

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

И. Ю. Якупов

**Отчет по лабораторной работе
«Использование генетических алгоритмов для
построения управляющих автоматов»**

Вариант №30

Санкт-Петербург
2011

Оглавление

Оглавление	2
Введение	3
1. Постановка задачи	3
1.1. Задача о роботе, обходящем препятствия	3
2. Реализация.....	4
2.1. Структура программы	4
2.2. Функция приспособленности	4
2.3. Оператор мутации	4
2.4. Оператор большой мутации	4
2.5. Метод генерации очередного поколения	5
3. Результаты работы.....	5
Заключение.....	10
Источники	10

Введение

Цель лабораторной работы — найти зависимость эффективности работы генетического алгоритма построения конечного автомата Мура, решающего задачу о роботе, обходящем препятствия, от периода большой мутации (ПБМ). Большая мутация (БМ) — это замена 90% особей в популяции на случайные.

При выполнении лабораторной работы не использовались какие-либо сторонние библиотеки для генетических алгоритмов. Все исходные коды в данной работе написаны на языке программирования C++ с использованием фреймворка Qt. Для построения графиков использована сторонняя библиотека Qwt [1].

1. Постановка задачи

Задача данной лабораторной работы — исследовать влияние периода большой мутации (БМ) на эффективность работы генетического алгоритма, строящего автомат Мура 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. 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.

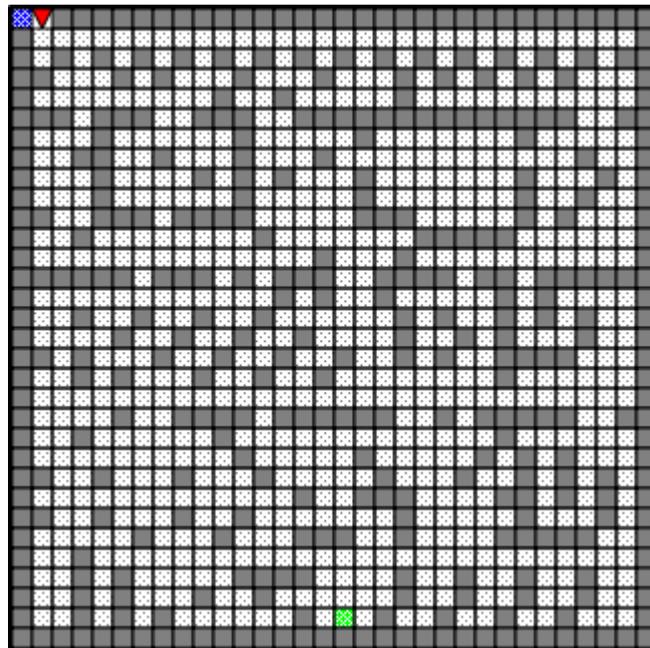


Рис. 1 — Игровое поле. Синий сектор — старт, зеленый — финиш.

2. Реализация

2.1. Структура программы

Программа написана на языке C++ и состоит из десяти классов:

- RoboMainWidget — виджет главного окна;
- MapWalk — поток, управляющий визуализатором;
- GraphicalField — виджет визуализатора;
- Field — игровое поле, хранящее информацию о расположении робота;
- MultiEvolver — класс, управляющий потоками и выполняющий миграцию особей;
- xGAGenome — геном;
- xGAPopulation — популяция;
- xGAOneThreadGA — однопоточный генетический алгоритм;
- xGAStatistics — статистика;
- xGARoboFuncFactory — функции, необходимые для оценки автомата.

2.2. Функция приспособленности

Было решено не использовать расстояние до финиша при оценке приспособленности. Как показало тестирование, возможны ситуации, когда расстояние до финиша минимально, но кратчайший путь до финиша занимает в десятки раз больше ходов. Такие ситуации возникают, когда робот и финиш находятся по разные стороны от сплошной стены.

Тестирование также показало, что практически все особи, способные добраться до финиша, делают это за менее чем 400 ходов. Поэтому, была выбрана следующая функция приспособленности (ФП):

$$F(A) = 400 - s,$$

где A — оцениваемая особь, s — число ходов, за которое робот добрался до финиша. Все особи со значением ФП, большим или равным двумстам, считаются успешными.

2.3. Оператор мутации

Оператор мутации реализован следующим образом. Случайно выбранные семь процентов генов инвертируются. Измениться могут начальное состояние автомата, переходы и действия.

2.4. Оператор большой мутации

Оператор большой мутации реализован следующим образом. 90% особей заменяется на случайные, остальные — мутируют. Затем популяция оценивается, и применяется элитизм: пять процентов лучших особей из прошлого поколения заменяют собой худших особей текущего поколения.

2.5. 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 десяти. Для генерации очередного поколения используется классический генетический алгоритм.

Опишем используемый алгоритм. Сначала, с заданной вероятностью, выбираются автоматы, которые будут участвовать в скрещивании. После этого выполняются скрещивания. Затем остальные автоматы мутируют. Наконец, полученная популяция оценивается, и применяется элитизм.

3. Результаты работы

Было произведено по десять запусков алгоритма при различных значениях периода большой мутации. Критерием эффективности работы алгоритма было решено считать номер популяции, в которой появилась особь, доходящая до финиша за менее чем 100 ходов. Тестирование показало, что эффективность работы алгоритма не зависит от периода большой мутации. Тестирование проводилось при следующих значениях ПБМ: 100000, 50000, 10000, 500, 100, 50 и при отключенной БМ. Искомые особи появлялись спустя 50-200 тысяч поколений, вне зависимости от выбора параметра. Было протестировано два способа мутации: спустя фиксированное число поколений вне зависимости от результатов, а также спустя фиксированное число поколений, на протяжении которого приспособленность лучшей особи не изменилась.

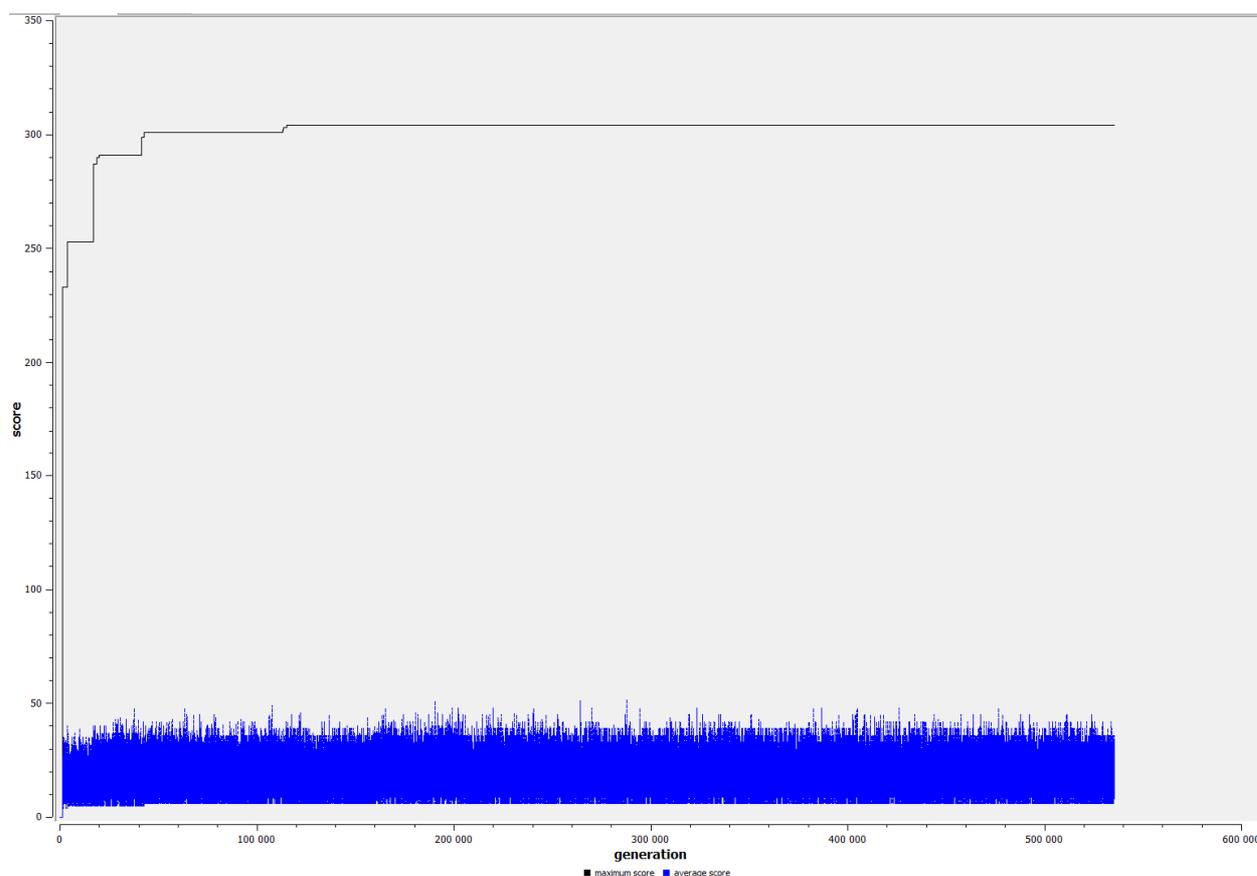


Рис. 1 — большая мутация отключена

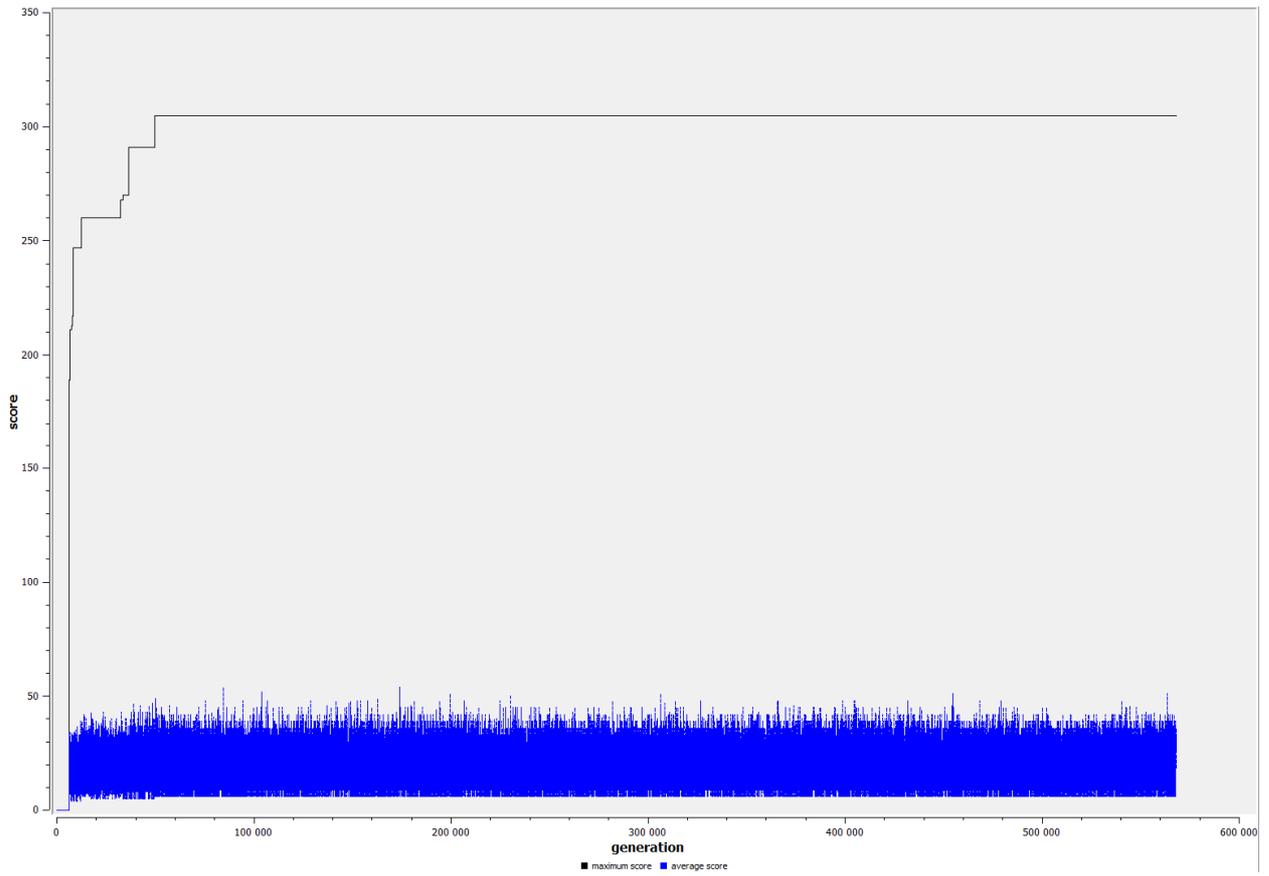


Рис. 2 — ПБМ составляет 100000 поколений

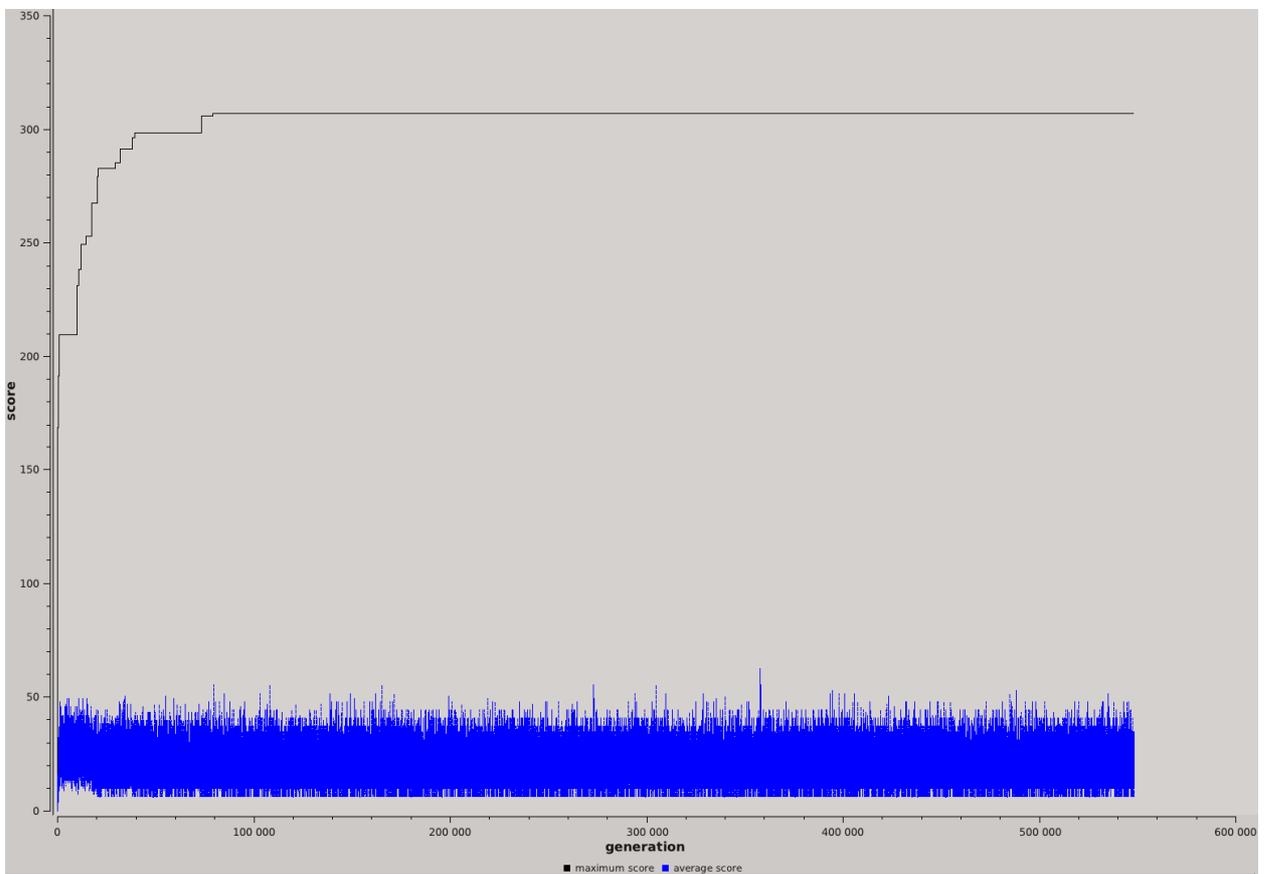


Рис. 3 — ПБМ составляет 50000 поколений, если значение ФП лучшей особи не изменилось

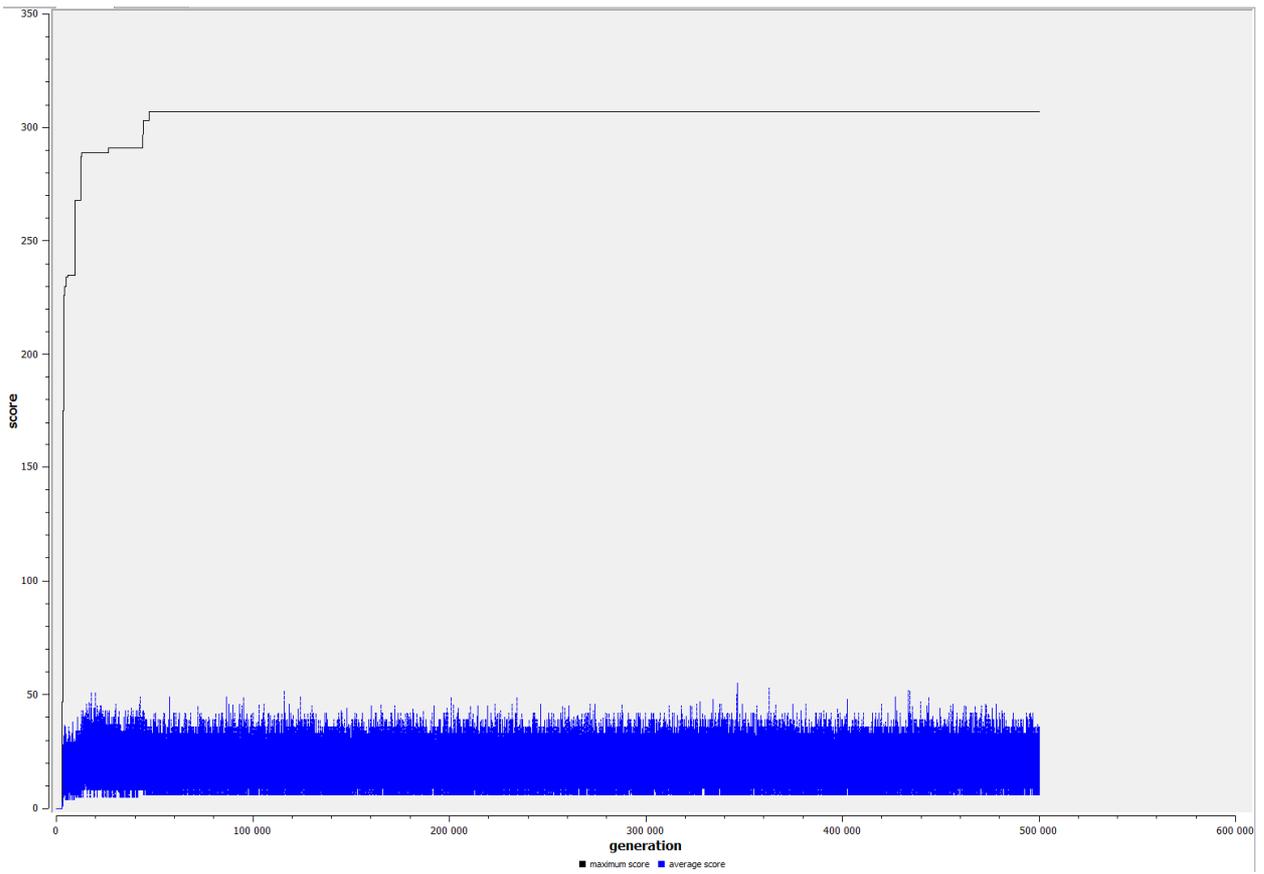


Рис. 4 — ПБМ составляет 10000 поколений

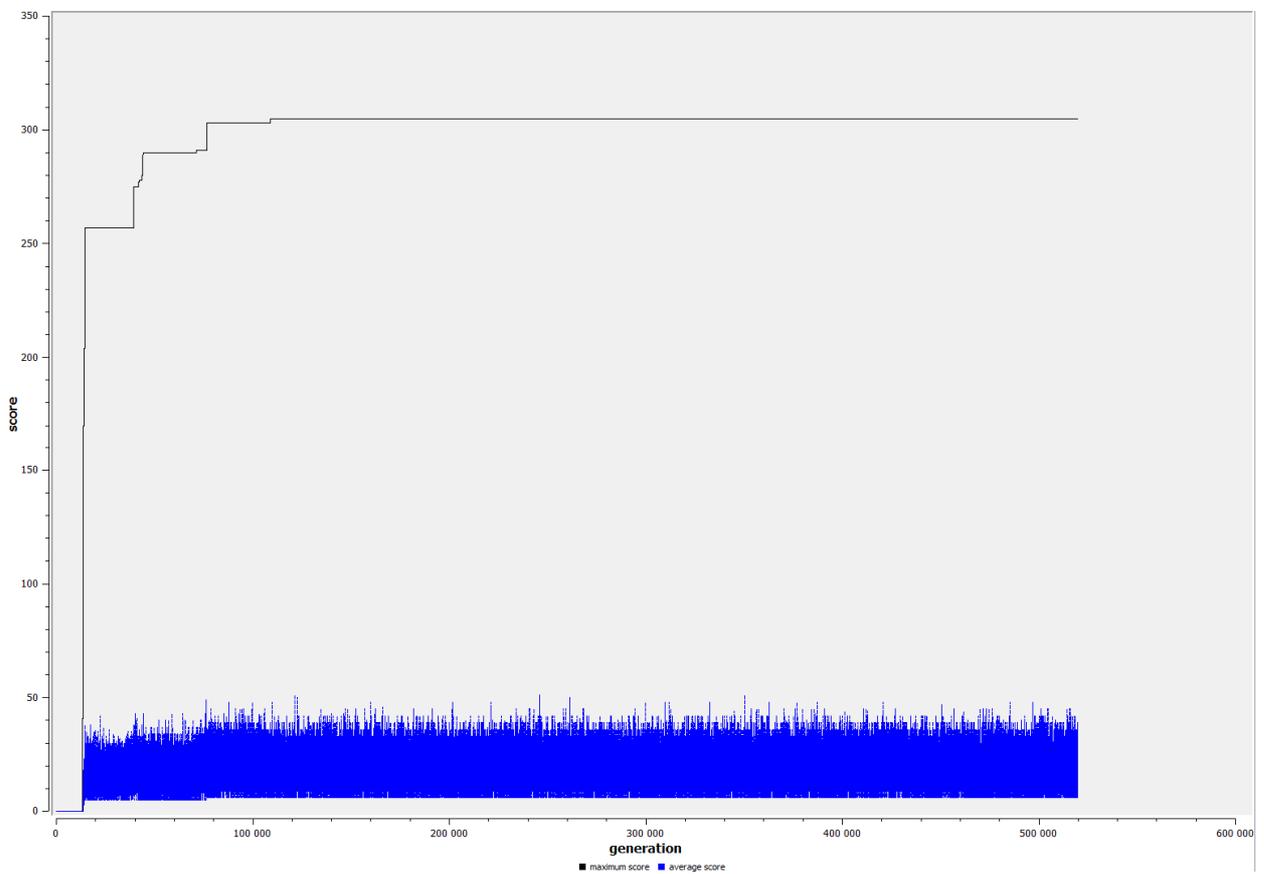


Рис. 5 — ПБМ составляет 500 поколений

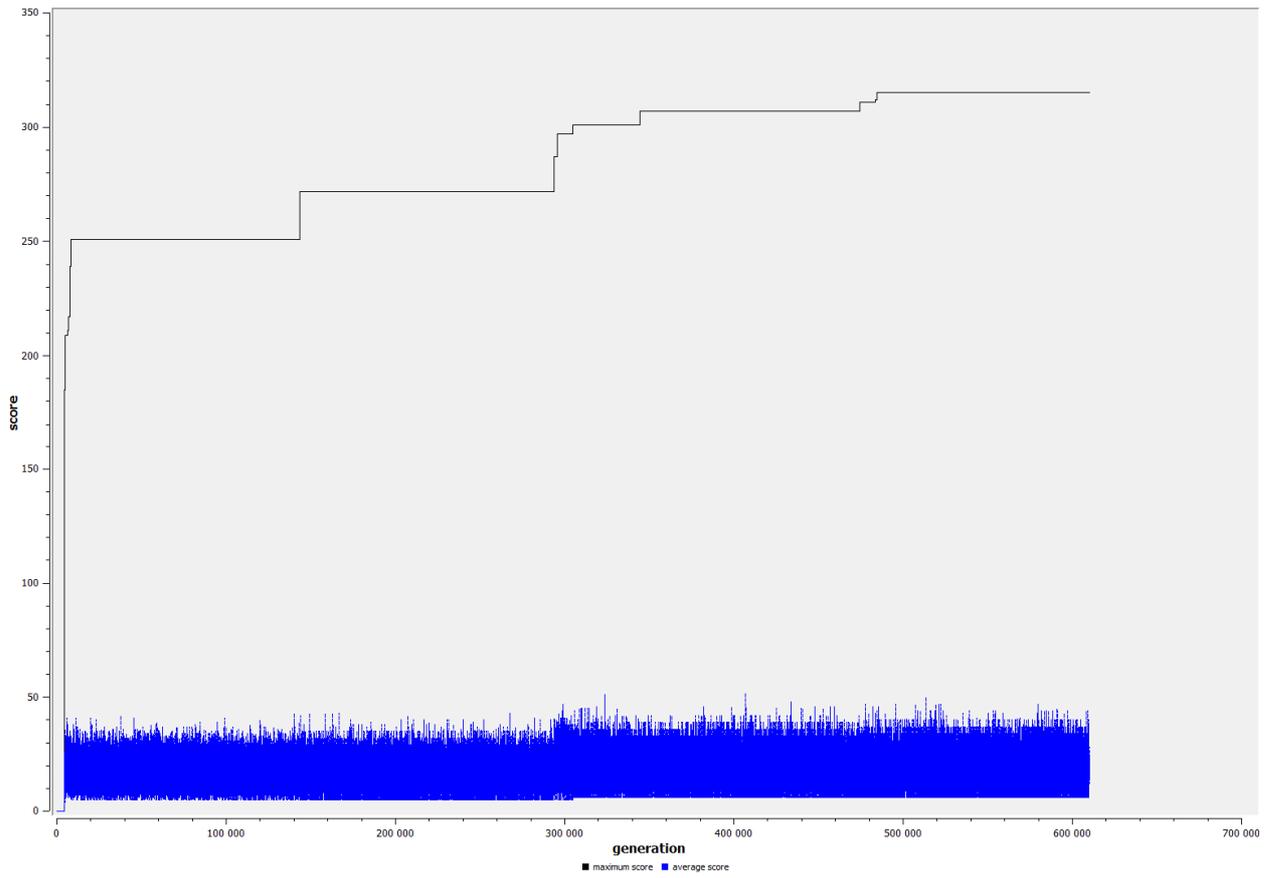


Рис. 6 — ПБМ составляет 100 поколений

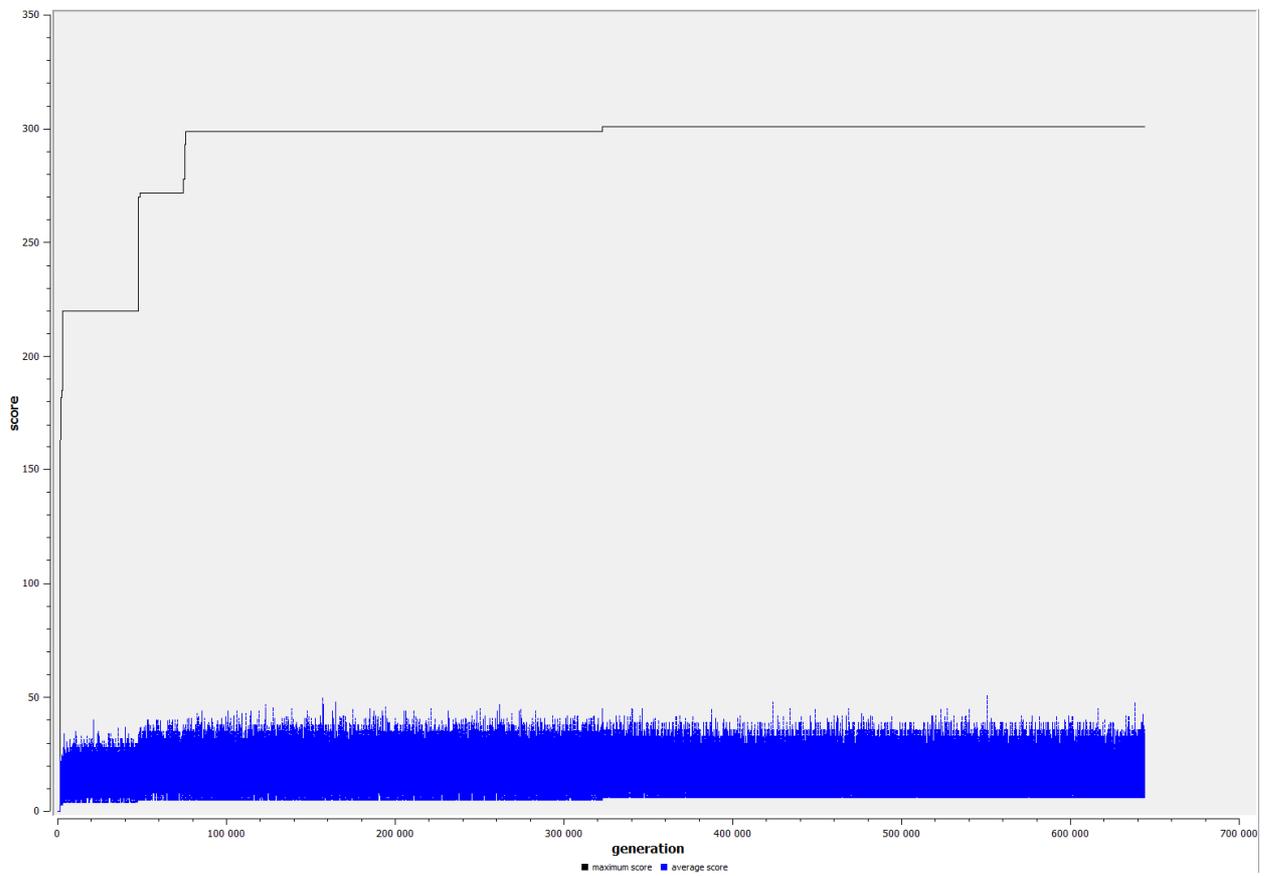


Рис. 7 — ПБМ составляет 50 поколений

Заключение

В результате выполнения лабораторной работы получено, что при построении конечного детерминированного автомата Мура из восьми состояний для решения задачи о роботе, обходящем препятствия, с помощью генетического алгоритма, эффективность работы алгоритма не зависит от периода большой мутации. Можно порекомендовать не использовать большую мутацию при решении данной задачи.

Источники

1. Qwt — Qt Widgets for Technical Applications
<http://qwt.sourceforge.net/>