

Санкт-Петербургский государственный университет
информационных технологий, механики и оптики

Кафедра «Компьютерные технологии»

Р. З. Хамитов

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

Вариант № 7

Санкт-Петербург
2009

Оглавление

Введение	3
1. Постановка задачи	4
1.1. Автомат Мили	4
1.2. Задача об «Умном муравье»	4
2. Реализация	5
2.1. Описание используемого представления автоматов	5
2.2. Описание метода скрещивания	5
2.3. Описание метода мутации	6
2.4. Описание метода генерации очередного поколения	6
2.5. Описание способа вычисления функции приспособленности	6
3. Результаты работы	6
3.1. Граф переходов	7
3.2. График максимального значения функции приспособленности	7
3.3. График среднего значения функции приспособленности	8
Заключение	9
Источники	9

Введение

В лабораторной работе изучается применение генетических алгоритмов для генерации конечных автоматов. В качестве примера взята задача об «Умном муравье». Результатом работы является автомат Мили, построенный с помощью генетического алгоритма и представляющий логику муравья.

При выполнении лабораторной работы использовался программный модуль «Виртуальная лаборатория» [1], написанный студентами кафедры «Компьютерные технологии» СПбГУ ИТМО. Модуль позволяет реализовывать генетические алгоритмы и особей для них в виде подключаемых плагинов.

1. Постановка задачи

Цель данной лабораторной работы – построить автомат Мили, близкий к оптимальному и решающий задачу об «Умном муравье» [2]. Оптимальность заключается в том, что автомат должен иметь минимальное число состояний, и муравей, управляемый данным автоматом, должен съесть как можно больше еды, выполнив при этом как можно меньше шагов.

1.1. Автомат Мили

Автомат Мили – это конечный автомат, генерирующий выходные воздействия в зависимости от текущего состояния и входного воздействия. Пример диаграммы переходов автомата приведен на рис. 1. Как видно из приведенной диаграммы, над каждой дугой расположена пара значений – входное и выходное воздействия. При этом выходное воздействие зависит не только от состояния, в котором находится автомат, но и от входного воздействия.

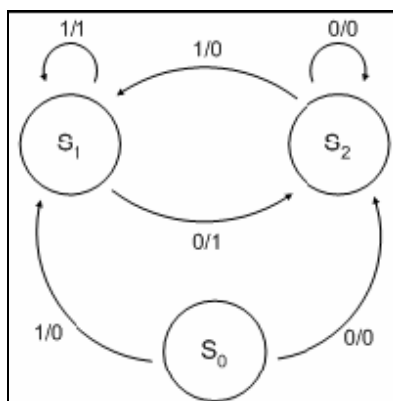


Рис. 1

1.2. Задача об «Умном муравье»

В задаче об «Умном муравье» рассматривается поле (рис. 2), состоящее из клеток. Поле имеет размеры 32x32 клеток 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, помеченной как «Start».

За один ход муравей может определить, есть ли в клетке перед ним яблоко, и выполнить одно из следующих действий:

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

Максимальное число ходов – 200. Цель работы – создать муравья с фиксированным числом состояний, который съест как можно больше яблок.

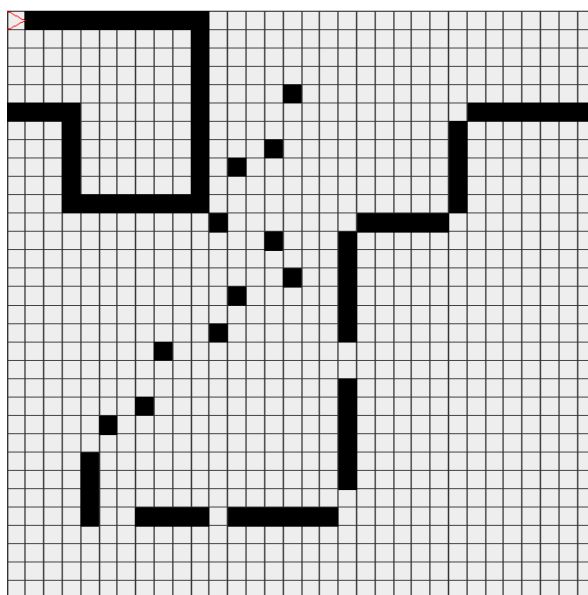


Рис. 2

2. Реализация

Виртуальная лаборатория состоит из ядра и подключаемых плагинов. Для решения поставленной задачи требуется создать два плагина. Первый из них реализует генетический алгоритм, использующий традиционный генетический алгоритм (Классический генетический алгоритм [3]). Второй плагин должен реализовывать «особь» – конечный автомат Мили, решающий задачу об «Умном муравье». Для «особи» необходимо реализовать операции мутации и скрещивания.

2.1. Используемое представление автоматов

Особями в данном алгоритме являются автоматы Мили. Автомат Мили представляется в виде графа переходов. Непосредственно в коде программы используется двумерный массив, который хранит переход (новое состояние и выходное действие) для каждой пары, состоящей из текущего состояния автомата и входного воздействия. Значениями входной переменной являются нуль и единица – присутствие или отсутствие еды в клетке перед муравьём.

2.2. Метод скрещивания

Оператор скрещивания получает на вход две особи и выдает также две особи. Число состояний в автоматах фиксировано и одинаково для всех особей. Процесс скрещивания происходит следующим образом.

1. Создаем два новых автомата с тем же числом состояний в каждом.
2. Случайным образом либо первый сын получает номер начального состояния от первого родителя, второй – от второго, либо наоборот.

3. Рассмотрим состояние с номером i в каждом родителе. Для каждого состояния есть два элемента Transition (индексы $[i][0]$ и $[i][1]$). Таким образом, от двух родителей получаем четыре элемента Transition: $p10$ – нулевой элемент первого родителя, $p11$ – первый элемент первого родителя, $p20$ – нулевой элемент второго родителя, $p21$ – первый элемент второго родителя. Аналогично для того же i для детей имеем: $s10$, $s11$, $s20$, $s21$. Теперь выбираем случайно один набор из четырех: $(p10, p01, p00, p11)$ $(p00, p11, p10, p01)$ $(p10, p11, p00, p01)$ $(p00, p01, p10, p11)$, и присваиваем $s00$, $s01$, $s10$, $s11$ этим набором.

4. Повторяем шаг 3 для всех состояний.

2.3. Метод мутации

Мутация особи-автомата заключается в случайном выборе состояния и случайном изменении одного из переходов из данного состояния. Также с вероятностью 0.5 изменяется начальное состояние автомата на случайно выбранное.

2.4. Метод генерации очередного поколения

Генерация очередного поколения осуществляется с помощью метода «рулетки»: всем автоматам ставятся в соответствие отрезки длины, равной значению *Fitness*-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

В качестве функции приспособленности (*Fitness*-функции) будем использовать выражение вида:

$$\text{Fitness} = \text{Apples} - \text{Steps} / 200,$$

где *Apples* – число съеденных муравьем яблок за 200 шагов, *Steps* – ход, на котором муравей последний раз съел яблоко. Эмуляция действий муравья выполняется до тех пор пока он либо не съест все яблоки на поле, либо не превысит максимально допустимое число шагов.

3. Результаты работы

В результате работы генетического алгоритма был построен автомат Мили, состоящий из пяти состояний. Значение функции приспособленности для данного автомата равно 82.02. Из приведенного выше соотношения следует, что муравей съедает 83 яблока за 196 ходов.

3.1. Граф переходов

На рис. 3 представлен граф переходов автомата, наилучшего получившегося результата. Граф переходов получен с помощью визуализатора «Виртуальной лаборатории для генерации автоматов с помощью генетических алгоритмов» [1]. Вершины графа обозначают состояния автомата. Ребра графа – переходы между состояниями. На ребрах написано входное и выходное воздействия при переходе.

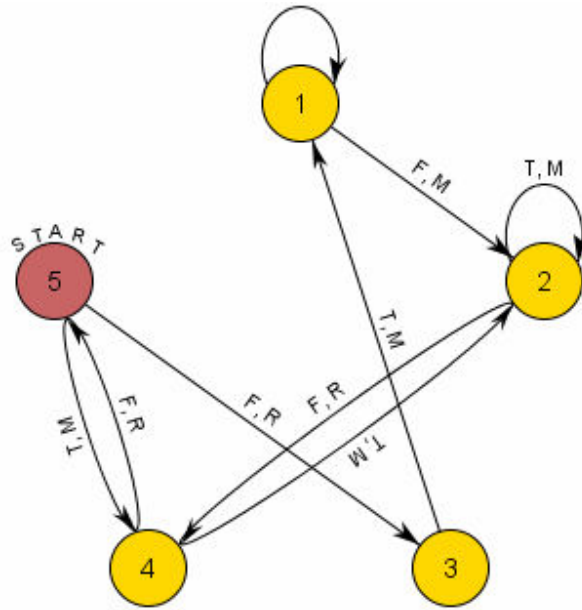


Рис. 3

3.2. График максимального значения функции приспособленности

На рис. 4 приведен график зависимости максимального значения функции приспособленности среди особей поколения. Из графика видно, что выбранный генетический алгоритм быстро сходится, так как уже после 20 поколений максимальное значение функции приспособленности перестает улучшаться.

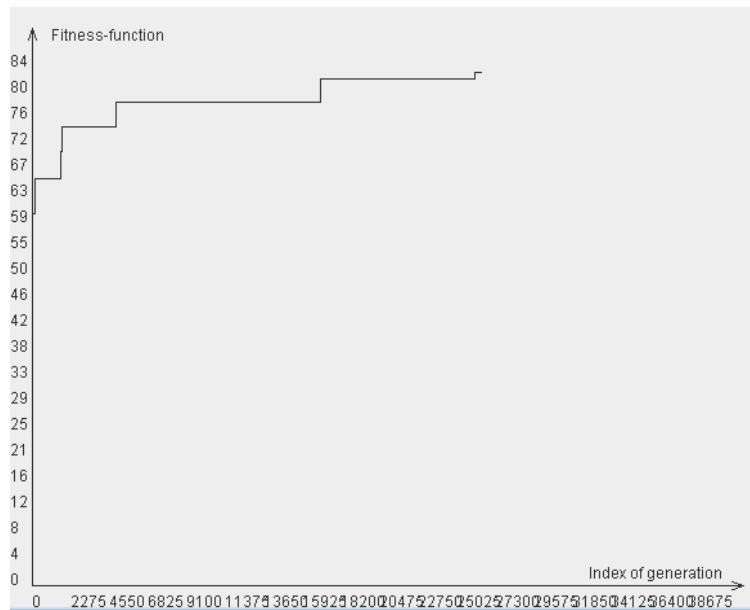


Рис. 4

3.3. График среднего значения функции приспособленности

На рис. 5 приведен график зависимости среднего значения функции приспособленности среди особей поколения. Из графика следует, что средняя приспособленность особей в каждом поколении стремится к максимальному значению функции приспособленности.

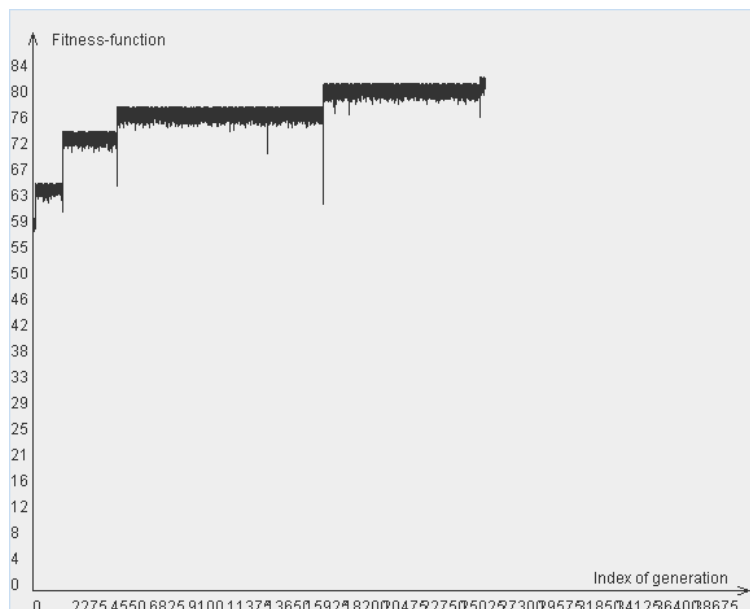


Рис. 5

Заключение

Результаты лабораторной работы показали, что используемый метод имеет низкую эффективность для решения рассматриваемой задачи. Хотя реализованный алгоритм генерирует автомат лучше, чем при полном переборе. Заметим, что известен автомат с пятью состояниями, построенный для этой же задачи вручную, который управляет муравьем, съедающим 81 яблоко [2]. Низкая эффективность вытекает из числа итераций алгоритма, в котором ответ был найден примерно лишь на 250000-ой итерации.

Источники

1. *Давыдов А. А., Соколов Д. О., Царев Ф. Н., Шалыто А. А.* Описание виртуальной лаборатории. http://is.ifmo.ru/works/_2_93_davidov_sokolov.pdf
2. *Бедный Ю. Д., Шалыто А. А.* Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей». http://is.ifmo.ru/works/_ant.pdf
3. *Яминов Б.* Генетические алгоритмы. <http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005>
4. *Инструкция по созданию plugin'ов к виртуальной лаборатории.* http://svn2.assembla.com/svn/not_instrumental_tool/docs/pdf/interface_manual.pdf