

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

Чуприков П. С.

**Отчет по лабораторной работе  
«Построение управляющих автоматов с  
помощью эволюционных стратегий»**

Вариант № 13

Санкт-Петербург  
2011

# Содержание

<b>ВВЕДЕНИЕ</b>	<b>2</b>
<b>1. Постановка задачи</b>	<b>2</b>
1.1. Задача «Умный муравей» . . . . .	2
1.2. Особи. Функция приспособленности . . . . .	3
1.3. Автомат Мили . . . . .	3
<b>2. Реализация</b>	<b>4</b>
2.1. Эволюционная стратегия . . . . .	5
2.2. Детали реализации . . . . .	5
2.3. Особи . . . . .	5
2.4. Мутация . . . . .	6
2.5. Отображение результатов . . . . .	6
<b>3. Результаты</b>	<b>6</b>
3.1. Графики . . . . .	7
<b>ЗАКЛЮЧЕНИЕ</b>	<b>9</b>
<b>РЕСУРСЫ</b>	<b>10</b>
<b>ЛИТЕРАТУРА</b>	<b>10</b>

# ВВЕДЕНИЕ

В рамках лабораторной работы требуется исследовать один из параметров эволюционных стратегий, а именно максимальное допустимое количество поколений, в течении которых может не изменяться функция приспособленности. В качестве особи предлагается использовать управляющий автомат Мили, решающий задачу «Умный муравей».

Лабораторная работа выполнялась с использованием библиотеки «Watchmaker» 1, для которой была написана реализация эволюционной стратегии. Для визуализации результата использовались модули виртуальной лаборатории «3genetic» 2.

## 1. Постановка задачи

В лабораторной работе требуется исследовать нижеописанный подход, используемый при практическом использовании эволюционных стратегий.

Экспериментально замечено, что эволюционные стратегии очень быстро «сходятся», т. е. особи в поколении быстро достигают локального максимума, и во время дальнейшей эволюции функция приспособленности перестает расти. В связи с этим, если в течении определенного числа  $K$  поколений лучшее значение функции приспособленности не изменяется, то алгоритм перезапускается. Нужно оценить эффективность работы эволюционной стратегии для разных значениях  $K$  и выбрать из них наилучшее.

### 1.1. Задача «Умный муравей»

Дано поле размером  $32 \times 32$ , расположенное на поверхности тора (рис. 1). В некоторых клетках расположена еда, расположение которой заранее известно. Муравей управляется автоматом и начинает свое движение в клетке, помеченной как «Start». За один ход муравей может выполнить следующие действия:

- пойти вперед (съесть еду, если она там есть);
- повернуть направо;
- повернуть налево;
- ничего не делать.

Муравей может сделать не более чем 200 ходов. Также муравей может сообщить управляющему автомату: есть перед ним еда или нет. Задача съесть максимальное количество еды за минимальное число шагов.

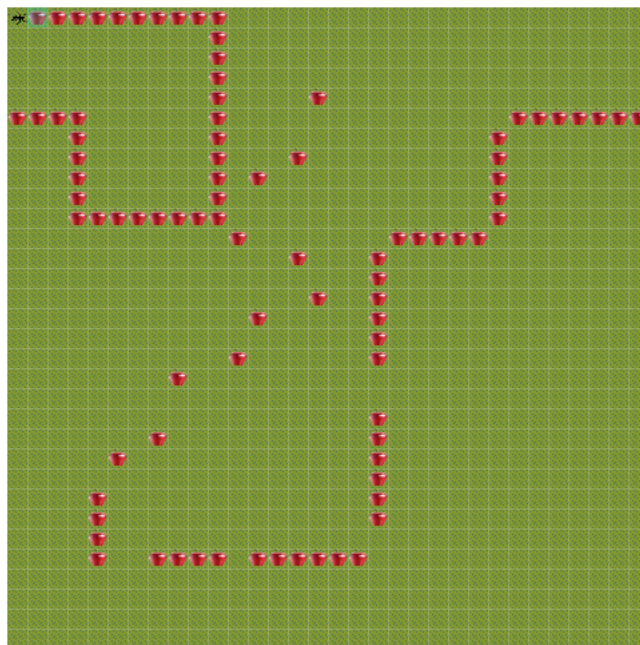


Рис. 1. Поле для задачи «Умный муравей». Муравей находится на стартовой позиции.

## 1.2. Особи. Функция приспособленности

В качестве особей для эволюционной стратегии используются автоматы Мили, управляющие муравьем. Функция приспособленности (*Fitness*) вычисляется следующим образом:

$$Fitness = F + \frac{200 - S}{200}$$

где  $F$  — число съеденной еды,  $S$  — число сделанных шагов.

## 1.3. Автомат Мили

Автомат Мили — это конечный автомат, выходные воздействия которого зависят и от входных воздействий, и от состояния (подробнее в [1]). Заметим, что в данной задаче входными воздействиями являются {«впереди

есть еда», «впереди нет еды»}, а выходными воздействиями — множество действий, выполняемых муравьем.

В памяти компьютера автомат Мили хранится в виде графа переходов, который в свою очередь реализуется с помощью списков смежности. Графически граф переходов в отчете будет изображаться как на рис. 2.

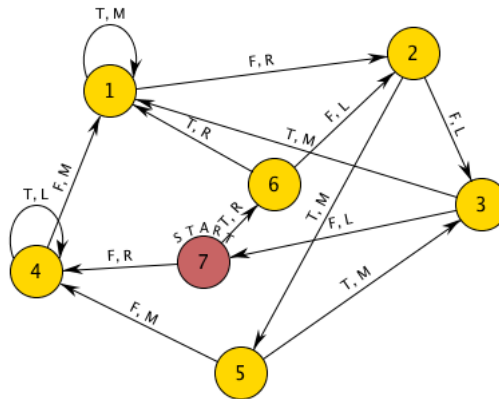


Рис. 2. Автомат мили

Пометку на ребре  $(C, A)$  следует понимать так: сначала указывается входное воздействие, при котором осуществляется переход по данному ребру, затем через запятую следует выходное воздействие.

Входные воздействия:

- T - перед муравьем есть еда;
- F - перед муравьем нет еды.

Выходные воздействия:

- M - сделать шаг вперед и съесть еду в клетке, если она (еда) там есть;
- L - повернуть налево;
- R - повернуть направо.

## 2. Реализация

Данная лабораторная работа представляет собой набор модулей для библиотеки «Watermark», реализованные на языке Java. Для графического оформления интерфейса использовалась библиотека «Swing» языка

Java. Графики результатов отрисовывались с помощью свободной библиотеки «JFreeChart».

## 2.1. Эволюционная стратегия

В задачах работы требовалось реализовать эволюционную стратегию, которую кратко можно описать следующим образом. Сначала генерируется случайным образом начальное поколение, затем на каждом шаге из поколения отбирается  $\lambda$  кандидатов для мутации. Над каждым кандидатом  $\mu/\lambda$  раз осуществляется мутация, формируя дочернее поколение. Далее в  $(\mu, \lambda)$  версии алгоритма следующим поколением становится дочернее, а в случае  $(\lambda + \mu)$  следующее поколение получается объединением дочернего и кандидатов.

В ходе практических экспериментов было выведено так называемое правило «одной пятой»: если более одной пятой детей лучше родителей, то нужно усилить мутацию, иначе — ослабить. Подробнее об эволюционных стратегиях можно узнать из [2] или [3].

## 2.2. Детали реализации

Были реализованы как  $(\mu, \lambda)$ , так и  $(\lambda + \mu)$  стратегии. Для поддержания разнообразия в поколении, селекция производилась сначала по лучшим, но различным особям, и только потом добирались лучшие из оставшихся. Правило *одной пятой* поддерживалось с помощью повышения (понижения) процента мутации особи.

Также был использован подход, перезапускающий алгоритм, если в течение длительного времени мутации не могли улучшить текущий максимум функции приспособленности. Стоит заметить, что именно данный подход характеризуется параметром  $K$ , который подлежит исследованию.

## 2.3. Особи

В лабораторной работе используется реализация автомата Мили из виртуальной лаборатории «3genetic». Визуализация особи осуществлялась с помощью модуля «sAnt», входящего в состав «3genetic» (с небольшими изменениями).

## 2.4. Мутация

Оператор мутации является модулем к «Watchmaker» и осуществляет мутацию автомата Мили с параметром  $Pr$ , характеризующим степень мутации автомата. Суть мутации следующая: алгоритм перебирает все возможные переходы автомата и с вероятностью  $Pr$  выполняет одно из следующих действий:

- меняет состояние, в которое ведет переход, на случайное;
- меняет на противоположное условие (входное воздействие) на переходе;
- меняет выходное воздействие на случайное.

С вероятностью, пропорциональной степени мутации, начальное состояние заменяется на случайное.

## 2.5. Отображение результатов

Возможные значения параметра, исследуемые в лабораторной работе:  $1 \leq K \leq 21$ . Для графического представления результатов выбран следующий ряд графиков:

1. График зависимости среднего значения времени выполнения одного запуска алгоритма от параметра  $K$ ;
2. График зависимости среднего значения функции приспособленности от параметра  $K$ ;
3. График зависимости лучшего значения функции приспособленности от времени выполнения алгоритма для различных значений  $K$ .

Последний график для увеличения стабильности усредняется по 10 запускам.

## 3. Результаты

Для проведения экспериментов была выбрана стратегия  $(\mu + \lambda)$ . Это позволяет сохранять наилучших отобранных особей на протяжении всей эволюции. Параметр  $\mu$  был выбран отличный от единицы, что вместе

с принудительным копированием особей с различной функцией приспособленности должно было дать разнообразие в популяции. Итак, эволюционная стратегия была настроена следующим образом:

$$\mu = 4;$$

$$\lambda = 20.$$

Такие параметры позволяют осуществлять достаточно широкий поиск и поддерживать различные наиболее приспособленные особи во время эволюции.

### 3.1. Графики

Усредненная статистика по 100 запускам: время работы — рис. 3, средняя приспособленность — рис. 4. Один запуск завершается, если в течении  $K$  поколений лучшее значение функции приспособленности не изменяется.

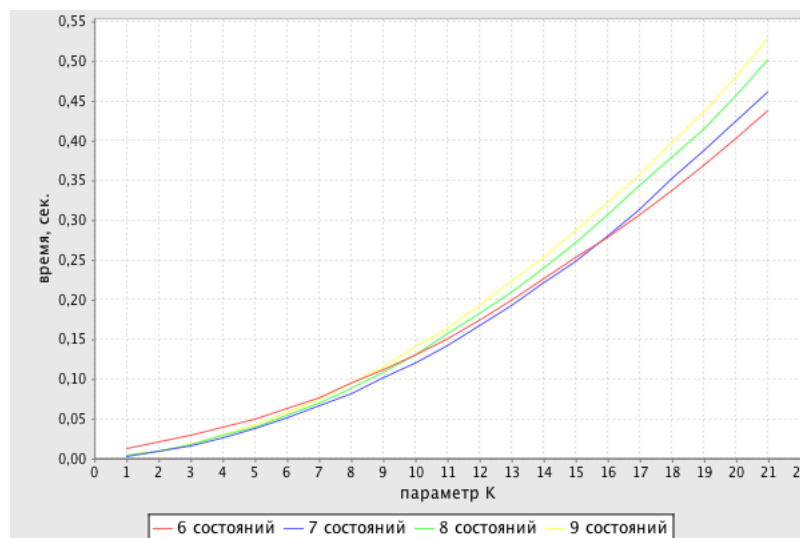


Рис. 3. График зависимости среднего значения времени работы алгоритма от параметра  $K$  при  $1 \leq K \leq 21$



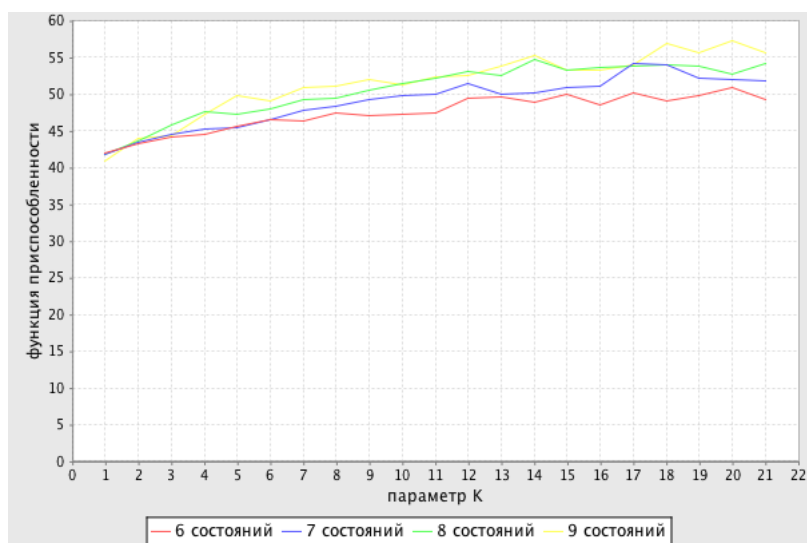


Рис. 4. График зависимости среднего значения функции приспособленности от параметра  $K$  при  $1 \leq K \leq 21$

Сравнение результатов работы алгоритма для различных  $K$  проводилось для семи состояний — рис. 5

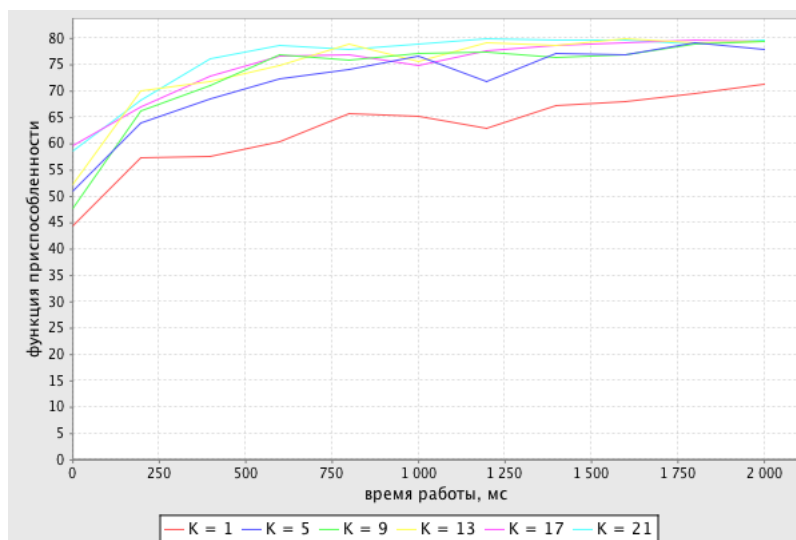


Рис. 5. График зависимости лучшего значения функции приспособленности от времени работы алгоритма при  $1 \leq K \leq 21$  и семи состояниях в автомате

## ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы был выявлен ряд особенностей поведения эволюционной стратегии в зависимости от параметра  $K$ , а именно:

- Среднее время одного запуска алгоритма растет *экспоненциально* с ростом параметра  $K$  (рис. 3). К примеру если при  $K = 10$  время работы едва превышало одну десятую секунды, то при  $K = 20$  алгоритм работал уже порядка полминуты.
- Наблюдается небольшой, но стабильный рост средней функции приспособленности с ростом параметра  $K$  (рис. 4). И хотя данное значение остается небольшим (порядка 50), это означает, что алгоритм с ростом  $K$  даже самые безнадежные случайные популяции «вытягивает» до все лучших автоматов, съедающих около половины еды.
- Важный с практической точки зрения результат — это то, что начиная с  $K \approx 10$  и времени  $t \approx 1.7$  сек. алгоритм стабильно находит автоматы из семи состояний, съедающие 80 единиц еды (рис. 5).

Практическая польза от применения параметра  $K$  была также подтверждена в ходе исследований.

## РЕСУРСЫ

1. Библиотека «Watchmaker» — <http://watchmaker.uncommons.org>.
2. Виртуальная лаборатория «3genetic» — <http://rain.ifmo.ru/~buzdalov/lab-2011/3genetic.zip>.
3. Исходные коды лабораторной работы и отчета — <https://bitbucket.org/pschuprikov/evolutionstrategy/src>

## ЛИТЕРАТУРА

- [1] Н. И. Поликарпова, А. А. Шалыто. Автоматное программирование. Санкт-Петербург, 2008.
- [2] Jason Brownlee. Clever Algorithms — Nature-Inspired Programming Recipes. First Edition. LuLu, January 2011.
- [3] Mitchel Melanie. An Introduction to Genetic Algorithms. MIT Press. Fifth printing, 1999.