

Санкт-Петербургский государственный университет
информационных технологий, механики и оптики

Кафедра компьютерных технологий

Р. Р. Валитов

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

Вариант № 3

Санкт-Петербург
2009

Оглавление

Введение	3
1. Постановка задачи	4
2. Реализация.....	5
2.1. Представление автоматов	5
2.2. Метод скрещивания.....	5
2.3. Метод мутации.....	5
2.4. Метод генерации очередного поколения	5
3. Результаты работы.....	6
3.1. График зависимости функции приспособленности от времени	6
3.2. Полученный автомат	6
Заключение.....	7
Источники	8

Введение

Цель лабораторной работы – построить с помощью генетических алгоритмов конечный автомат Мили, решающий задачу об «Умном муравье».

При выполнении работы использовалась среда «Программный комплекс для изучения методов глобальной оптимизации *GlOpt*» [1], разработанная студентами кафедры «Компьютерные технологии» СПбГУ ИТМО. Среда позволяет реализовывать генетические алгоритмы в виде исполняемых плагинов и изучать процесс их выполнения.

1. Постановка задачи

Задачей лабораторной работы является построение с помощью генетических алгоритмов конечного автомата Мили, решающего задачу об «Умном муравье». При этом автомат необходимо представить в виде битовых строк, также должен быть использован островной генетический алгоритм и метод рулетки для генерации очередного поколения.

Игра происходит на поверхности тора размером 32x32 клетки. В некоторых клетках находится 89 яблок. За ход муравей может выполнить следующие действия:

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

Игра длится 200 ходов. Цель игры – найти муравья, который за минимальное число ходов съедает как можно больше яблок. Пример игрового поля представлен на рис. 1.

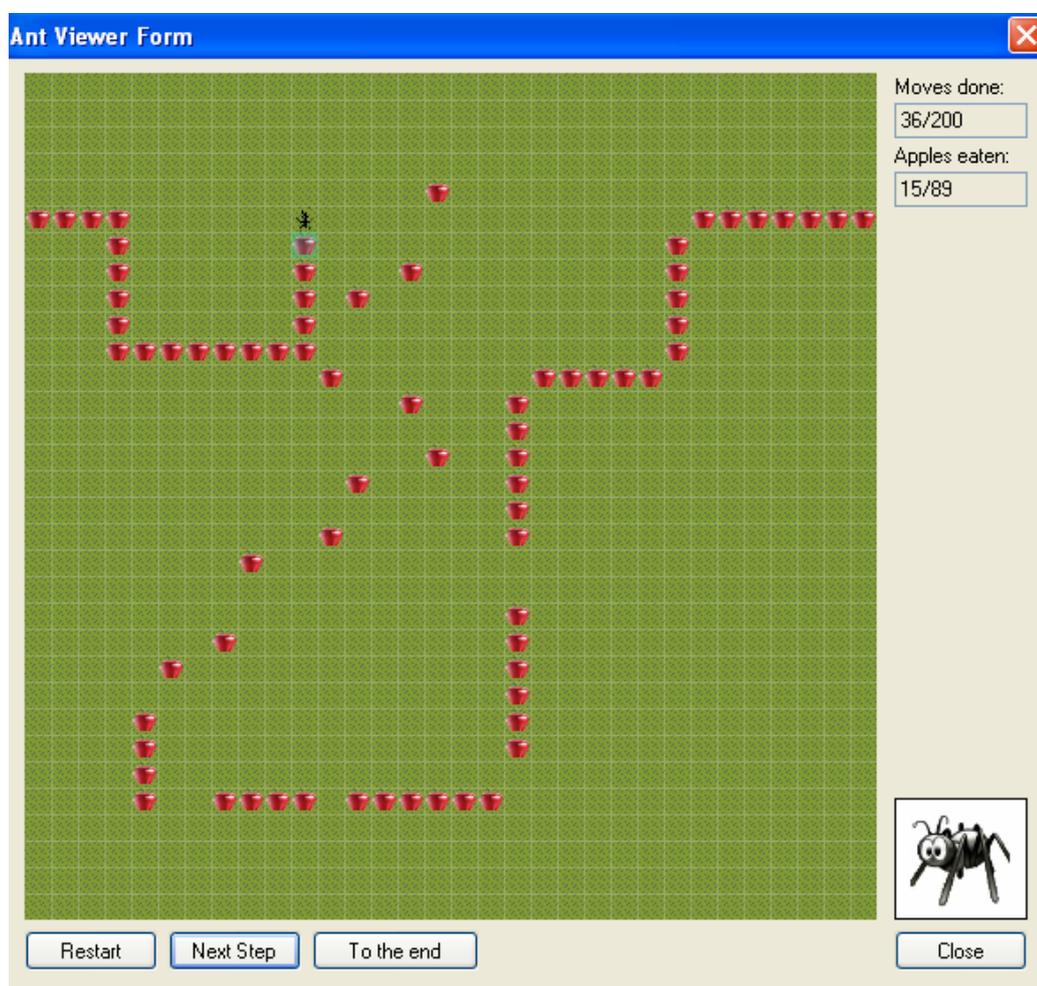


Рис. 1. Игровое поле

2. Реализация

Для использования собственного алгоритма в программном комплексе *GlOpt* необходимо создать плагин, реализовав два класса, наследованные соответственно от интерфейсов `SearchOperator` и `GeneticOptimizationAlgorithm`.

2.1. Способ представления автоматов

Автомат, управляющий муравьем, представляется массивом бит. По числу состояний автомата `StateCount` определяется необходимое число бит для хранения номера состояния в двоичном представлении. Битовый массив представляется следующим образом: первая половина массива содержит `StateCount` записей вида – номер состояния перехода и номер действия в случае отсутствия впереди муравья яблока. Вторая половина массива содержит `StateCount` записей вида – номер состояния перехода и номер действия в случае наличия впереди яблока.

2.2. Метод скрещивания

Вероятность скрещивания определяется параметром `CrossoverProbability`. При скрещивании два предка порождают два потомка. Битовая строка каждого предка делится на две части (длина этих частей выбирается случайным образом). Одна из частей битовой строки предка заменяется на часть строки другого предка такой же длины. Полученная строка и является битовой строкой потомка.

2.3. Метод мутации

Каждый бит в автомате инвертируется с вероятностью `MutationProbability`.

2.4. Метод генерации очередного поколения

Вначале генерируется популяция случайных особей. Затем особи размещаются на острове.

На каждом шаге алгоритма над особями с одного острова производится операция скрещивания по принципу метода рулетки. Этот принцип заключается в следующем: особи размещаются на рулетке, причем каждой особи отводится сектор, размер которого пропорционален величине функции приспособленности особи. Запуская рулетку $2n$ раз (где n – размер популяции на острове), особи разбиваются на n пар. Затем применяется операция скрещивания.

На следующем этапе ко всем особям применяется операция мутации.

На последнем этапе особь с максимальным значением функции приспособленности с каждого острова помещается на оставшиеся острова, заместив там случайную особь.

Если значение функции приспособленности не изменилось в течение 500 поколений, выполняется большая мутация: на каждый остров поселяется особь с максимальным значением функции, а остальные особи заменяются новыми, построенными случайным образом.

3. Результаты работы

3.1. График зависимости функции приспособленности от времени

На рис. 2 представлен график зависимости функции приспособленности лучшей особи от времени.

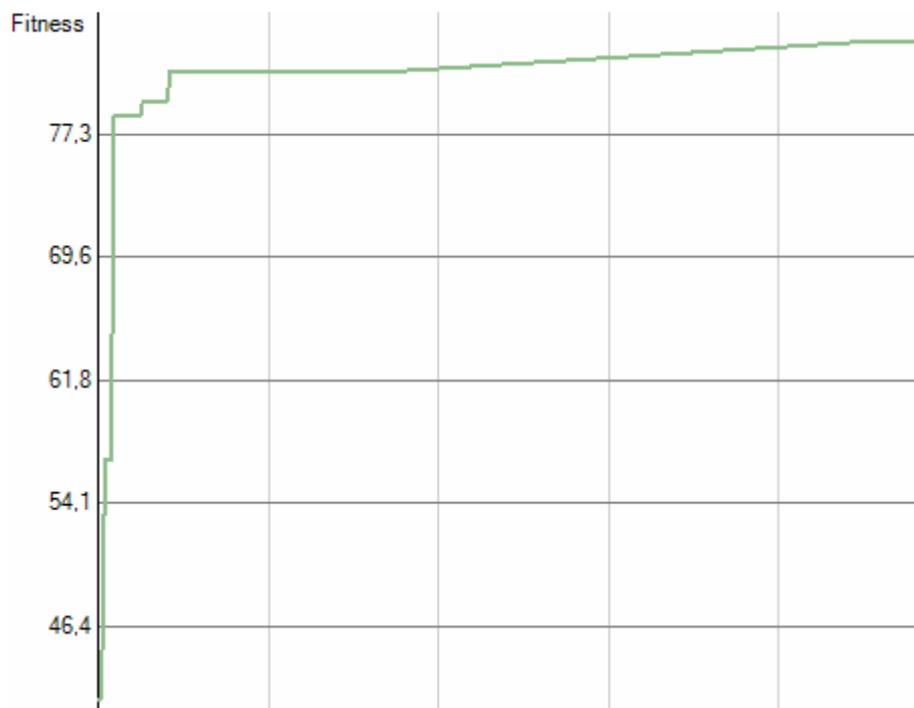


Рис. 2

3.2. Полученный автомат

В результате работы генетического алгоритма был получен автомат Мили из восьми состояний. Значение функции приспособленности полученного муравья равно 85,01.

На рис. 3 изображен полученный автомат с восемью состояниями. Начальное состояние имеет номер «1». Из каждого состояния ведут два перехода: по ребру «0», если впереди нет еды и по ребру «1», если она есть. Буквами *M* (Move), *R* (Right), *L* (Left) обозначены действия муравья при переходе.

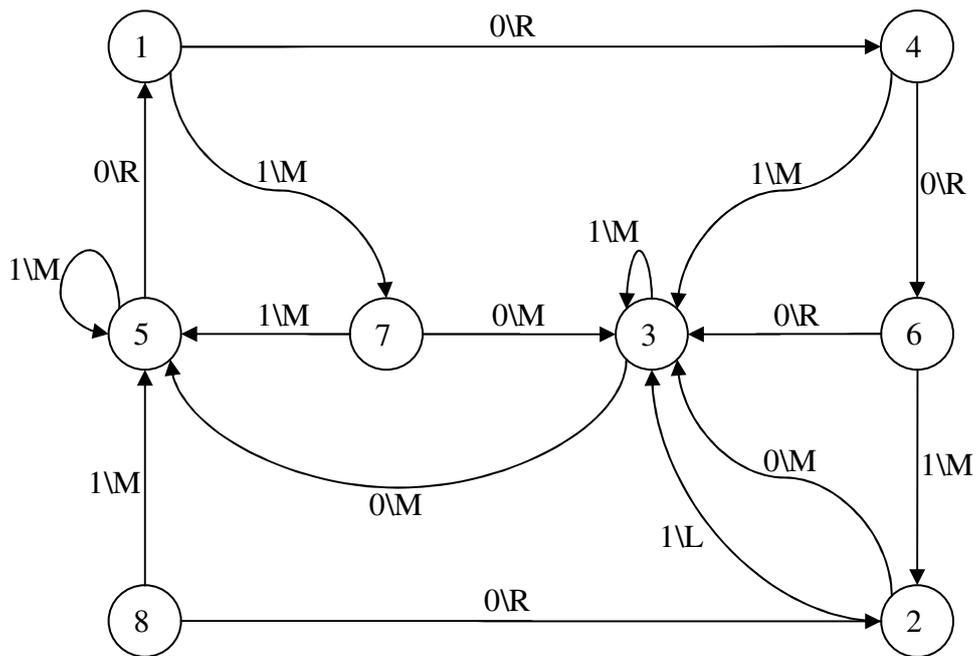


Рис. 3. Граф переходов автомата

Заключение

В результате лабораторной работы был построен автомат Мили из восьми состояний, задающий муравья, который успевает съесть 86 яблок из 89.

Данный результат был получен за восемь минут генетического поиска, что существенно меньше времени полного перебора. Таким образом, выбранный метод – эффективен.

Источники

1. Документация к комплексу для изучения методов глобальной оптимизации *GIOpt*.
http://is.ifmo.ru/courses/_giopt-src.rar
2. Бедный Ю. Д., Шалыто А. А. Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей». <http://is.ifmo.ru/works/ant>
3. Яминов Б. Генетические алгоритмы.
<http://rain.ifmo.ru/cat/view.php/theory/unordered/genetic-2005>