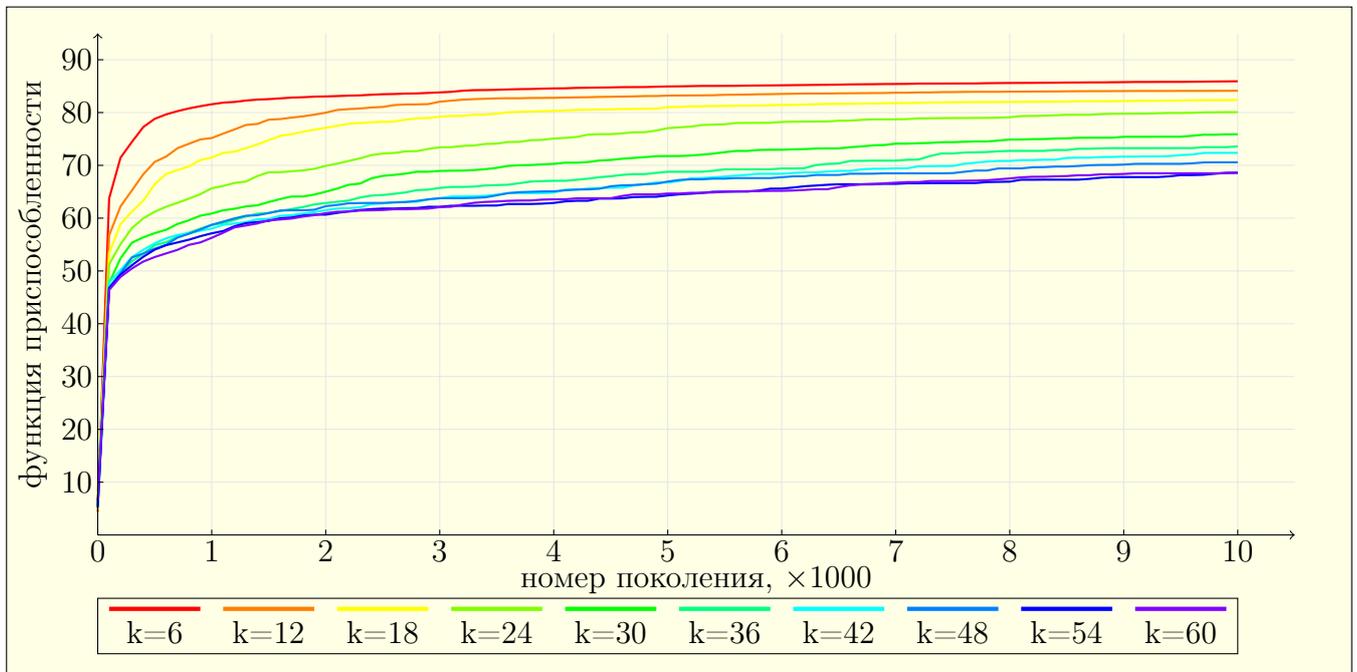


График 4. Зависимость функции приспособленности от номера поколения при использовании k -оператора

Таким образом, k -оператор работает наиболее эффективно при количестве мутирующих генов близком к 6. Для уточнения этих результатов было произведено по 500 запусков, в каждом из которых строилось по 10000 поколений при $k = 1, 2, \dots, 11$. Результаты вычислений представлены на графике 5.

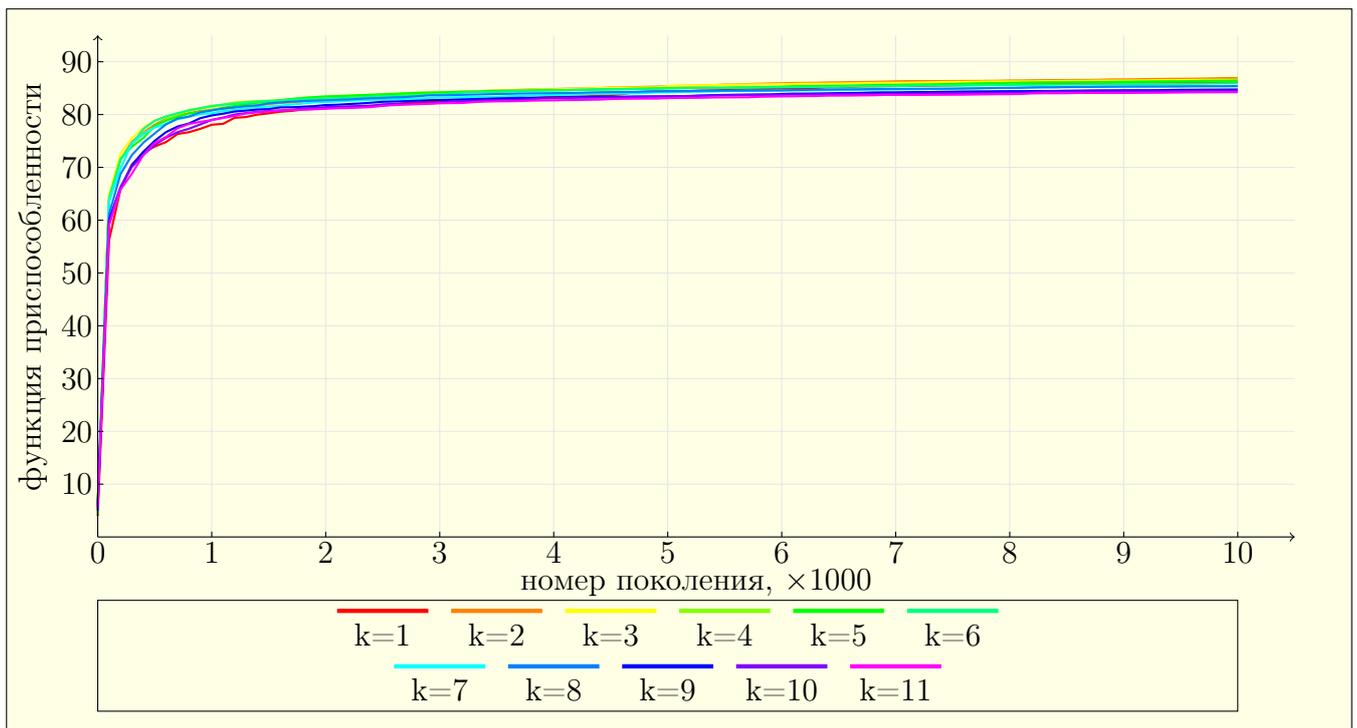


График 5. Зависимость функции приспособленности от номера поколения при использовании k -оператора

Итоговый результат предварительного исследования k -оператора представлен на диаграмме 1.

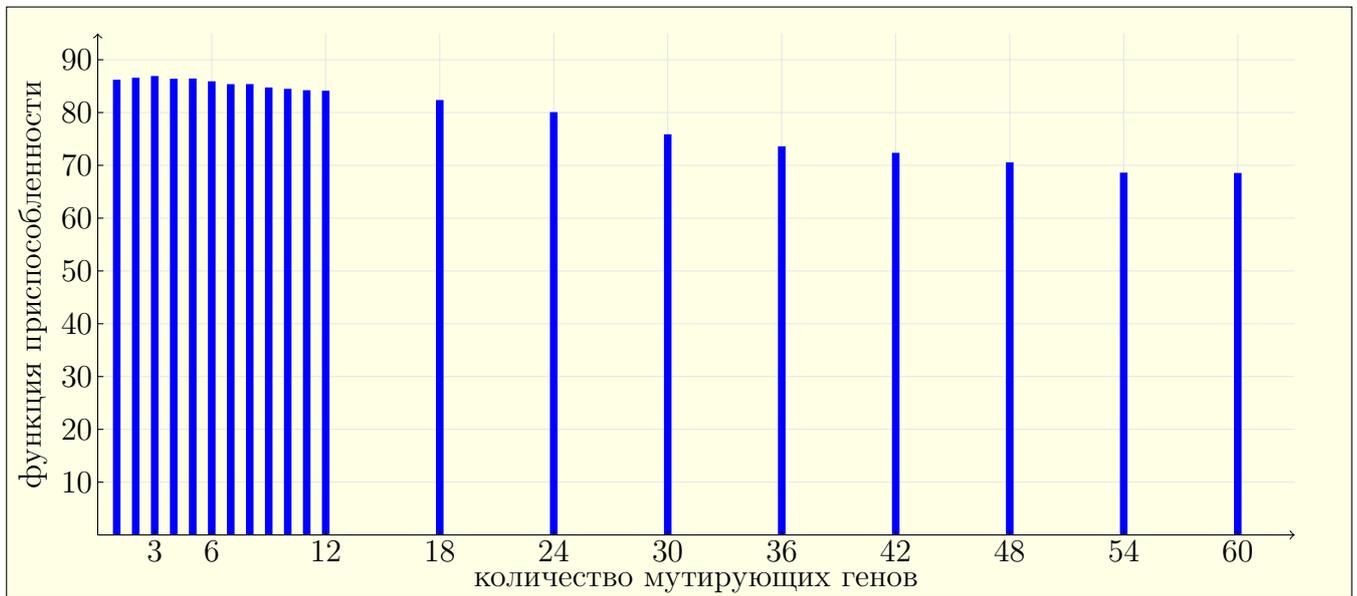


Диаграмма 1. Функция приспособленности последнего поколения при различном количестве мутирующих генов при использовании k -оператора

На основании полученных результатов для более подробного исследования оператора было взято значение $k = 3$.

3.3 Сравнение операторов

При более подробном исследовании операторов значение функции приспособленности усреднялось по 1000 запускам, в каждом из которых строилось по 10000 поколений. На графиках 6 и 7 приведены результаты для первых 1000 и 10000 поколений соответственно.

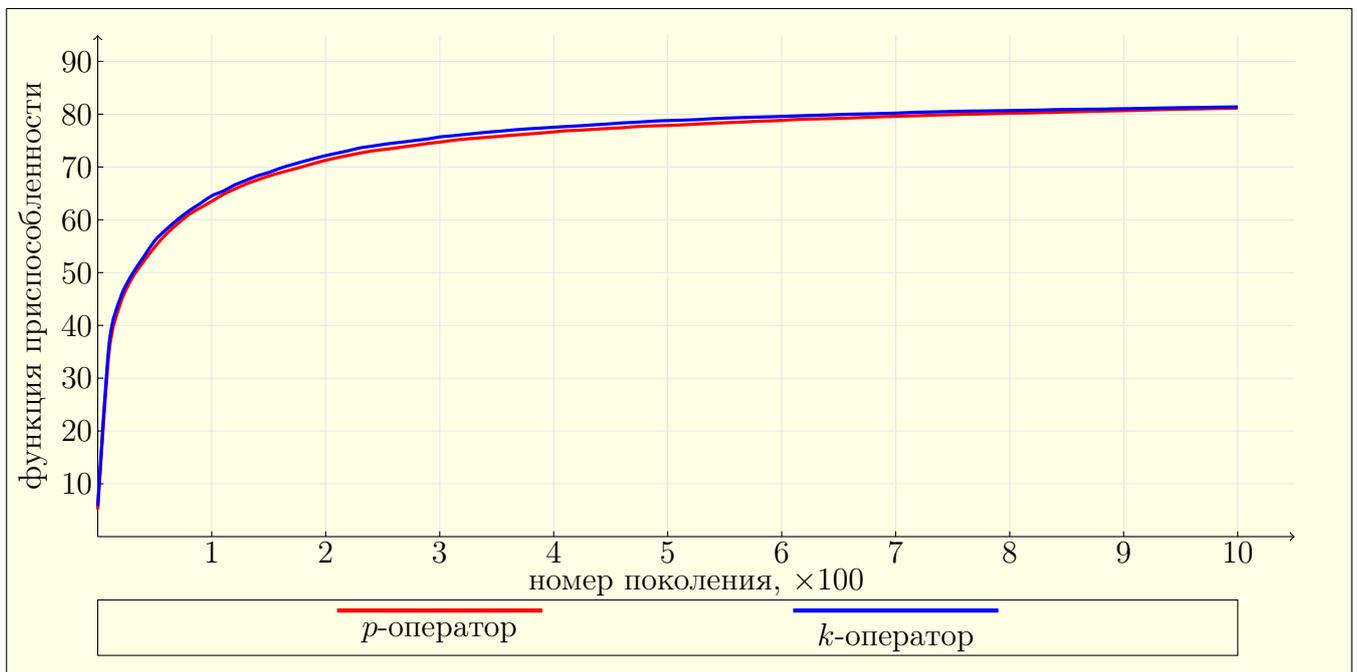


График 6. Сравнение функций приспособленности p - и k -операторов за первые 1000 поколений

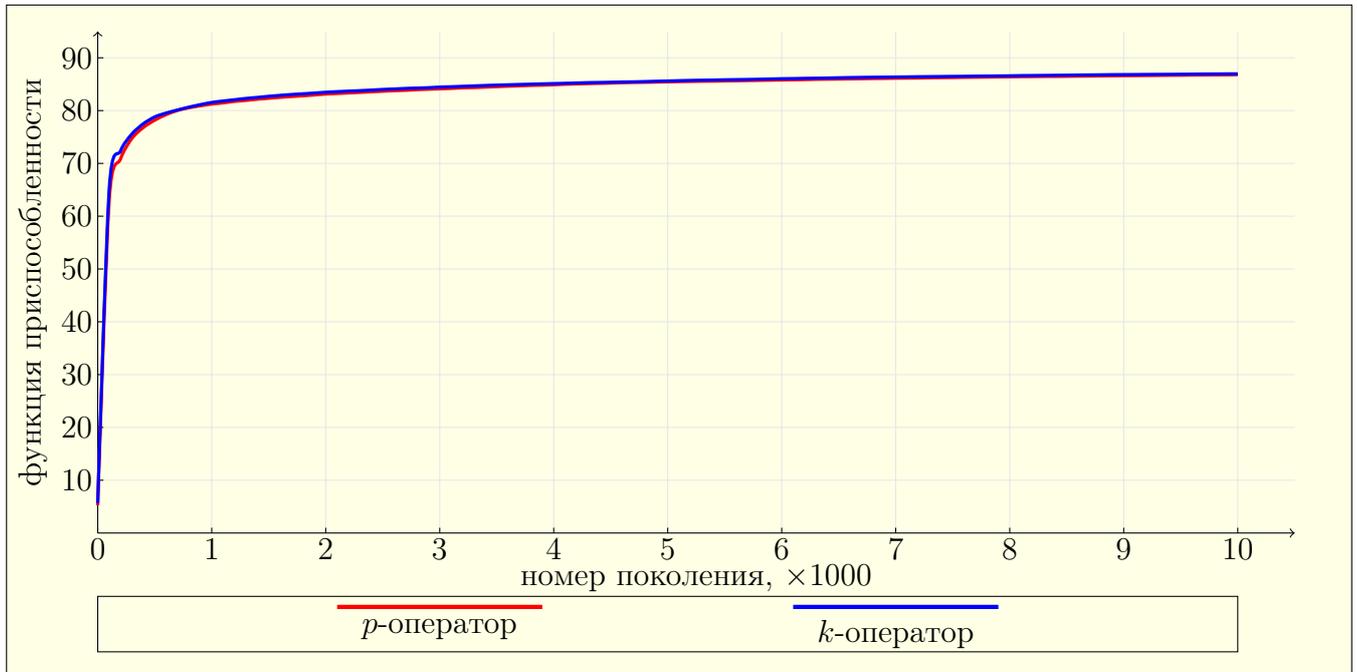


График 7. Сравнение функций приспособленности p - и k -операторов за первые 10000 поколений

Как видно из графиков, отличие между p - и k -операторами незначительно.

Заключение

Все вычисления производились на компьютере с процессором Intel Core 2 Duo 2.1 GHz, объемом оперативной памяти 3 GB под управлением ОС Arch Linux 32-bit. Общее время вычислений составило более 6 часов. Всего было рассмотрено 950 миллионов автоматов.

Результаты исследований, проведенных в лабораторной работе, показали, что эффективность операторов мутации, мутирующего каждый ген хромосомы с вероятностью p и мутирующего случайно выбранные k генов хромосомы, отличается незначительно, и оба оператора можно с одинаковой эффективностью применять на практике.

Источники

- [1] Bäck T., Hoffmeister F., Schwefel H.-P. *A Survey of Evolutionary Strategies*.
- [2] Поликарпова Н.И., Шалыто А.А. *Автоматное программирование*.
- [3] Документация пакета Watchmaker
<http://watchmaker.uncommons.org/api/index.html>
- [4] Документация пакета Uncommons Maths
<http://maths.uncommons.org/api/index.html>

