

Санкт-Петербургский государственный университет
информационных технологий, механики и оптики
Факультет информационных технологий и программирования
Кафедра компьютерных технологий

В.О. Якорев

Отчет по лабораторной работе
«Автоматное программирование и генетические алгоритмы на
примере задачи об «Умном муравье»

Вариант № 6

Санкт-Петербург
2010

Оглавление	Error! Bookmark not defined.
Введение	3
1. Постановка задачи	4
1.1. Задача об «Умном муравье»	4
1.2. Автомат Мура	5
2. Реализация	6
2.1. Представление автоматов	6
2.2. Метод скрещивания	6
2.3. Метод мутации	6
2.4. Метод генерации очередного поколения	6
2.5. Способ вычисления функции приспособленности	7
3. Результаты работы модуля	8
3.1. Графики максимального, среднего и минимального значений функции приспособленности	8
3.2. Граф переходов полученного автомата	8
Заключение	9
Источники	10

Введение

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

При выполнении работы использовался программный комплекс для изучения методов глобальной оптимизации «GIOpt», разработанный студентами кафедры «Компьютерные технологии» СПбГУ ИТМО. Данный комплекс позволяет реализовывать генетический алгоритмы в виде исполняемых плагинов и изучать процесс их выполнения.

1. Постановка задачи

Цель лабораторной работы – построение конечного автомата Мура, решающего задачу об «Умном муравье», с помощью генетических алгоритмов. В данном варианте необходимо использовать представление автоматов с помощью графов переходов, традиционный генетический алгоритм и метод «рулетки» для генерации очередного поколения. При фиксированном числе состояний необходимо получить автомат, представляющий муравья, съедающего как можно больше еды.

1.1. Задача об «Умном муравье»

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

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

Игра длится 200 ходов. Цель игры – найти муравья, который за минимальное число ходов съест как можно больше яблок. Всего яблок 89. Желательно чтобы в автомате, представляющем муравья, число состояний было минимально. Вид игрового поля показан на рис. 1.

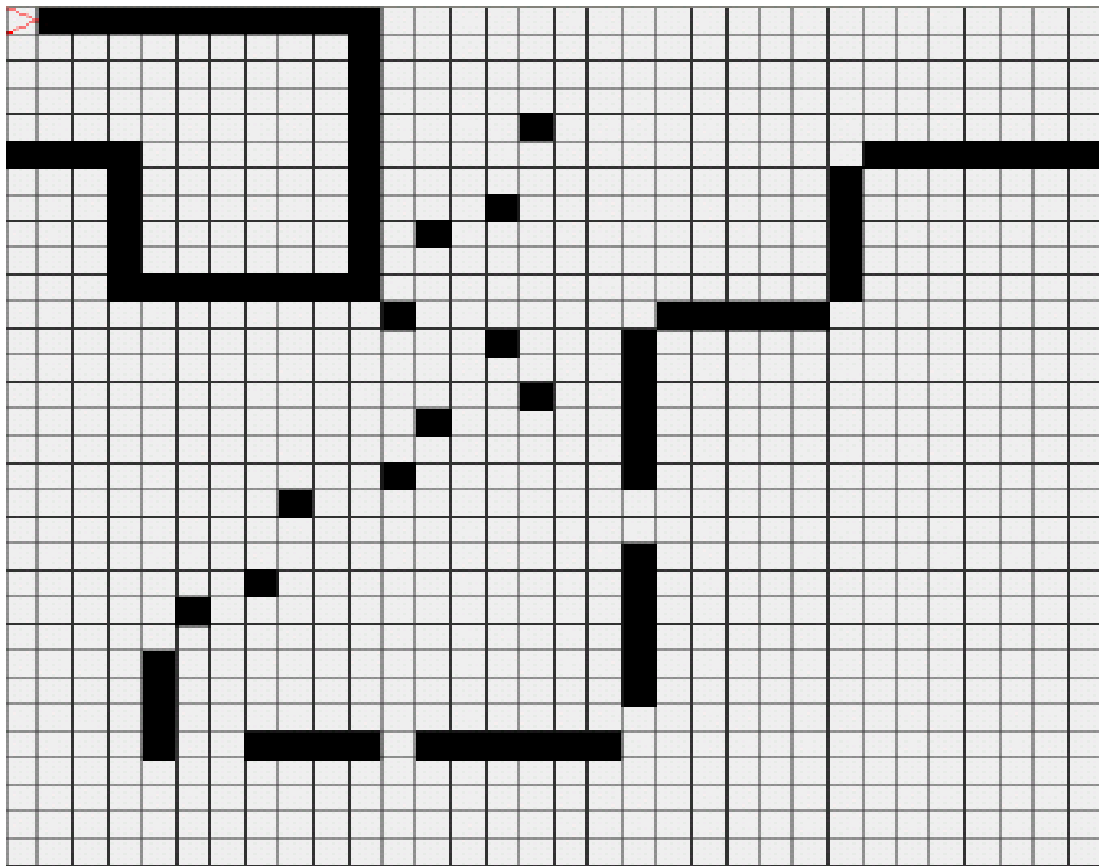


Рис. 1

1.2. Автомат Мура

В данном варианте лабораторной работы необходимо построить автомат Мура. Автомат Мура – конечный автомат, который генерирует выходные воздействия в зависимости от состояния, в которое делается переход. Формально автомат Мура – это совокупность 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 сигнала: перед ним есть еда или нет еды. Автомат, исходя из текущего состояния и воздействия, переходит в следующее состояние (отображение δ). На следующем этапе по отображению μ выбирается действие муравья.

2. Реализация

Виртуальная лаборатория состоит из ядра и подключаемых модулей. Для решения поставленной задачи требуется создать два плагина: плагин генетического алгоритма, использующий традиционный генетический алгоритм и метод рулетки для генерации очередного поколения и плагин особи, представляющий конечный автомат Мура. В плагине особи необходимо реализовать классы мутации и скрещивания.

2.1. Представление автоматов

Особями в данном варианте являются автоматы Мура. Автомат Мура представляется в виде графа переходов. Граф переходов представляется в виде двумерного массива `Automaton.Transition[][] transitions`. Для каждого состояния хранятся два перехода, соответствующие двум входным сигналам. В задаче об «Умном муравье» значениями входного сигнала являются ноль и единица – присутствие или отсутствие еды в клетке перед муравьем. Также хранится одномерный массив, в котором для каждого состояния хранится соответствующее действие.

2.2. Метод скрещивания

Оператор скрещивания получает на вход две родительские особи и выдает две дочерние особи. Число состояний одинаково для всех особей и является настройкой плагина. Процесс скрещивания происходит следующим образом:

1. Создаются две особи, эквивалентные родителям.
2. С одинаковой вероятностью либо первый потомок получает номер начального состояния от первого родителя, второй – от второго, либо первый – от второго родителя, второй – от первого.
3. Для каждого состояния с вероятностью 50% происходит обмен действиями, соответствующими состояниям.
4. Для каждого состояния происходит один из равновероятных вариантов обмена переходами из состояния между потомками.

2.3. Метод мутации

Метод мутации особи заключается в случайном изменении одного из переходов случайного состояния. Также с вероятностью 50% изменяется начальное состояние на случайно выбранное.

2.4. Метод генерации очередного поколения

Для генерации очередного поколения используется традиционный генетический алгоритм (канонический) и метод рулетки. Сначала с помощью метода «рулетки» из текущего поколения выбирается определенное число особей (является параметром алгоритма и задает долю особей, попадающих в первое промежуточное поколение). Алгоритм работает по аналогии с колесом рулетки, разбитым на сектора. Число секторов

равно числу особей, размер секторов пропорционален функции приспособленности особей. Сначала генерируется случайное число, затем находится сектор, в который оно попало. Так как случайные числа выпадают равномерно, в промежуточное поколение попадет больше особей с высоким значением функции приспособленности. Далее, чтобы до конца заполнить промежуточное поколение, повторяется следующая процедура. Случайным образом выбираются две пары особей. Лучшая особь из первой пары скрещивается с лучшей из второй пары (выбор лучшей особи определяется приспособленностью). Потомки попадают в следующее промежуточное поколение. На последнем этапе с некоторой вероятностью каждая особь мутирует (вероятность является параметром алгоритма). Таким образом, мы получаем новое поколение.

2.5. Способ вычисления функции приспособленности

Для того, чтобы вычислить функцию приспособленности, мы эмулируем поведения муравья, описываемого данным автоматом. Муравей будет двигаться пока не сделает 200 шагов или не съест всю еду. Значение функции приспособленности (Fitness) вычисляется по следующей формуле:

$$Fitness = Apples - Steps / 200,$$

где Apples – количество съеденной муравьем еды, Steps – ход, на котором муравей последний раз съел еду.

3. Результаты работы модуля

В результате запуска генетического алгоритма был получен автомат Мура, решающий задачу об «Умном муравье».

3.1. Графики максимального, среднего и минимального значений функции приспособленности

На рис. 2 приведены графики зависимости максимального и среднего значений функции приспособленности.

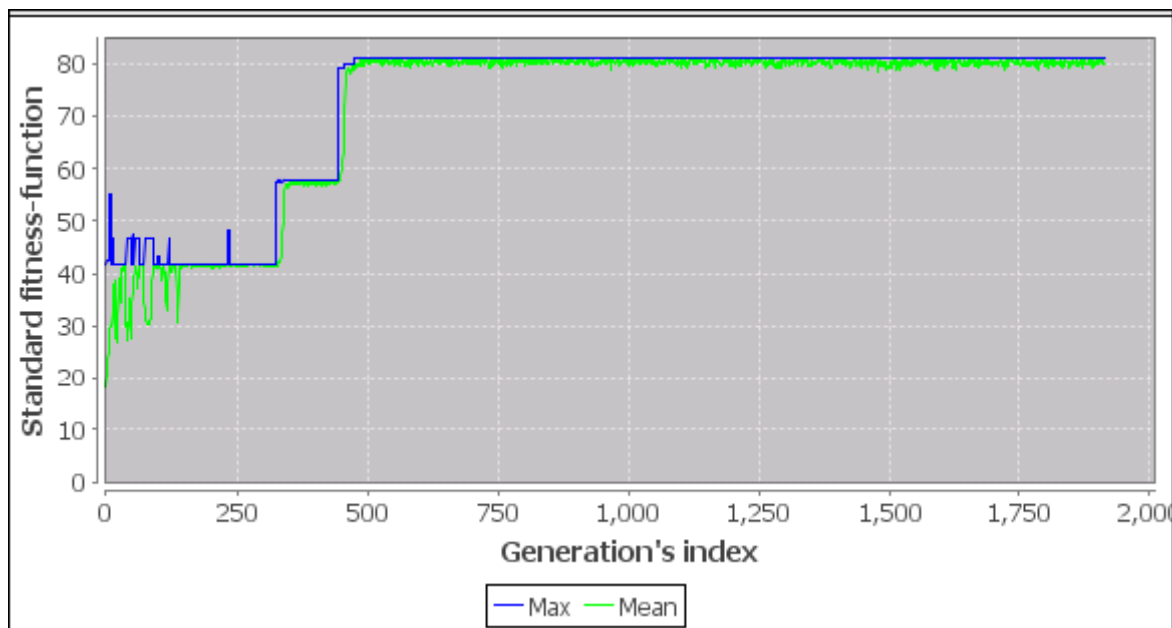


Рис. 2

Из рассмотрения графика, мы видим, что значение функции приспособленности равно 81,275.

3.2. Граф переходов полученного автомата

Полученный конечный автомат Мура, содержащий пять состояний, представлен в виде графа переходов. Вершины графа являются состояниями автомата, ребра графа – переходы между состояниями. На ребрах обозначено входное воздействие, в вершине написано соответствующее ей действие.

На рис. 3 приведен граф переходов полученного в ходе лабораторной работы автомата. Число съедаемых единиц еды – 82.

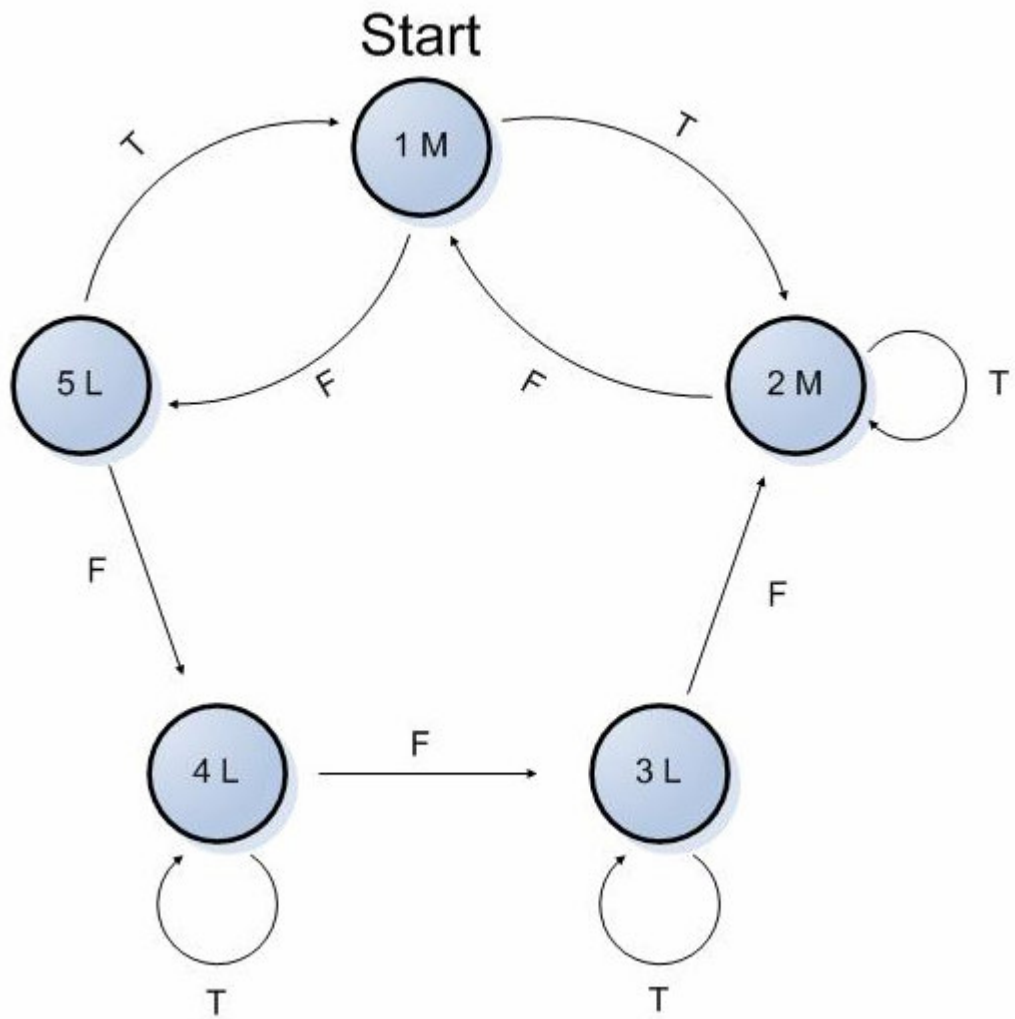


Рис. 3

Заключение

В ходе лабораторной работы был получен автомат Мура из пять состояний. Результаты лабораторной работы показали, что используемый метод достаточно эффективен при решении задачи об «Умном муравье». Полученный муравей съедает 82 единицы еды.

Источники

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