

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

Е.А. Лукьянец

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

Вариант № 9

Санкт-Петербург
2011 г.

Оглавление

Введение.....	3
1. Постановка задачи.....	3
Задача о роботе, обходящем препятствия.....	3
Автомат Мили.....	4
2. Генетический алгоритм.....	4
Создание нового поколения.....	4
Оператор мутации.....	4
Операторы кроссовера.....	4
«Стандартный» кроссовер (DefaultCrossover).....	4
«Элитный» кроссовер (EliteCrossover).....	4
Кроссовер «по состояниям» (StateCrossover).....	5
Функция приспособленности.....	5
3. Результаты работы.....	5
Заключение.....	5
Приложение. Исходный код.....	7

Введение

В лабораторной работе необходимо найти зависимость эффективности работы генетического алгоритма построения автомата, решающего задачу о роботе, обходящем препятствия, от вероятности применения кроссовера. Для решения задачи была написана программа на языке программирования *Java*, полностью реализующая генетический алгоритм.

1. Постановка задачи

Задача лабораторной работы — исследовать влияние вероятности применения различных операторов кроссовера на эффективность работы генетического алгоритма, строящего автомат Мили из 5 состояний, решающий задачу о «Роботе, обходящем препятствия» («Obstacle-avoiding robot problem»). Критерий оценки автомата состоит в том, что автомат, имея фиксированное число состояний, должен приводить к тому, что робот, управляемый автоматом, доходит из стартовой точки до цели за минимальное число шагов.

1.1. Задача о роботе, обходящем препятствия

Имеется игровое поле 32×32 клетки, ограниченное забором. В определенных местах поля размещены препятствия. Робота необходимо проложить путь из начальной позиции на карте до цели, обходя препятствия. За ход робот может выполнить следующие действия:

- повернуть налево;
- повернуть направо;
- сделать шаг вперед, если в новой клетке нет препятствия или забора;
- ничего не делать.

Игра длится 200 ходов. Цель игры — построить робота, который за минимальное число ходов сумеет добраться из стартовой точки до цели. На рис. 1 изображен пример поля.

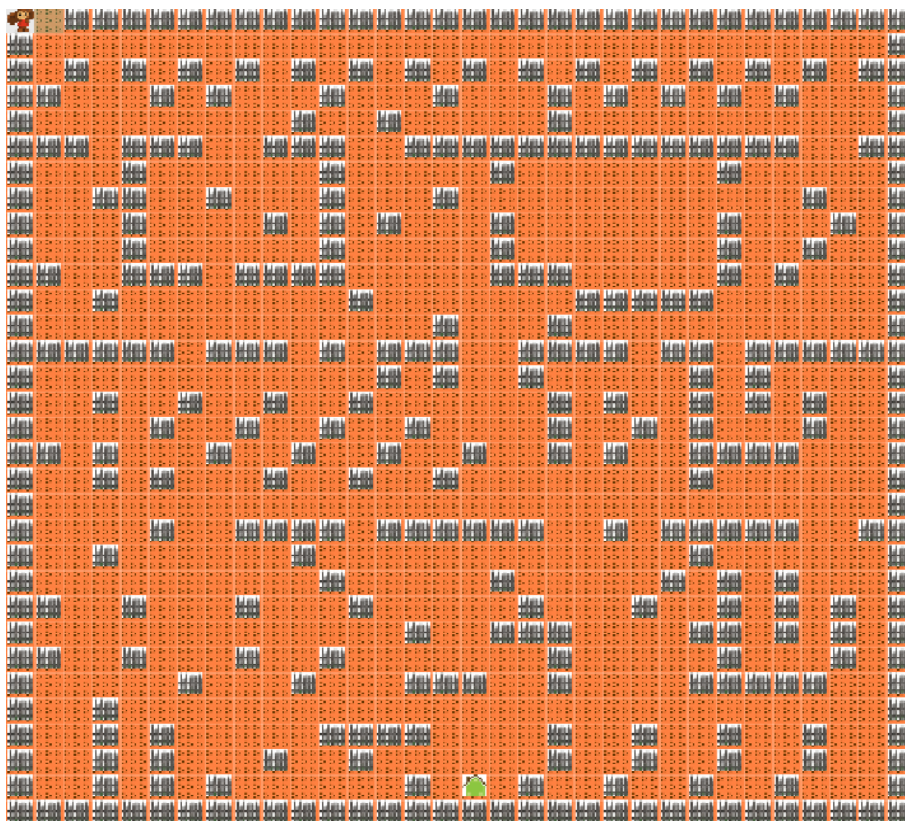


Рис. 1. Поле, по которому ходит робот. Старт в левом верхнем углу, конец обозначен «домиком», серые клетки — стены, остальные клетки свободные

1.2. Автомат Мили

Автомат Мили — это конечный автомат, генерирующий выходные воздействия в зависимости от текущего состояния автомата и входных сигналов.

В данном случае входной сигнал единственный — есть впереди робота препятствие или нет.

2. Генетический алгоритм

Для поиска наилучшего автомата, управляющего роботом, используется генетический алгоритм. Работа генетического алгоритма состоит из нескольких фаз: вначале создаётся первое поколение особей, далее алгоритм, используя операторы мутации и кроссовера, по данному поколению строит следующие.

2.1. Создание нового поколения

Начальное поколение состоит из фиксированного числа случайно созданных автоматов. Все автоматы во всех запусках имеют одинаковое число состояний.

Переход к новому поколению состоит из двух стадий: некоторая часть особей скрещивается между собой, после чего какая-то часть поколения мутирует. Отбор при этом не используется.

Также используется элитизм: часть наиболее хороших в условии задачи особей переносится в следующее поколение без применения к ним оператора мутации. Благодаря этому достигается более стабильная работа генетического алгоритма. Стоит заметить, что без применения элитизма этот алгоритм перестает работать из-за отсутствия «давления»: из поколения в поколение приспособленность особей не улучшается.

2.2. Оператор мутации

Мутация реализована следующим образом: каждая особь, кроме «элитных» с некоторой наперед заданной вероятностью мутирует одним из 5 возможных способов:

- стартовое состояние меняется на случайное;
- у одного из состояний на одном из переходов действие меняется на случайное;
- у одного из состояний случайно меняется состояние, в которое ведет переход;
- у одного из состояний меняются местами состояния, в которые ведут переходы;
- случайным образом создается новая особь.

2.3. Операторы кроссовера

В программе реализовано три различных кроссовера:

1. «Стандартный» кроссовер (*DefaultCrossover*). Наперед заданная часть неэлитных особей скрещивается между собой. При этом в кроссовере участвует две неэлитные особи. Для каждого состояния производится одно из четырех равновероятных действий:
 - особи меняются состояниями, в которые ведут переходы, и действиями на переходах по первому значению сигнала (впереди нет стены);
 - особи меняются состояниями, в которые ведут переходы, и действиями на переходах по второму значению сигнала (впереди есть стена);
 - особи меняются состояниями, в которые ведут переходы, и действиями на переходах по обоим значениям сигнала;
 - ничего не происходит.
2. «Элитный» кроссовер (*EliteCrossover*). Наперед заданная часть неэлитных особей скрещиваются с элитными. При этом в кроссовере участвует две особи: неэлитная, которая в результате применения оператора меняется, и элитная, которая не изменяется. Для каждого состояния неэлитной особи производится одно из четырех равновероятных действий:

- состояние, в которое ведет переход, и действие на переходе по первому значению сигнала (впереди нет стены) заменяются на соответствующие состояние и действие на переходе по первому значению сигнала в элитной особи;
 - состояние, в которое ведет переход, и действие на переходе по второму значению сигнала (впереди есть стена) заменяются на соответствующие состояние и действие на переходе по второму значению сигнала в элитной особи;
 - состояние, в которое ведет переход, и действия на переходах по обоим значениям сигнала заменяются на соответствующие состояние и действия на переходах в элитной особи;
 - ничего не происходит.
3. Кроссовер «по состояниям» (*StateCrossover*). Наперед заданная часть неэлитных особей скрещивается между собой. В этом кроссовере участвует две неэлитные особи. В процессе кроссовера они меняются двумя случайными состояниями, то есть: если меняются состояния под номером i из первого автомата и под номером j из второго, то все переходы и действия по обоим значениям сигнала состояния i меняются на соответствующие им переходы и действия состояния j и наоборот.

2.4. Функция приспособленности

В качестве функции приспособленности использовалось выражение $L + \frac{k}{500}$, где:

- L — манхеттенское расстояние от поля, в котором остановился робот после 200 шагов, до поля, являющегося концом пути. Если робот дошел до конца не более, чем за 200 шагов, L равно 0;
- k — количество шагов, которое потребовалось роботу, чтобы дойти до цели. Если за 400 шагов он до конца не дошел, k равно 400.

В процессе генетического алгоритма функция приспособленности будет уменьшаться.

3. Результаты работы

Для каждого из трех кроссоверов и для каждой вероятности (0,0, 0,1, ..., 1,0) были проведены эксперименты, каждый — по 100 запусков генетического алгоритма, размер поколения при этом составлял 300 особей. В каждом запуске бралась величина минимальной функции приспособленности в соответствующем поколении. После 100 запусков результаты усреднялись. Результаты для каждого из видов кроссовера отражены на рис. 2—4.

Остальные параметры генетического алгоритма: 10000 поколений, вероятность мутации — 0,1, часть «элитных» особей — 0,04.

Можно заметить, что для «стандартного» кроссовера наибольшая эффективность достигается при вероятности около 0,8, для «элитного» кроссовера — при вероятности около 0,1, для кроссовера «по состояниям» — при вероятности около 0,5.

Заключение

Результаты лабораторной работы показали, что наиболее эффективное построение автомата Мили с 5 состояниями, решающего задачу о роботе, обходящем препятствия, наблюдается при применении так называемого «элитного» кроссовера при значениях вероятности около 0,1.

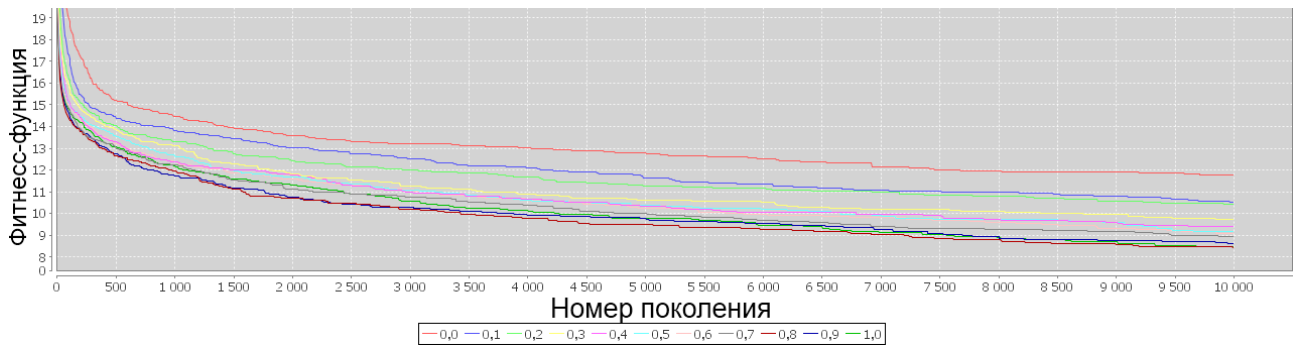


Рис. 2. График зависимости значения функции приспособленности от номера поколения при использовании «стандартного» кроссовера

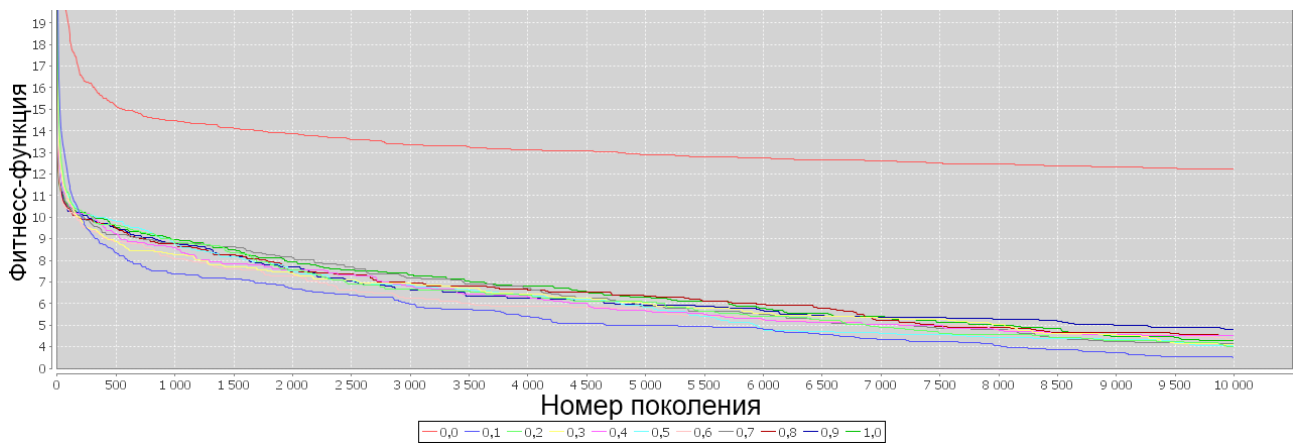


Рис. 3. График зависимости значения функции приспособленности от номера поколения при использовании «элитного» кроссовера

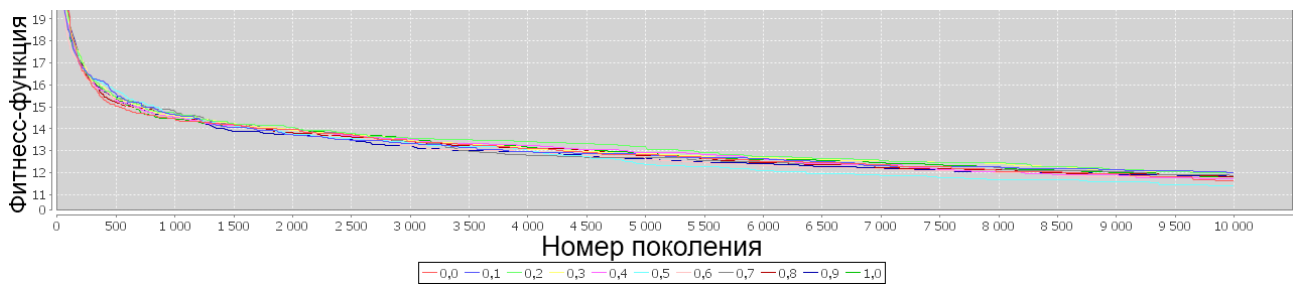


Рис. 4. График зависимости значения функции приспособленности от номера поколения при использовании кроссовера «по состояниям»

Приложение. Исходный код

Исходный код (оформленный как проект для среды разработки *Eclipse*), а также некоторые результаты запусков генетического алгоритма могут быть найдены по ссылке: <https://github.com/leugenea/AutomataOARobot>.