

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

Г. О. Чикишев

Отчет по лабораторной работе
«Использование генетических алгоритмов для построения
управляющих автоматов»
Вариант № 14

Санкт-Петербург
2009

Оглавление

Введение	3
1. Постановка задачи	4
1.1 Задача об «Умном муравье-3».....	4
1.2 Автомат Мура	5
1.3 Представление автомата	5
2. Генетический алгоритм.....	6
2.1. Создание начального поколения.....	6
2.2. Мутация.....	6
2.3. Скрещивание.....	6
2.4. Отбор особей для формирования следующего поколения.....	6
2.5. Вычисление функции приспособленности	6
2.6. Настраиваемые параметры алгоритма.....	7
3. Результаты.....	8
Заключение.....	11
Источники	12

Введение

В данной работе исследуется применение генетических алгоритмов для генерации конечных автоматов. В качестве примера взята задача «Умный муравей-3». Результатом работы является автомат Мура, сгенерированный с помощью генетического алгоритма и представляющий логику муравья.

При выполнении работы использовалась программа «Виртуальная лаборатория», написанная студентами кафедры «Компьютерные технологии» и позволяющая реализовывать генетические алгоритмы и особи для них в виде подключаемых модулей – плагинов [1].

1. Постановка задачи

Задача данной работы – построить близкий к оптимальному автомат Мура, решающий задачу об «Умном муравье-3». Оптимальность в *широком смысле* следует понимать следующим образом: автомат должен иметь минимальное число состояний, и муравей, управляемый данным автоматом, должен съесть как можно больше еды, выполнив при этом как можно меньше шагов. В работе ставится цель достигнуть максимально возможного значения второго параметра (количество еды) при фиксированных остальных параметрах (число шагов и число состояний автомата).

1.1 Задача об «Умном муравье-3»

Приведем описание задачи об «Умном муравье-3» [2]. Игра происходит на поверхности тора размером 32 на 32 клетки (рис. 1). В некоторых клетках (обозначены на рис. 1 черным цветом) находится еда. Клетки поля, в которых нет еды, обозначены серым цветом. Для каждой клетки вероятность появления еды в ней равна заранее определенной величине ρ .

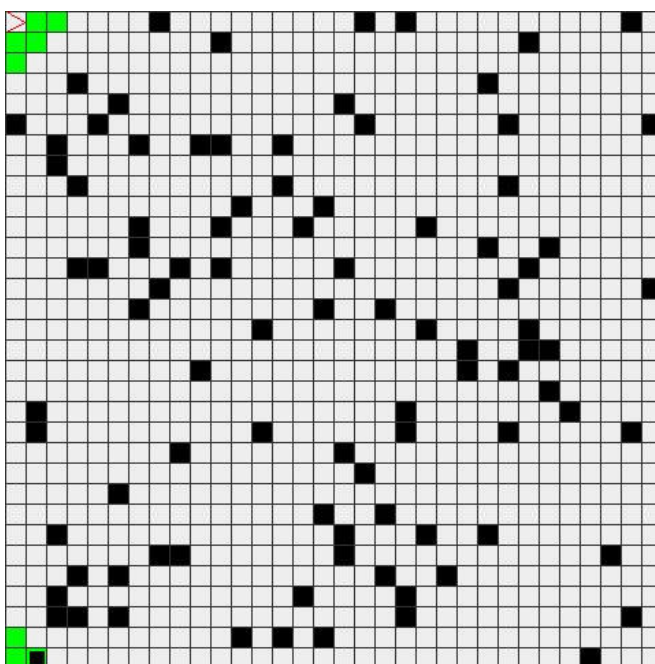


Рис. 1. Игровое поле

В левой верхней клетке в начале игры находится муравей (на рис.1 он обозначен красным треугольником, который также показывает направление его первоначального движения). Он занимает одну клетку и смотрит в одном из четырех направлений (север, юг, запад, восток). В начале игры муравей смотрит на восток. Муравей умеет определять, в каких из восьми находящихся рядом с ним и помеченных зеленым цветом клетках находится еда. Таким образом, в каждой клетке для муравья существует 256 входных воздействий (условий). За один игровой ход муравей может совершить одно из трех действий:

- сделать шаг вперед, съедая при этом еду, если она там находится (M – move);
- повернуть налево (L – left);
- повернуть направо (R – right).

Съеденная муравьем еда не восполняется, муравей жив на протяжении всей игры, еда не является необходимым ресурсом для его жизни. Игра длится 200 ходов, на каждом из которых муравей совершает одно из четырех действий. По окончании игры подсчитывается количество еды, съеденной муравьем. Это значение и есть результат

игры. Цель игры – создать муравья, который съест как можно больше еды. При этом отметим, что муравьи, съедающие всю еду, заканчивают игру с одинаковым результатом, который не зависит от того, сколько ходов им на это потребовалось.

1.2 Автомат Мура

В рассматриваемом варианте автомат для управления муравьем является автоматом Мура – совокупностью пяти объектов $A = \{S, X, Y, \delta, \mu\}$, где S – множество вершин, X – множество входных воздействий, Y – множество выходных воздействий, δ – отображение $S \times X \rightarrow S$, μ – отображение $S \rightarrow Y$.

Муравей получает входное воздействие в виде сообщения, описывающего восемь ближайших клеток, в которых либо есть еда, либо ее нет. Автомат, исходя из текущего состояния и воздействия, переходит в следующую вершину (отображение δ). На следующем этапе по отображению μ выбирается действие муравья – его ход.

1.3 Представление автомата

В рассматриваемом варианте муравей представляется массивом битов. По числу состояний S определяется необходимое число битов для хранения номера состояния, исходя из его двоичного представления. В начале массива располагается номер стартового состояния. Далее S раз повторяется следующая конструкция: номер вершины для перехода в случае отсутствия пищи на клетке перед муравьем, номер вершины в случае наличия пищи, выходное воздействие (ход вперед, поворот налево, направо, остаться на месте). Ход кодируется двумя битами: 00 – остаться на месте, 01 – повернуться налево, 10 – повернуться направо, 11 – сделать ход вперед.

2. Генетический алгоритм

Разработанный алгоритм генетического программирования [3] состоит из пяти частей:

- создание начального поколения;
- мутация;
- скрещивание (кроссовер);
- отбор особей для формирования следующего поколения;
- вычисление функции приспособленности (фитнес-функции).

2.1. Создание начального поколения

Каждая особь представляет собой некоторый конечный автомат Мура с заданным числом состояний, описывающий поведение муравья.

Начальное поколение состоит из фиксированного числа (по умолчанию – 200) случайно сгенерированных автоматов Мура с заранее определенным числом состояний.

2.2. Мутация

При мутации стартовым состоянием становится случайное состояние из автомата, затем случайно выбирается один из двух равновероятных вариантов:

- изменение действия, совершаемого муравьем, который находится в новом стартовом состоянии – случайно и равновероятно выбирается выходной сигнал из множества Y .
- изменение состояния, в которое ведет переход, – случайно и равновероятно выбирается переход. После этого состояние, в которое ведет переход, заменяется на случайно выбранное состояние.

Также в генетическом алгоритме предусмотрена операция «большой мутации», которая осуществляет генерацию полностью новых случайных автоматов. Она может быть использована, когда в популяции поведение муравьев почти идентично.

2.3. Скрещивание

Оператор скрещивания получает на вход две особи и выдает также две особи. Процесс скрещивания происходит следующим образом. Создаются два потомка, эквивалентные родителям, затем для каждого состояния равновероятно либо производится обмен между потомками выходного сигнала и действиями по входным сигналам для этого состояния, либо не производится. Таким образом, состояния родителей перемешиваются в следующем поколении.

2.4. Отбор особей для формирования следующего поколения

Генерация очередного поколения осуществляется с помощью метода «рулетки»: всем автоматам ставятся в соответствие отрезки длины, равной значению фитнес-функции, затем выбирается случайная точка, которая будет принадлежать одному из отрезков, соответствующий автомат добавляется в следующее поколение. Выбор осуществляется столько раз, сколько особей должно быть в поколении. Затем производится скрещивание пар автоматов, и, наконец, мутация каждого из получившихся детей.

2.5. Вычисление функции приспособленности

Функция приспособленности (фитнес-функция) вычисляется следующим образом: проводится эксперимент с участием муравья с поведением, заданным текущим автоматом Мура. С помощью лаборатории эмулируются его перемещения в зависимости от расположения еды: он проходит по полю несколько раз, каждый раз подсчитывается

число съеденных яблок. Полученные результаты усредняются (сумма делится на количество попыток) и выдаются в качестве ответа.

2.6. Настраиваемые параметры алгоритма

Следующие параметры алгоритма генетического программирования могут быть изменены:

- размер поколения;
- количество состояний;
- количество экспериментов над каждым муравьем;
- вероятность мутации;
- время до «большой» мутации поколения;
- вероятность появления еды в каждой клетке.

Как показывают вычислительные эксперименты, изменение некоторых из этих параметров может существенно влиять на время поиска автомата, позволяющего муравью съесть всю еду. Отметим, что от распределения еды зависит число состояний автомата, который «съедает» наибольшее количество еды. Чем меньше еды – тем больше состояний потребуется, если все поле заполнено едой, то хватит двух состояний (с помощью одного муравей идет вперед, другое позволяет повернуть с прямой траектории).

3. Результаты

В качестве фиксированных параметров автомата и генетического алгоритма были выбраны следующие значения:

- число различных случайных игровых полей – 100;
- вероятность появления еды – 0.05;
- число состояний автомата – 5.

На рис.2 представлен график зависимости максимального значения фитнес-функции от номера поколения.

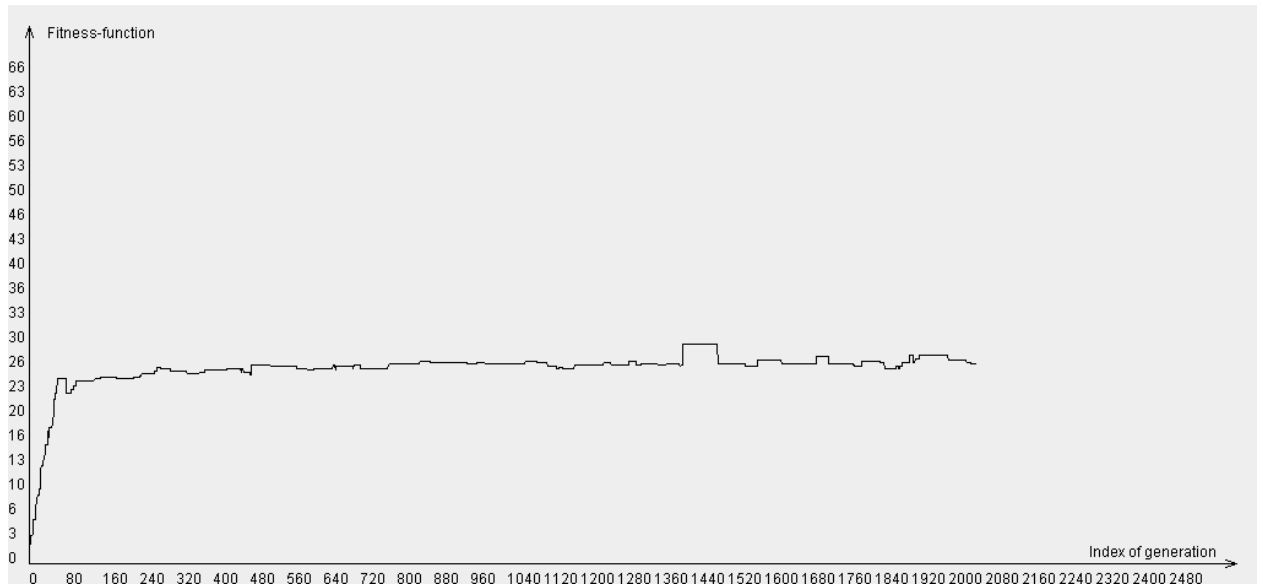


Рис. 2. График максимального значения фитнес-функции

На рис. 3 представлен график зависимости усредненного значения фитнес-функции от номера поколения.

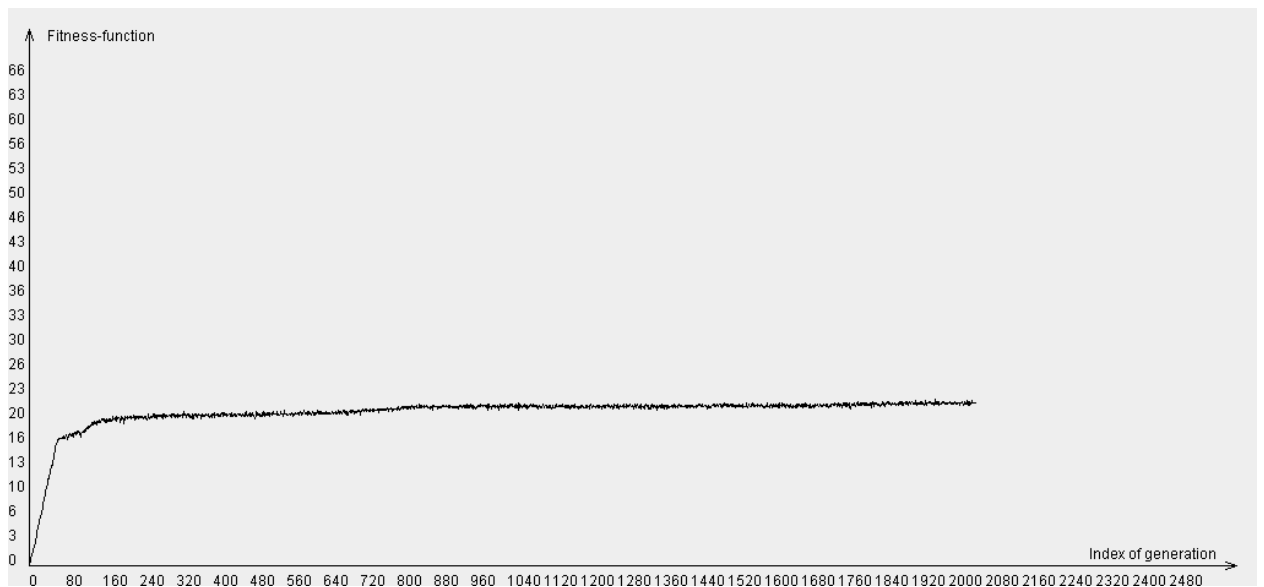


Рис. 3. График среднего значения фитнес-функции

Ниже приведен граф переходов для полученного автомата Мура с пятью состояниями, сгенерированный с помощью виртуальной лаборатории. Для каждого состояния показано выходное воздействие, а затем таблица переходов по всем 256 входным воздействиям.

5 states (initial=2), fitness=20.5

Action for this state=M, transitions=

1, 2, 2, 2, 2, 3, 5, 3, 4, 2, 3, 1, 1, 4, 2, 4, 2, 2, 1, 5, 1, 2, 5,
 2, 2, 3, 3, 3, 5, 1, 1, 5, 5, 5, 1, 1, 3, 2, 2, 5, 5, 1, 2, 4, 3, 1,
 4, 3, 5, 4, 1, 1, 3, 3, 1, 1, 3, 1, 2, 5, 1, 4, 4, 3,
 4, 4, 3, 1, 2, 2, 1, 3, 2, 4, 5, 5, 2, 4, 1, 5, 4, 5, 3, 4, 4, 4, 3,
 5, 4, 4, 2, 5, 5, 3, 2, 2, 5, 1, 5, 1, 4, 1, 4, 2, 1, 5, 1, 1, 3, 1,
 4, 2, 5, 2, 2, 2, 4, 1, 3, 1, 4, 1, 1, 3, 2, 5, 5, 1,
 5, 5, 5, 4, 3, 4, 1, 4, 1, 3, 4, 1, 1, 3, 2, 1, 1, 1, 5, 3, 1, 3, 1,
 4, 4, 5, 4, 2, 2, 5, 3, 4, 1, 1, 2, 4, 1, 5, 3, 3, 5, 5, 2, 2, 5, 2,
 4, 4, 1, 4, 2, 4, 5, 4, 4, 3, 4, 2, 1, 2, 2, 2, 2, 4,
 4, 1, 1, 2, 4, 4, 5, 4, 1, 3, 4, 4, 4, 5, 2, 4, 1, 2, 4, 3, 4, 5, 4,
 2, 5, 1, 3, 3, 2, 4, 4, 4, 4, 1, 1, 3, 4, 4, 3, 3, 5, 4, 4, 2, 4, 3,
 5, 3, 2, 5, 5, 3, 3, 4, 4, 5, 3, 5, 2, 2, 4, 4, 4, 3

Action for this state=M, transitions=

1, 4, 3, 2, 1, 5, 1, 5, 2, 3, 2, 2, 4, 5, 2, 4, 5, 3, 4, 3, 5, 1, 4,
 2, 4, 1, 3, 2, 4, 5, 4, 4, 1, 4, 1, 2, 1, 4, 3, 2, 2, 5, 4, 3, 3, 5,
 3, 2, 4, 4, 1, 3, 5, 3, 3, 4, 5, 5, 1, 2, 1, 3, 1, 2,
 1, 3, 5, 1, 1, 1, 4, 1, 1, 5, 5, 4, 3, 2, 4, 5, 2, 3, 2, 5, 2, 4, 3,
 4, 3, 4, 4, 3, 5, 4, 5, 5, 1, 2, 2, 4, 5, 1, 4, 3, 1, 5, 2, 1, 5, 5,
 1, 2, 2, 1, 2, 4, 4, 1, 2, 4, 2, 3, 5, 5, 2, 5, 1, 4,
 5, 5, 2, 2, 3, 1, 3, 2, 2, 2, 4, 5, 5, 3, 1, 3, 5, 1, 4, 1, 2, 3, 1,
 5, 4, 1, 4, 5, 1, 3, 2, 5, 1, 2, 5, 1, 4, 1, 5, 4, 5, 2, 5, 3, 5, 1,
 2, 3, 1, 2, 2, 2, 5, 5, 2, 2, 1, 3, 5, 1, 5, 2, 2, 1,
 5, 3, 4, 1, 1, 1, 4, 4, 3, 2, 3, 1, 5, 1, 2, 1, 2, 4, 2, 1, 1, 5, 3,
 5, 3, 2, 5, 4, 1, 5, 2, 3, 4, 4, 3, 1, 4, 4, 2, 1, 3, 5, 5, 5, 1, 2,
 5, 2, 4, 2, 1, 3, 2, 1, 5, 3, 2, 3, 4, 4, 3, 4, 4, 2

Action for this state=M, transitions=

3, 1, 3, 1, 2, 4, 2, 4, 1, 2, 3, 2, 1, 4, 5, 1, 1, 4, 2, 1, 3, 4, 3,
 3, 3, 4, 5, 3, 4, 1, 2, 4, 4, 5, 5, 1, 4, 3, 1, 3, 4, 1, 1, 1, 4, 5,
 2, 3, 3, 5, 2, 4, 3, 2, 1, 2, 3, 1, 5, 1, 2, 1, 5, 1,
 2, 3, 3, 3, 5, 3, 3, 1, 5, 3, 2, 4, 1, 1, 4, 3, 2, 1, 5, 2, 3, 2, 4,
 3, 2, 3, 3, 3, 1, 5, 2, 5, 3, 4, 2, 5, 1, 5, 2, 4, 1, 2, 4, 4, 1, 5,
 1, 3, 2, 3, 4, 5, 1, 1, 4, 4, 3, 3, 3, 5, 1, 5, 3, 4,
 3, 3, 5, 2, 2, 2, 2, 3, 2, 3, 5, 2, 2, 4, 3, 5, 4, 2, 5, 1, 1, 5, 2,
 5, 2, 4, 3, 3, 4, 3, 4, 3, 1, 2, 3, 3, 1, 1, 3, 5, 4, 5, 5, 4, 5, 5,
 3, 5, 3, 3, 1, 2, 2, 1, 5, 5, 4, 3, 5, 4, 4, 4, 3, 2,
 5, 1, 5, 3, 5, 3, 1, 2, 1, 4, 4, 4, 4, 3, 4, 2, 4, 1, 3, 1, 2, 3, 5,
 2, 5, 4, 4, 4, 3, 2, 2, 3, 4, 1, 1, 3, 2, 2, 3, 5, 1, 4, 5, 4, 5, 2,
 1, 2, 3, 5, 1, 1, 5, 2, 2, 4, 2, 2, 1, 1, 4, 4, 4, 4

Action for this state=R, transitions=

1, 5, 3, 5, 1, 3, 2, 1, 2, 2, 1, 4, 2, 1, 3, 5, 3, 3, 5, 5, 2, 5, 2,
 1, 1, 2, 2, 3, 3, 4, 5, 5, 1, 1, 4, 2, 2, 5, 1, 2, 1, 3, 5, 2, 3, 5,
 2, 4, 3, 2, 3, 4, 2, 4, 4, 2, 4, 5, 1, 3, 5, 2, 2, 1,
 2, 2, 3, 1, 4, 1, 5, 1, 1, 4, 5, 5, 5, 2, 3, 1, 3, 2, 1, 5, 4, 1, 5,
 4, 5, 4, 5, 2, 5, 4, 3, 5, 3, 2, 4, 4, 3, 4, 3, 1, 5, 2, 1, 5, 5, 5,
 5, 3, 1, 5, 4, 5, 3, 1, 5, 3, 2, 1, 3, 2, 3, 3, 3, 3,
 3, 2, 2, 1, 4, 2, 4, 5, 4, 1, 1, 3, 4, 1, 1, 1, 4, 3, 5, 5, 2, 2, 4,
 5, 5, 3, 1, 3, 3, 3, 2, 5, 3, 3, 2, 4, 5, 3, 4, 3, 1, 2, 5, 2, 5, 3,
 3, 1, 1, 3, 2, 2, 4, 1, 1, 1, 1, 2, 3, 4, 4, 4, 2, 3,

1, 1, 3, 1, 4, 3, 4, 3, 3, 1, 1, 3, 2, 5, 3, 3, 4, 4, 2, 1, 3, 2, 2,
3, 3, 1, 2, 5, 2, 4, 4, 1, 1, 3, 1, 1, 2, 3, 2, 5, 5, 2, 2, 1, 2, 2,
5, 4, 4, 2, 5, 4, 4, 5, 5, 4, 5, 5, 1, 5, 3, 5, 1, 1

Action for this state=L, transitions=

1, 4, 4, 1, 1, 4, 2, 4, 1, 5, 3, 4, 3, 1, 4, 4, 3, 5, 4, 5, 3, 3, 2,
3, 3, 1, 2, 1, 4, 1, 5, 3, 4, 1, 3, 3, 2, 4, 2, 2, 5, 3, 5, 5, 4, 5,
3, 3, 3, 4, 4, 4, 5, 1, 5, 1, 1, 4, 4, 4, 4, 5, 3, 1,
3, 5, 3, 1, 4, 2, 2, 3, 4, 5, 4, 4, 3, 4, 4, 5, 2, 5, 2, 5, 3,
2, 2, 5, 1, 4, 4, 2, 3, 4, 3, 1, 1, 1, 3, 4, 5, 3, 1, 2, 5, 3, 1, 3,
1, 1, 2, 3, 1, 4, 1, 3, 2, 1, 5, 4, 4, 5, 4, 3, 5, 1,
3, 5, 5, 2, 5, 3, 1, 1, 5, 1, 3, 4, 5, 1, 2, 4, 2, 4, 1, 3, 2, 1, 5,
4, 5, 5, 3, 3, 5, 1, 4, 5, 3, 3, 5, 3, 1, 4, 1, 5, 5, 3, 1, 2, 2, 5,
3, 5, 2, 4, 5, 4, 1, 5, 3, 5, 4, 4, 2, 1, 1, 2, 3, 1,
1, 3, 5, 5, 4, 4, 1, 2, 3, 5, 5, 4, 4, 3, 5, 1, 4, 4, 3, 3, 1, 5, 5,
2, 2, 3, 1, 4, 5, 2, 2, 3, 2, 3, 2, 2, 2, 5, 5, 1, 1, 2, 5, 3, 5, 4,
1, 5, 3, 4, 3, 1, 2, 2, 4, 3, 3, 1, 1, 4, 1, 5, 2, 3

Заключение

В результате выполнения данной работы были получены следующие значения:

- максимальное значение фитнес-функции – 30;
- среднее значение фитнес-функции – 21.8.

Поскольку вероятность появления еды на клетке равна 0.05, то ожидаемое общее число клеток, занятых едой, составляет 51. Графики показывают, что был получен автомат съедающий половину еды на поле, что является хорошим результатом.

Использование метода рулетки плохо подходит для выделения наилучших особей при скрещивании, поскольку значения фитнес-функции для особей отличаются на небольшую величину. Поэтому даже у самой приспособленной особи есть шанс не попасть в следующее поколение. Это хорошо заметно на графике максимального значения фитнес-функции.

К достоинствам традиционного генетического алгоритма можно отнести высокую скорость генерации очередного поколения (по сравнению, например, с островным алгоритмом [4]). В результате получается автомат, достаточно быстро и эффективно решающий задачу «Умный муравей-3».

Источники

1. Инструкция по созданию plugin'ов к виртуальной лаборатории
http://svn2.assembla.com/svn/not_instrumental_tool/docs/pdf/interface_manual.pdf
2. Бедный Ю. Д., Шалыто А. А. Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей» и ее модификациях
http://is.ifmo.ru/works/_ant.pdf
3. Яминов Б. Генетические алгоритмы
<http://rain.ifmo.ru/cat/view.php/theory/unordered/genetic-2005>
4. Царев Ф. Н., Шалыто А. А. Применение генетических алгоритмов для построения автоматов с минимальным числом состояний для задачи об «Умном муравье» Тезисы научно-технической конференции «Научно-программное обеспечение в образовании и научных исследованиях». СПбГУ ПУ. 2008, с. 209–215.
http://is.ifmo.ru/download/2008-02-25_tsarev_shalyto.pdf