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

Н. М. Паничев

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

Вариант № 4

Санкт-Петербург
2009

Введение 3
1. Постановка задачи 4
1.1. Задача об «Умном муравье» 4
1.2. Автомат Мура 4
2. Реализация 5
2.1. Способ представления автоматов 5
2.2. Метод скрещивания 5
2.3. Метод мутации 5
2.4. Метод генерации следующего поколения 5
3. Результаты работы 6
Заключение 7
Источники 8

Введение

Цель лабораторной работы – изучение генетических алгоритмов для построения конечного автомата. Ведется построение конечного автомата Мура, решающего задачу об «Умном муравье».

При выполнении работы использовался программный комплекс для изучения методов глобальной оптимизации «*GlOpt*»[1], разработанный студентами кафедры «Компьютерные технологии» СПбГУ ИТМО.

1. Постановка задачи

Задачей лабораторной работы является построение с помощью генетического алгоритма конечного автомата Мура, решающего задачу об «Умном муравье». Автомат должен быть представлен в виде битовых строк. Также необходимо использовать островной генетический алгоритм и метод «рулетки» для генерации очередного поколения.

1.1. Задача об «Умном муравье»

Игра происходит на поле размером 32x32 клетки, замкнутом в тор. В некоторых клетках находится 89 яблок. Муравью дается 200 ходов. За каждый ход муравей может выполнить из следующих действий:

- повернуться налево;
- повернуться направо;
- шагнуть на переднюю клетку и, при наличии яблока, съесть его;
- ждать.

Необходимо построить управляющий автомат для муравья, который съест как можно больше яблок за наименьшее число ходов.

1.2. Автомат Мура

Автомат, у которого выходное воздействие зависит только от состояния и не зависит от входного воздействия, называется автоматом Мура. Существует два вида автоматов Мура: автомат первого рода и автомат второго рода. К первым относят автоматы, которые формируют выходные воздействия на основе текущих значений внутренних переменных. Ко вторым – автоматы, которые предварительно обновляют свое состояние, а затем на его основе формирует выходное воздействие. В задаче используется автомат Мура второго рода. Пример такого автомата изображен на рис. 1 (здесь z_i – входные воздействия, w_i – выходные воздействия, a_i – состояния).

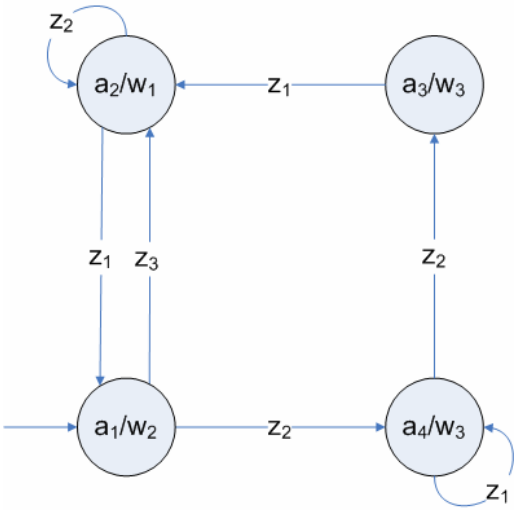

Рис. 1. Автомат Мура второго рода

2. Реализация

При реализации алгоритма с помощью «*GlOpt*» необходимо использовать классы `GeneticOptimizationAlgorithm` и `SearchOperator`. Используемая в алгоритме фитнес-функция – $F + \frac{200 - S}{S}$, где F – число съеденных муравьем яблок, S – число шагов, сделанных муравьем до съедания последнего яблока.

2.1. Способ представления автоматов

Автомат, управляющий муравьем, представляется в программе битовым массивом, содержащим `StateCount*(2*StateCountLength+2)` бит. Каждое состояние кодируется выполняемым действием, состоянием перехода при отсутствии яблока и состоянием перехода при наличии яблока в данной клетке.

2.2. Метод скрещивания

Используется одноточечный кроссовер. При скрещивании из двух предков алгоритм порождает два потомка. Битовая строка первого предка делится на две части случайной длины. Поочередно заменяя их на аналогичные части второго предка, получаем двух потомков. Скрещивание происходит с вероятностью `CrosoverProbability`.

2.3. Метод мутации

Используется два вида мутации: малая и большая.

Малая мутация применяется во время генерации поколения. Каждый бит в автомате особи инвертируется с вероятностью `MutationProbability`.

Большая мутация запускается, если фитнес-функция не изменялась в течение 500 поколений. При этом в каждый остров поселяется самая приспособленная особь, а все остальные заменяются на новых, сгенерированных случайным образом особей.

2.4. Метод генерации следующего поколения

Применяется островной генетический алгоритм [2]. Сначала на `QuantityIslands` островов распределяется `GenerationSize` особей. На каждой итерации алгоритма над особями с одного острова производится операция скрещивания по принципу метода рулетки. Каждой особи на рулетке отводится сектор, размер которого пропорционален величине фитнес-функции данной особи. Запуская рулетку 2*[`размер популяции на острове`], разбиваем особи на пары. Затем применяется одноточечный кроссовер. Далее следует мутация.

На последнем этапе самая приспособленная особь с каждого острова замещает собой случайную особь на остальных островах.

3. Результаты работы

График роста фитнес-функции изображен на рис. 2. Из него следует, что спустя 10 минут работы значение фитнес-функции достигло своего максимума – 85,03. В течение следующих 13 часов работы алгоритма значение фитнес-функции не менялось.

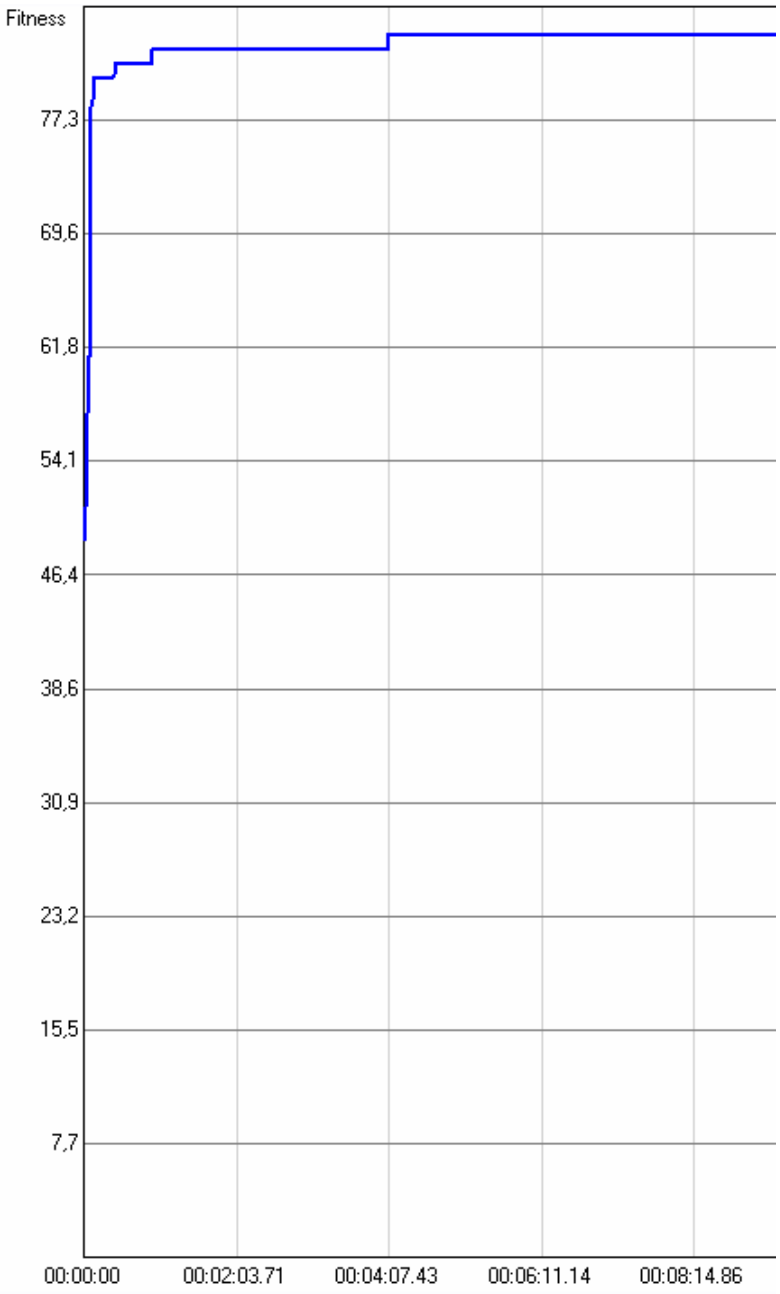

Рис. 2. График роста фитнес-функции

Заключение

В результате работы генетического алгоритма был получен автомат Мура из восьми состояний, который съедает 86 яблок из 89. Автомат представлен на рис. 3.

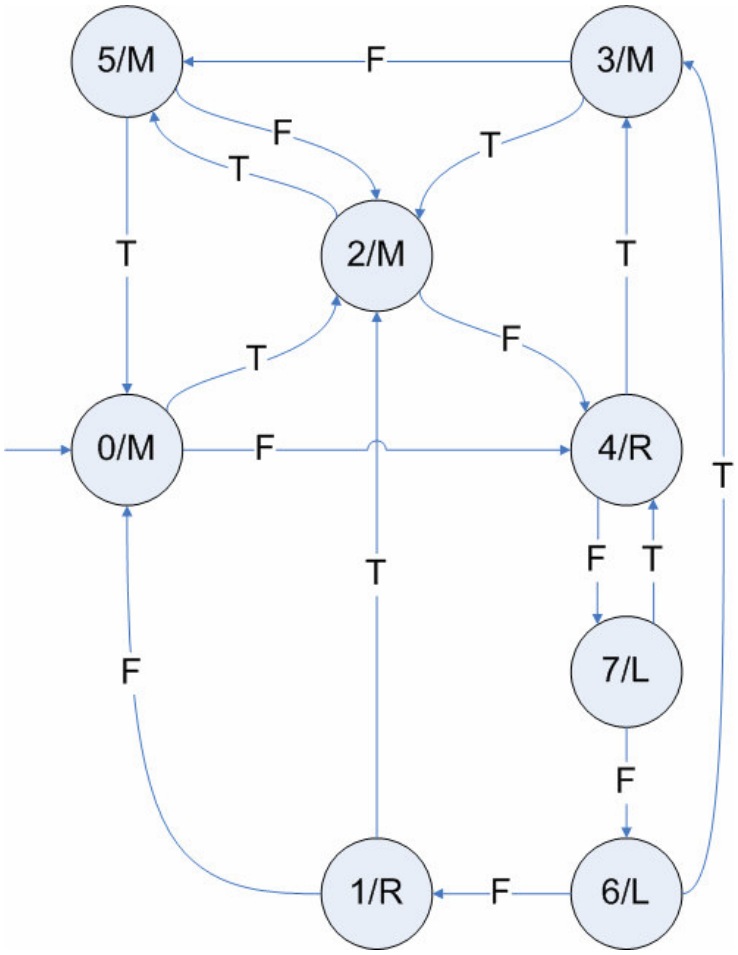

Рис. 3. Полученный автомат

Источники

1. Документация к комплексу для изучения методов глобальной оптимизации «*GlOpt*».
http://is.ifmo.ru/courses/_giopt-src.rar
2. Яминов Б. Генетические алгоритмы.
http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005