

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

А.С. Тяhti

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

Вариант №11

Санкт-Петербург
2009

Оглавление

Введение.....	3
<u>1.</u> Постановка задачи.....	Error! Bookmark not defined.
<u>1.1.</u> Автомат Мили	Error! Bookmark not defined.
<u>1.2.</u> Задача об «Умном муравье».....	5
<u>2.</u> Реализация.....	<u>6</u>
<u>2.1.</u> Представление особи.....	<u>6</u>
<u>2.2.</u> Описание метода скрещивания	<u>6</u>
<u>2.3.</u> Описание метода мутации	<u>6</u>
<u>2.4.</u> Генерация нового поколения	<u>7</u>
<u>2.5.</u> Вычисление функции приспособленности.....	<u>7</u>
<u>3.</u> Результаты работы построенного модуля	<u>8</u>
<u>3.1.</u> Граф переходов	<u>8</u>
<u>3.2.</u> График максимального значения функции приспособленности	<u>9</u>
<u>3.3.</u> График среднего значения функции приспособленности	<u>9</u>
Заключение	11 <u>0</u>
Источники	11 <u>1</u>

Введение

Цель лабораторной работы – применение генетических алгоритмов для построения конечных автоматов. В качестве примера рассматривается построение конечного автомата Мили, решающего задачу об «Умном муравье».

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

1. Постановка задачи

Задача данной лабораторной работы – построить автомат Мили, решающий задачу об «Умном муравье». При построении необходимо стремиться к тому, чтобы полученный автомат был близок к оптимальному. Критерий оценки оптимальности решения заключается в следующем: муравей, управляемый автоматом, который состоит из конечного числа состояний, съедает всю еду на поле за минимальное число шагов.

1.1. Автомат Мили

Автомат Мили – конечный автомат, который генерирует выходные действия в зависимости от текущего состояния и входного воздействия. Пример представления автомата Мили в виде диаграммы переходов приведен на рисунке (рис. 1).

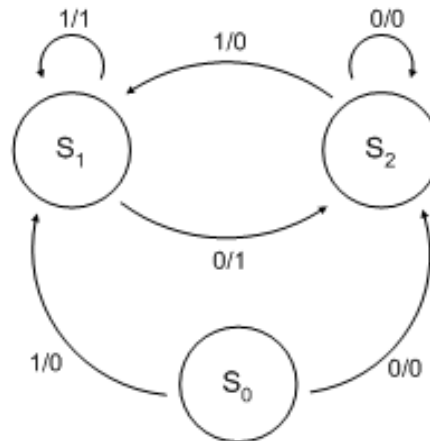


Рис. 1

Над каждой дугой расположена пара значений – входное и выходное воздействия. При этом выходное воздействие зависит от состояния автомата и от входного воздействия.

1.2. Задача об «Умном муравье»

Действие происходит на поверхности тора размером 32x32 клетки. В некоторых клетках находится еда. Муравей начинает движение из клетки, помеченной «Start».

За ход муравей может выполнить следующие действия:

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

Игра длится 200 ходов. Цель игры – создать муравья «с минимальным числом состояний», который за минимальное число ходов съест как можно больше яблок.

2. Реализация

Виртуальная лаборатория представляет собой ядро и набор подключаемых модулей. В рамках решения поставленной задачи необходимо создать два модуля. Первый – модуль генетического алгоритма, использующий клеточный (cellular) генетический алгоритм и метод рулетки для генерации очередного поколения. Второй модуль должен реализовывать «особь» – конечный автомат Мили, который обеспечивает решение задачи об «Умном муравье». Для «особи» необходимо реализовать операции мутации и скрещивания [3].

2.1. Представление особи

В данном алгоритме особи представляют собой автоматы Мили. Автомат будем задавать с помощью графа переходов. В программе для этих целей используется матрица, хранящая для каждой пары состояния автомата и входного воздействия переход (новое состояние и выходное воздействие). Значение входной переменной в задаче об «Умном муравье» показывает (0 или 1) есть ли в клетке перед муравьем еда или нет.

2.2. Описание метода скрещивания

Имеются два автомата Мили ($p1$ и $p2$), и необходимо получить следующие два. Для этого создадим два новых автомата с таким же числом состояний, как у исходных. Начальные состояния детей задаются следующим образом: либо первый ребенок получает начальное состояние первого родителя, а второй – второго, либо наоборот, первый – второго родителя, а второй ребенок – начальное состояние первого родителя.

Далее определяем переходы для детей. У каждого родителя есть два перехода из i -ого состояния – $p1[i][0]$, $p1[i][1]$ для первого родителя и $p2[i][0]$, $p2[i][1]$ – для второго. Распределим случайным образом эти четыре родительских перехода среди детей (четыре возможных случая) и повторим данную операцию для всех состояний. В результате получаем пару новых особей.

2.3. Описание метода мутации

Метод мутации особи заключается в случайном выборе состояния и случайном изменении одного из переходов из данного состояния. Также с вероятностью 50% изменяется на случайное начальное состояние автомата.

2.4. Генерация нового поколения

Генерация нового поколения осуществляется с использованием клеточного генетического алгоритма и метода рулетки.

Суть алгоритма заключается в следующем – пусть имеется замкнутая с обоих концов сетка (тор) в каждой клетке которой помещается особь. Каждая особь может взаимодействовать только с одним из четырех своих соседей (верхняя, нижняя, правая и левая клетки относительно данной).

В ходе процесса генерации очередного поколения для каждой особи выбирается наиболее приспособленный (с наибольшим значением fitness-функции) из ее соседей. Далее осуществляется скрещивание этих двух особей, в результате которого либо один из детей помещается в клетку родителя, либо родитель остается на месте.

2.5. Вычисление функции приспособленности

Значение функции приспособленности (*Fitness-функции*) рассчитывается по следующей формуле:

$$\mathbf{Fitness} = \mathbf{Apples} - (\mathbf{Steps} / 200), \text{ где}$$

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 построенного модуля

В результате работы генетического алгоритма был построен автомат Мили, состоящий из пяти состояний. Значение функции приспособленности для данного автомата равно 82,04 – муравей, управляемый данным автоматом, съедает 83 яблока за 192 хода.

3.1. Граф переходов

На рис. 2 представлен граф переходов автомата из пяти состояний, задающий поведение муравья, который съедает 83 яблока за 192 ходов. Граф переходов получен с помощью визуализатора «Виртуальной лаборатории» [1].

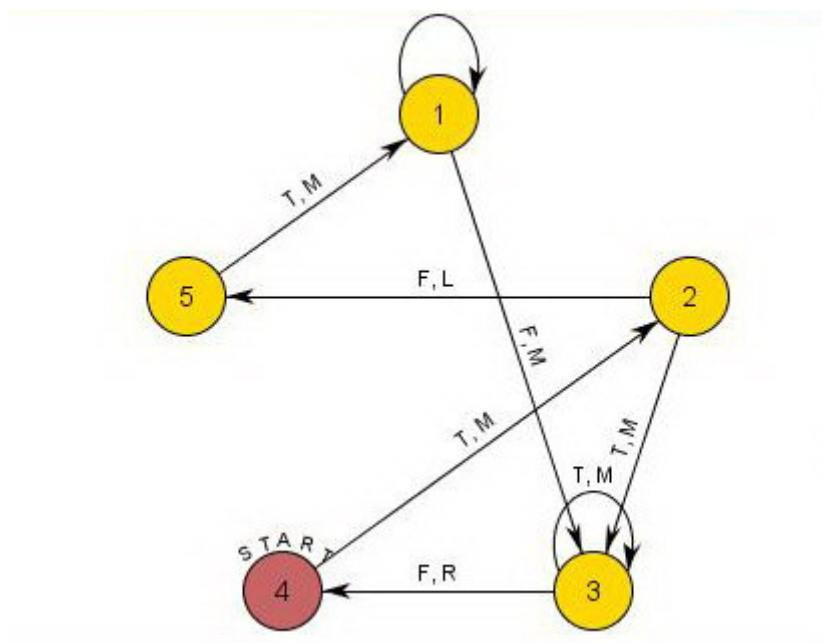


Рис. 2

Вершины графа обозначают состояния автомата. Ребра графа – переходы между состояниями. На ребрах указаны входное и выходное воздействия при переходе.

Используются следующие обозначения:

1. F – входное воздействие – впереди пусто;
2. T – входное воздействие – впереди есть еда;
1. L – выходное воздействие – поворот налево;
2. R – выходное воздействие – поворот направо;
3. M – выходное воздействие – шаг вперед;

3.2. График максимального значения функции приспособленности

График на рис. 3 показывает максимум значения функции приспособленности среди особей данного поколения в зависимости от номера поколения.

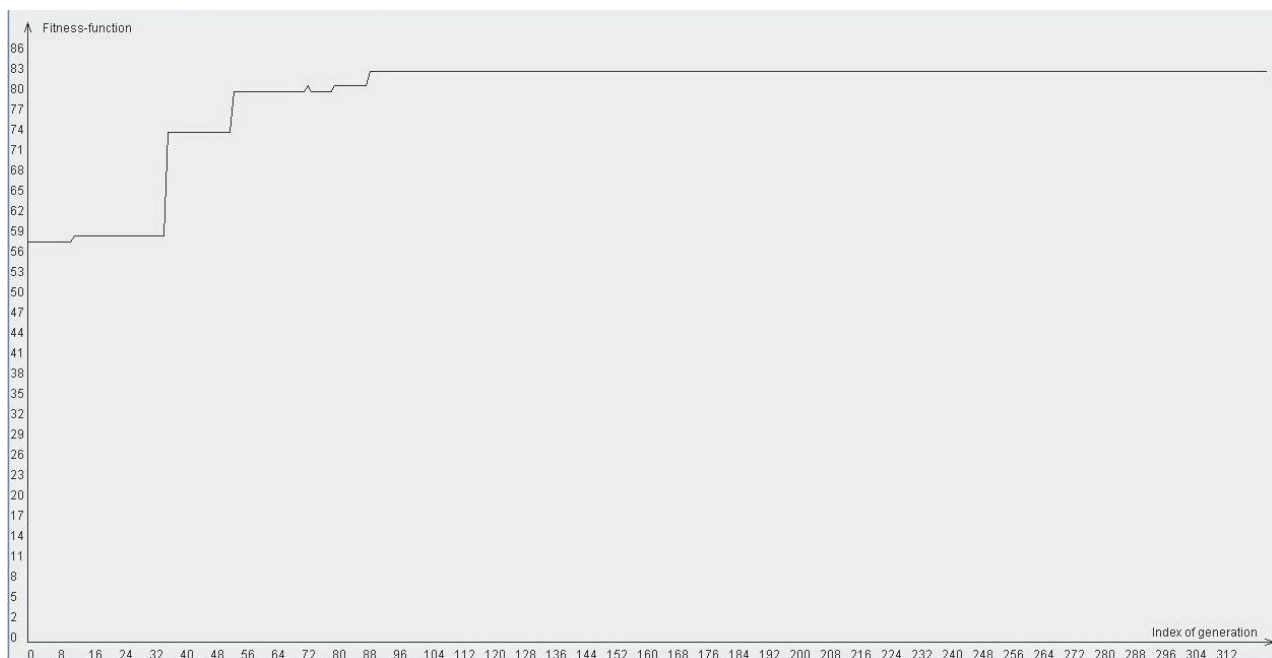


Рис. 3

3.3. График среднего значения функции приспособленности

График (рис. 4) показывает среднее значения функции приспособленности среди особей данного поколения в зависимости от номера поколения.

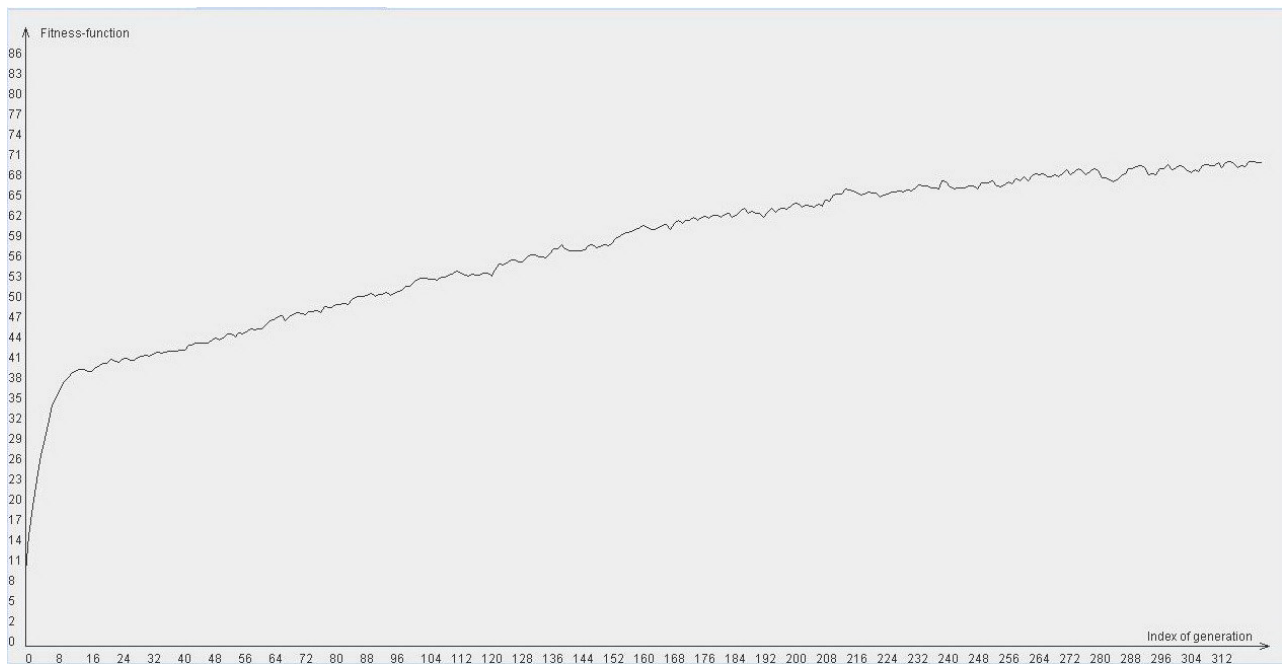


Рис. 4

Заключение

Результаты лабораторной работы показали, что используемые методы достаточно эффективны для построения автомата Мили с пятью состояниями для муравья, который съедает 83 яблока, так как полный перебор позволяет построить автомат с таким же результатом. При этом заметим, что известен автомат, построенный для этой задачи вручную, который управляет муравьём, съедающим 81 яблоко [2].

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».
http://is.ifmo.ru/works/_ant.pdf
3. *Яминов Б.* Генетические алгоритмы.
<http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005>