

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
(МИНОБРНАУКИ)

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ  
И ОПТИКИ»  
(СПбГУ ИТМО)

УТВЕРЖДАЮ  
Ректор СПбГУ ИТМО,  
докт. техн. наук, профессор  
В. Н. Васильев

\_\_\_\_\_ 2008 г.

ПРОГРАММНОЕ СРЕДСТВО 3GENETIC

ОПИСАНИЕ ПРИМЕНЕНИЯ

ЛИСТ УТВЕРЖДЕНИЯ

7.190.00001-01 31 01-ЛУ

Декан факультета «Информационные  
технологии и программирование»  
докт. техн. наук, профессор  
\_\_\_\_\_ В. Г. Парфенов

Руководитель темы  
заведующий кафедрой «Технологии программирования»,  
докт. техн. наук, профессор  
\_\_\_\_\_ А. А. Шалыто

Имя, И. подл.	Подп. и дата
В зам. имя, И	
Имя, И дубл.	
Подп. и дата	

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
(МИНОБРНАУКИ)

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ  
И ОПТИКИ»  
(СПбГУ ИТМО)

УТВЕРЖДЕНО  
7.190.00003-01 31 01-ЛУ

ПРОГРАММНОЕ СРЕДСТВО 3GENETIC

ОПИСАНИЕ ПРИМЕНЕНИЯ

7.190.00001-01 31 01

Листов 10

Имя. N подл.	Подп. и дата	Взам. имя. N	Имя. N дубл.	Подп. и дата

## **АННОТАЦИЯ**

В данном документе приводится описание применения программного средства 3Genetic.

## СОДЕРЖАНИЕ

1.	Назначение программы .....	4
2.	Условия применения.....	4
2.1.	Требования к специалистам, обеспечивающим установку и эксплуатацию программных средств, разработанных на основе технологии разработки мультимедиа приложений .....	4
2.2.	Требования к разработчикам, применяющим программное средство для решения задач..	4
3.	описание.....	5
3.1.	Модульность системы .....	5
3.2.	Модуль, содержащий представление особи.....	6
3.3.	Модуль, содержащий генетический алгоритм.....	6
4.	Входные и выходные данные .....	7

## 1. НАЗНАЧЕНИЕ ПРОГРАММЫ

Программное средство 3Genetic предназначено для решения задач различных областей при помощи алгоритмов генетического программирования, а также как средство тестирования и сравнения различных генетических алгоритмов. В частности в данном программном средстве уже реализовано решение задачи управления беспилотными летательными аппаратами при помощи конечных автоматов.

## 2. УСЛОВИЯ ПРИМЕНЕНИЯ

Для нормального функционирования данного программного средства необходимо, чтобы аппаратное обеспечение персональной ЭВМ удовлетворяло следующим требованиям:

- процессор *Intel Pentium IV* или совместимый;
- тактовая частота процессора 2ГГц, не менее;
- оперативная память 1024 МВ, не менее;
- дисковый накопитель объемом 1 GB, не менее;
- отображающее устройство (монитор) с поддержкой разрешения 1024x768;
- устройства ввода клавиатура и мышь (трекбол, тачпад);

### 2.1. Требования к специалистам, обеспечивающим установку и эксплуатацию программных средств, разработанных на основе технологии разработки мультимедия приложений

Администрирование установки и эксплуатации разработанных программных средств должно осуществляться одним специалистом, имеющим:

- опыт поддержания работоспособности ПЭВМ IBM PC и совместимых с ними ПЭВМ под управлением операционной системы Microsoft Windows 2000, Microsoft Windows XP либо Microsoft Windows 2003 Server и совместимых.

### 2.2. Требования к разработчикам, применяющим программное средство для решения задач

Для решения задач при помощи 3Genetic необходимо иметь навыки программирования на языке Java версии 1.06.

### 3. ОПИСАНИЕ

#### 3.1. Модульность системы

Возможности и, соответственно, элементы интерфейса ядра программного средства *3Genetic* можно разделить на две группы: выбор используемого генетического алгоритма, и просмотр результатов его работы. К первой группе относятся:

- выбор алгоритма (из встроенных или реализованных пользователем);
- выбор типа особи;
- выбор способа представления особи, соответствующего выбранному типу особи.

Ко второй группе относятся:

- просмотр графиков зависимости значения функции приспособленности от номера поколения (наибольшего значения и среднего значения по поколению);
- сохранение графиков в формате PNG;
- просмотр особей текущего поколения и эмуляция действий системы со сложным поведением под их управлением;
- сохранение 50 лучших особей поколения;
- просмотр лучших особей (на протяжении всей работы алгоритма) и эмуляция действий системы со сложным поведением под их управлением.

Программное средство *3Genetic*, как отмечено выше, написано на языке программирования *Java*, поэтому подключаемые модули, это *JAR*-архивы (**J**ava **A**rchive), помещенные в определенные директории и содержащие определенные *Java*-классы. Все упоминаемые ниже интерфейсы содержатся в файле *common.jar*. Некоторые необязательные, но, возможно, полезные, при написании собственного модуля классы, содержатся в файле *util.jar*.

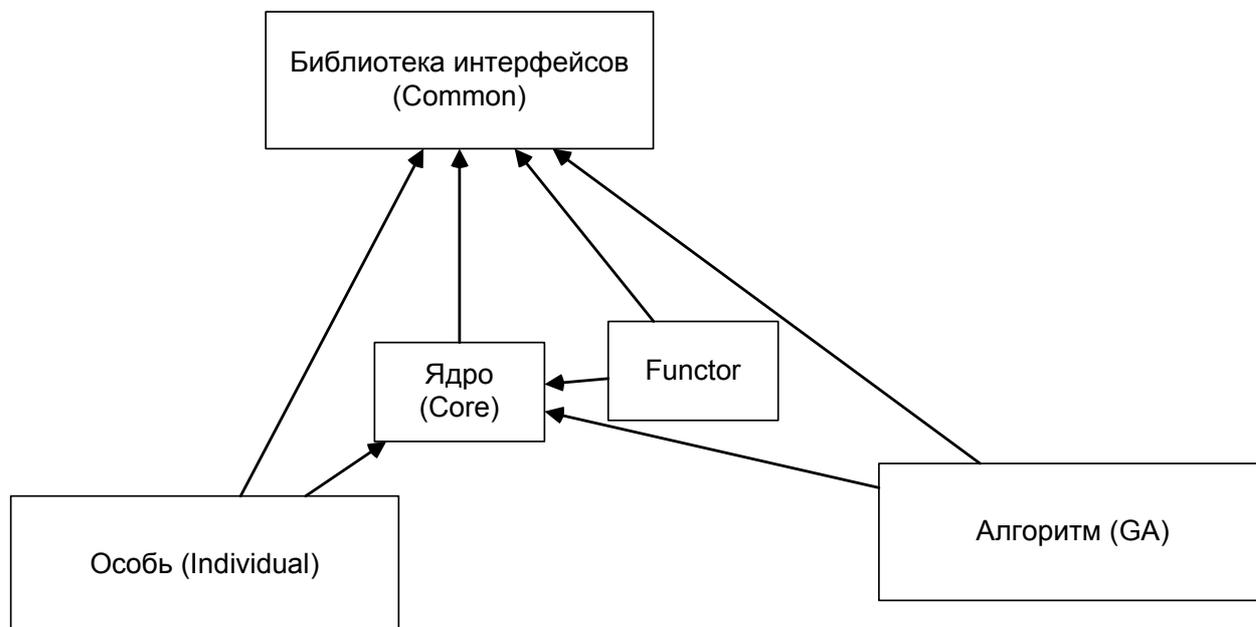


Рисунок 1. Граф переходов конечного автомата, построенного с помощью метода

сокращенных таблиц

### 3.2. Модуль, содержащий представление особи

Особь – это все то, с чем может работать генетический алгоритм – сущность, для которой определены операторы кроссовера и мутации и фитнес-функция. В терминах средства *3Genetic* – это экземпляр класса реализующего интерфейс `Individual`:

```
public interface Individual extends Comparable<Individual> {  
  
    public double fitness();  
    public Individual mutate(Random r);  
    public Individual[] crossover(Individual p, Random r);  
  
}
```

Метод `toString()` используется при сохранении особи в файл.

Для подключения модуля к ядру *3Genetic* *JAR*-файл с реализацией особи должен быть помещен в директорию `individuals`. В манифесте архива должны присутствовать следующие параметры: `Main-Class`, `Extension-Name`, `Comment`. `Main-Class` – это имя класса, реализующего интерфейс `Loader<IndividualFactory>`:

```
public interface Loader<Plugin> {  
  
    public Plugin load(Object... args);  
    Properties getProperties();  
  
}
```

`IndividualFactory` – это следующий интерфейс:

```
public interface IndividualFactory {  
    public Individual randomIndividual();  
}
```

Возвращаемые значения метода `load` (в данном случае он вызывается без параметров), загруженного `Loader`'ом экземпляра класса, реализующего интерфейс `IndividualFactory`, передаются системой генетическому алгоритму. Метод `getProperties()` возвращает свойства модуля, настраиваемые во время работы программного средства *3Genetic*. Класс, реализующий интерфейс `Loader`, должен иметь конструктор от одного аргумента типа `JarFile` (собственно *JAR*-файла, содержащего данный модуль).

Параметр `Extension-Name` содержит имя особи, отображаемое в диалоге выбора особи. Параметр `Comment` должен иметь вид «`arg1 arg2 ||| комментарий`, отображаемый в диалоге выбора особи». Здесь `arg1` и `arg2` – параметры, необходимые для рисования графиков, максимальное значение и число знаков после десятичной точки, отображаемых в значении функции приспособленности.

### 3.3. Модуль, содержащий генетический алгоритм

Генетический алгоритм в программном средстве *3Genetic* – это экземпляр класса *Java* реализующего интерфейс генетического алгоритма:

```
public interface GA {
```

```

public List<Individual> getGeneration();
public void nextGeneration();
public void bigMutation();
public Individual getBest();

}

```

Для подключения модуля к ядру программного средства *3Genetic JAR*-файл с реализацией генетического алгоритма должен быть помещен в директорию *gas*. В манифесте архива должны быть следующие параметры: *Main-Class*, *Extension-Name*, *Comment*.

Параметр *Main-Class* – это имя класса, реализующего интерфейс *Loader<GA>*. В данном случае метод *load* будет принимать один параметр – экземпляр *IndividualFactory*. Авторам неизвестны реализации генетических алгоритмов, которые не используют в своей работе (хотя бы при создании начального поколения) случайно сгенерированные особи. Параметры *Extension-Name* и *Comment* используются при отображении в диалоге выбора алгоритма.

#### 4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

В качестве входных данных используются информация из подключаемых модулей. В качестве выходных используется текстовое представление решения. Формат данного представления определяется разработчиком подключаемых модулей.

К примеру, в случае сокращенных таблиц переходов выходные данные имеют следующий формат:

- первая строка содержит два числа: число  $n$  состояний и номер начального состояния;
- вторая строка содержит три числа: общее число переменных, число  $k$  значимых переменных, число различных выходных воздействий;
- далее следуют  $n$  блоков, описывающих состояния;
- первая строка каждого такого блока содержит  $2^k$  чисел –  $i$ -ое из них равно номеру состояния, в которое осуществляется переход в случае, если набор значений значимых переменных совпадает с двоичной записью числа  $i$ ;
- каждый такой блок содержит  $2^k$  строк, каждая из которых соответствует определенному набору значений значимых переменных. В каждой из них содержится вектор номеров действий, выполняемый при соответствующей комбинации значений значимых переменных. Завершается описание состояния вектором значимых переменных.

Приведем пример выходных данных в таком формате.

```

6 5
6 3 7
4 4 0 0 0 2 5 0
0 0 0 0 0 0 0
0 0 0 1 1 1 0
1 0 0 0 0 0 0
1 0 0 0 0 0 0
0 1 0 1 0 1 0
0 0 1 0 0 0 1
1 1 0 0 0 0 1
0 0 0 0 0 0 0
1 0 1 1 0 0
1 5 0 2 0 4 4 1

```

```
1 0 0 0 0 0 0
0 0 0 0 0 0 1
1 0 0 0 0 1 0
0 0 1 0 1 0 1
1 1 1 1 0 0 0
0 1 1 0 0 0 0
0 1 0 0 0 0 0
0 0 0 0 1 1 1
1 0 1 0 1 0
4 0 5 3 4 0 5 3
0 0 0 1 0 0 1
0 0 0 0 1 0 0
0 0 0 1 0 1 0
0 1 1 0 0 1 1
0 0 0 0 1 0 0
0 1 0 0 1 1 0
1 0 0 1 0 0 0
0 1 0 0 0 0 0
1 1 0 0 1 0
5 3 2 3 3 3 5 2
0 0 0 0 0 0 0
0 1 0 0 0 0 0
1 1 0 0 1 1 0
1 0 0 0 1 0 0
0 0 0 0 0 0 1
0 0 0 1 0 1 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 1 1 0 1 0
1 4 0 2 3 2 2 1
1 0 0 0 0 0 1
0 0 0 0 0 0 1
0 0 1 1 1 1 0
0 0 0 1 0 0 0
0 1 0 0 1 0 1
0 1 0 0 1 0 0
0 0 0 1 1 0 0
0 0 0 0 0 0 1
1 0 1 0 1 0
0 0 0 0 0 0 0
0 0 0 0 0 0 1
0 1 0 0 0 1 0
1 0 0 0 1 0 0
1 1 0 0 1 0 1
1 1 0 0 0 0 1
0 1 1 0 1 1 1
0 0 0 0 0 0 0
0 1 0 0 0 1 1
1 1 1 0 0 0
```

Эти выходные данные описывают автомат, граф переходов которого приведен на рисунке.

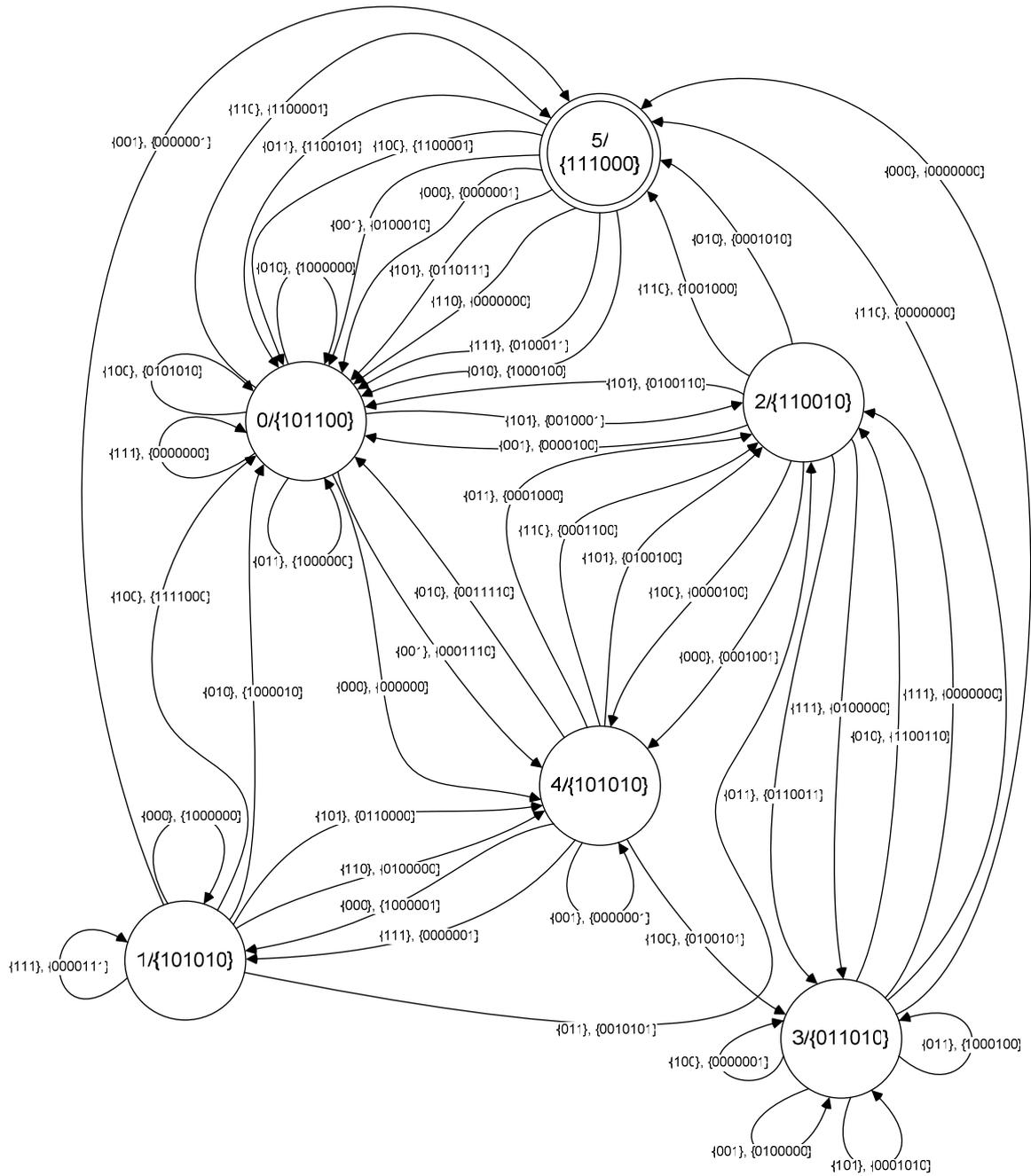


Рисунок 2. Граф переходов конечного автомата, построенного с помощью метода сокращенных таблиц

