

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

Д. Е. Родиков

**Анализ использования нескольких функций  
приспособленности для построения автоматов с  
помощью генетических алгоритмов на примере  
задачи «Умный муравей – 3»**

Санкт-Петербург  
2009

# Оглавление

Введение .....	3
1. Постановка задачи .....	4
1.1. Автомат Мили .....	4
1.2. Задача «Умный муравей 3» .....	4
2. Генетический алгоритм .....	5
2.1. Представление особи .....	5
2.2. Отбор особей и генерация нового поколения .....	5
2.3. Метод скрещивания особей .....	6
2.4. Мутация особей .....	6
2.5. Мутация поколения .....	6
2.6. Функции приспособленности .....	6
2.7. Совместное применение функций приспособленности .....	7
3. Построенный автомат .....	7
3.1. Результаты .....	7
3.2. Сравнение с известными результатами .....	8
3.3. Динамика изменения приспособленности лучшей особи .....	8
Заключение .....	9
Источники .....	<a href="#">10</a>

## **Введение**

Одна из основных проблем генетических алгоритмов – попадание в локальные экстремумы. Одним из перспективных путей решения данной проблемы, является использование нескольких функций приспособленности, вместо одной. В данной работе изучается применение нескольких функций приспособленности для построения автоматов с помощью генетических алгоритмов на примере задачи «Умный муравей – 3». Результатом работы является сравнительная характеристика методов использующих различное число функций приспособленности. Тестирующая программа написана на языке программирования *Java*, тестирование проводилось под операционной системой *Windows XP*.

# 1. Постановка задачи

Задача данной лабораторной работы – построить автомат Мили, решающий задачу об «Умном муравье – 3». Оптимальность заключается в том, что при заданном числе состояний автомата, муравей, управляемый данным автоматом, должен съесть как можно больше еды за 200 шагов.

## 1.1. Автомат Мили

Автомат Мили – это конечный автомат, генерирующий выходные воздействия в зависимости от текущего состояния и значения входного воздействия. Пример диаграммы переходов автомата Мили приведен на рис. 1.

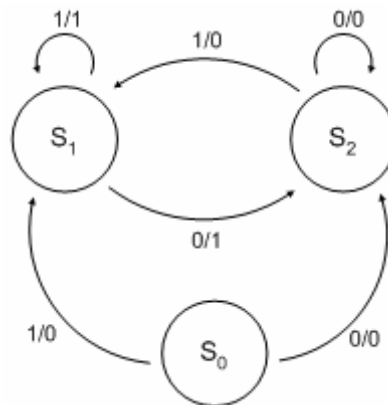


Рис. 1. Пример автомата Мили

Как видно из приведенной диаграммы, над каждой дугой расположена пара значений – входное и выходное воздействие. При этом выходное действие зависит не только от состояния, в котором находится автомат, но и от входного воздействия.

## 1.2. Задача «Умный муравей – 3»

Задача «Умный муравей – 3» является обобщением задачи «Умный муравей» [1]. Муравей видит не одну клетку как раньше, а восемь. В случае наперед заданного поля эта модификация упрощает задачу – при таком же числе состояний автомата, легче найти особь, съедающую все яблоки за 89 ходов.



Рис. 2. Видимые муравью клетки

Рассмотрим обобщение задачи на случайные карты, когда еда в каждой клетке существует с заданным параметром  $\mu$ . При этом необходимо найти такую стратегию поведения муравья, при которой математическое ожидание числа съеденных яблок максимально. В дальнейшем будем называть эту задачу задачей об «Умном муравье – 3»[1].

За один ход муравей может определить, есть ли в клетке перед ним яблоко, и выполнить одно из следующих действий:

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

Цель работы – создать муравья с фиксированным числом состояний, который за 200 шагов съест максимальное число яблок.

## 2. Генетический алгоритм

Для задачи поиска оптимального автомата, управляющего муравьем, применяется генетический алгоритм [2]. Работа генетического алгоритма состоит из нескольких фаз. В начале происходит генерация первого поколения особей. Далее алгоритм начинает выполнение итеративного процесса – на каждой итерации алгоритм строит следующее поколение из предыдущего. При этом применяются операции:

- отбор – из предыдущего поколения выбирается часть особей, для сравнения особей между собой алгоритм использует функцию приспособленности;
- скрещивание – по двум особям-родителям создаются две новые особи;
- мутация – случайным образом изменяется строение особи.

Опишем более подробно алгоритм, применяемый в данной работе.

### 2.1. Представление особи

Особями в данном алгоритме являются автоматы Мили с восемью состояниями. Для представления автомата в коде программы используется массив из деревьев, каждое из которых задает состояние автомата. Каждое дерево в каждой вершине содержит номер клетки, проверяемой на наличие еды (от 0 до 7). Из вершины возможны два перехода в зависимости от наличия еды в клетке. Каждый лист дерева содержит информацию о выполняемом переходе: выходное действие (поворот налево, поворот направо или движение вперед) и номер состояния, в которое перейдет автомат. В задаче об «Умном муравье – 3» входные воздействия образуют массив из восьми элементов, содержащий информацию о наличии еды в клетках, видимых муравью. Использование деревьев для представления состояний автомата рассмотрено в работе [3].

### 2.2. Отбор особей и генерация нового поколения

Начальное поколение генерируется из особей, созданных случайным образом. Оно разбито на несколько групп особей. В каждой группе используется своя функция приспособленности.

Рассмотрим процесс генерации нового поколения одной. Для каждой группы особей, использующих общую функцию приспособленности, выполняются следующие действия.

1. Реализуется метод отбора, называемого «элитизм». Некоторая часть лучших особей ( $eliteNumber \sim 5\%$ ) копируется из предыдущего поколения без изменений.

2. Реализуется скрещивание. Оставшаяся часть популяции генерируется с помощью скрещивания особей предыдущего поколения. Для скрещивания используется часть лучших особей предыдущего поколения (`crossNumber` ~ 25%).
3. С некоторой вероятностью (`MUTATE_POSSIBILITY` ~ 10%) производится мутация особей, не отобранных в пункте 1.

### 2.3. Метод скрещивания особей

При скрещивании алгоритм порождает две новые особи из двух особей-родителей. Состояния автоматов нумеруются числами от 0 до  $N - 1$  (у всех автоматов одинаковое число состояний). При процедуре скрещивания выполняются шаги:

1. У каждого автомата особей-родителей произвольно выбирается состояние.
2. В каждом состоянии произвольно выбирается переход.
3. Особи-родители обмениваются выбранными переходами.
4. Полученные особи являются результатом скрещивания.

### 2.4. Мутация особей

Мутация особи-автомата заключается в случайном выборе состояния и случайном изменении одного из переходов из данного состояния. Также с вероятностью 0.2 изменяется начальное состояние автомата.

### 2.5. Мутация поколения

Каждые 67 поколений производится обновление особей. Все особи, кроме лучшей особи каждого поколения, генерируются заново. Этот метод используется для выхода из локальных экстремумов, вероятность попадания в которые достаточно велика к данному этапу.

### 2.6. Функции приспособленности

Генетический алгоритм использует четыре функции приспособленности.

1. Стандартная функция приспособленности. Вычисляется по формуле

$$Fitness = Apples + (200 - Steps)/200,$$

где *Apples* – число яблок, съедаемых муравьем за 200 шагов, *Steps* – номер шага, на котором муравей съедает последнее яблоко. Вычисления производятся 10 раз на различных полях с яблоками. Среднее значение и считается значением функции приспособленности.

2. Половина функции приспособленности определяется числом съедаемых муравьем яблок, вторая половина – процентом использованных состояний:

$$Fitness = Apples / 2 + 100 * usedStates / statesNumber,$$

где *usedStates* – число использованных в процессе работы состояний автомата, *statesNumber* – число состояний автомата.

3. Функция приспособления определяется числом состояний, достижимых из начального состояния автомата. Она позволяет отсекаать автоматы, не использующие все свои состояния.

4. Функция приспособленности учитывает число сделанных поворотов (чем меньше, тем лучше). Позволяет максимизировать число клеток, посещенных муравьем. Вычисляется по формуле:

$$Fitness = Turns + Apples + (200 - Steps)/200,$$

где Turns – число совершенных поворотов за 200 шагов.

## 2.7. Совместное применение функций приспособленности

Набор применяемых функций приспособленности обязан содержать первую функцию, так как она требуется для итоговой оценки получаемых особей. Для реализации совместного использования функций приспособленности каждое поколение делится на группы. Число групп равно числу применяемых функций приспособленности. В каждой группе используется своя функция и реализуется генетический алгоритм, приведенный в разд. 2.2 – 2.5. Каждые семь поколений делается обмен особей между группами. Перед обменом особи сортируются в своих группах по значению функции приспособленности. Особи, находящиеся на четных позициях в отсортированном списке особей каждой группы, меняют свою группу. Номер новой группы особи равен номеру старой группы, увеличенному на единицу. Особи последней группы переходят в первую. Затем продолжается независимая работа генетического алгоритма в каждой из групп.

## 3. Построенный автомат

В результате работы генетического алгоритма с использованием функций приспособленности 1, 3, 4 был получен автомат Мили, который решает задачу об «Умном муравье–3». Автомат состоит из восьми состояний, значение стандартной функции приспособленности (функции 1) равно 20.0. Муравей, управляемый данным автоматом, съедает в среднем 20 яблок за 200 шагов. Вероятность появления яблока в клетке поля равна 4%. Число особей в популяции 1500, число итераций генерации нового поколения 100.

### 3.1. Результаты

Генетический алгоритм был протестирован для случаев использования различного числа функций приспособленности и различной вероятности появления еды в клетках поля. Результаты приведены в табл. 1.

Таблица 1. Результаты экспериментов

Номера использованных функций приспособленности / вероятность появления еды	0,01	0,02	0,03	0,04
1, 2, 3, 4	4,41	8,39	12,1	17,6
1, 3, 4	3,71	6,90	14,7	19,7
1, 2	3,16	7,51	13,6	19,4
1, 3	3,53	9,00	13,9	16,9
1, 4	3,67	8,27	14,1	19,7
1	3,66	8,59	13,0	15,9

В первой колонке указаны номера функций приспособленности, использованных алгоритмом, нумерация этих функций согласно их описанию в разд. 2.6. В первой строке указана вероятность появления еды в клетке поля. В остальных клетках таблицы указаны лучшие из полученных значений стандартной функции (функции 1) приспособленности. Использовались следующие параметры генетического алгоритма: число особей в популяции 300, число поколений 100. Последнее измерение функции приспособленности проводилось на случайном наборе из 1000 карт.

Анализ результатов показал, что нельзя однозначно выделить несколько функций приспособления, совместное использование которых будет давать однозначно лучший результат, чем использование одной функции приспособленности.

С другой стороны, есть комбинации функций, на одних тестах почти не уступающие использованию одной функции приспособленности, а на других – заметно превосходящие. Например, комбинация функций 1, 4 при вероятности появления еды 0,01 – 0,02 дает одинаковый результат с результатом, полученным при использовании первой функции приспособленности, а при вероятности появления еды 0,03 – 0,04 заметно превосходит его. Так же неплохо выглядит и комбинация трех функций 1, 3, 4, так как единственный неудачный результат получен при вероятности появления еды 0,02. Лучшие из полученных результатов превосходят результаты, приведенные в работе [3], что подтверждает эффективность метода.

### 3.2. Сравнение с известными результатами

В работе [3] приведены результаты экспериментов с применением различных способов представления автоматов для решения задачи «Умный муравей-3». Использовались такие способы как битовые строки, таблицы переходов, двоичные деревья. Тестирование проводилось на полях с вероятностями появления еды 0,01, 0,02, 0,03, 0,04. Лучшие из результатов для автоматов с восемью состояниями таковы: 3,74, 7,95, 13,77, 18,60 соответственно.

В табл. 2 лучшими результатами являются 4,41, 9,00, 14,7, 19,7 для вероятностей появления еды 0,01, 0,02, 0,03, 0,04 соответственно.

Таблица 2. Результаты сравнения

Вероятность появления еды в клетке поля	Работа [3]	Данная работа	Разница, %
0,01	3,74	4,41	18
0,02	7,95	9,00	13
0,03	13,8	14,7	6,5
0,04	18,6	19,7	6,0

### 3.3. Динамика изменения приспособленности лучшей особи

Интересно проследить динамику изменения функции приспособленности лучшей особи. Использовались следующие параметры генетического алгоритма: вероятность еды в клетке – 0.04 (4%), число особей в популяции – 1500, число поколений – 100. Измерения функции приспособленности после каждых десяти поколений производились на случайном наборе из 1000 карт. Результаты приведены в табл. 3.



Таблица 3. Результаты экспериментов

Номера использованных функций приспособленности / номер поколения	10	20	30	40	50	60	70	80	90	100
1,3,4	15,6	18,2	18,8	19,5	19,5	19,5	19,7	19,9	20,0	20,0
1,3	13,2	15,5	16,7	17,7	17,8	18,2	19,5	19,5	19,8	20,0
1	15,5	17,5	17,9	18,2	18,2	18,5	18,3	18,3	18,4	18,7

Из табл. 3 видно, что результаты, полученные при использовании нескольких функций приспособленности, лучше, чем при использовании одной первой (20,0 против 18,7).

Также можно установить, что значение лучшей приспособленности при использовании одной функции приспособленности между 40 и 90 поколениями стояло практически на месте, а иногда и уменьшалось. Это можно объяснить тем, что при пересчете функции приспособленности на 1000 карт не находилось хороших особей других типов (для других типов карт), а показатели прежних лучших особей уменьшались из-за пересчета. Таким образом, можно предположить, что был найден некоторый локальный экстремум, практически без возможности выйти за его пределы.

С другой стороны, при использовании нескольких функций приспособления лучшая приспособленность только улучшалась, причем несколько быстрее, чем при использовании одной функции. Видимо, метод использования нескольких функций приспособления позволяет не попадать в один экстремум большинству хороших особей. Таким образом, поиск лучших особей идет по более широкой области автоматов, но в то же время дополнительные функции приспособленности позволяют отсекал заведомо неудачные автоматы.

Таким образом, динамика изменения лучшей приспособленности лучше при использовании нескольких функций приспособленности.

## Заключение

Целью работы была оценка эффективности использования нескольких функций приспособленности при генерации автоматов. Она была оценена по двум критериям: лучшая приспособленность полученной особи и динамика изменения лучшей приспособленности от поколения к поколению. В целом результаты при использовании нескольких функций приспособленности были лучше, чем при применении одной. Предложенный метод направлен в первую очередь на предотвращение сужения области поиска лучшей особи до некоторого локального экстремума. Определенные трудности представляет создание различных содержательных функций приспособленности. В сравнении с ранее предложенными методами, метод использования нескольких функций приспособленности дал лучший результат.

Весьма перспективно выглядит использование предложенного метода на задачах, в которых есть естественная возможность создания нескольких функций приспособленности.

## Источники

1. Бедный Ю. Д., Шалыто А. А. Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей»  
<http://is.ifmo.ru/works/ant.pdf>
2. Яминов Б. Генетические алгоритмы  
<http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005>
3. Данилов В. Р. Технология генетического программирования для генерации автоматов управления системами со сложным поведением  
[http://is.ifmo.ru/download/danilov\\_bachelor.pdf](http://is.ifmo.ru/download/danilov_bachelor.pdf)