

Санкт-Петербургский государственный университет информационных  
технологий, механики и оптики

Факультет информационных технологий и программирования

Кафедра «Компьютерные технологии»

Р. Б. Галинский

**Отчет по лабораторной работе  
«Использование генетических  
алгоритмов для построения  
управляющих автоматов»**

Вариант № 23

Санкт-Петербург

2009

## Оглавление

Введение.....	3
1. Постановка задачи.....	4
1.1. Автомат Мили .....	4
1.2. Задача «Умный муравей-3».....	4
2. Реализация .....	4
2.1. Представление автоматов в программе.....	5
2.2. Метод скрещивания .....	5
2.3. Метод мутации .....	5
2.4. Метод генерации очередного поколения.....	5
2.5. Способ вычисления функции приспособленности .....	6
3. Результаты работы .....	6
3.1. График максимального значения функции приспособленности.....	6
3.2. График среднего значения функции приспособленности .....	7
3.3. Таблица переходов полученного автомата.....	8
Конфигурационные файлы.....	9
Заключение .....	9
Источники.....	9

## **Введение**

Цель лабораторной работы – применение генетических алгоритмов для построения конечных автоматов. В качестве примера рассматривается построение конечного автомата Мили, решающего задачу «Умный муравей-3». При выполнении лабораторной работы использовалась программа «Виртуальная лаборатория» [1], написанная студентами кафедры «Компьютерные технологии» СПбГУ ИТМО и позволяющая реализовывать генетические алгоритмы и особи для них в виде подключаемых модулей.

## 1. Постановка задачи

Построить с помощью генетических алгоритмов конечный автомат Мили, решающий задачу «Умный муравей-3». Использовать представление автоматов с помощью графов переходов и сокращенных таблиц. Способ скрещивания выбрать самостоятельно. Использовать клеточный генетический алгоритм и метод «рулетки» для генерации очередного поколения.

### 1.1. Автомат Мили

Автомат типа Мили – конечный автомат, который генерирует выходные действия в зависимости от текущего состояния и входного воздействия. Пример представления автомата Мили в виде диаграммы переходов приведен на рисунке (рис. 1).

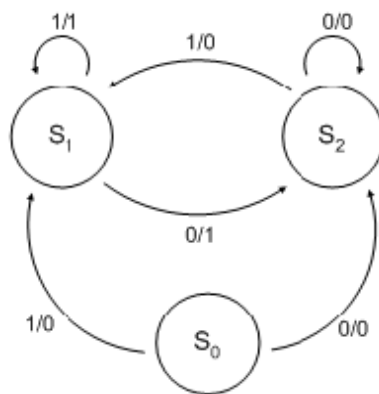


Рис. 1. Автомат Мили

Над каждой дугой расположена пара значений – входное действие и выходное действие. При этом выходное действие зависит от состояния автомата и от входного воздействия.

### 1.2. Задача «Умный муравей-3»

Игра происходит на поверхности тора размером 32x32 клетки. В некоторых клетках находится еда. Муравей начинает движение из клетки, помеченной «Start».

За каждый ход, муравей может:

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

Игра длится 200 ходов. Цель игры – создать муравья «с минимальным числом состояний», который за минимальное число ходов ест как можно больше яблок.

## 2. Реализация

Виртуальная лаборатория состоит из ядра и подключаемых модулей. Для решения поставленной задачи требуется создать два модуля. Первый модуль – модуль генетического алгоритма, использующий клеточный генетический алгоритм и метод рулетки для генерации очередного поколения. Второй модуль должен реализовывать «особь» – конечный автомат,

решающий задачу «Умный муравей». Для «особи» необходимо реализовать операции мутации и скрещивания.

### 2.1. Представление автоматов в программе

В программе автомат представлен в виде набора состояний. Для каждого состояния хранится таблица соответствий вида:

$$[\text{входное воздействие} > (\text{действие} + \text{новое состояние})],$$

определяющая поведение автомата. Также хранится номер начального состояния. Автомат представлен в виде сокращенных таблиц, каждая из которых является полной для одного состояния. Число элементов таблицы ограничивается некоторой константой, задаваемой в конфигурационном файле.

### 2.2. Метод скрещивания

Функция скрещивания получает на вход две особи и выдает также две особи. При этом используются сокращенные таблицы первого и второго родителей. Совпадающие строки этих таблиц случайным образом распределяются между детьми. Остальные строки дописываются уже к определенному ребенку, но не случайным образом.

### 2.3. Метод мутации

При мутации с некоторой вероятностью, стартовым состоянием становится случайное состояние автомата. Затем изменяется либо выходное воздействие, связанное с этим состоянием, либо переход по какому-либо входному воздействию.

### 2.4. Метод генерации очередного поколения

Для генерации очередного поколения используется клеточный *генетический алгоритм* (*Cellular Genetic Algorithms*) и метод рулетки. *Cellular Genetic Algorithms* – модель параллельных генетических алгоритмов.

Пусть дано 2500 процессов, расположенных на сетке размером 50×50 ячеек, как показано на рис. 2 (левая сторона замыкается с правой, а верхняя с нижней, в результате получается тор).

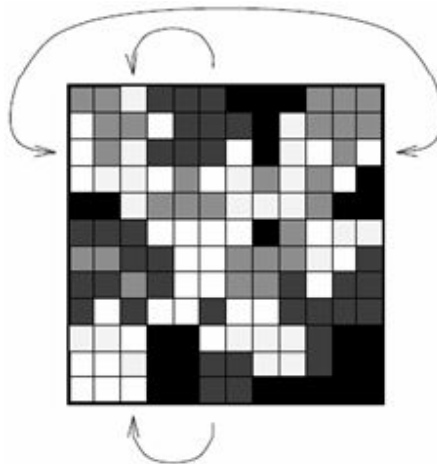


Рис. 2. Модель клеточного генетического алгоритма

Каждый процесс может взаимодействовать только с четырьмя своими соседями (сверху, снизу, слева, справа). В каждой ячейке находится ровно одна особь. Каждый процесс будет выбирать лучшую особь среди своих соседей, скрещивать с ней особь из своей ячейки и одного полученного ребенка помещать в свою ячейку вместо родителя.

Метод рулетки – способ отбора новых особей. Пусть особи располагаются на колесе рулетки, так что размер сектора каждой особи пропорционален ее приспособленности. Изначально промежуточная популяция пуста. Запуская рулетку  $N$  раз, выберем требуемое число особей для записи в промежуточную популяцию. Ни одна выбранная особь не удаляется с рулетки. Промежуточная популяция – это набор особей, которые получили право размножаться. После отбора особи промежуточной популяции случайным образом разбиваются на пары. Каждая из них с вероятностью один процент скрещивается методом описанным выше.

### 2.5. Способ вычисления функции приспособленности

Функция приспособленности вычисляется следующим образом: полученный автомат используется для управления муравьем, подсчитывается число съеденных яблок и выдается в качестве ответа.

## 3. Результаты работы

В результате работы рассматриваемого генетического алгоритма был получен автомат, который решает задачу «Умный муравей-3». Автомат содержит семь состояний, значение функции приспособленности равно 23.15.

### 3.1. График максимального значения функции приспособленности

На рис. 3 приведен график зависимости максимального значения функции приспособленности среди особей поколения.

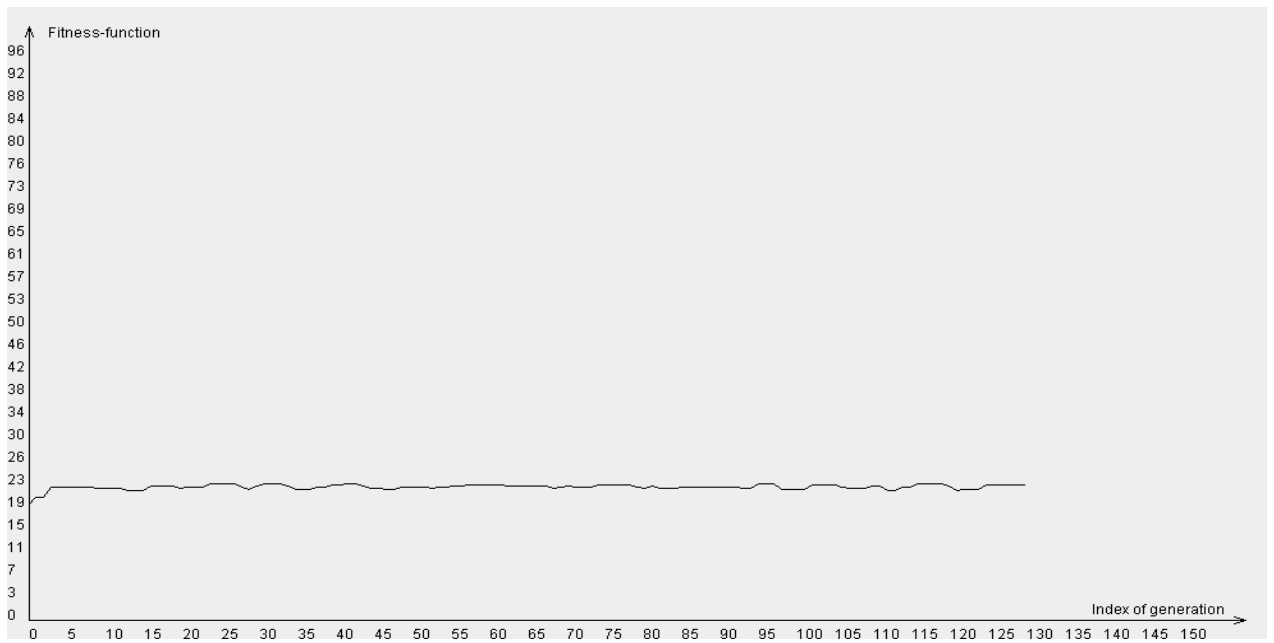


Рис. 3. График зависимости максимального значения функции приспособленности среди особей поколения

### 3.2. График среднего значения функции приспособленности

На рис. 4 приведен график зависимости среднего значения функции приспособленности среди особей поколения

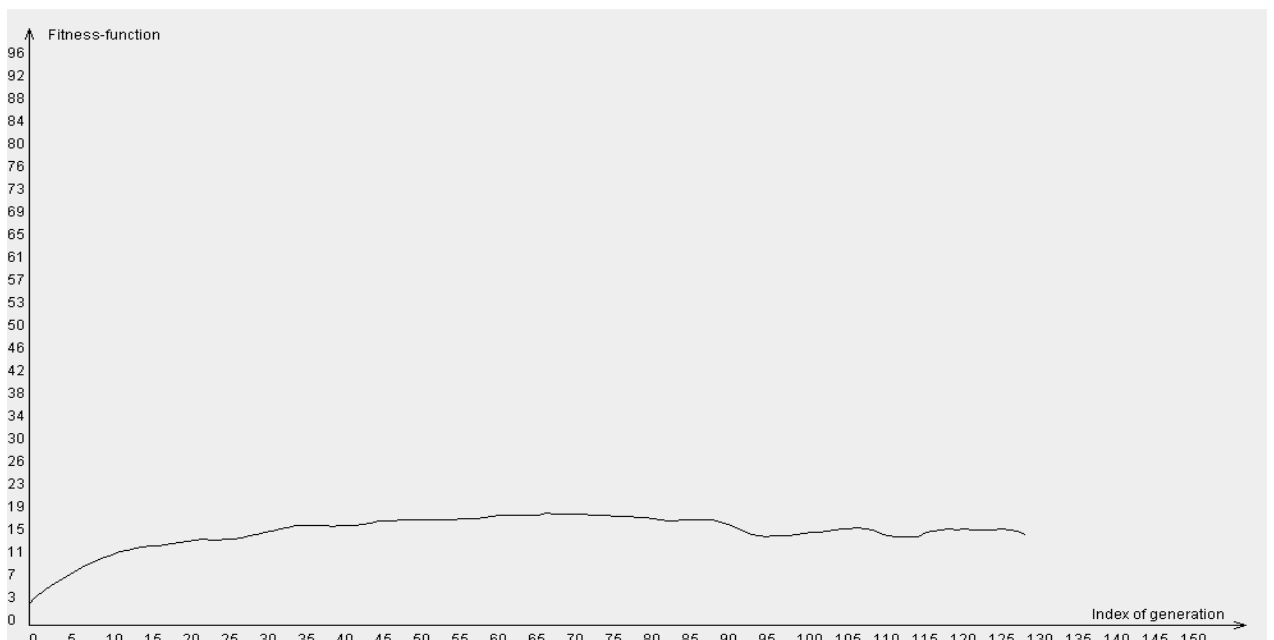


Рис. 3. График зависимости среднего значения функции приспособленности среди особей поколения

### 3.3. Таблица переходов полученного автомата

State count: 7. Initial state: 3. Fitness: 23.15.

STATE(0)	STATE(1)	STATE(2)	STATE(3)	STATE(4)	STATE(5)	STATE(6)
139(M, 3)	93(M, 5)	140(M, 3)	89(L, 1)	234(L, 3)	92(L, 6)	46(L, 3)
233(M, 5)	138(R, 3)	139(M, 6)	88(L, 1)	139(R, 6)	45(L, 5)	140(L, 4)
42(M, 6)	137(M, 3)	185(R, 1)	182(R, 3)	86(L, 2)	185(M, 6)	234(M, 3)
87(L, 4)	88(R, 0)	232(R, 0)	229(L, 6)	85(M, 3)	184(R, 1)	45(M, 6)
227(M, 3)	135(M, 3)	38(M, 3)	179(M, 1)	179(M, 2)	135(M, 1)	184(L, 2)
85(L, 1)	182(R, 4)	131(R, 0)	130(M, 5)	226(R, 1)	84(L, 0)	231(L, 1)
223(M, 0)	181(L, 4)	36(R, 3)	129(M, 6)	84(R, 0)	82(M, 1)	43(L, 4)
222(M, 3)	133(M, 0)	83(R, 2)	80(R, 1)	35(L, 2)	223(L, 1)	227(R, 1)
174(R, 2)	83(L, 3)	224(R, 4)	32(M, 6)	81(L, 3)	81(L, 4)	224(M, 6)
173(R, 2)	127(L, 5)	80(R, 5)	31(R, 3)	128(M, 0)	175(R, 2)	220(M, 4)
31(M, 2)	32(M, 1)	78(M, 1)	78(M, 6)	173(L, 0)	222(M, 1)	31(L, 6)
170(R, 0)	120(L, 6)	171(L, 5)	30(L, 0)	31(L, 1)	30(L, 6)	125(R, 2)
74(L, 2)	119(M, 5)	76(L, 5)	121(R, 3)	125(R, 5)	76(M, 0)	170(R, 4)
121(M, 1)	24(M, 0)	27(M, 4)	213(R, 6)	29(L, 3)	167(R, 0)	75(M, 3)
215(M, 1)	70(R, 0)	72(L, 1)	118(R, 3)	76(R, 4)	72(M, 5)	27(R, 3)
162(M, 1)	210(M, 1)	165(M, 2)	212(L, 2)	28(L, 5)	165(L, 3)	121(M, 5)
114(M, 4)	115(R, 1)	116(R, 3)	164(L, 2)	26(R, 1)	162(M, 3)	212(L, 0)
16(M, 3)	18(M, 6)	22(R, 5)	117(M, 5)	209(R, 1)	19(R, 2)	163(M, 6)
251(L, 6)	64(L, 2)	208(R, 2)	69(L, 6)	208(M, 0)	251(R, 2)	162(M, 6)
156(M, 5)	251(L, 4)	254(L, 5)	68(L, 6)	205(L, 3)	203(L, 0)	68(L, 2)
14(M, 3)	110(L, 6)	19(R, 1)	65(L, 5)	110(L, 4)	109(L, 5)	59(L, 6)
13(M, 2)	61(R, 6)	160(R, 1)	252(L, 5)	156(M, 3)	156(M, 0)	153(L, 4)
200(M, 4)	13(M, 0)	253(M, 0)	64(L, 2)	249(R, 2)	108(L, 6)	244(M, 2)
105(R, 1)	154(R, 0)	251(R, 2)	251(L, 6)	247(R, 4)	247(R, 5)	9(R, 4)
104(L, 2)	59(R, 1)	248(R, 2)	61(M, 3)	104(R, 1)	200(R, 0)	8(L, 6)
151(R, 5)	153(L, 5)	13(R, 5)	108(L, 2)	9(L, 5)	11(R, 2)	52(M, 4)
8(M, 5)	105(R, 6)	104(R, 2)	200(R, 1)	103(L, 3)	103(M, 2)	4(R, 0)
242(M, 4)	197(L, 3)	100(M, 0)	105(L, 2)	102(M, 6)	146(L, 4)	97(L, 5)
240(M, 0)	146(M, 2)	238(R, 5)	149(L, 3)	53(L, 5)	3(M, 3)	238(L, 6)
4(M, 0)	52(L, 0)	2(L, 2)	239(R, 5)	50(R, 3)	190(M, 5)	191(R, 4)
239(R, 0)	97(R, 3)	1(M, 1)	4(L, 3)	190(M, 1)	237(R, 2)	237(L, 0)
143(M, 0)	2(R, 5)	141(M, 4)	97(M, 5)	236(M, 1)	95(L, 2)	1(L, 0)

Ячейки имеют вид *входное\_воздействие (действие, номер\_нового\_состояния)*. Следует понимать, что входное воздействие – восьмибитное число, характеризующее расположение еды в области видимости муравья.



## Конфигурационные файлы

Для сборки модулей виртуальной лаборатории был использован `ant` – скрипт, который находится в файле `build.xml`, и пакет `NetBeans 6.5`.

Модуль «особи»:

- `automaton.conf` – конфигурация задания;
- `Count.states` – число состояний автомата;
- `Count.attempts` – число попыток муравья;
- `Mu` – вероятность того, что еда есть в некоторой клетке;
- `Count.significantpredicates` – максимальное число значимых входных данных для автомата.

Настройки модуля генетического алгоритма совпадают с таковыми для стандартного клеточного алгоритма, реализованного в виртуальной лаборатории.

## Заключение

Результаты лабораторной работы показали, что используемые методы достаточно эффективны для построения автомата Мили, заданного сокращенными таблицами, который решает задачу «Умный муравей-3». Кроме того, как показывают графики функции приспособленности, достаточно неплохие результаты получены уже на первых поколениях особей, что говорит об эффективности представления автомата и клеточной модели генетического алгоритма. К недостаткам описанного варианта работы можно отнести большое время генерации очередного поколения. С другой стороны, получен автомат, который как раз наиболее быстро и эффективно выполняет задачу «Умный муравей-3».

## Источники

1. *Инструкция по созданию plugin'ов к виртуальной лаборатории.*  
[http://svn2.assembla.com/svn/not\\_instrumental\\_tool/docs/pdf/interface\\_manual.pdf](http://svn2.assembla.com/svn/not_instrumental_tool/docs/pdf/interface_manual.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](http://is.ifmo.ru/works/_ant.pdf)
3. *Яминов Б.* Генетические алгоритмы.  
<http://rain.ifmo.ru/cat/view.php/theory/unordered/genetic-2005>