

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

П.В. Авдеев

Отчет по лабораторной работе

«Применение генетических алгоритмов для построения автоматов в задаче об
умном муравье»

Вариант №1

Санкт Петербург

2011

Оглавление:

Введение.....	3
1. Постановка задачи.....	4
1.1. Задача об умном муравье.....	4
1.2. Автомат Мили.....	5
2. Генетический алгоритм.....	6
2.1. Представление особи.....	6
2.2. Оператор скрещивания.....	6
2.3. Оператор мутации.....	6
2.4. Метод генерации очередного поколения.....	7
2.5. Функция приспособленности.....	7
3. Результаты.....	8
Заключение.....	12
Источники.....	13

Введение

В данной лабораторной работе изучается зависимость эффективности работы генетического алгоритма построения автомата, решающего задачу об умном муравье, от вероятности мутации.

При выполнении этой работы использовались исходные коды фреймворка для работы с генетическими алгоритмами «Watchmaker» [1]. На основе примеров, содержащихся в этом пакете, был написан новый модуль, который решает поставленную задачу. Также для представления автомата был использован визуализатор из виртуальной лаборатории 3genetic [2]. Все исходные коды в данной работе написаны на языке программирования Java.

1. Постановка задачи

Задача данной лабораторной работы – исследовать влияние вероятности мутации на эффективность работы алгоритма, строящего автомат Мили, решающий задачу об умном муравье. Критерий оценки автомата заключается в том, что автомат, имея фиксированное число состояний, должен приводить к тому, что муравей, управляемый автоматом, съедает всю еду на поле за ограниченное число шагов.

1.1. Задача об умном муравье

В задаче об умном муравье рассматривается игровое поле, состоящее из клеток (рис. 1).

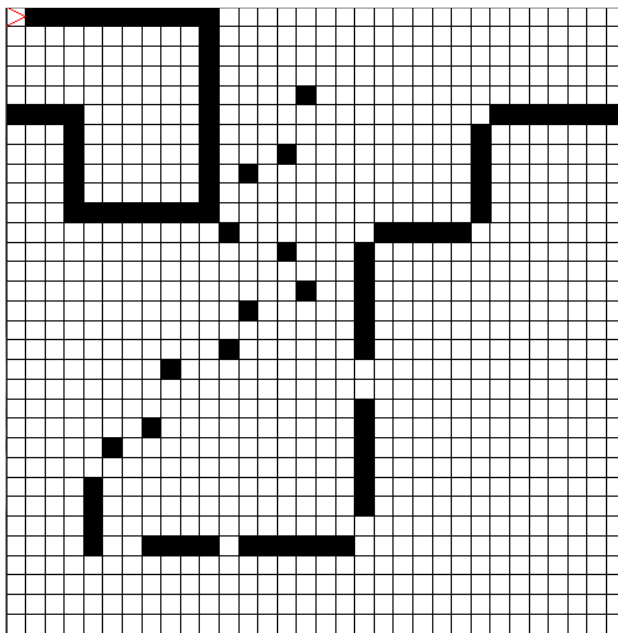


Рис. 1 – Игровое поле

Поле имеет размеры 32 на 32 клетки и располагается на поверхности тора. Некоторые клетки поля пусты, некоторые содержат по одному яблоку. Всего на поле 89 яблок.

Муравей начинает свое движение из клетки «Start». За один ход муравей может определить, есть ли в клетке перед ним яблоко, и выполнить одно из следующих действий:

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

Всего делается 200 ходов. Требуется построить автомат с определенным 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. На рис. 2, 3 приведены автоматы, решающие задачу об умном муравье.

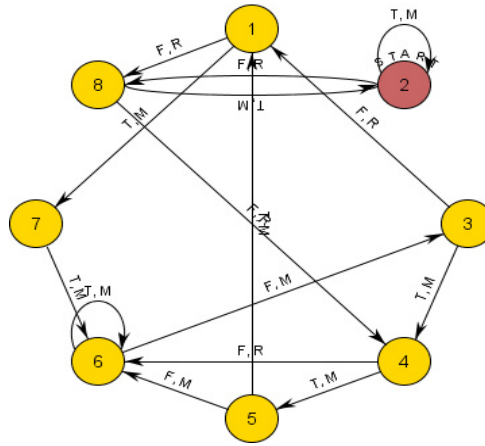


Рис. 2 – Автомат Мили с восемью состояниями, съедающий 89 яблок

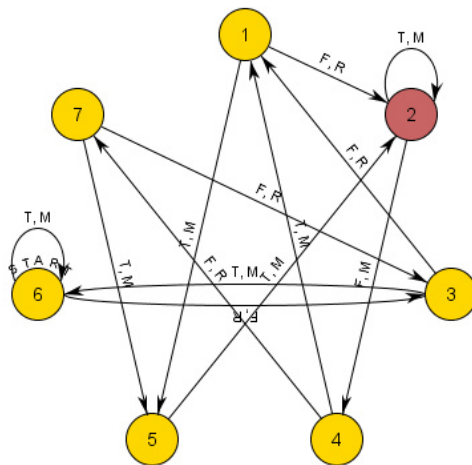


Рис. 3 – Автомат Мили с семью состояниями, съедающий 88 яблок

Над каждой дугой расположены два значения: входное и выходное воздействия.

Входные воздействия:

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

Выходные воздействия обозначаются так:

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

2. Генетический алгоритм

Для задачи поиска оптимального автомата, управляющего муравьём, применяется генетический алгоритм. Работа генетического алгоритма состоит из нескольких фаз. Вначале происходит генерация первого поколения особей. Далее алгоритм начинает выполнение итеративного процесса – на каждой итерации строится следующее поколение из предыдущего. При этом применяются процедуры:

- отбор – из предыдущего поколения выбирается часть особей; для сравнения особей между собой алгоритм использует функцию приспособленности, сопоставляющую каждой особи число, определяющее ее приспособленность;
- скрещивание – по двум особям-родителям создаются две новые особи;
- мутация – случайным образом изменяется строение особи.

Опишем более подробно алгоритм, применяемый в данной работе.

2.1. Представление особи

Особями в данном алгоритме являются автоматы Мили. Автомат Мили представляется в виде графа переходов. Непосредственно в коде программы используется двумерный массив `Transition[][] transitions`, который хранит переход (новое состояние и выходное воздействие) для каждой пары, состоящей из текущего состояния автомата и входного сигнала. В задаче об умном муравье значениями входной переменной являются 0 и 1 – присутствие или отсутствие еды в клетке перед муравьем.

2.2. Оператор скрещивания

При скрещивании алгоритм порождает две новые особи из двух особей родителей. Состояния автоматов нумеруются числами от 1 до N (у всех автоматов одинаковое число состояний). Рассмотрим процедуру скрещивания: для всех i (i от 1 до N) выполняются шаги:

1. Рассматривается состояние с номером i из первого родителя и состояние с тем же номером из второго; запоминаются переходы из этих состояний.
2. Рассматривается состояние с одинаковым номером i в автоматах – потомках; случайным образом переходы особей-родителей распределяются среди состояний особей-потомков.

2.3. Оператор мутации

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

2.4. Метод генерации очередного поколения

Начальное поколение генерируется из особей, заданных случайным образом. Рассмотрим процесс генерации нового поколения из текущего.

- 1) В новое поколение отправляется определенная часть всей популяции (1/4 от размера популяции) без применения операторов – элитизм. Таким способом удастся добиться более стабильной работы генетического алгоритма.
- 2) С помощью метода рулетки из исходного поколения выбирается количество недостающих особей до размера поколения.
- 3) К выбранным на шаге №2 особям применяется оператор скрещивания и мутации.
- 4) Восстанавливается размер популяции отсечением худших особей.

2.5. Функция приспособленности

Функция приспособленности вычисляется по формуле:

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

где Apples – число яблок, съедаемых муравьем за 200 шагов, Steps – номер шага, на котором муравей съедает последнее яблоко.

3. Результаты

Для автомата, состоящего из N ($N = 6, 7, 8$) состояний, были произведены запуски с вероятностями от 0,1 до 0,8 с шагом 0,1. Было произведено 100 запусков с размером популяции 150 особей и числом поколений не более 500. В каждом запуске бралась максимальная величина функции приспособленности в соответствующем поколении. После получения результатов ста запусков данные значения усреднялись.

Результаты запусков отражены на рис. 4 (для автомата с шестью состояниями) и рис. 6 (для автомата с семью состояниями) и рис. 8 (для автомата с восемью состояниями).

Можно заметить, что наибольшая эффективность достигается при p от 0,1 до 0,2. Рассмотрим этот промежуток подробнее с шагом вероятности 0,01.

По графикам, изображенным на рис. 5 (для автомата с шестью состояниями), рис.7 (для автомата с семью состояниями) и рис.9 (для автомата с восемью состояниями) можно сделать вывод, что при выборе вероятности в пределах от 0,08 до 0,12 генетический алгоритм работает наиболее эффективно.

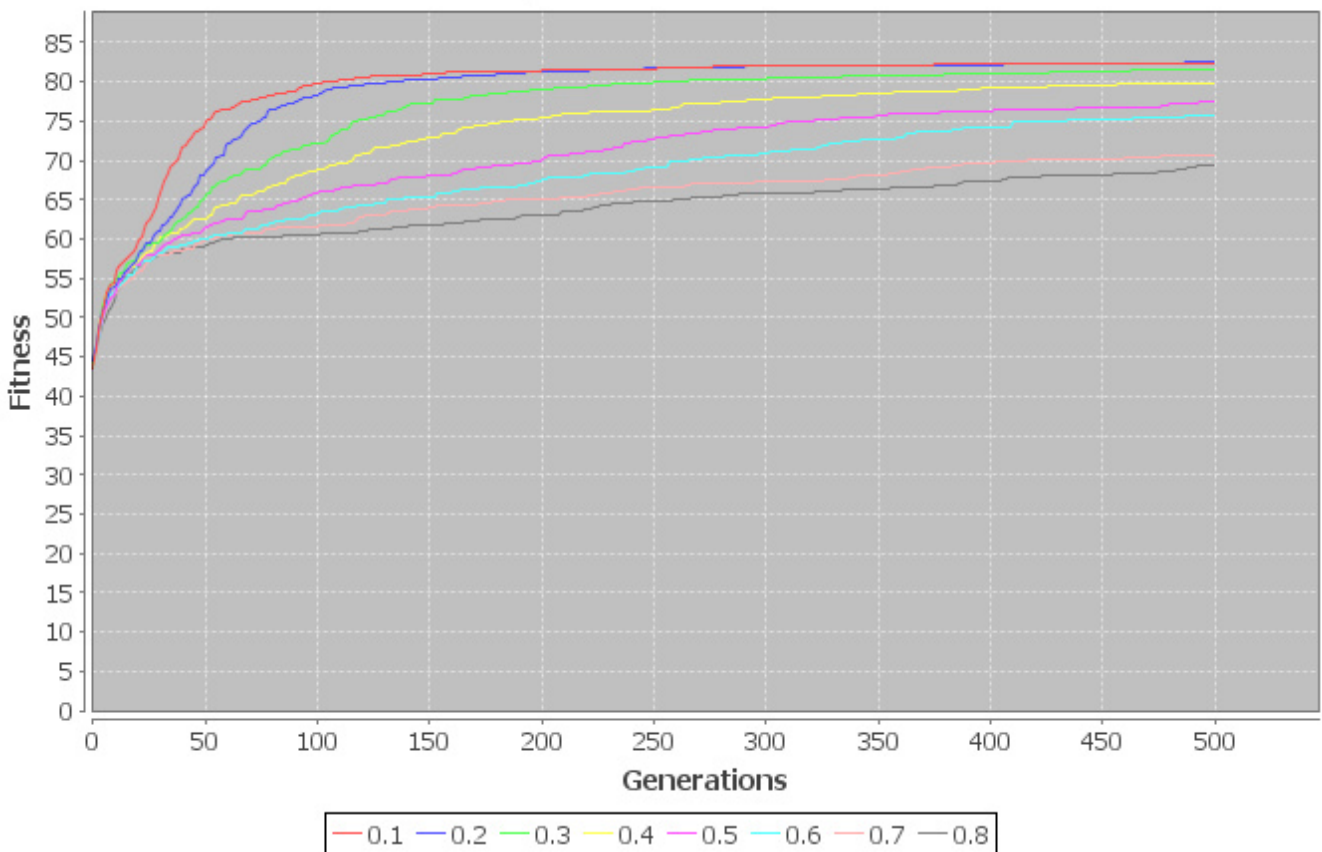


Рис. 4 – График функции приспособленности для автомата с шестью состояниями: 100 запусков, размер популяции 200, число поколений 500

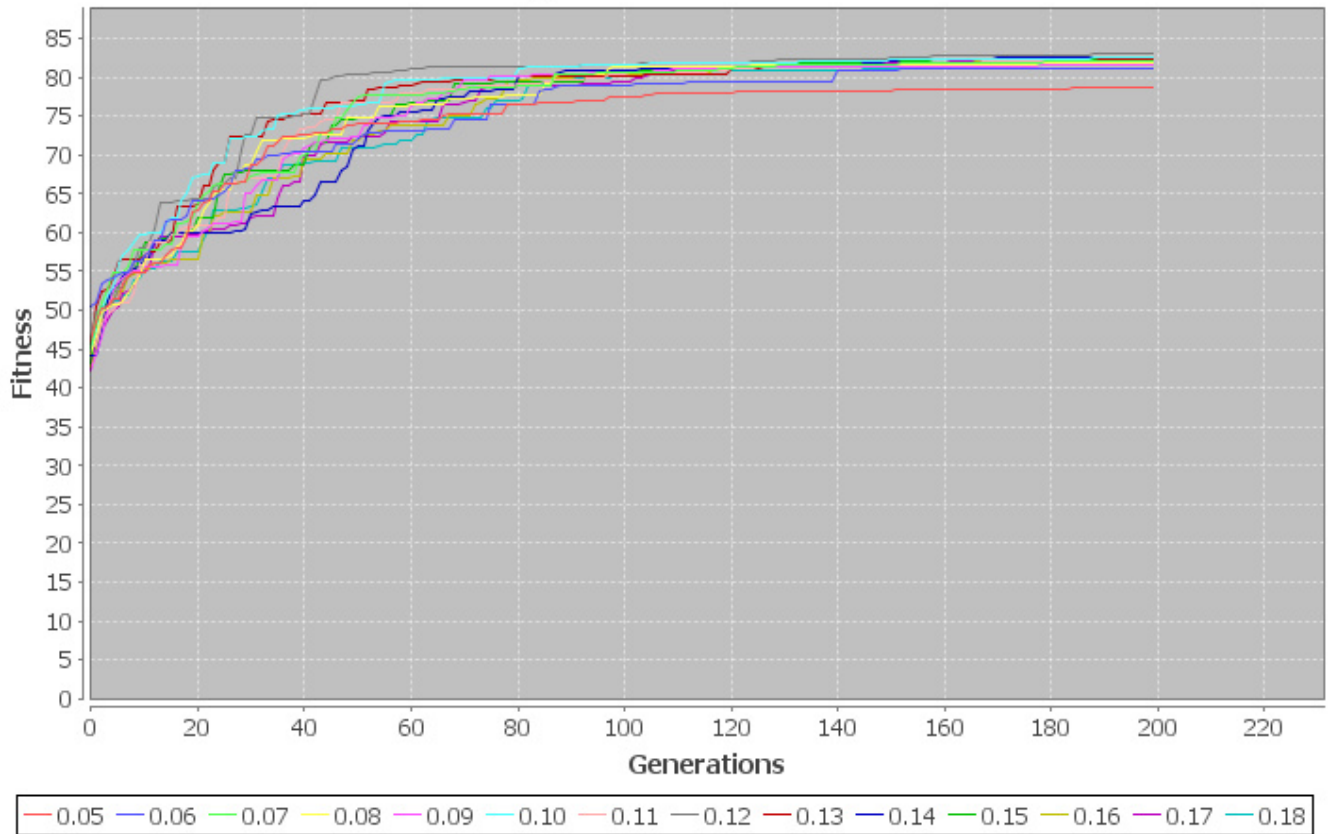


Рис. 5 – График функции приспособленности для автомата с шестью состояниями: 10 запусков, размер популяции 350, число поколений 200

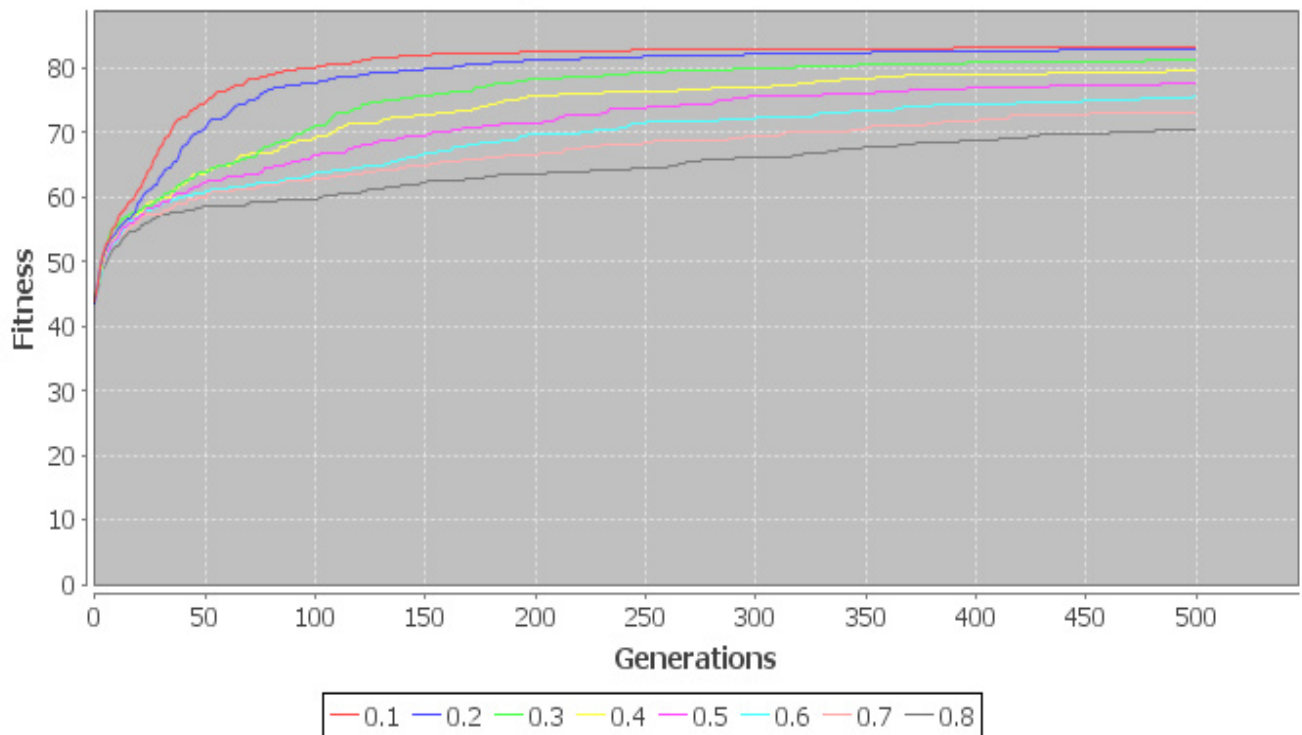


Рис. 6 – График функции приспособленности для автомата с семью состояниями: 100 запусков, размер популяции 200, число поколений 500

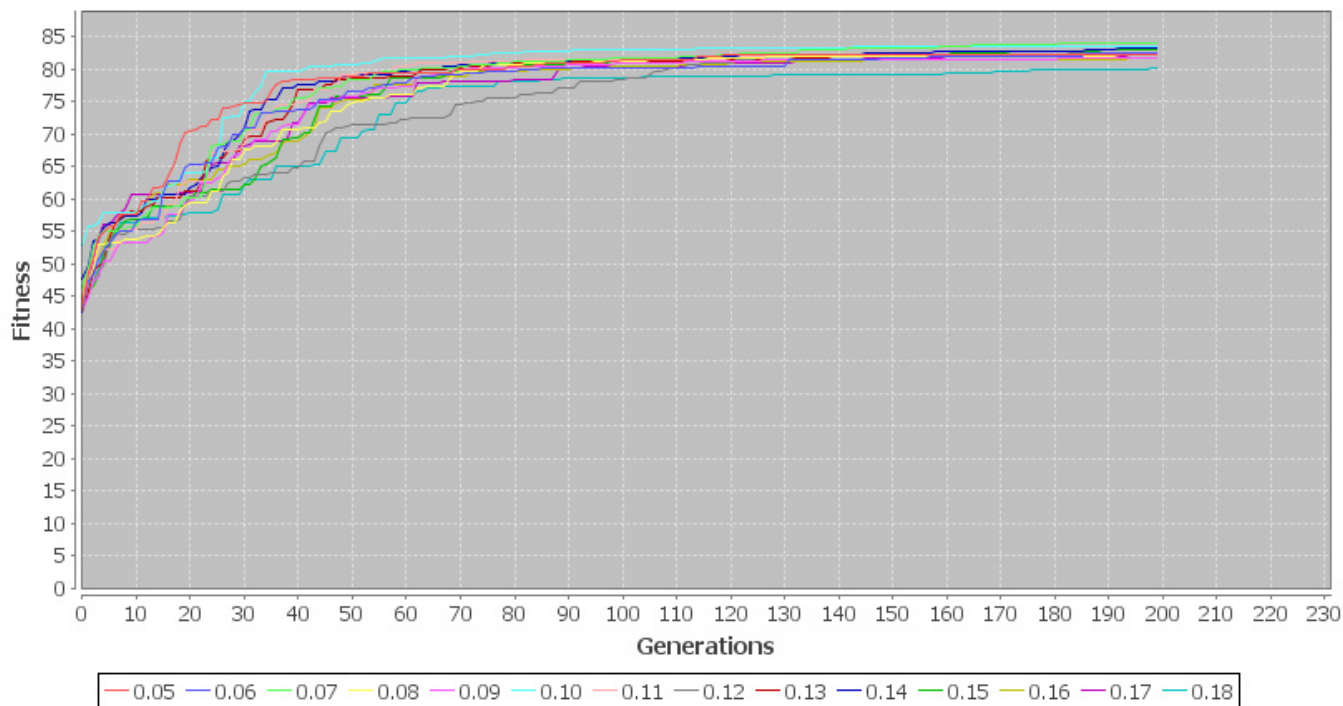


Рис. 7 – График функции приспособленности для автомата с семью состояниями: 10 запусков, размер популяции 350, число поколений 200

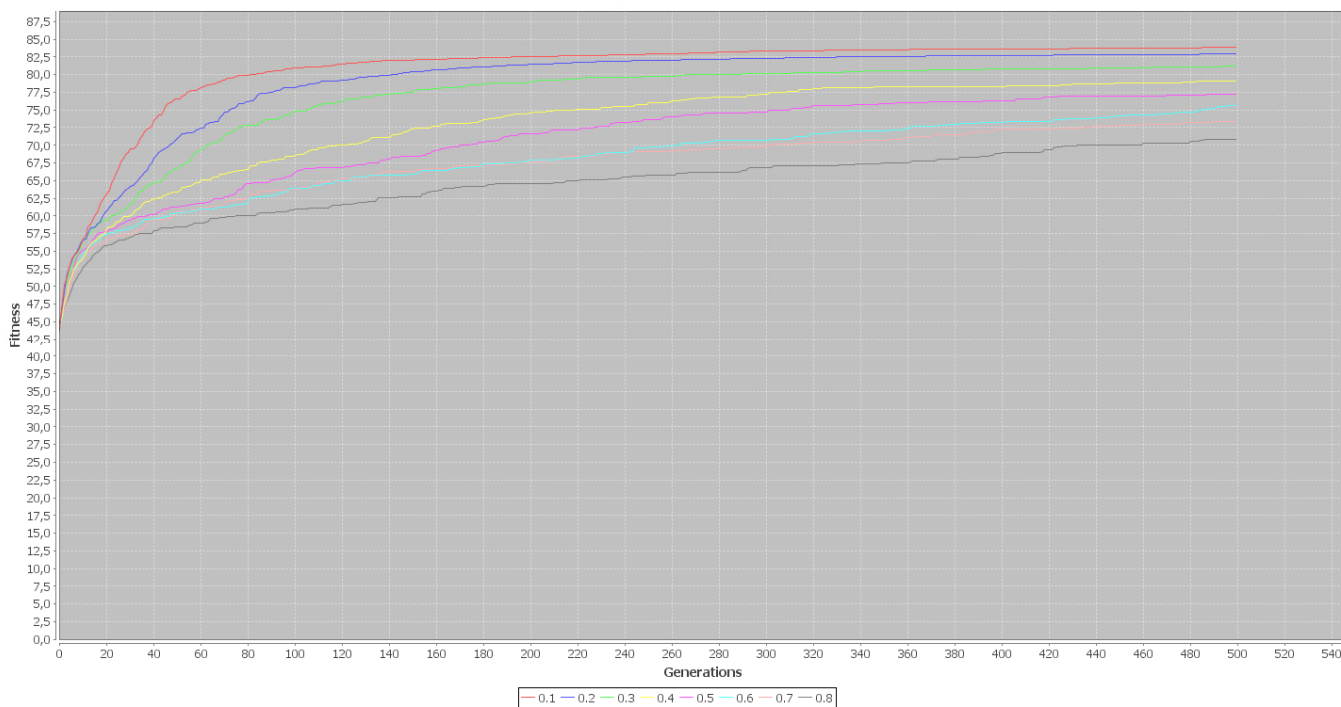


Рис. 8 – График функции приспособленности для автомата с восемью состояниями: 100 запусков, размер популяции 200, число поколений 500

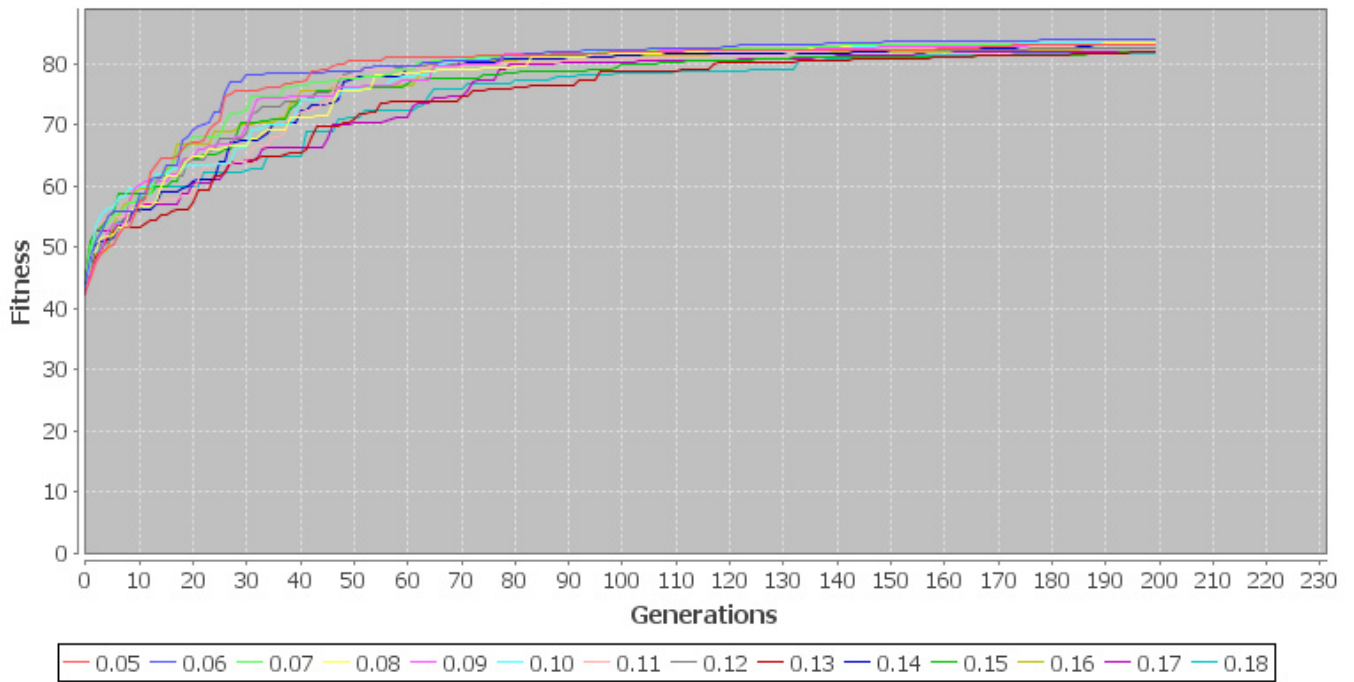


Рис. 9 – Графики функции приспособленности для автомата с восемью состояниями: 10 запусков, размер популяции 350, число поколений 200

Заключение

Результаты лабораторной работы показали, что эффективное построение автомата Мили с шестью, семью, восемью состояниями, который решает задачу об умном муравье, наблюдается при значениях вероятности мутации в диапазоне от 0,05 до 0,15. В качестве оптимального значения вероятности для описанного в работе оператора мутации можно порекомендовать значение 0,1.

Источники

1. Документация пакета «Watchmaker».

<http://watchmaker.uncommons.org/manual/index.html>

2. Инструкция по созданию plugin'ов к виртуальной лаборатории:

http://svn2.assembla.com/svn/not_instrumental_tool/docs/pdf/interface_manual.pdf