

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

А. Р. Хамзин

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

Вариант №27

Санкт-Петербург  
2011 г.

<b>Введение</b> .....	3
<b>1. Постановка задачи</b> .....	4
1.1. Задача о работе, обходящем препятствия .....	4
<b>2. Генетический алгоритм</b> .....	5
2.1. Функция приспособленности .....	5
2.2. Ранговый отбор .....	5
2.3. Оператор мутации.....	5
2.4. Оператор большой мутации.....	6
2.5. Оператор скрещивания.....	6
2.6. Метод генерации очередного поколения .....	6
<b>3. Результаты работы</b> .....	7
<b>Заключение</b> .....	8
<b>Источники</b> .....	9

## **Введение**

В данной лабораторной работе требуется исследовать эффективность работы генетического алгоритма построения автомата, решающего задачу о роботе, который обходит препятствия, в зависимости от вероятности применения оператора кроссовера(или скрещивания), при этом используя, сравнительно небольшую долю элитизма – в диапазоне от двух до пяти процентов.

При выполнении работы использовался программный комплекс для изучения методов глобальной оптимизации *G/Opt* [1], разработанный студентами кафедры «Компьютерные Технологии» НИУ ИТМО.

# 1. Постановка задачи

Задача данной лабораторной работы — исследовать влияние вероятности применения оператора кроссовера на эффективность работы генетического алгоритма, строящего автомат Мили из восьми состояний для решения задачи о роботе, обходящем препятствия. Представленный данным автоматом робот должен доходить до цели не более чем за 200 шагов.

## 1.1. Задача о роботе, обходящем препятствия

Дано поле 32×32 клетки, представляющее собой фиксированный лабиринт с препятствиями (см. рис. 1). Робот видит только клетку впереди себя.

За один ход робот может:

- пойти вперед (если впереди препятствие, то ничего не произойдет);
- повернуть налево;
- повернуть направо.

Задача робота — добраться до цели за наименьшее число ходов, не превышающее двухсот.

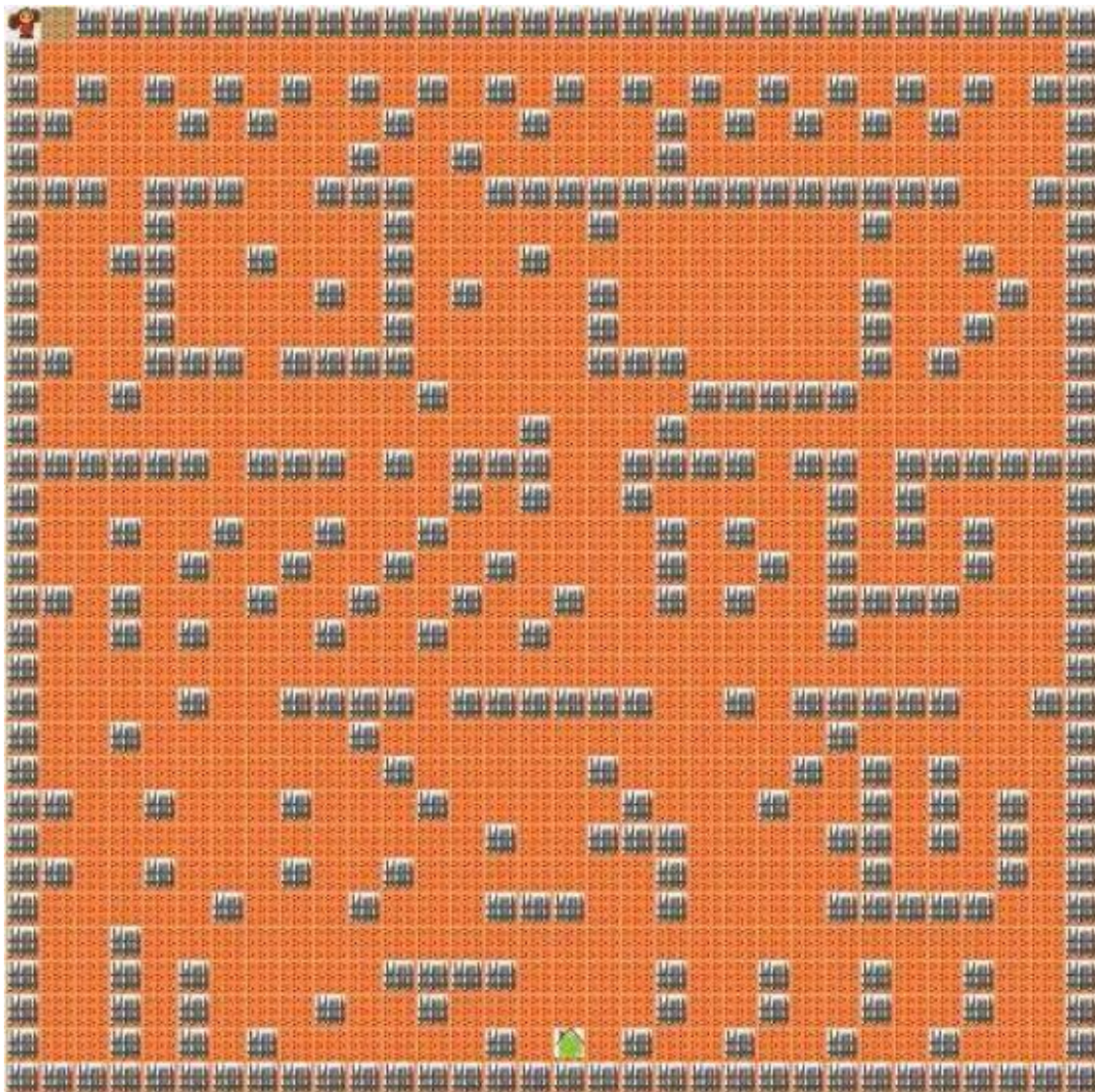


Рис. 1. Поле, по которому ходит робот

## 2. Генетический алгоритм

Для решения поставленной задачи, применяется генетический алгоритм. Работа генетического алгоритма состоит из нескольких фаз. Вначале происходит генерация первого поколения особей. Далее алгоритм начинает выполнение итеративного процесса построения следующего поколения из предыдущего.

При этом применяются процедуры:

- отбор – из предыдущего поколения выбирается часть особей; в данной работе используется ранговый отбор;
- скрещивание – из двух особей-родителей создаются две новые особи;
- мутация – случайным образом изменяется строение особи.
- большая мутация – каждая особь в поколении с заданной вероятностью заменяется новой особью, сгенерированной случайным образом.

Опишем более подробно алгоритм, применяемый в данной работе.

### 2.1. Функция приспособленности

Функция приспособленности (ФП) имеет следующий вид:

$$fitness(A) = \begin{cases} 200 + \min_{\text{path}}(|x_2 - x_1| + |y_2 - y_1|), & \text{робот не доходит до цели} \\ k, & \text{робот доходит до цели} \end{cases},$$

где  $A$  — автомат, принадлежащий пространству поиска,  $\text{path}$  — путь, который проходит робот за 200 ходов,  $(x_1, y_1)$  — координаты цели,  $(x_2, y_2)$  — координаты робота,  $k$  — число ходов, за которое робот доходит до цели.

### 2.2. Ранговый отбор

Особи сортируются на основе их ФП таким образом, чтобы

$$f_i \geq f_j \text{ для } i \geq j.$$

Затем каждой особи назначается ранг  $r_i$ , численно равный  $gSize - i$ , где  $gSize$  — размер поколения. Далее при отборе особей используется принцип: вероятность попадания особи в промежуточное поколение пропорциональна её рангу.

### 2.3. Оператор мутации

Оператор мутации реализован следующим образом. Для каждого перехода из каждого состояния действие на этом переходе и номер следующего состояния с вероятностью 0,05 изменяются на случайные. С той же вероятностью может измениться начальное состояние автомата.

## 2.4. Оператор большой мутации

Как уже упоминалось выше, оператор большой мутации реализован таким образом, что при его применении каждая особь в поколении с заданной вероятностью, численно равной 0,5, заменяется на новую. Замена происходит таким образом, что вновь созданный путём генерации случайной особи экземпляр заменяет одну особь в поколении. В данной работе оператор большой мутации имеет период равный пятистам поколениям, то есть большая мутация происходит через каждые пятьсот поколений.

## 2.5. Оператор скрещивания

При скрещивании алгоритм порождает две новые особи из двух особей родителей.

Рассмотрим процедуру скрещивания. Пусть оператор на вход получает две особи:  $p_1$  и  $p_2$ . Это, так называемые, особи родителей. Сделав итеративный проход по всем состояниям данных особей, создаются две новые —  $s_1$  и  $s_2$ , при этом для каждой итерации, для входных воздействий, численно равных 0 и 1, соответствующие выходные воздействия  $s_1$  и  $s_2$  с равной вероятностью наследуются от выходных воздействий особей  $p_1$  и  $p_2$ .

## 2.6. Метод генерации очередного поколения

Начальное поколение состоит из фиксированного числа случайно сгенерированных автоматов. Все автоматы в поколении имеют одинаковое наперед заданное число состояний. В данной работе число состояний равно восьми.

Для генерации очередного поколения используется классический генетический алгоритм.

Опишем используемый алгоритм.

- Сохранение элитной части особей:  $m = n * elitePart$ , где  $n$  — число особей в поколении,  $elitePart$  — доля элитных особей в поколении. В ходе исследований было установлено, что для наиболее эффективной работы алгоритма  $elitePart$  следует взять равной 0,05.
- Формирование промежуточного поколения из  $n - m$  особей посредством рангового отбора, при этом элита тоже участвует в отборе.
- Разбиение особей из промежуточного поколения на  $\frac{n-m}{2}$  случайных пар. Разбиение происходит путём генерации случайной перестановки чисел от 0 до  $n - m$  в массиве, в результате чего, пара подряд идущих элементов данного массива являет собой пару случайных чисел, при этом никакие из чисел не повторяются. Таким образом, рассматривая пары особей из промежуточного поколения с индексами, численно равными паре подряд идущих элементов массива, и было получено случайное разбиение на  $\frac{n-m}{2}$  пар.
- Применение с заданной вероятностью оператора кроссовера к полученным парам особей, замена пар особей родителей парами особей детей.
- Применение оператора мутации к каждой особи из промежуточного поколения.
- Формирование нового поколения из сохраненной элиты и промежуточного поколения.

### 3. Результаты работы

Для каждого из следующих значений вероятности применения оператора кроссовера (см. таблицу) было произведено 100 запусков. При каждом запуске алгоритм заканчивал работу, построив тысячу поколений. При этом запоминалось среднее значение ФП для каждого запуска. Строился график зависимости среднего значения ФП от вероятности применения оператора кроссовера (см. рис. 2).

Таблица. Результаты эксперимента

$p$	$\langle f \rangle$
0	163
0,1	210
0,2	148
0,3	116
0,4	157
0,5	210
0,6	179
0,7	179
0,8	203
0,9	180
1	139

Здесь:  $p$  – вероятность применения оператора кроссовера,  $\langle f \rangle$  - среднее значение ФП для ста запусков

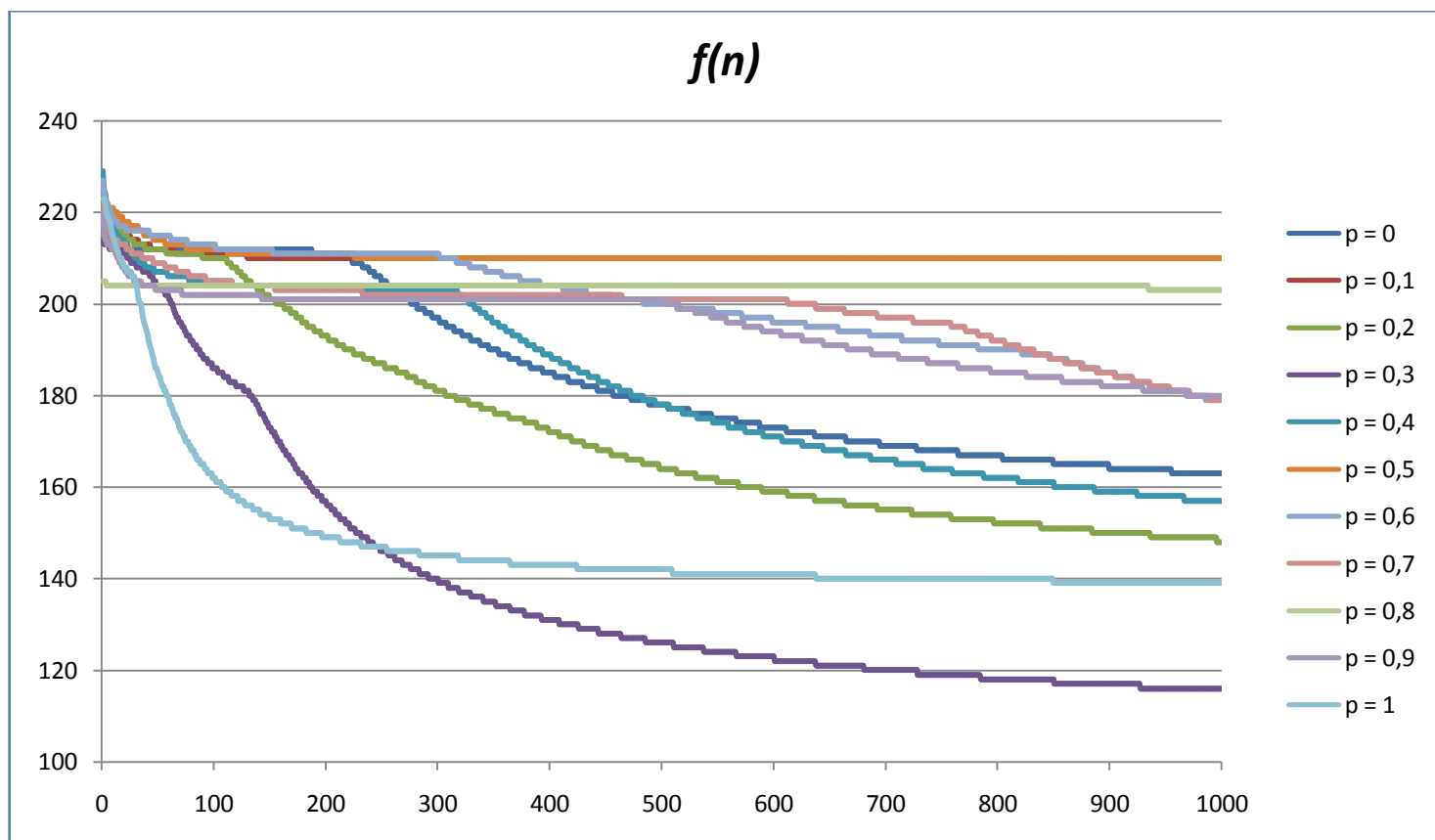


Рис. 2. График зависимости среднего значения ФП от номера поколения  $n$  при различных значениях вероятности применения кроссовера

## **Заключение**

В результате выполнения лабораторной работы получено, что при построении конечного детерминированного автомата Мили из восьми состояний для решения задачи о роботе, обходящем препятствия, с помощью генетического алгоритма, наилучшие результаты достигаются при значении вероятности применения оператора кроссовера, равном 0,3.



## Источники

1. Исходный код и документация к комплексу для изучения методов глобальной оптимизации GLOpt.

[http://is.ifmo.ru/courses/\\_giopt-src.rar](http://is.ifmo.ru/courses/_giopt-src.rar)

2. Яминов Б. Генетические алгоритмы.

<http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005>