

Санкт-Петербургский государственный университет информационных
технологий, механики и оптики

Факультет информационных технологий и программирования

Кафедра «Компьютерные технологии»

А. В. Шестаков

Отчет по лабораторной работе

**«Использование генетических алгоритмов для построения управляющих
автоматов»**

Вариант № 12

Санкт-Петербург

2009

Оглавление

Введение	3
1. Постановка задачи	4
1.1. Текст задания лабораторной работы	4
1.2. Конечный автомат Мура	4
1.3. Постановка задачи об «Умном муравье»	5
1.4. Клеточный генетический алгоритм	5
1.5. Метод «рулетки»	6
2. Реализация	6
2.1. Описание используемого способа представления автоматов в генетическом алгоритме	7
2.2. Описание метода скрещивания	7
2.3. Описание оператора мутации	7
2.4. Описание метода генерации очередного поколения	7
2.5. Описание способа вычисления фитнес-функции	8
3. Полученный результат	9
3.1. График значений фитнес-функции	9
3.2. Список переходов полученного автомата	9
3.3. Представление полученного автомата в виде диаграммы переходов	11
Заключение	11
Источники	13

Введение

Цель данной лабораторной работы – построение управляющего автомата с помощью генетического алгоритма. В данном варианте задания требуется построить генетическим алгоритмом автомат Мура, который бы эффективно решал задачу об «Умном муравье».

Для выполнения поставленной задачи был использован программный комплекс для изучения методов глобальной оптимизации *GlOpt* [1], созданный студентами кафедры «Компьютерные технологии» А. С. Тяhti и А. А. Чебатуркиным.

1. Постановка задачи

1.1. Текст задания лабораторной работы

«Постройте с помощью генетических алгоритмов конечный автомат Мура, решающий задачу об «Умном муравье». Используйте представление автоматов с помощью графов переходов. Способ скрещивания выберите самостоятельно. Используйте клеточный генетический алгоритм и метод «рулетки» для генерации очередного поколения».

1.2. Конечный автомат Мура

Согласно книге «Автоматное программирование» [2], автоматом Мура называется автомат, у которого выходное воздействие зависит только от состояния и не зависит от входного воздействия. Различают два типа автомата Мура: автомат первого рода формирует выходные воздействия на основе текущих значений внутренних переменных, в то время как автомат второго рода сначала обновляет свое состояние, а затем на основе обновленного состояния формирует выходное воздействие. Для выполнения задания данной лабораторной работы было решено использовать автомат Мура второго рода, пример такого автомата изображен в виде диаграммы переходов на рис. 1.

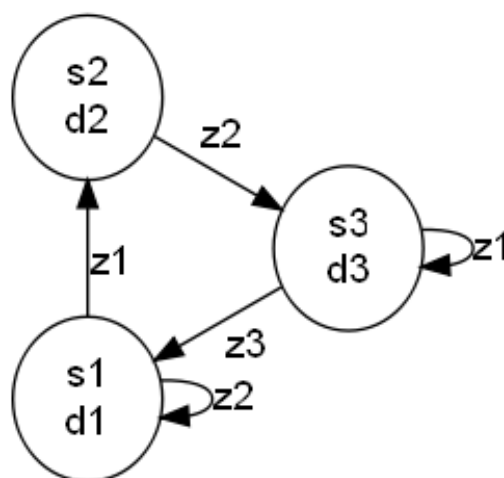


Рис. 1. Пример автомата Мура. s – состояния, z – входные воздействия, d – действия, соответствующие состояниям

В каждой вершине написано соответствующее ей выходное воздействие. Таким образом, оно не зависит от входного воздействия, а зависит только от текущего состояния автомата.

1.3. Постановка задачи об «Умном муравье»

На тороидальном поле размером 32 на 32 находятся 89 ячеек с едой. Еда расположена вдоль ломаной линии («тропы»), некоторые ячейки которой пусты. Расположение ячеек с едой фиксировано. Муравей должен за 200 ходов съесть всю еду на линии. Требуется построить конечный автомат, моделирующий поведение такого муравья.

В начале муравей находится в левом верхнем углу поля и смотрит направо. У муравья есть сенсор, позволяющий ему определить, есть ли еда в ячейке, находящейся непосредственно перед ним. За один ход муравей может выполнить одно из следующих действий – повернуться налево, повернуться направо, подвинуться на ячейку вперед. Для того чтобы съесть еду, муравью необходимо попасть в ячейку с ней. Ячейки с едой, в которых муравей уже побывал, неотличимы для него от исходно пустых ячеек.

1.4. Клеточный генетический алгоритм

Клеточный генетический алгоритм (Cellular Genetic Algorithm) – модель параллельного генетического алгоритма [3]. Предполагается, что в каждой клетке квадратного поля, замкнутого в тор, располагается одна особь (рис. 2).

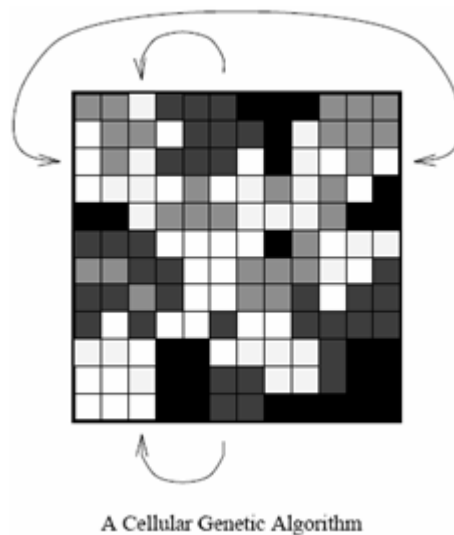


Рис. 2

По аналогии с «классическим генетическим алгоритмом», на каждой итерации работы алгоритма происходят отбор, скрещивание и мутация. Каждая особь может быть скрещена только с одной особью из соседней клетки (соседними считаются клетки, расположенные сверху, снизу, слева, справа от данной).

1.5. Метод «рулетки»

Метод «рулетки» – метод определения, какие члены популяции выбираются для размножения. При использовании этого метода оценка каждой особи пропорциональна величине ее фитнес-функции. Выбор производится в результате последовательных запусков колеса рулетки. После остановки вращения выбирается та особь, на участок которой указывает стрелка рулетки.

2. Реализация

Для реализации был использован программный комплекс для изучения методов глобальной оптимизации *GlOpt* [1]. Он подробно описан в прилагающейся к нему документации. Для выполнения задачи потребовалось создать программу, включающую в себя реализацию заданного алгоритма,

оператора скрещивания и оператора мутации. При построении этой программы искался автомат с девятью состояниями.

2.1. Описание используемого способа представления автоматов в генетическом алгоритме

Каждый автомат представлен в программе в виде таблицы переходов, таблицы действий, соответствующих состояниям автомата и номера стартового состояния.

2.2. Описание метода скрещивания

Используемый метод скрещивания создает копию первого родителя, затем для каждого состояния автомата копии с вероятностью 50% изменяет один из переходов на случайный переход по тому же входному воздействию из второго родителя. Также изменяется действие в состоянии, в которое совершается переход для того, чтобы частично сохранить логику переходов второго родителя. Начальное состояние нового автомата наследуется от одного из родителей равновероятно.

2.3. Описание оператора мутации

Оператор мутации равновероятно выполняет над автоматом одно из следующих действий:

1. Изменяет начальное состояние автомата на случайно выбранное.
2. Для случайно выбранного состояния изменяет переход из него по одному из входных воздействий на случайно выбранный переход.
3. Для случайно выбранного состояния изменяет соответствующее ему действие на случайное.

2.4. Описание метода генерации очередного поколения

Для генерации очередного поколения используется представление списка индивидуумов как квадратного поля (по одному индивидууму на

клетку), замкнутого в тор. Для каждого индивидуума выполняются следующие действия:

1. Происходит скрещивание с четырьмя соседями (сверху, снизу, слева, справа).
2. К каждому из четырех полученных индивидуумов с вероятностью 15% применяется оператор мутации.
3. Методом «рулетки» определяется кандидат на переход в следующее поколение (вероятность перехода пропорциональна значению фитнес-функции индивидуума).
4. Если кандидат обладает лучшим значением фитнес-функции, чем исходный индивидуум, то он занимает его место.

2.5. Описание способа вычисления фитнес-функции

Фитнес-функция (функция приспособленности) вычисляется по формуле: $F + \frac{200-S}{S}$. Здесь F – число яблок, съеденных муравьем, S – число шагов, сделанных муравьем до момента съедения последнего яблока. Вычисление число сделанных шагов и съеденных яблок производится путем эмуляции поведения муравья, управляемого автоматом.

3. Полученный результат

3.1. График значений фитнес-функции

График значений фитнес-функции, полученный в результате работы программного комплекса, представлен на рис. 3. Из рассмотрения графика и левого поля пользовательского интерфейса следует, что удалось построить автомат, управляющий муравьем, который съедает все 89 яблок. Этот результат достигается сравнительно быстро – за 6 мин 47 с.

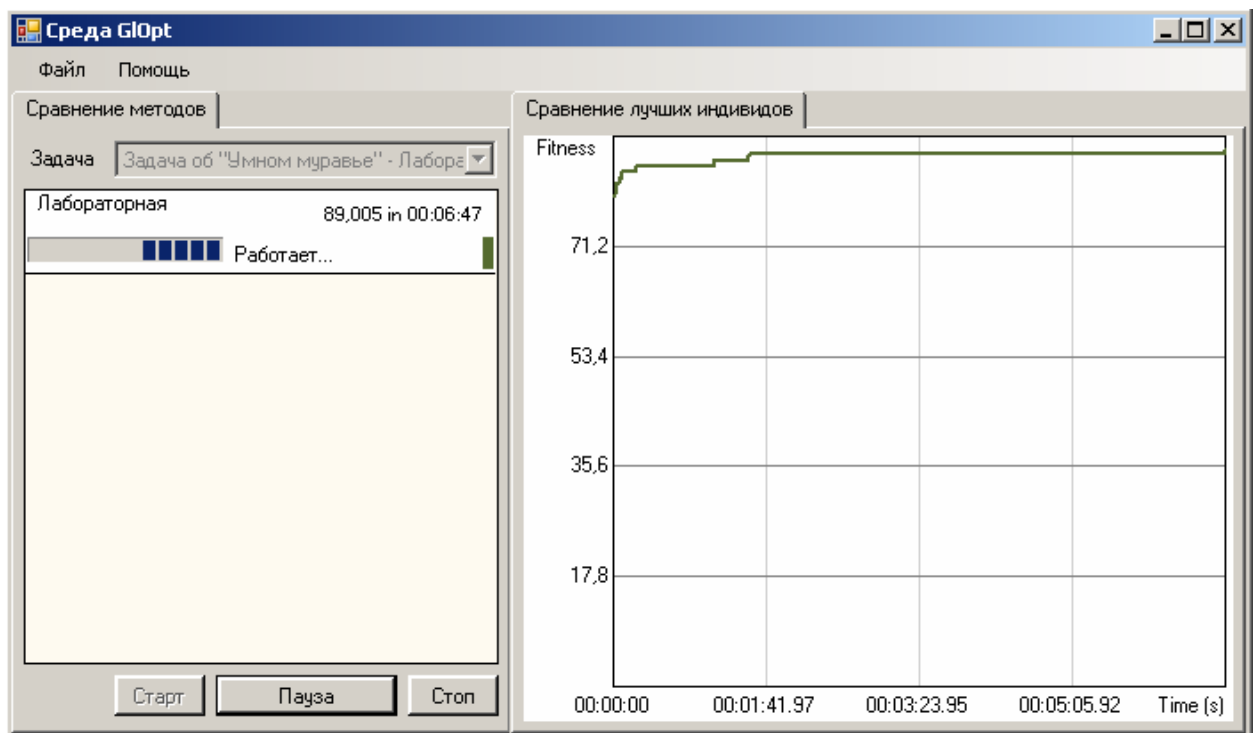


Рис. 3

3.2. Список переходов сгенерированного автомата

Начальное состояние автомата № 4, список его состояний и переходов:

Состояние № 1:

Действие – идти прямо.

Впереди еда – переход в состояние 1.

Впереди нет еды – переход в состояние 5.

Состояние № 2:

Действие – идти прямо.

Впереди еда – переход в состояние 1.

Впереди нет еды – переход в состояние 1.

Состояние № 3:

Действие – поворот направо.

Впереди еда – переход в состояние 2.

Впереди нет еды – переход в состояние 4.

Состояние № 4:

Действие – поворот направо.

Впереди еда – переход в состояние 9.

Впереди нет еды – переход в состояние 6.

Состояние № 5:

Действие – идти прямо.

Впереди еда – переход в состояние 7.

Впереди нет еды – переход в состояние 3.

Состояние № 6:

Действие – поворот направо.

Впереди еда – переход в состояние 2.

Впереди нет еды – переход в состояние 8.

Состояние № 7:

Действие – идти прямо.

Впереди еда – переход в состояние 2.

Впереди нет еды – переход в состояние 8.

Состояние № 8:

Действие – поворот направо.

Впереди еда – переход в состояние 5.

Впереди нет еды – переход в состояние 5.

Состояние № 9:

Действие – идти прямо.

Впереди еда – переход в состояние 9.

Впереди нет еды – переход в состояние 4.

3.3. Представление полученного автомата в виде диаграммы переходов

На рис. 4 полученный автомат Мура представлен в виде диаграммы переходов, где в вершинах написаны соответствующие им действия. Переходы обозначены стрелками различных цветов. Красный – перед муравьем нет еды, зеленый – перед муравьем находится еда, фиолетовый – переход в любом случае.

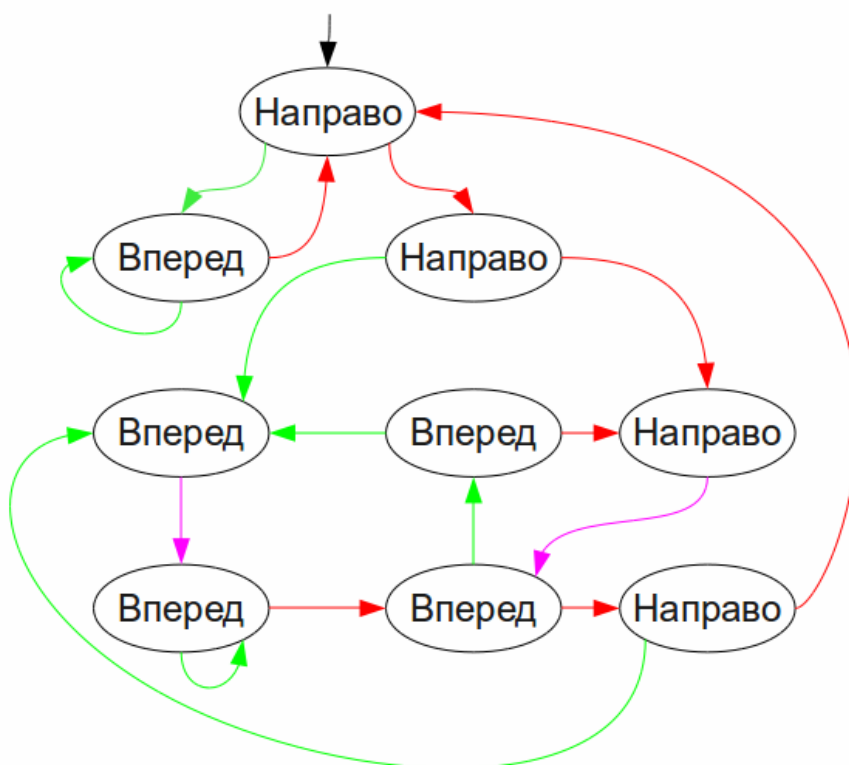


Рис.4

Заключение

В результате выполнения данной лабораторной работы с помощью генетического алгоритма был построен автомат Мура, решающий задачу об «Умном муравье». Скорость построения автомата (чуть менее семи минут) свидетельствует о том, что данный генетический алгоритм позволяет быстро и эффективно решать поставленную задачу.

В результате работы алгоритма был получен конечный автомат Мура второго рода (его выходное воздействие однозначно определяется

состоянием, в которое автомат перешел) с девятью состояниями. Этот автомат эффективно решает задачу об «Умном муравье». Под его управлением муравей съедает все яблоки, величина его фитнес-функции 89,005.

Источники

1. *Тяхти А. С., Чебатуркин А. А.* Программный комплекс для изучения методов глобальной оптимизации *GIOpt*.
http://is.ifmo.ru/courses/_giopt-src.rar
2. *Поликарпова Н. И., Шалыто А. А.* Автоматное программирование. СПб.: Питер, 2009. http://is.ifmo.ru/books/_book.pdf
3. *Яминов Б.* Генетические алгоритмы.
<http://rain.ifmo.ru/cat/view.php/theory/unordered/genetic-2005>