

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

А. В. Савченко

Отчет по лабораторной работе  
«Применение генетических алгоритмов для построения автоматов в  
задаче об умном муравье»  
Вариант №23

Санкт Петербург

2011

## Оглавление:

Введение.....	3
1. Постановка задачи.....	4
1.1. Задача об умном муравье-3.....	4
1.2. Автомат Мили.....	5
2. Эволюционная стратегия.....	6
2.1. Представление автоматов.....	6
2.2. Случайное порождение автомата.....	7
2.3. Мутация автомата 1.....	7
2.4. Мутация автомата 2.....	8
2.5. Удаление недостижимых ветвей.....	8
2.6. Метод генерации очередного поколения.....	9
3. Результаты.....	10
Заключение.....	12
Источники.....	13

## **Введение**

В данной работе изучается зависимость эффективности работы эволюционной стратегии для решения задачи об умном муравье-3 от типа мутации.

При выполнении работы использовался фреймворк «Watchmaker» [1] для работы с генетическими алгоритмами. Все исходные коды в данной работе написаны на языке программирования Java.

## 1. Постановка задачи

Задача данной лабораторной работы – исследовать влияние типа мутации на эффективность работы алгоритма, строящего автомат Мили, который решает задачу об умном муравье-3 [2].

### 1.1 Задача об умном муравье-3

Дано поле 32×32 клетки, расположенное на поверхности тора (рис. 1). На поле случайно расположено некоторое количество яблок, определяемое вероятностью нахождения еды в клетке. Муравей видит восемь клеток (рис. 2).

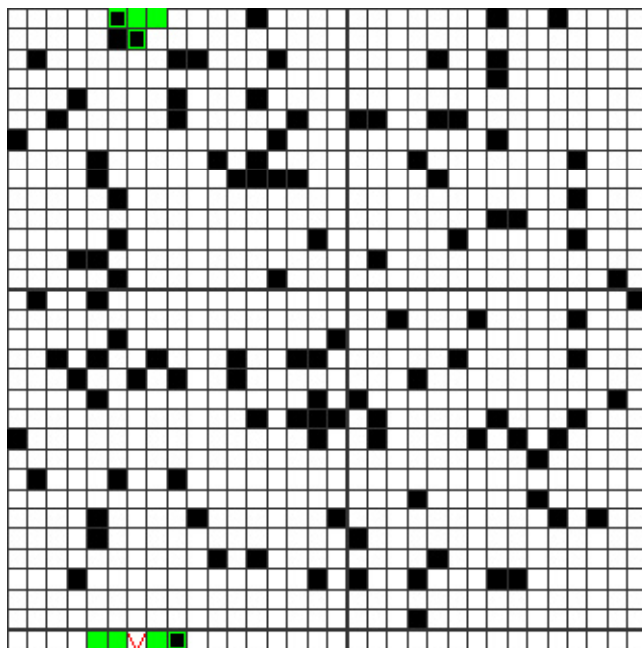


Рис. 1. Игровое поле



Рис. 2. Область видимости муравья

За один ход муравей может определить, есть ли в клетке перед ним яблоко, и выполнить одно из следующих действий:

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

Всего делается 200 ходов. Требуется построить автомат с фиксированным числом состояний, который за минимальное число ходов ест как можно больше яблок.

## 1.2 Автомат Мили

Автомат Мили – это конечный автомат, генерирующий выходные воздействия в зависимости от текущего состояния и входного воздействия. Как видно из диаграммы на рис. 3, над каждой дугой расположена пара значений: входное и выходное воздействия.

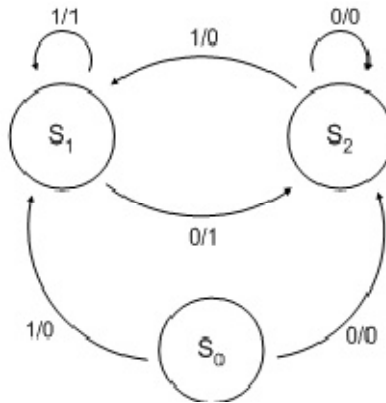


Рис. 3. Пример диаграммы переходов автомата мили из трех состояний

Входные воздействия для муравья:

- Т – перед муравьем есть еда;
- F – перед муравьем нет еды.

Выходные воздействия для муравья:

- М – сделать шаг вперед и съесть еду в новой клетке, если она там есть;
- R – повернуть направо;
- L – повернуть налево.

## 2. Эволюционная стратегия

Для поиска оптимального автомата, управляющего муравьём, применяется эволюционная стратегия ( $\mu+\lambda$ ) [3]. Генерация  $\mu$  особей первого поколения происходит случайным образом. Далее выполняется повторение действий:

- скрещивание особей из текущего поколения;
- мутация особей из текущего поколения;
- отбор наилучших  $\mu$  особей из  $\mu+\lambda$  в следующее поколение.

В данной работе применяется стратегия (1+5), поэтому операция скрещивания отсутствует; в результате мутации одной особи получаются все пять потомков, из которых затем выбирается лучший.

### 2.1 Представление автоматов

Особью в данном алгоритме является автомат Мили, который представлен в виде дерева решений [4, 5]: вместо задания функций переходов для автомата в целом, задается функция для каждого состояния, выраженная деревом решений. В этих деревьях переменными являются входные переменные автомата, а множеством значений – все возможные пары (*Новое состояние, Действие*). Таким образом, автомат в целом задается упорядоченным набором деревьев решений.

На рис. 4 приведен пример автомата Мили. При этом  $a, b, c$  – входные переменные булевого типа, а  $L, R, M$  – выходные воздействия. Этот автомат, представленный деревьями решений, приведен на рис. 5 [5].

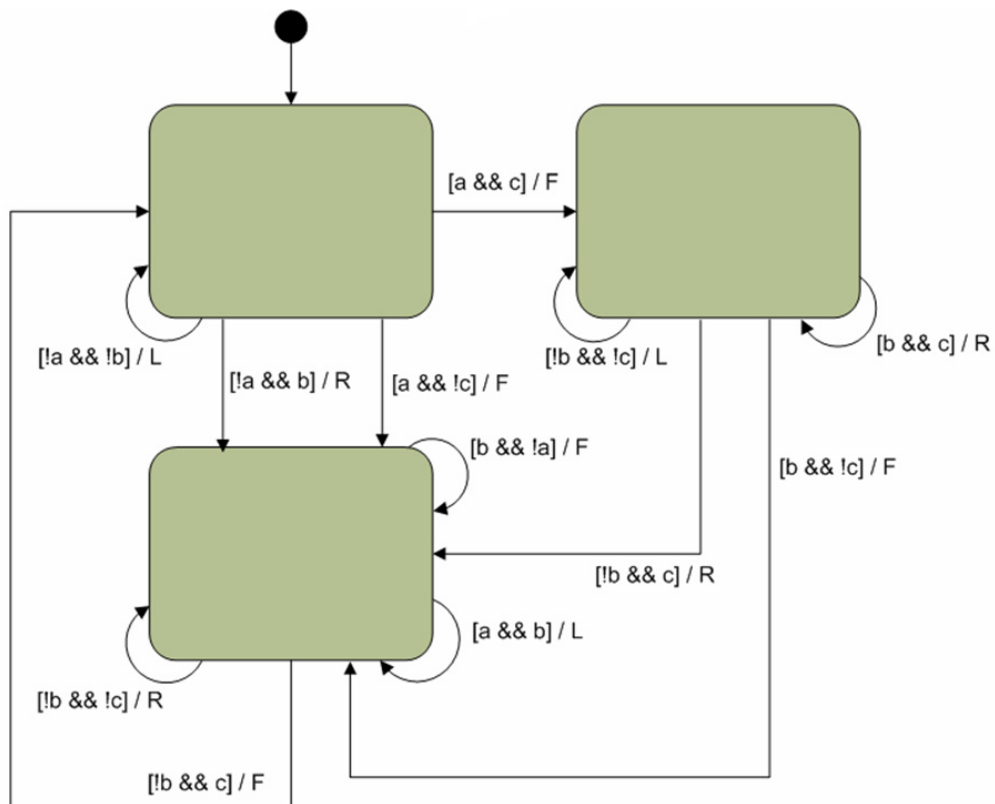


Рис. 4. Пример автомата Мили

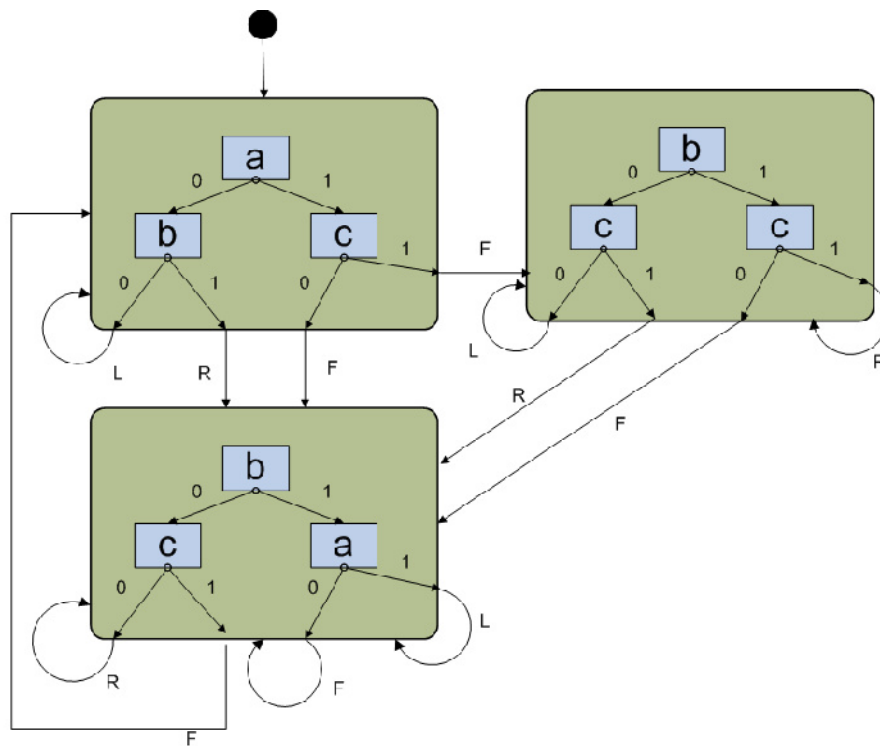


Рис. 5. Представление автомата Миля с помощью деревьев решений

В задаче об умном муравье-3 значениями восьми входных переменных являются 0 и 1 – отсутствие или присутствие еды в соответствующей клетке из области видимости муравья.

## 2.2 Случайное порождение автомата

В каждом состоянии создается случайное дерево решений – случайным образом выбирается метка: либо одна из переменных, либо одно из значений функции переходов. После этого создается вершина, помеченная выбранной меткой. При этом, если была выбрана переменная, рекурсивно генерируются дети вершины, иначе вершина становится листом дерева.

## 2.3 Мутация автомата 1

В случайном дереве решений происходит мутация: выбирается случайный узел в поддереве, после чего поддерево, соответствующее выбранному узлу, заменяется случайно сгенерированным (рис. 6).

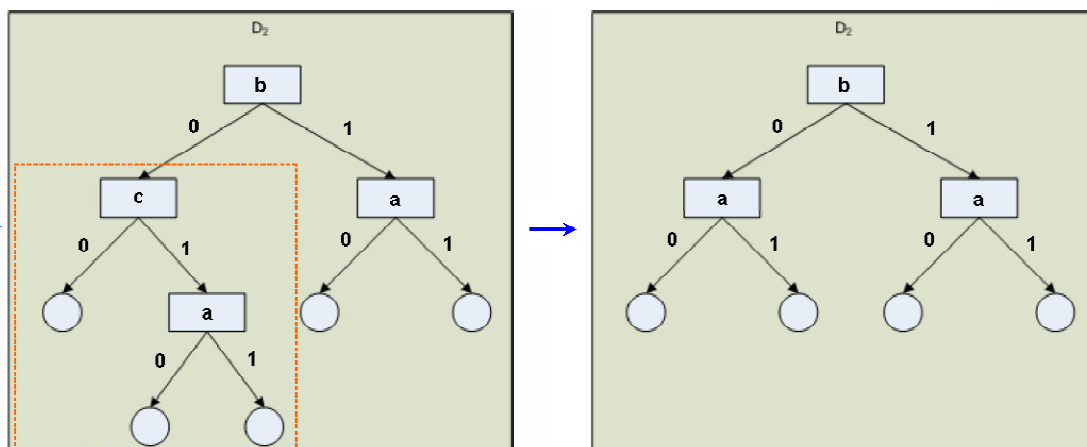


Рис. 6. Мутация деревьев решений

## 2.4 Мутация автомата 2

Цель данной лабораторной работы предусматривает наличие альтернативной операции мутации: с некоторой вероятностью происходит обмен поддеревьями между различными состояниями автомата.

Также в работе исследуется использование комбинации мутаций 1 и 2.

## 2.5 Удаление недостижимых ветвей

Заданные операции могут порождать деревья, в которых некий атрибут на пути от корня до листа встречается дважды (рис. 7). Такие деревья являются корректными, но имеют недостижимые вершины, что увеличивает объем занимаемой памяти, а также замедляет генерацию при копировании таких деревьев в потомков.

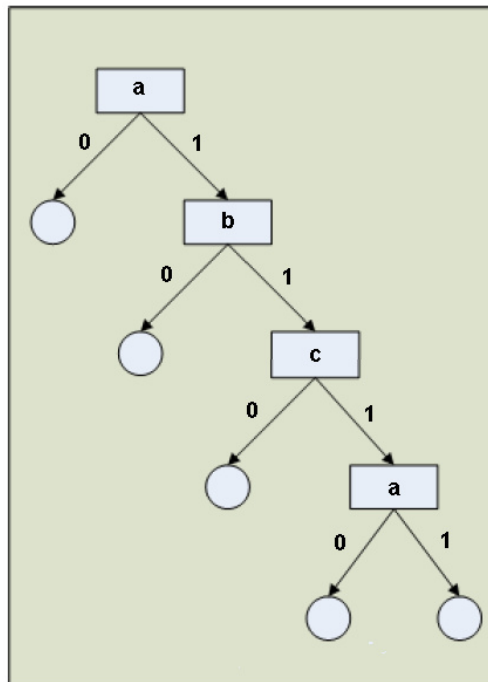


Рис. 7. Пример дерева решений с недостижимой вершиной

Таким образом, необходимо ввести операцию обрезки – удаление недостижимых ветвей. Операция выполнена следующим образом: узел, одна из дочерних вершин которого недостижима, заменяется своей достижимой дочерней вершиной (рис. 8).

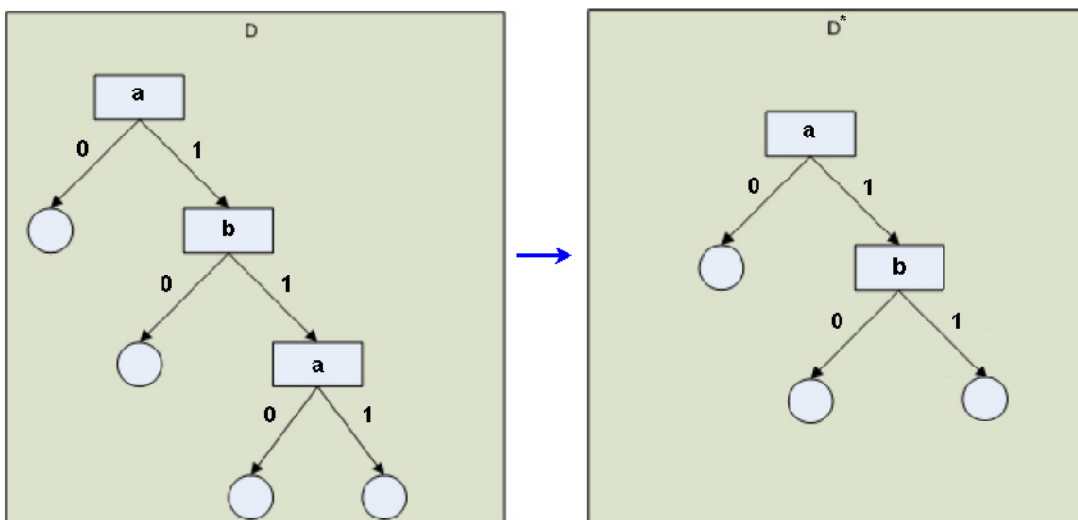


Рис. 8. Пример обрезки недостижимых ветвей



## 2.6 Метод генерации очередного поколения

Начальное поколение генерируется из особей, заданных случайным образом (2.2). Процесс генерации нового поколения из текущего заключается в следующем: из множества особей текущего поколения ( $\mu$  штук) и их потомков ( $\lambda$  штук) выбирается  $\mu$  наиболее приспособленных особей. Приспособленность определяется фитнес-функцией:

$$Fitness = Apples - Steps/200;$$

где Apples – число яблок, съедаемых муравьем за 200 шагов, Steps – номер шага, на котором муравей съедает последнее яблоко. Таким образом, следующее поколение состоит из особей с максимальным значением функции приспособленности.

### 3. Результаты

Для исследования использовались автоматы с восемью состояниями. Функция приспособленности вычислялась по 50 различным полям с вероятностью нахождения еды в клетке 0,11.

Было произведено 100 запусков с числом поколений 200 и вероятностью обмена поддеревьями для альтернативной мутации 0,1. Результаты представлены на рис. 9.

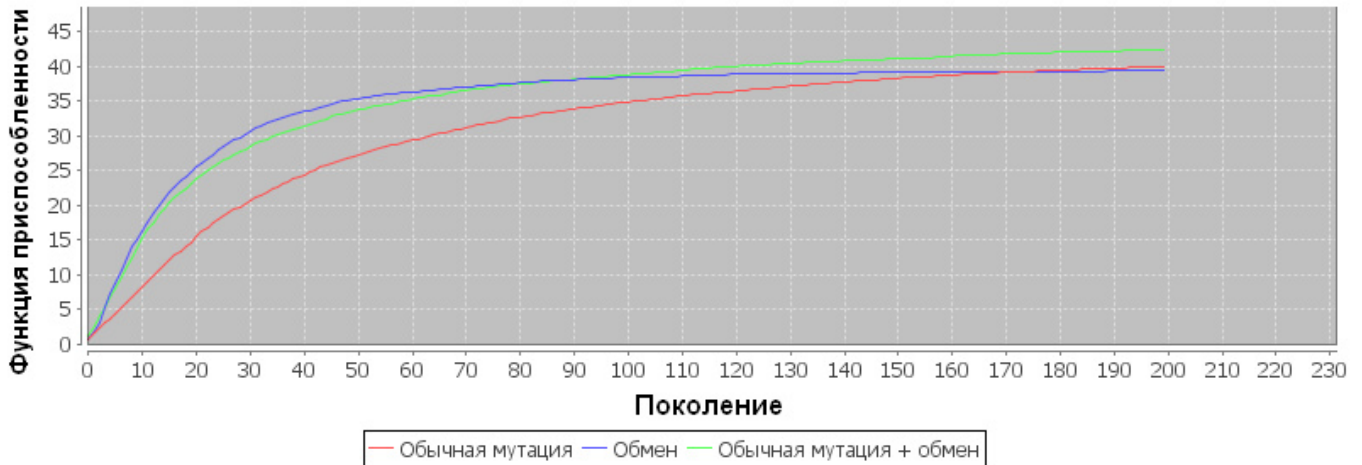


Рис. 9. График зависимости среднего значения функции приспособленности от номера поколения с использованием разных мутаций (вероятность обмена 0,1)

Очевидно, что обычная мутация дает худшие результаты, по сравнению с модифицированной (обычная+обмен). Для более подробного исследования было произведено 100 запусков до 500 поколения с теми же параметрами (рис. 10).

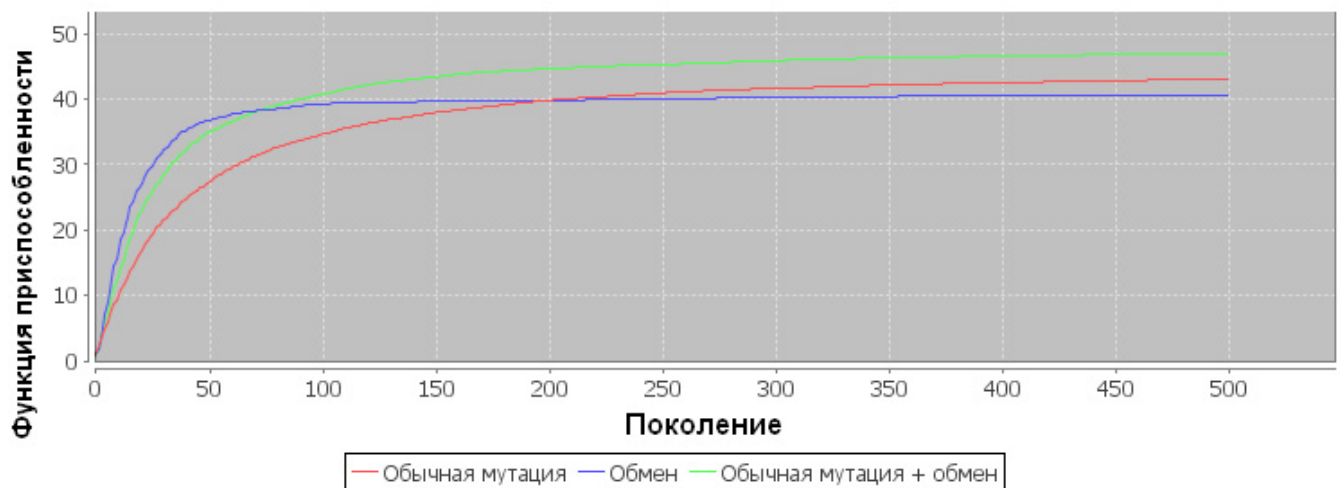


Рис. 10. График зависимости среднего значения функции приспособленности от номера поколения с использованием разных мутаций (вероятность обмена 0,1)

Из графика видно, что обмен поддеревьями в автомате из восьми состояний быстро приводит к созданию ограниченного числа особей, из-за чего функция приспособленности не растет, и обычная мутация в итоге выигрывает. Также видно, что модифицированная мутация превосходит обычную в каждом из поколений.

Чтобы продолжить сравнение мутаций, было выполнено 100 запусков до 200 поколения для различных вероятностей обмена поддеревьями – от 0,1 до 0,9 с шагом 0,1 (рис. 11).

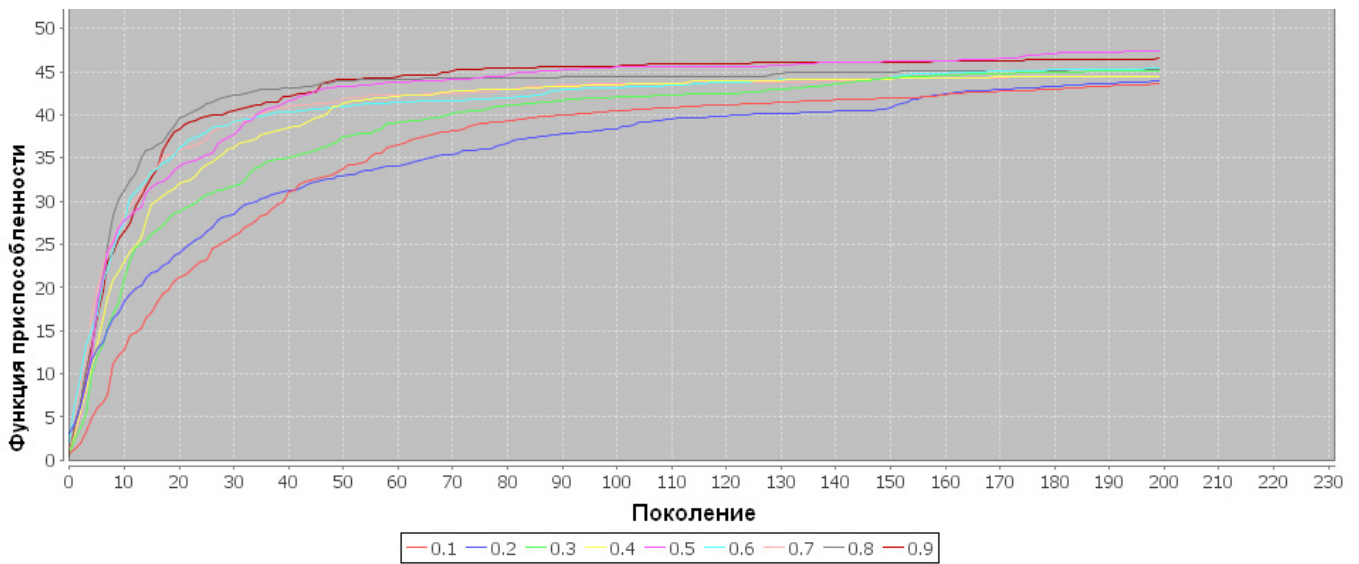


Рис. 11. График зависимости среднего значения функции приспособленности от вероятности обмена поддеревьями

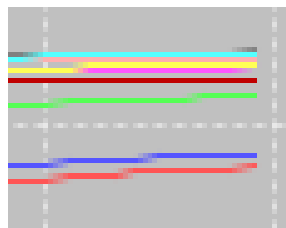


Рис. 12. График зависимости среднего значения функции приспособленности от вероятности обмена поддеревьями

График показывает, что почти одинаковый результат достигается при вероятности из промежутка от 0,4 до 0,8, лучший – при 0,8. Для этого значения вероятности было сделано 100 запусков до 200 поколения (рис. 12).

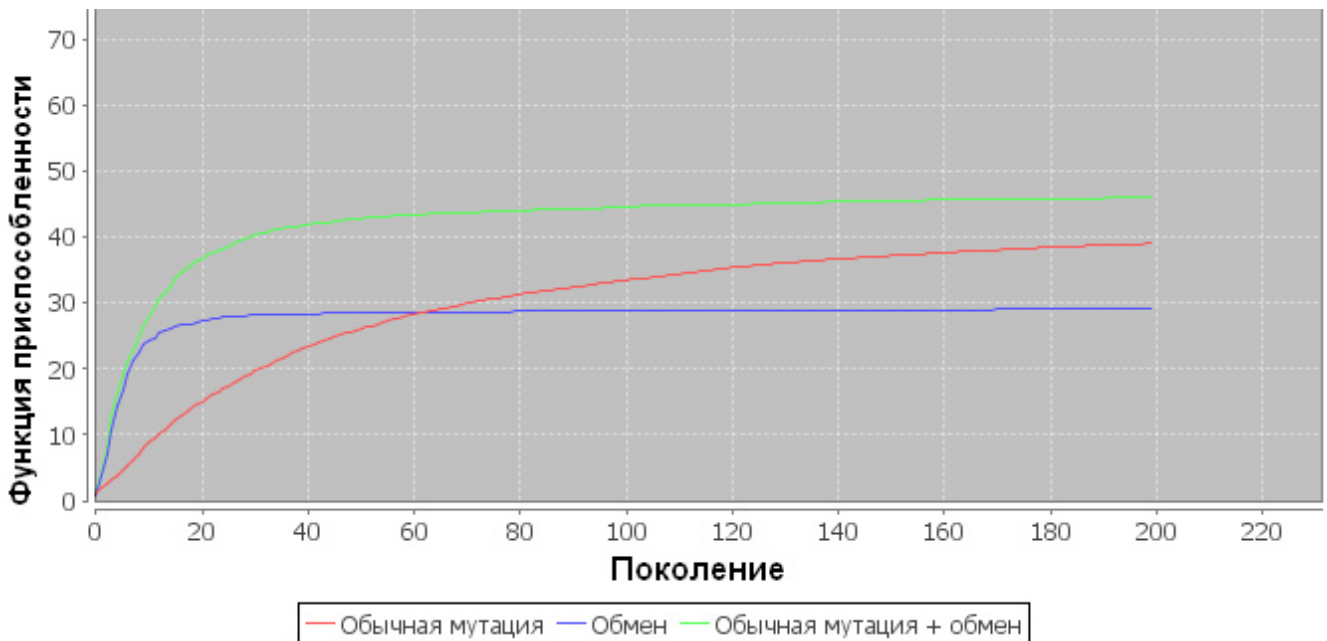


Рис. 13. График зависимости среднего значения функции приспособленности от номера поколения с использованием разных мутаций (вероятность обмена 0,8)

Здесь модифицированная мутация значительно обгоняет обычную уже на первых поколениях и сохраняет свое превосходство и далее.

## **Заключение**

Результаты лабораторной работы показали, что при использовании мутации, основанной на обмене поддеревьями, функция приспособленности быстро достигает своего максимума и далее не меняется. Однако при использовании мутации обменом вместе с обычной мутацией достигаются лучшие результаты. Для решения задачи об умном муравье-3 рекомендуется использовать обычную мутацию вместе с мутацией обменом с вероятностью обмена от 0,4 до 0,8.

## Источники

1. Документация пакета «Watchmaker»  
<http://watchmaker.uncommons.org/manual/index.html>
2. Буздалов М.В. Лабораторные работы по автоматному программированию  
<http://rain.ifmo.ru/~buzdalov/lab-2011/presentations/labs-presentation.pdf>
3. Back T., Hoffmeister F., Schwefel H.-P. A Survey of Evolutionary Strategies  
<http://rain.ifmo.ru/~buzdalov/lab-2011/books/es-survey.pdf>
4. Данилов В.Р., Шалыто А.А. Метод представления функций переходов деревьями решений для генерации автоматов с помощью генетического программирования  
<http://is.ifmo.ru/works/ 2009 08 01 danilov.pdf>
5. Данилов В.Р. Методы представления функции переходов при генерации автоматов управления на основе генетического программирования  
<http://is.ifmo.ru/papers/ 2010 02 25 danilov.pdf>