

Санкт-Петербургский государственный университет информационных технологий,
механики и оптики

Кафедра компьютерных технологий

А. И. Агапова

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

Вариант № 8

Санкт-Петербург
2009

Оглавление

Введение	3
1. Постановка задачи	4
1.1 Задача об «Умном муравье».....	4
1.2 Конечный автомат Мура.....	5
2. Реализация	6
2.1 Функция приспособленности.....	6
2.2 Метод мутации.....	6
2.3 Метод скрещивания.....	6
2.4 Генерация нового поколения.....	7
3. Результаты работы	8
3.1 Граф переходов.....	8
3.2 Графики максимального и среднего значения функции приспособленности.....	9
Заключение	10
Источники	11

Введение

В данной работе изучается применение генетических алгоритмов для построения конечных автоматов на примере задачи «Умный муравей». В результате необходимо построить конечный автомат Мура, управляющий поведением муравья таким образом, чтобы задача решалась наиболее эффективным образом.

При выполнении работы использовался программный модуль («Виртуальная лаборатория»), позволяющий реализовывать генетические алгоритмы и особей для них в виде подключаемых плагинов.

1. Постановка задачи

1.1 Задача об умном муравье

В задаче об «Умном муравье» рассматривается фиксированное поле 32 на 32 клетки (рис.1), расположенное на поверхности тора. Большая часть клеток пуста, остальные 89 содержат пищу. Имеется агент, муравей, способный видеть, есть ли перед ним пища, или нет. Он начинает движение с клетки, помеченной как «старт» и может совершать следующие действия:

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

Максимальное число ходов – 200. Муравей должен съесть как можно больше пищи, совершив при этом как можно меньше ходов.

В данной лабораторной работе эта задача решается путем построения с помощью генетических алгоритмов конечного автомата, управляющего действиями муравья. При этом желательно минимизировать число состояний этого автомата.

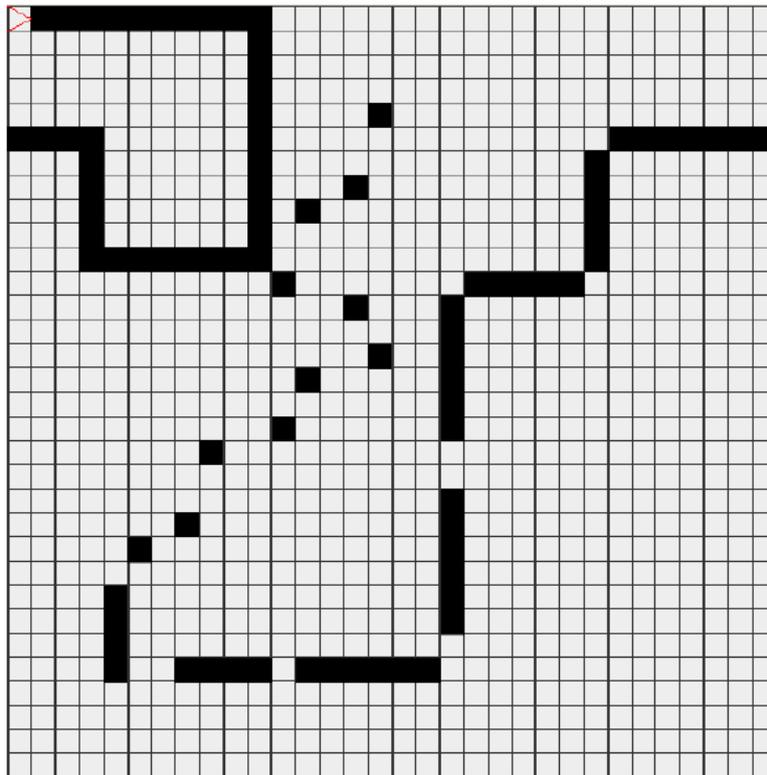


Рис. 1. Поле

1.2 Конечный автомат Мура

Это конечный автомат, генерирующий выходные воздействия в зависимости от текущего состояния. Функция переходов зависит от входных воздействий и текущего состояния.

Формально, конечным автоматом Мура называют совокупность $\langle S, X, Y, s_0, \delta, \mu \rangle$, где:

S – конечное множество состояний;

X – конечное множество входных воздействий;

Y – конечное множество выходных воздействий;

s_0 – стартовое состояние;

δ – функция перехода. Ее аргументы – состояние и входной символ, результат – состояние:

$$s(t+1) = \delta(s(t), x(t));$$

μ – функция, ставящая выходное воздействие в соответствие состоянию: $y(t) = \mu(s(t))$;

Здесь $t = 0, 1, 2, \dots$ – абстрактное время.

В данном случае входными воздействиями являются два взаимоисключающих случая: есть ли перед муравьем еда в данный момент времени, или нет. Выходные воздействия – это те четыре действия, которые может совершать муравей. Каждому состоянию ставится в соответствие одно из них.

В итоге автомат хранится в виде графа переходов, где вершины – это состояния, ребра – это переходы. Из каждой вершины выходит по два ребра. Переход по первому осуществляется, когда перед муравьем есть еда, переход по второму – если ее нет. При переходе в очередное состояние агент выполняет соответствующее ему действие.

2. Реализация

В данной работе используется традиционный генетический алгоритм. В роли особей выступают конечные автоматы в вышеуказанном представлении.

2.1. Функция приспособленности

Функция приспособленности в генетическом алгоритме вычисляется для каждой особи и является мерой эффективности решения этой особью итоговой задачи.

Для того чтобы оценить, насколько хорошо полученный автомат управляет действиями муравья, необходимо смоделировать поведение агента и оценить результаты. Таким образом, запустив муравья на имеющемся поле, можно получить следующие показатели:

a – количество съеденной за 200 ходов пищи;

t – номер последнего из ходов, на которых была съедена пища (ходы нумеруются с единицы).

Примем за значение функции приспособленности следующую величину:

$$F = a - t/200.$$

Из рассмотрения формулы видно, что эта функция определена вполне корректна, и значение ее тем выше, чем эффективней автомат решает задачу.

2.2. Метод мутации

Мутация в генетическом алгоритме – это случайное изменение некоторых признаков особи.

При имеющемся представлении особей у них можно изменить четыре признака:

- начальное состояние;
- действие, поставленное в соответствие случайно выбранному состоянию;
- состояние, в которое ведет случайно взятое ребро;
- соответствие переходов «есть пища» и «нет пищи» для случайно выбранного состояния.

Последнее означает, что те два ребра, которые ведут из случайной вершины, обмениваются состояниями, в которые они ведут.

Все указанные варианты действий равновероятны.

2.4. Метод скрещивания

Оператор скрещивания принимает на вход две особи – родителей, и возвращает также две особи – потомков. Обозначим первых как $S1$ и $S2$, а вторых – как $P1$ и $P2$.

Скрещивание происходит следующим образом:

- 1) Если обозначить начальное состояние особи A как $A.start$, то с равной вероятностью либо $P1.start = S1.start$ и $P2.start = S2.start$, либо $P1.start = S2.start$ и $P2.start = S1.start$ (рис. 2).

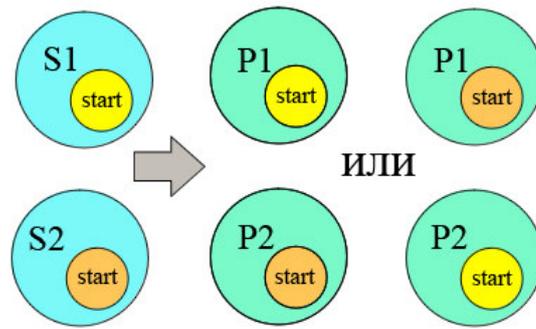


Рис. 2. Скрещивание. Этап 1

2) Если обозначить переход из состояния i в автомате A по входному воздействию «вперед есть пища» как $A(i, 0)$, а по «вперед нет пищи» – как $A(i, 1)$, то с равной вероятностью будет справедливо одно из четырех (рис. 3):

$$\begin{aligned}
 &P1(i, 0) = S1(i, 0), P1(i, 1) = S1(i, 1), P2(i, 0) = S2(i, 0), P2(i, 1) = S2(i, 1); \\
 &P1(i, 0) = S1(i, 0), P1(i, 1) = S2(i, 1), P2(i, 0) = S2(i, 0), P2(i, 1) = S1(i, 1); \\
 &P1(i, 0) = S2(i, 0), P1(i, 1) = S1(i, 1), P2(i, 0) = S1(i, 0), P2(i, 1) = S2(i, 1); \\
 &P1(i, 0) = S2(i, 0), P1(i, 1) = S2(i, 1), P2(i, 0) = S1(i, 0), P2(i, 1) = S1(i, 1).
 \end{aligned}$$

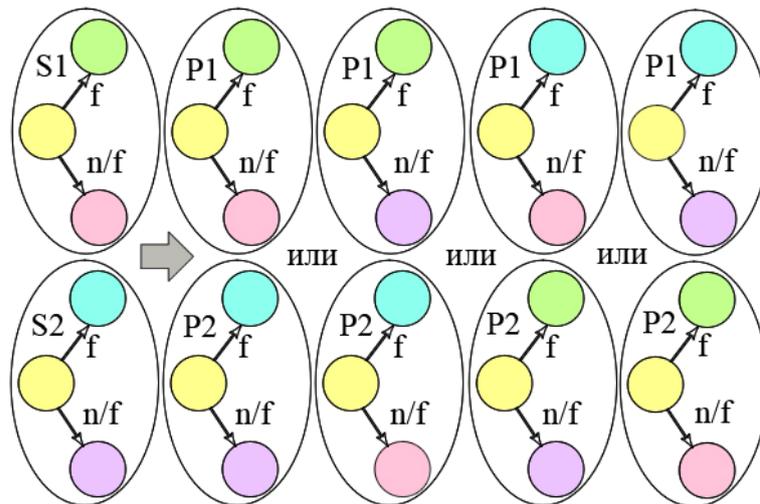


Рис. 3. Скрещивание. Этап 2

2.3. Генерация очередного поколения

Размер поколения, вероятность мутации и доля особей, попадающих в новое поколение из предыдущего, являются настраиваемыми параметрами алгоритма.

Для того чтобы выбрать тех особей, которые попадут в новое поколение из предыдущего, используется «метод рулетки». Он состоит в следующем.

Расположим всех особей на колесе рулетки таким образом, что размер сектора, принадлежащего каждой из них, пропорционален ее значению функции приспособленности. Запустив рулетку k раз, отберем k особей. Таким образом, вероятность попадания особи в новое поколение будет пропорциональна ее значению функции приспособленности.

Новое поколение дополняется до необходимого размера следующим образом.

Пока размер поколения меньше требуемого, выполняются следующие действия:

1) Случайно генерируются две пары особей, в каждой из пар выбирается лучшая (имеющая большее значение функции приспособленности). Далее полученные две особи

скрещиваются, и их потомки добавляются в новое поколение.

2) Если размер поколения все еще меньше требуемого, в него добавляется случайно выбранная из старого поколения особь, к которой предварительно применяется оператор мутации.

Далее с вероятностью, задаваемой параметром «вероятность мутации», ко всем особям нового поколения применяется оператор мутации.

Если после смены достаточно большого числа поколений не происходит увеличения приспособленности, применяется так называемая «большая мутация»: все особи поколения заменяются случайно сгенерированными.

3. Результаты работы

Алгоритм был запущен с параметрами:

- шесть состояний в автомате;
- вероятность мутации 0.02;
- размер поколения равен 200 особям;
- доля особей, попадающих в новое поколение из старого 0.7.

В результате получен конечный автомат из шести состояний, достаточно эффективно решающий поставленную задачу: муравей съедает 83 единицы пищи за 198 ходов.

3.1 Граф переходов

На рис. 4 представлен граф переходов полученного конечного автомата.

Вершины графа - это состояния автомата. В них записаны обозначения действий:

R – повернуть направо;

L – повернуть налево;

N – ничего не делать;

S – сделать шаг вперед,

которые следует выполнить муравью, как только автомат окажется в данном состоянии. Начальное состояние помечено желтым цветом.

Ребра графа означают переходы. Буква f , записанная на ребре, означает переход по условию «есть пища», n/f — по условию «нет пищи».

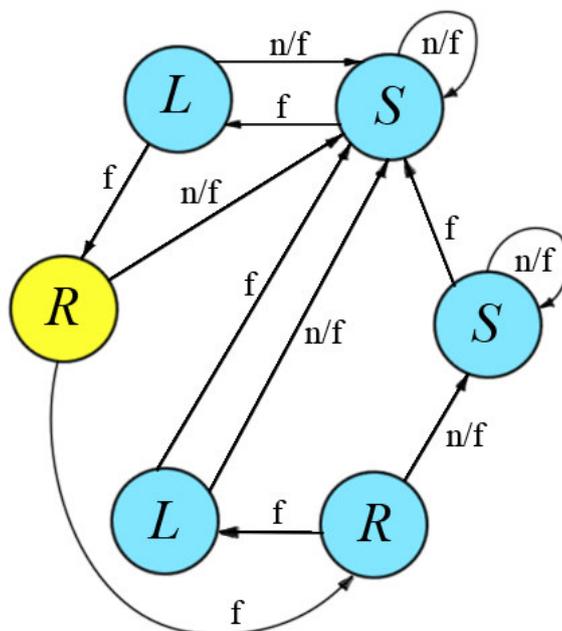


Рис. 4. Граф переходов

3.2 Графики максимального и среднего значения функции приспособленности

На рис. 5 изображены графики максимального (синий) и среднего (зеленый) значений функции приспособленности на протяжении 12000 поколений.

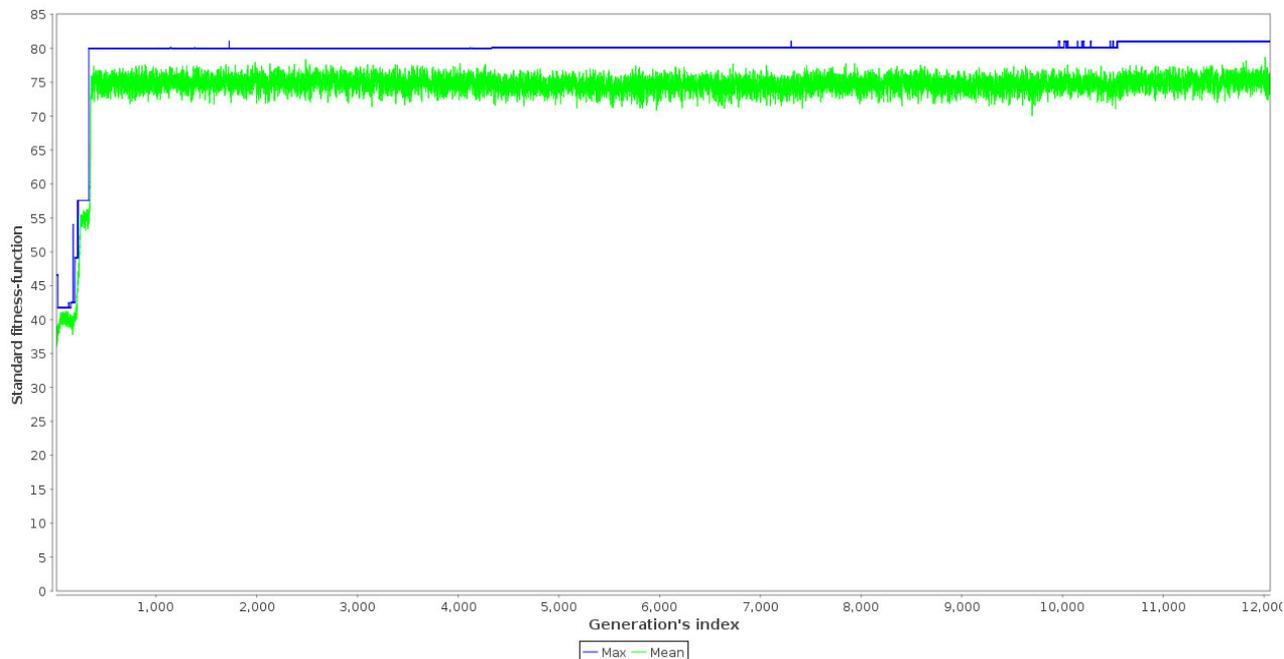


Рис. 5. Графики функции приспособленности

Заметим, что выбранный генетический алгоритм довольно быстро сходится: уже после нескольких сотен поколений значение функции приспособленности перестает значительно увеличиваться. Эксперименты показывают, что выбранная доля особей, переходящих в новое поколение, близка к оптимальной по сходимости алгоритма: при меньших значениях теряется слишком много хороших особей, при больших – поколения быстро вырождаются и заклиниваются около одного значения.

В то время как максимальное значение функции приспособленности остается постоянным на протяжении многих поколений, среднее значение заметно колеблется и по величине ощутимо меньше максимального. Это происходит вследствие наличия случайно сгенерированных особей в каждом поколении.

Заключение

Результаты лабораторной показывают, что использованный метод имеет сравнительно невысокую эффективность для решения данной задачи, хотя и значительно превосходит по скорости построения автомата метод полного перебора.

Сгенерировать автомат, решающий задачу «Умный муравей» лучше, чем полученный, возможно, но число итераций алгоритма при этом возрастет во много раз.

Кроме того, известен [2] автомат с пятью состояниями, построенный вручную для решения этой задачи. Управляемый этим автоматом муравей способен съесть 81 единицу пищи.

Источники

1. *Давыдов А. А., Соколов Д. О., Царев Ф. Н., Шалыто А. А.* Описание виртуальной лаборатории /Сборник докладов III Международной научно-практической конференции «Современные информационные технологии и ИТ-образование». ВМК МГУ. М.: МАКС Пресс. 2008, с. 179 – 183.
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 плагинов к виртуальной лаборатории.*
http://svn2.assembla.com/svn/not_instrumental_tool/docs/pdf/interface_manual.pdf