

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

В. С. Смирнов

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

Вариант №7

Санкт-Петербург  
2009

## **Содержание**

Введение .....	2
1. Постановка задачи .....	3
2. Реализация алгоритма .....	3
2.1. Представление автоматов в генетическом алгоритме .....	3
2.2. Функция приспособленности .....	4
2.3. Метод скрещивания .....	4
2.4. Метод мутации .....	4
2.5. Способ генерации очередного поколения .....	5
3. Результаты эксперимента .....	5
Заключение .....	7
Библиография .....	8

## **Введение**

В данной лабораторной работе изучается применение генетических алгоритмов для построения конечных автоматов на примере задачи об «Умном муравье». В результате работы автоматически построен автомат Мили, управляющий муравьем на поле игры.

Для выполнения работы использована «Виртуальная лаборатория» [1], созданная студентами кафедры «Компьютерные технологии» СПбГУ ИТМО, позволяющая реализовывать генетические алгоритмы и особи для них в виде отдельных подключаемых модулей.

## 1. Постановка задачи

Задание сформулировано следующим образом:

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

В задаче об «Умном муравье» [2] задано клеточное поле, по которому может передвигаться агент – муравей. Поле имеет размеры  $32 \times 32$ , оно лежит на поверхности двумерного тора. Это означает, что муравей, делая шаг за край поля, оказывается на его противоположном конце. В некоторых клетках поля размещена еда. Муравей имеет направление движения. За один ход он совершает одно из трех действий: перейти в находящуюся перед собой клетку, развернуться влево, или повернуть направо. При этом, ему известно, имеется ли еда в клетке перед ним. Задача муравья – начав движение из угла поля, за 200 ходов съесть, как можно больше единиц еды – посетить максимальное число содержащих их клеток.

Автомат Мили [3] – разновидность конечного автомата, в которой переход определяется текущим состоянием и значением входной переменной. Цель лабораторной работы – с использованием генетического алгоритма, построить автомат Мили, управляющий муравьем.

Общие принципы работы генетических алгоритмов, их типы, основные термины, применяемые для их описания, рассмотрены в статье [4]. В данной работе будут приведены только отличия реализованного генетического алгоритма от известных стандартных моделей.

## 2. Реализация алгоритма

### 2.1. Представление автоматов в генетическом алгоритме

В задании предъявлено требование, чтобы в особях генетического алгоритма автоматы были представлены в виде графов переходов. Этот способ записи также известен под названием «представление с помощью полных таблиц переходов». Его описание можно найти в работе [5].

Автомат рассматривается как ориентированный граф. Его вершины представляют состояния. Переходы автомата отображаются в дуги, помеченные соответствующими значениями входного и выходного символов. Матрица смежности такого графа вместе с номером начального состояния является представлением автомата в особи. В конкретной программной реализации, матрица смежности хранится в двумерном массиве, в ячейках которого записаны номер нового состояния и значение выходного символа. Рассмотренный способ представления иллюстрирует рис. 1.

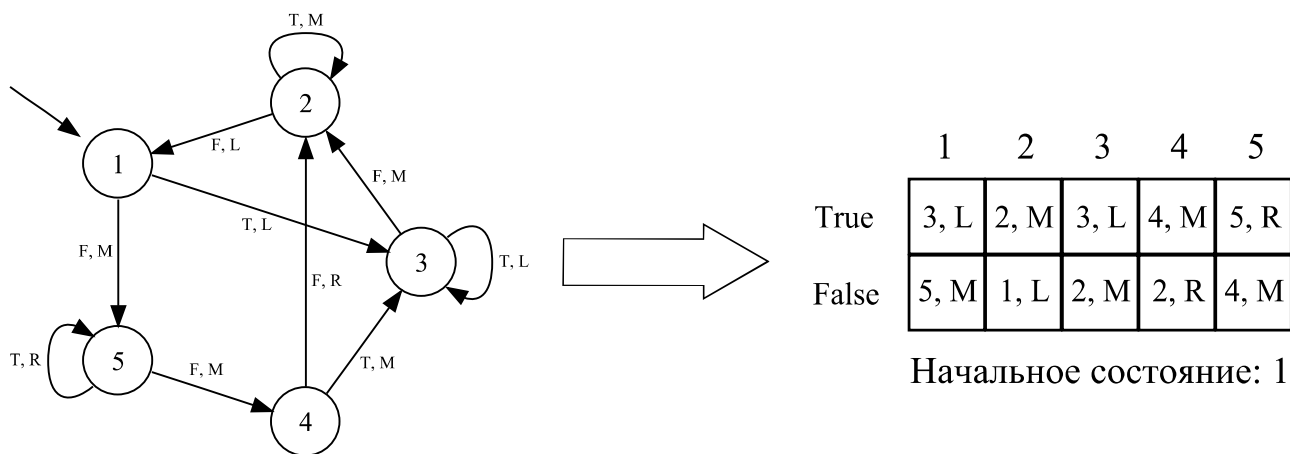


Рис. 1. Представление автомата в виде графа переходов

## 2.2. Функция приспособленности

Функция приспособленности отражает эффективность особи в решении задачи. Муравей, в соответствии с правилами, устанавливается в верхнюю левую клетку поля, и испытываемому автомату передается управление. Муравей совершает шаги до тех пор, пока не съест всю еду, или же не сделает двухсотый шаг. По итогам испытания подсчитывается количество съеденной еды и совершенных шагов. Функция приспособленности вычисляется по формуле:  $fitness = eaten - steps/200$ , где *eaten* — количество съеденной еды, а *steps* — совершенных шагов (нумерация начинается с нуля).

## 2.3. Метод скрещивания

В качестве метода скрещивания выбран один из наиболее простых способов: одноточечный кроссовер. Случайно выбирается номер позиции из чисел от двух до  $n - 1$ , где  $n$  — число состояний. Особи сохраняют свои начальные состояния, и обмениваются данными массивов, хранящих информацию о переходах, начиная с выбранной позиции.

## 2.4. Метод мутации

Выбор оператора мутации также осуществлялся из соображений простоты. Изменению у особи подвергается только один переход автомата: из случайно выбранного состояния, при одном из возможных значений входной переменной. Данные соответствующей ячейки массива переходов заменяются случайно сгенерированными. Помимо этого, с некоторой заданной вероятностью  $\alpha$ , номер начального состояния заменяется случайным.

## **2.5. Способ генерации очередного поколения**

В соответствии с заданием, реализованный генетический алгоритм — островной, с отбором особей по методу «рулетки». В дополнение к поставленным требованиям, в алгоритме применен элитизм и периодическое уничтожение части популяции.

Метод рулетки предназначен для выбора особи из поколения с учетом ее приспособленности. Как подсказывает название, метод действует подобно рулетке, если отвести на ней по сектору для каждой особи, ширина которого соответствовала бы ее приспособленности. Таким образом, у каждой особи есть шанс быть выбранной, но у лучших он выше. Важно отметить, что выбор особи не исключает ее из популяции: применяя метод несколько раз подряд, может быть выбрана одна и та же особь.

Все генетические алгоритмы объединяет процесс получения решения: поколение за поколением, из предыдущей популяции производится следующая популяция. Для этого каким-нибудь способом отбираются особи, и к ним применяются операторы скрещивания и мутации. Островной алгоритм делает эту схему сложнее: в нем популяция разделена на несколько равных частей («островов»), в каждой из которых процесс эволюции производится отдельно, в изоляции от других. Периодически, по прошествии в эволюции некоторого числа поколений, острова обмениваются лучшими особями, этот процесс называется миграцией. От выбора его частоты зависит эффективность алгоритма: слишком редкие миграции приводят к преждевременной сходимости, а слишком частые — к тому, что теряются отличия в действии островной модели от канонической.

В данной работе реализация островного алгоритма дополнена элитизмом и периодическим уничтожением части популяции. Элитизм — метод построения нового поколения, при котором часть мест в нем занимают лучшие особи предыдущего без изменений, а оставшиеся особи производятся обычным способом: скрещиванием и мутацией. Благодаря элитизму, лучшие результаты алгоритма не теряются в результате случайных операций, что ускоряет сходимость. Другое реализованное дополнение — раз в заданное число поколений популяция каждого острова с некоторой вероятностью уничтожается, и заменяется случайно созданными особями. Этот процесс может стимулировать дальнейшую эволюцию, если та остановилась. Для того чтобы лучшие решения, достигнутые к этому моменту алгоритмом, при этом не терялись, элита последнего поколения включается в новую популяцию острова.

## **3. Результаты эксперимента**

Реализованный алгоритм был испытан на практике. Приведенный на рис. 2 график построен «Виртуальной лабораторией», он описывает произошедший процесс эволюции приблизительно до трехсотого поколения. Дальнейшая работа программы в этом конкретном запуске не привела к улучшению результата.

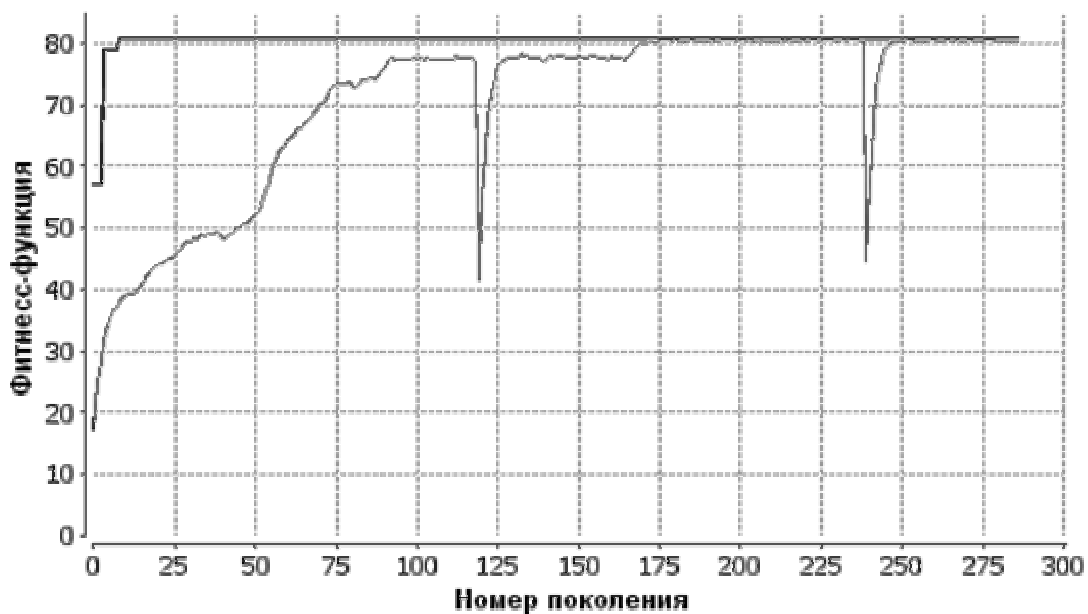


Рис. 2. Среднее и максимальное по популяции значения фитнес-функции

На графике верхняя линия отражает максимальные значения фитнес-функции в поколении, а нижняя — средние. Наибольшее значение функции приспособленности составило 81,02. В качестве примера рассмотрим одну из особей, получивших такую оценку. Достигнутый ею результат — съедено 82 единицы еды за 196 шагов. На рис. 3 приведен граф переходов соответствующего автомата.

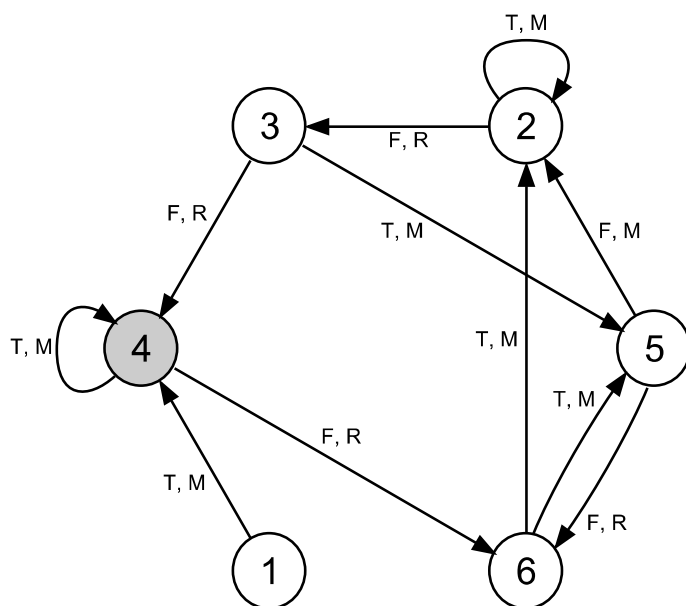


Рис. 3. Граф переходов автомата — одной из лучших особей

Состояние номер четыре является начальным. Соответствующая ему вершина графа выделена серым цветом. Подписи ребер составлены из двух символов. Первый обозначает условие, по которому производится переход: наличие еды перед муравьем описывается буквой «Т», а ее отсутствие — буквой «F». Второй символ описывает действие, совершаемое при данном переходе автомата. «М» — движение вперед, «R» — поворот вправо. Третье возможное действие, поворот влево «L», не оказалось использованным ни в одном случае.

## **Заключение**

В выполненной работе при помощи генетического алгоритма был автоматически построен конечный автомат, управляющий поведением агента в задаче «Умный муравей». Был достигнут результат в 82 съеденных единицы еды за 196 шагов. Хотя полученный автомат имеет шесть состояний, одно из них недостижимо. Это позволяет сравнивать его с известными результатами построения автоматов с пятью состояниями. Так, полный перебор позволяет получить автомат, под управлением которого муравей съедает 83 единицы еды [6]. Кроме того, известен построенный вручную автомат из пяти состояний, с которым муравей достигает результата в 81 единицу [2]. Отсюда можно заключить, что генетические алгоритмы можно считать эффективным способом построения управляющих автоматов в задаче «Умный муравей».



## Библиография

1. *Давыдов А.А., Соколов Д.О., Царев Ф.Н., Шалыто А.А.* Виртуальная лаборатория обучения генетическому программированию для генерации управляющих конечных автоматов. [http://is.ifmo.ru/works/2\\_93\\_davidov\\_sokolov.pdf](http://is.ifmo.ru/works/2_93_davidov_sokolov.pdf)
2. *Бедный Ю. Д., Шалыто А. А.* Применение генетических алгоритмов для построения конечных автоматов в задаче «Умный муравей». [http://is.ifmo.ru/works/\\_ant.pdf](http://is.ifmo.ru/works/_ant.pdf)
3. *Поликарпова Н. И., Шалыто А. А.* Автоматное программирование. СПб.: Питер, 2010.
4. *Яминов Б.* Генетические алгоритмы. <http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005>
5. Технология генетического программирования для генерации автоматов управления системами со сложным поведением. Промежуточный отчет по I этапу «Выбор направления исследований и базовых компонентов», с. 47–51
6. [http://is.ifmo.ru/genalg/2007\\_01\\_report-genetic.pdf](http://is.ifmo.ru/genalg/2007_01_report-genetic.pdf)
7. *Царев Ф. Н., Шалыто А. А.* О построении автоматов с минимальным числом состояний для задачи об «умном муравье» / Сборник докладов X международной конференции по мягким вычислениям и измерениям. СПбГЭТУ «ЛЭТИ» Т.2, 2007, с. 88–91. [http://is.ifmo.ru/download/ant\\_ga\\_min\\_number\\_of\\_state.pdf](http://is.ifmo.ru/download/ant_ga_min_number_of_state.pdf)