

**Федеральное агентство по образованию  
Архангельский государственный технический университет**

**Методические указания по выполнению  
лабораторных работ и курсового проекта по дисциплине  
«Интеллектуальные информационные системы»**

**Архангельск 2010**

Рецензент  
А.Т. Гурьев

к.т.н., доцент, директор института информационных технологий АГТУ.

УДК 004.89

*Алешко Р.А., Бачурин И.В.* Методические указания по выполнению лабораторных работ и курсового проекта по дисциплине «Интеллектуальные информационные системы». – Архангельск: Изд-во Арханг. гос. техн. ун-та, 2010. – 26 с.

Подготовлены кафедрой информационных технологий АГТУ.

В методических указаниях представлена теоретическая составляющая, необходимая для выполнения лабораторных работ и курсового проекта, а также варианты заданий.

Предназначены для студентов, обучающихся по специальности 230201.65 – Информационные системы и технологии (направление подготовки 230000 - Информатика и вычислительная техника)

© Архангельский государственный  
технический университет, 2010  
© Алешко Р.А., Бачурин И.В., 2010

## **Оглавление**

### **Введение**

#### **1. Теоретическая часть**

1.1. Генетические алгоритмы и традиционные методы оптимизации

1.2. Основные понятия генетических алгоритмов

1.3. Классический генетический алгоритм

1.4. Иллюстрация выполнения классического генетического алгоритма

#### **2. Указания к выполнению курсового проекта**

2.1. Оформление пояснительной записки

2.2. Физическая реализация проекта

2.3. Этапы выполнения курсового проекта

#### **3. Задания на курсовой проект по дисциплине «Интеллектуальные информационные системы»**

#### **Литература**

## Введение

Цель курсового проектирования – приобретение практических навыков реализации интеллектуальных информационных систем (ИИС) и, в частности, генетических алгоритмов.

Генетические алгоритмы возникли в результате наблюдения и попыток копирования естественных процессов, происходящих в мире живых организмов, в частности, эволюции и связанной с ней селекции (естественного отбора) популяций живых существ. Конечно, при подобном сопоставлении следует обращать внимание на принципиально различную длительность протекания упоминаемых естественных процессов, т.е. на чрезвычайно быструю обработку информации в нервной системе и очень медленный процесс естественной эволюции. Однако при компьютерном моделировании эти различия оказываются несущественными.

Идею генетических алгоритмов высказал Дж. Холланд в конце шестидесятых - начале семидесятых годов XX века. Он заинтересовался свойствами процессов естественной эволюции (в том числе фактом, что эволюционируют хромосомы, а не сами живые существа). Холланд был уверен в возможности составить и реализовать в виде компьютерной программы алгоритм, который будет решать сложные задачи так, как это делает природа - путем *эволюции*. Поэтому он начал трудиться над алгоритмами, оперируя последовательностями двоичных цифр (единиц и нулей), получившими название *хромосом*. Эти алгоритмы имитировали эволюционные процессы в поколениях таких хромосом. В них были реализованы механизмы *селекции* и *репродукции*, аналогичные применяемым при естественной эволюции. Так же, как и в природе, генетические алгоритмы осуществляли поиск «хороших» хромосом без использования какой-либо информации о характере решаемой задачи. Требовалась только некая оценка каждой хромосомы, отражающая ее *приспособленность*. Механизм *селекции* заключается в выборе хромосом с наивысшей оценкой (т.е. наиболее приспособленных), которые репродуцируют чаще, чем особи с более низкой оценкой (хуже приспособ-

собленные). *Репродукция* означает создание новых хромосом в результате рекомбинации генов родительских хромосом. *Рекомбинация* - это процесс, в результате которого возникают новые комбинации генов. Для этого используются две операции: *скрещивание*, позволяющее создать две совершенно новые хромосомы потомков путем комбинирования генетического материала пары родителей, а также *мутация*, которая может вызывать изменения в отдельных хромосомах.

В генетических алгоритмах применяется ряд терминов, заимствованных из генетики, прежде всего *гены* и *хромосомы*, а также *популяция*, *особь*, *аллель*, *генотип*, *фенотип*.

Генетические алгоритмы применяются при разработке программного обеспечения, в системах искусственного интеллекта, оптимизации, в искусственных нейронных сетях и в других отраслях знаний.

Методические указания предназначены для студентов, обучающихся по специальности 230201.65 – Информационные системы и технологии (направление подготовки 230000 - Информатика и вычислительная техника).

## 1. Теоретическая часть

### 1.1. Генетические алгоритмы и традиционные методы оптимизации

Генетический алгоритм представляет собой метод, отражающий естественную эволюцию методов решения проблем, и в первую очередь задач оптимизации. Генетические алгоритмы - это процедуры поиска, основанные на механизмах естественного отбора и наследования. В них используется эволюционный принцип выживания наиболее приспособленных особей. Они отличаются от традиционных методов оптимизации несколькими базовыми элементами.

Свойства генетических алгоритмов:

- 1) обрабатывают не значения параметров самой задачи, а их закодированную форму;
- 2) осуществляют поиск решения исходя не из единственной точки, а из их некоторой популяции;
- 3) используют только целевую функцию, а не ее производные либо иную дополнительную информацию,
- 4) применяют вероятностные, а не детерминированные правила выбора.

Перечисленные четыре свойства, которые можно сформулировать также как кодирование параметров, операции на популяциях, использование минимума информации о задаче и рандомизация операций приводят в результате к устойчивости генетических алгоритмов и к их превосходству над другими широко применяемыми технологиями.

### 1.2. Основные понятия генетических алгоритмов

При описании генетических алгоритмов используются определения, заимствованные из генетики.

**Популяция** - это конечное множество *особей*.

**Особи**, входящие в популяцию, в генетических алгоритмах представляются хромосомами с закодированным в них множествами *параметров задачи*, т.е. решений, которые иначе называются *точками в пространстве поиска (search points)*. В некоторых источниках особи на-

зываются *организмами*.

**Хромосомы** (другие названия - *цепочки* или *кодовые последовательности*) - это упорядоченные *последовательности генов*.

**Ген** (также называемый *свойством*, *знаком* или *детектором*) - это атомарный элемент *генотипа*, в частности, хромосомы.

**Генотип** или **структура** - это набор хромосом данной особи. Следовательно, особями популяции могут быть генотипы либо единичные хромосомы (довольно распространенный случай, когда генотип состоит из одной хромосомы).

**Фенотип** - это набор значений, соответствующих данному генотипу, т.е. *декодированная структура* или множество *параметров задачи* (*решение*, *точка пространства поиска*).

**Аллель** - это значение конкретного гена, также определяемое как *значение свойства* или *вариант свойства*.

**Локус** или *позиция* указывает место размещения данного гена в хромосоме (цепочке). Множество позиций генов - это *локи*.

Очень важным понятием в генетических алгоритмах считается **функция приспособленности** (*fitness function*), иначе называемая *функцией оценки*. Она представляет меру приспособленности данной особи в популяции. Эта функция играет важнейшую роль, поскольку позволяет оценить степень приспособленности конкретных особей в популяции и выбрать из них наиболее приспособленные (т.е. имеющие наибольшие значения функции приспособленности) в соответствии с эволюционным принципом выживания «*сильнейших*» (лучше всего приспособившихся). Функция приспособленности также получила свое название непосредственно из генетики. Она оказывает сильное влияние на функционирование генетических алгоритмов и должна иметь точное и корректное определение. В задачах оптимизации функция приспособленности, как правило, оптимизируется. В теории управления функция приспособленности может принимать вид *функции погрешности*, а в теории игр - *стоимостной функции*. На каждой итерации генетического алгоритма приспособленность каждой особи данной популяции оценивается при помощи функции приспособленности, и на этой основе соз-

дается следующая популяция особей, составляющих множество потенциальных решений проблемы, например, задачи оптимизации.

Очередная популяция в генетическом алгоритме называется *поколением*, а к вновь создаваемой популяции особей применяется термин «новое поколение» или «поколение потомков».

### **Пример 1.**

Рассмотрим функцию

$$f(x) = 2x^2 + 1 \quad (1)$$

и допустим, что  $x$  принимает целые значения из интервала от 0 до 15.

Задача оптимизации этой функции заключается в перемещении по пространству, состоящему из 16 точек со значениями 0, 1, ..., 15 для обнаружения той точки, в которой функция принимает максимальное (или минимальное) значение.

В этом случае в качестве *параметра задачи* выступает переменная  $x$ . Множество  $\{0, 1, \dots, 15\}$  составляет *пространство поиска* и одновременно - множество потенциальных решений задачи. Каждое из 16 чисел, принадлежащих к этому множеству, называется *точкой пространства поиска, решением, значением параметра, фенотипом*. Следует отметить, что решение, оптимизирующее функцию, называется *наилучшим* или *оптимальным* решением. Значения параметра  $x$  от 0 до 15 можно закодировать следующим образом:

0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111,  
1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111.

Это широко известный способ двоичного кодирования, связанный с записью десятичных цифр в двоичной системе. Представленные *кодовые последовательности* также называются *цепями* или *хромосомами*. В рассматриваемом примере они выступают и в роли *генотипов*. Каждая из хромосом состоит из 4 генов (иначе можно сказать, что двоичные последовательности состоят из 4 битов). Значение гена в конкретной позиции называется *аллелью*, принимающей в данном случае значения 0 или 1. *Популяция* состоит из *особей*, выбираемых среди этих 16 *хромосом*. Примером популяции с численностью, равной 6, может быть, например, множество хромосом  $\{0010, 0101, 0111, 1001, 1100,$



1110}, представляющих собой закодированную форму следующих фенотипов: {2, 5, 7, 9, 12, 14}. *Функция приспособленности* в этом примере задается выражением (1). Приспособленность отдельных хромосом в популяции определяется значением этой функции для значений  $x$ , соответствующих этим хромосомам, т.е. для *фенотипов*, соответствующих определенным *генотипам*.

### 1.3. Классический генетический алгоритм

Основной (классический) генетический алгоритм состоит из следующих шагов:

- 1) инициализация, или выбор исходной популяции хромосом;
- 2) оценка приспособленности хромосом в популяции;
- 3) проверка условия останова алгоритма;
- 4) селекция хромосом;
- 5) применение генетических операторов;
- 6) формирование новой популяции;
- 7) выбор «наилучшей» хромосомы.

Блок-схема основного генетического алгоритма изображена на рис.1. Рассмотрим конкретные этапы этого алгоритма более подробно.

**Инициализация**, т.е. формирование исходной популяции, заключается в случайном выборе заданного количества хромосом (особей), представляемых двоичными последовательностями фиксированной длины.

**Оценивание приспособленности** хромосом в популяции состоит в расчете функции приспособленности для каждой хромосомы этой популяции. Чем больше значение этой функции, тем выше «качество» хромосомы. Форма функции приспособленности зависит от характера решаемой задачи.

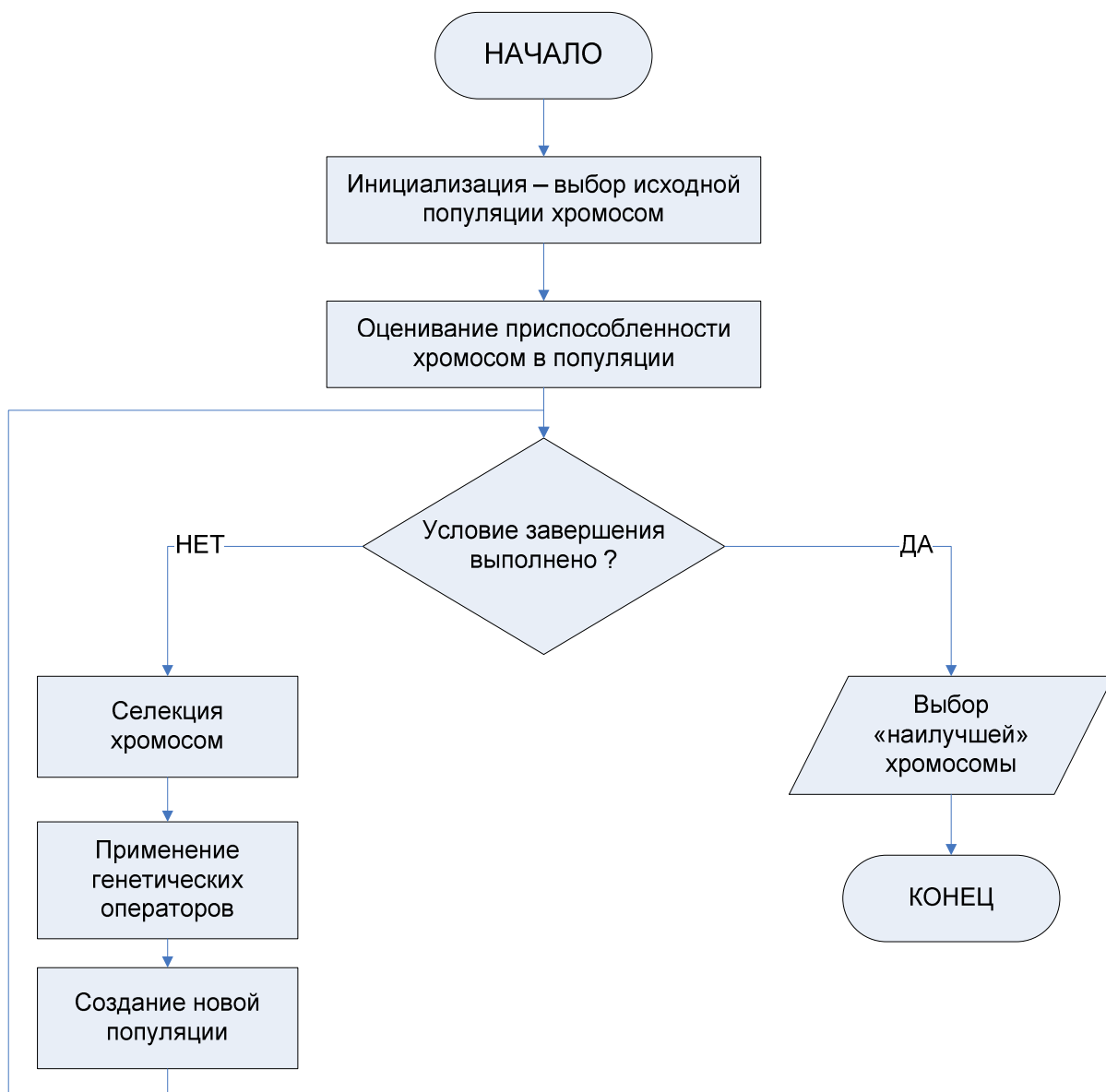


Рис.1. Блок-схема генетического алгоритма

**Проверка условия остановки алгоритма.** Определение условия остановки генетического алгоритма зависит от его конкретного применения. В оптимизационных задачах, если известно максимальное (или минимальное) значение функции приспособленности, то остановка алгоритма может произойти после достижения ожидаемого оптимального значения, возможно - с заданной точностью. Остановка алгоритма также может произойти в случае, когда его выполнение не приводит к улучшению уже достигнутого значения. Алгоритм может быть останов-

лен по истечении определенного времени выполнения либо после выполнения заданного количества итераций. Если условие остановки выполнено, то производится переход к завершающему этапу выбора «наилучшей» хромосомы. В противном случае на следующем шаге выполняется селекция.

**Селекция хромосом** заключается в выборе (по рассчитанным на втором этапе значениям функции приспособленности) тех хромосом, которые будут участвовать в создании потомков для следующей популяции, т.е. для очередного поколения. Такой выбор производится согласно принципу естественного отбора, по которому наибольшие шансы на участие в создании новых особей имеют хромосомы с наибольшими значениями функции приспособленности. Существуют различные методы селекции. Наиболее популярным считается так называемый *метод рулетки (roulette wheel selection)*, который свое название получил по аналогии с известной азартной игрой. Каждой хромосоме может быть сопоставлен сектор колеса рулетки, величина которого устанавливается пропорциональной значению функции приспособленности данной хромосомы. Поэтому, чем больше значение функции приспособленности, тем больше сектор на колесе рулетки. Все колесо рулетки соответствует сумме значений функции приспособленности всех хромосом рассматриваемой популяции, каждой хромосоме, обозначаемой  $ch_i$  для  $i=1,2,...,N$  (где  $N$  обозначает численность популяции) соответствует сектор колеса  $v(ch_i)$ , выраженный в процентах согласно формуле

$$v(ch_i) = p_s(ch_i) \cdot 100\% \quad (2)$$

где

$$p_s(ch_i) = \frac{F(ch_i)}{\sum_{i=1}^N F(ch_i)} \quad (3)$$

причем  $F(ch_i)$  - значение функции приспособленности хромосомы  $ch_i$ , а  $p_s(ch_i)$  - *вероятность селекции* хромосомы  $ch_i$ . Селекция хромосомы может быть представлена как результат поворота колеса рулетки, поскольку «выигравшая» (т.е. выбранная) хромосома относится к выпавшему сектору этого колеса. Очевидно, что чем больше сектор, тем больше вероятность «победы» соответствующей хромосомы. Поэтому

вероятность выбора данной хромосомы оказывается пропорциональной значению ее функции приспособленности. Если всю окружность колеса рулетки представить в виде цифрового интервала  $[0, 100]$ , то выбор хромосомы можно отождествить с выбором числа из интервала  $[a, b]$ , где  $a$  и  $b$  обозначают соответственно начало и окончание фрагмента окружности, соответствующего этому сектору колеса; очевидно, что  $0 \leq a < b \leq 100$ . В этом случае выбор с помощью колеса рулетки сводится к выбору числа из интервала  $[0, 100]$ , которое соответствует конкретной точке на окружности колеса.

В результате процесса селекции создается *родительская популяция*, также называемая *родительским пулом* (*mating pool*) с численностью  $N$ , равной численности текущей популяции.

**Применение генетических операторов** к хромосомам, отобраным с помощью селекции, приводит к формированию новой популяции потомков от созданной на предыдущем шаге родительской популяции.

В классическом генетическом алгоритме применяются два основных генетических оператора: *оператор скрещивания* (*crossover*) и *оператор мутации* (*mutation*). Однако следует отметить, что оператор мутации играет явно второстепенную роль по сравнению с оператором скрещивания. Это означает, что скрещивание в классическом генетическом алгоритме производится практически всегда, тогда как мутация - достаточно редко. Вероятность скрещивания, как правило, достаточно велика (обычно  $0,5 \leq p_c \leq 1$ ), тогда как вероятность мутации устанавливается весьма малой (чаще всего  $0 \leq p_m \leq 0,1$ ). Это следует из аналогии с миром живых организмов, где мутации происходят чрезвычайно редко

В генетическом алгоритме мутация хромосом может выполняться на популяции родителей перед скрещиванием, либо на популяции потомков, образованных в результате скрещивания.

**Оператор скрещивания.** На первом этапе скрещивания выбираются пары хромосом из родительской популяции (родительского пула). Это временная популяция, состоящая из хромосом, отобранных в результате селекции и предназначенных для дальнейших пре-

образований операторами скрещивания и мутации с целью формирования новой популяции потомков. На данном этапе хромосомы из родительской популяции объединяются в пары. Это производится случайным способом в соответствии с вероятностью скрещивания  $p_c$ . Далее для каждой пары отобранных таким образом родителей разыгрывается позиция гена (*локус*) в хромосоме, определяющая так называемую *точку скрещивания*. Если хромосома каждого из родителей состоит из  $L$  генов, то очевидно, что точка скрещивания  $l_k$  представляет собой натуральное число, меньшее  $L$ . Поэтому фиксация точки скрещивания сводится к случайному выбору числа из интервала  $[1, L-1]$ . В результате скрещивания пары родительских хромосом получается следующая пара потомков:

1) потомок, хромосома которого на позициях от 1 до  $l_k$  состоит из генов первого родителя, а на позициях от  $l_k + 1$  до  $L$  - из генов второго родителя;

2) потомок, хромосома которого на позициях от 1 до  $l_k$  состоит из генов второго родителя, а на позициях от  $l_k + 1$  до  $L$  - из генов первого родителя.

**Оператор мутации** с вероятностью  $p_m$  изменяет значение гена в хромосоме на противоположное (т.е. с 0 на 1 или обратно). Например, если в хромосоме  $[100110101010]$  мутации подвергается ген на позиции 7, то его значение, равное 1, изменяется на 0, что приводит к образованию хромосомы  $[100110001010]$ . Как уже упоминалось выше, вероятность мутации обычно очень мала, и именно от нее зависит, будет данный ген мутировать или нет. При выборе вероятности  $p_m$  мутации можно применять, например, случайные числа из интервала  $[0, 1]$  для каждого гена и отбор для выполнения этой операции тех генов, для которых разыгранное число оказывается меньшим или равным значению  $p_m$ .

**Формирование новой популяции.** Хромосомы, полученные в результате применения генетических операторов к хромосомам временной родительской популяции, включаются в состав новой популяции. Она становится так называемой текущей популяцией для данной итерации генетического алгоритма. На каждой очередной итерации рас-

считываются значения функции приспособленности для всех хромосом этой популяции, после чего проверяется условие остановки алгоритма и либо фиксируется результат в виде хромосомы с наибольшим значением функции приспособленности, либо осуществляется переход к следующему шагу генетического алгоритма, т.е. к селекции. В классическом генетическом алгоритме вся предшествующая популяция хромосом замещается новой популяцией потомков, имеющей ту же численность.

**Выбор «наилучшей» хромосомы.** Если условие остановки алгоритма выполнено, то следует вывести результат работы, т.е. представить искомое решение задачи. Лучшим решением считается хромосома с наибольшим значением функции приспособленности.

В завершение следует признать, что генетические алгоритмы унаследовали свойства естественного эволюционного процесса, состоящие в генетических изменениях популяций организмов с течением времени.

Главный фактор эволюции - это естественный отбор (т.е. природная селекция), который приводит к тому, что среди генетически различающихся особей одной и той же популяции выживают и оставляют потомство только наиболее приспособленные к окружающей среде. В генетических алгоритмах также выделяется этап селекции, на котором из текущей популяции выбираются и включаются в родительскую популяцию особи, имеющие наибольшие значения функции приспособленности. На следующем этапе, который иногда называется *эволюцией*, применяются генетические операторы скрещивания и мутации, выполняющие рекомбинацию генов в хромосомах.

Операция скрещивания заключается в обмене фрагментами цепочек между двумя родительскими хромосомами. Пары родителей для скрещивания выбираются из родительского пула случайным образом так, чтобы вероятность выбора конкретной хромосомы для скрещивания была равна вероятности  $p_c$ . Например, если в качестве родителей случайным образом выбираются две хромосомы из родительской популяции численностью  $N$ , то  $p_c = 2/N$ . Аналогично, если из родительской популяции численностью  $N$  выбирается  $2z$  хромосом ( $z \leq N/2$ ), которые

образуют  $z$  пар родителей, то  $p_c = 2z/N$ . Обратим внимание, что если все хромосомы текущей популяции объединены в пары до скрещивания, то  $p_c = 1$ . После операции скрещивания родители в родительской популяции замещаются их потомками.

Операция мутации изменяет значения генов в хромосомах с заданной вероятностью  $p_m$  способом, представленным при описании соответствующего оператора. Это приводит к инвертированию значений отобранных генов с 0 на 1 и обратно. Значение  $p_m$ , как правило, очень мало, поэтому мутации подвергается лишь небольшое количество генов. Скрещивание - это ключевой оператор генетических алгоритмов, определяющий их возможности и эффективность. Мутация играет более ограниченную роль. Она вводит в популяцию некоторое разнообразие и предупреждает потери, которые могли бы произойти вследствие исключения какого-нибудь значимого гена в результате скрещивания.

Основной (классический) генетический алгоритм известен в литературе в качестве инструмента, в котором выделяются три вида операций: *репродукции*, *скрещивания* и *мутации*. Термины *селекция* и *репродукция* в данном контексте используются в качестве синонимов. При этом репродукция в данном случае связывается скорее с созданием копий хромосом родительского пула, тогда как более распространенное содержание этого понятия обозначает процесс формирования новых особей, происходящих от конкретных родителей. Если мы принимаем такое толкование, то операторы скрещивания и мутации могут считаться операторами репродукции, а селекция - отбором особей (хромосом) для репродукции.

#### **1.4. Иллюстрация выполнения классического генетического алгоритма**

Рассмотрим выполнение описанного в предыдущем разделе классического генетического алгоритма на как можно более простом примере. Проследим последовательность выполнения его этапов, соответствующих блок-схеме на рис.1.

## Пример 2.

Рассмотрим сильно упрощенный и довольно искусственный пример, состоящий в нахождении хромосомы с максимальным количеством единиц. Допустим, что хромосомы состоят из 12 генов, а популяция насчитывает 8 хромосом. Понятно, что наилучшей будет хромосома, состоящая из 12 единиц. Посмотрим, как протекает процесс решения этой весьма тривиальной задачи с помощью генетического алгоритма.

**Инициализация, или выбор исходной популяции хромосом.** Необходимо случайным образом сгенерировать 8 двоичных последовательностей длиной 12 битов. Таким образом можно сформировать исходную популяцию:

$$\begin{array}{ll} \text{ch}_1 = [111001100101] & \text{ch}_5 = [010001100100] \\ \text{ch}_2 = [001100111010] & \text{ch}_6 = [010011000101] \\ \text{ch}_3 = [011101110011] & \text{ch}_7 = [101011011011] \\ \text{ch}_4 = [001000101000] & \text{ch}_8 = [000010111100] \end{array}$$

**Оценка приспособленности хромосом в популяции.** В рассматриваемом упрощенном примере решается задача нахождения такой хромосомы, которая содержит наибольшее количество единиц. Поэтому функция приспособленности определяет количество единиц в хромосоме. Обозначим функцию приспособленности символом  $F$ . Тогда ее значения для каждой хромосомы из исходной популяции будут такие:

$$\begin{array}{ll} F(\text{ch}_1) = 7 & F(\text{ch}_5) = 4 \\ F(\text{ch}_2) = 6 & F(\text{ch}_6) = 5 \\ F(\text{ch}_3) = 8 & F(\text{ch}_7) = 8 \\ F(\text{ch}_4) = 3 & F(\text{ch}_8) = 5 \end{array}$$

Хромосомы  $\text{ch}_3$  и  $\text{ch}_7$  характеризуются наибольшими значениями функции принадлежности. В этой популяции они считаются наилучшими кандидатами на решение задачи. Если в соответствии с блок-схемой генетического алгоритма (рис.1) условие остановки алгоритма не выполняется, то на следующем шаге производится селекция хромосом из текущей популяции.

**Селекция хромосом.** Селекция производится методом рулетки. На основании формул (2) и (3) для каждой из 8 хромосом текущей популя-



ции (в нашем случае - исходной популяции, для которой  $N = 8$ ) получаем секторы колеса рулетки, выраженные в процентах (рис.2)

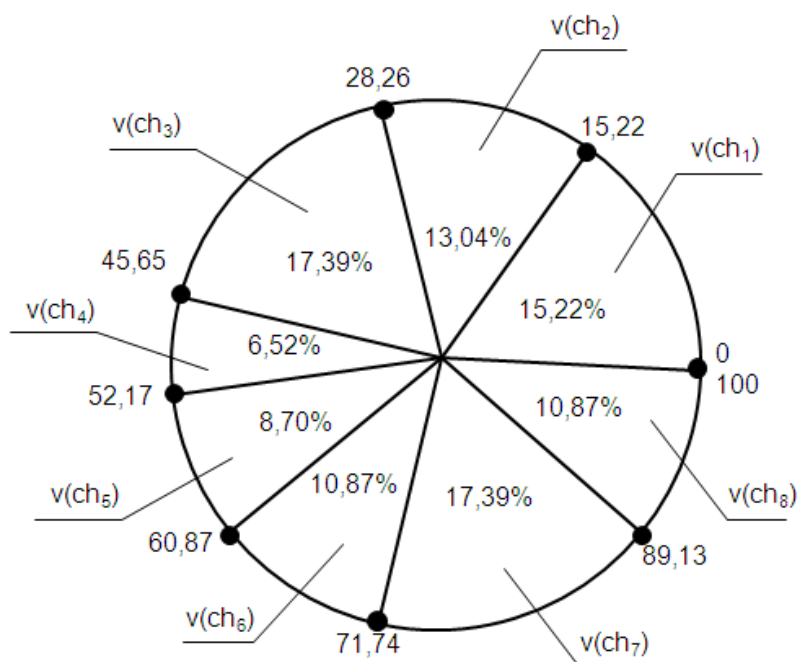


Рис.2. Колесо рулетки для селекции

$v(ch_1) = 15,22$	$v(ch_5) = 8,70$
$v(ch_2) = 13,04$	$v(ch_6) = 10,87$
$v(ch_3) = 17,39$	$v(ch_7) = 17,39$
$v(ch_4) = 6,52$	$v(ch_8) = 10,87$

Розыгрыш с помощью колеса рулетки сводится к случайному выбору числа из интервала  $[0, 100]$ , указывающего на соответствующий сектор на колесе, т.е. на конкретную хромосому.

Допустим, что разыграны следующие 8 чисел:

79 44 9 74 44 86 48 23

Это означает выбор хромосом:

$ch_7$   $ch_3$   $ch_1$   $ch_7$   $ch_3$   $ch_7$   $ch_4$   $ch_2$

Как видно, хромосома  $ch_7$  была выбрана трижды, а хромосома  $ch_3$  - дважды. Именно эти хромосомы имеют наибольшее значение функции приспособленности. Однако выбрана и хромосома  $ch_4$  с наименьшим значением функции приспособленности. Все выбранные таким образом хромосомы включаются в так называемый родительский пул.

**Применение генетических операторов.** Допустим, что ни одна из отобранных в процессе селекции хромосом не подвергается мутации, и все они составляют популяцию хромосом, предназначенных для скрещивания. Это означает, что вероятность скрещивания  $p_c = 1$ , а вероятность мутации  $p_m = 0$ . Допустим, что из этих хромосом случайным образом сформированы пары родителей

$ch_2$  и  $ch_7$ ,  $ch_1$  и  $ch_7$ ,  $ch_3$  и  $ch_4$ ,  $ch_3$  и  $ch_7$

Для первой пары случайным образом выбрана точка скрещивания  $l_k = 4$ , для второй  $l_k = 3$ , для третьей  $l_k = 11$ , для четвертой  $l_k = 5$ . При этом процесс скрещивания протекает так, как показано на рис.3.

Первая пара родителей [001100111010] [101011011011] $l_k = 4$	скрещивание →	Первая пара потомков [001111011011] [101000111010]
Вторая пара родителей [111001100101] [101011011011] $l_k = 3$	скрещивание →	Вторая пара потомков [111011011011] [101001100101]
Третья пара родителей [011101110011] [001000101000] $l_k = 11$	скрещивание →	Третья пара потомков [011101110010] [001000101001]
Четвертая пара родителей [011101110011] [101011011011] $l_k = 5$	скрещивание →	Четвертая пара потомков [011101011011] [101011110011]

Рис.3. Процесс скрещивания хромосом

В результате выполнения оператора скрещивания получаются 4 пары потомков. Если бы при случайном подборе пар хромосом для скрещивания были объединены, например,  $ch_3$  с  $ch_3$  и  $ch_4$  с  $ch_7$  вместо  $ch_3$  с  $ch_4$  и  $ch_3$  с  $ch_7$ , а другие пары остались без изменения, то скрещивание  $ch_3$  с  $ch_3$  дало бы две такие же хромосомы независимо от разыгранной точки скрещивания. Это означало бы получение двух потомков, идентичных своим родителям. Заметим, что такая ситуация наиболее вероятна для хромосом с наибольшим значением функции приспособленности, т.е. именно такие хромосомы получают наибольшие шансы на переход в новую популяцию.

**Формирование новой популяции.** После выполнения операции скрещивания мы получаем (согласно рис. 3) следующую популяцию потомков:

Ch1 = [001111011011]	Ch5 = [011101110010]
Ch2 = [101000111010]	Ch6 = [001000101001]
Ch3 = [111011011011]	Ch7 = [011101011011]
Ch4 = [101001100101]	Ch8 = [101011110011]

Для отличия от хромосом предыдущей популяции обозначения вновь сформированных хромосом начинаются с заглавной буквы С.

Согласно блок-схеме генетического алгоритма (рис.1) производится возврат ко второму этапу, т.е. к оценке приспособленности хромосом из вновь сформированной популяции, которая становится текущей. Значения функций приспособленности хромосом этой популяции составляют:

$F(Ch1) = 8$	$F(Ch5) = 7$
$F(Ch2) = 6$	$F(Ch6) = 4$
$F(Ch3) = 9$	$F(Ch7) = 8$
$F(Ch4) = 6$	$F(Ch8) = 8$

Заметно, что популяция потомков характеризуется гораздо более высоким средним значением функции приспособленности, чем популяция родителей. Обратим внимание, что в результате скрещивания получена хромосома  $Ch_3$  с наибольшим значением функции приспособленности, которым не обладала ни одна хромосома из родительской популяции. Однако могло произойти и обратное, поскольку после скрещива-

ния на первой итерации хромосома, которая в родительской популяции характеризовалась наибольшим значением функции приспособленности, могла просто «потеряться». Помимо этого, «средняя» приспособленность новой популяции все равно оказалась бы выше предыдущей, а хромосомы с большими значениями функции приспособленности имели бы шансы появиться в следующих поколениях.

## **2. Указания к выполнению курсового проекта**

В результате выполнения курсового проекта должны быть представлены пояснительная записка и программная реализация генетического алгоритма.

### **2.1. Оформление пояснительной записки**

Пояснительная записка оформляется на листах бумаги формата А4 в соответствии с требованиями СТО 01.04-2005 «Работы студентов. Общие требования и правила оформления». Объем пояснительной записки 20-25 страниц машинописного текста.

### **2.2 Техническое задание**

Одной из главных составляющих успешного создания автоматизированной системы (АС) является разработка технического задания (ТЗ). Согласно ГОСТ 34.602-89 ТЗ включает следующие разделы:

1. Общие сведения.
2. Назначение и цели создания (развития) системы.
3. Характеристика объектов автоматизации.
4. Требования к системе.
5. Состав и содержание работ по созданию системы.
6. Порядок контроля и приемки системы.
7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие.
8. Требования к документированию.
9. Источники разработки.

Суть ТЗ как основного документа в процессе создания АС заключается в проработке, выборе и утверждении основных технических, организационных, программных, информационно-логических и лингвистических решений, которые устанавливаются в разделе «Требования к системе».

Данный раздел, в свою очередь, состоит из трех подразделов:

1. Требования к системе в целом.
2. Требования к функциям (задачам), выполняемым системой.
3. Требования к видам обеспечения.

Требования к системе в целом отражают концептуальные параметры и характеристики создаваемой системы, среди которых указываются требования к структуре и функционированию системы, к надежности и безопасности, к численности и квалификации персонала и т. д.

Требования к функциям (задачам) содержат перечень функций, задач или их комплексов; временной регламент каждой функции, задачи или комплекса задач; требования к качеству реализации каждой функции; к форме представления выходной информации; характеристики необходимой точности и времени выполнения, требования одновременности выполнения группы функций; достоверности выдачи результатов.

Требования по видам обеспечения в зависимости от вида системы содержат требования по математическому, информационному, лингвистическому, программному, техническому, метрологическому, организационному, методическому и другим видам обеспечения системы. Для большинства разновидностей АС, в частности для всевозможных видов АИС, особое значение имеют решения и требования по информационному обеспечению. В данном подразделе, в частности, определяются требования:

1. К составу, структуре и способам организации данных в системе (информационно-логическая схема).
2. К информационному обмену между компонентами системы.
3. К информационной совместимости со смежными системами.
4. По использованию общероссийских и других классификаторов, унифицированных документов.

5. По применению систем управления базами данных.
6. К структуре процесса сбора, обработки, передачи данных в системе и представлению данных.
7. К защите данных от разрушений при авариях и сбоях в электропитании системы.
8. К контролю, хранению, обновлению и восстановлению данных.
9. К процедуре придания юридической силы документам, продуцируемым техническими средствами АС.

На основе установленных в ТЗ основных требований и технических решений на последующих этапах конкретизируются и непосредственно разрабатываются компоненты и элементы системы.

Разработка и документация программного обеспечения в процессе создания или комплектования автоматизированных систем регламентируются комплексом стандартов, объединенных в группу «Единая система программной документации (ЕСПД)».

### **2.3. Физическая реализация проекта**

Физическая реализация проекта (программное средство) представляется на любом носителе информации. Выбор среды разработки должен быть обоснован и согласован с преподавателем.

### **2.4. Этапы выполнения курсового проекта**

- 1) Выбор темы и утверждение задания.
- 2) Знакомство с теоретической частью.
- 3) Разработка концептуальной модели программного средства.
- 4) Тестирование программного средства и устранение возможных ошибок в его работе.
- 5) Оформление пояснительной записки.
- 6) Сдача проекта на проверку.
- 7) Защита проекта.

### 3. Задания на курсовой проект по дисциплине «Интеллектуальные информационные системы»

Разработать программную реализацию генетического алгоритма, решающего задачу коммивояжера на основе матриц расстояний.

Задача коммивояжера является классической оптимизационной задачей. Суть ее заключается в следующем. Дано множество из  $n$  городов и матрица расстояний между ними. Цель коммивояжера – объехать все эти города по кратчайшему пути. Причем в каждом городе он должен побывать один раз и свой путь закончить в том же городе, откуда начал.

Для решения предлагается следующая задача: имеется десять городов, расстояния между которыми представлены матрицей  $M$ , состоящей из 10 столбцов и 10 строк, где на пересечении строки и столбца находится расстояние между соответствующими городами:

**M**

№	1	2	3	4	5	6	7	8	9	10
1	0	1	8	8	2	3	6	2	2	8
2	1	0	1	5	7	3	2	1	5	7
3	8	1	0	2	5	7	4	1	2	6
4	8	5	2	0	1	7	2	4	3	9
5	2	7	5	1	0	9	2	2	1	8
6	3	3	7	7	9	0	9	6	9	3
7	6	2	4	2	2	9	0	7	2	1
8	2	1	1	4	2	6	7	0	5	2
9	2	5	2	3	1	9	2	5	0	2
10	8	7	6	9	8	3	1	2	2	0

Для решения задачи применим следующий генетический алгоритм. Решение представим в виде перестановки чисел от 1 до 10, отображающей последовательность посещения городов. А значение целевой функции будет равно стоимости всей поездки, вычисленной в соответствии с вышеприведенной матрицей. Заметим, что чем меньше значение целевой функции, тем лучше. То есть целью в данном случае является поиск минимума целевой функции.

В качестве оператора скрещивания выберем одну из процедур, описанных далее:

### **Частично отображаемое скрещивание.**

Для двух родительских хромосом  $ch_{1,1} = \{1,2,3,4,5,6,7,8,9,10,1\}$  и  $ch_{1,2} = \{1,4,6,2,9,8,7,3,10,5,1\}$  случайным образом находим две точки разрыва:  $p_1 = 4$  и  $p_2 = 8$  ( $1 \leq p_1 \leq p_2 \leq n$ ). Считаем, что разрыв произошел следующим образом:  $ch_{1,1} = \{1,2,3,4 \mid 5,6,7,8 \mid 9,10,1\}$  и  $ch_{1,2} = \{1,4,6,2 \mid 9,8,7,3 \mid 10,5,1\}$ . Напомним, что позиции 1 и 11 в операции не участвуют, как будто реальная хромосома расположена между этими позициями. При формировании потомков  $ch_{2,1} = \{1, x, x, x \mid x, x, x, x \mid x, x, 1\}$  и  $ch_{2,2} = \{1, x, x, x \mid x, x, x, x \mid x, x, 1\}$ , вначале производим обмен частей находящихся между точками разрыва:  $ch_{2,1} = \{1, x, x, x \mid 9,8,7,3 \mid x, x, 1\}$  и  $ch_{2,2} = \{1, x, x, x \mid 5,6,7,8 \mid x, x, 1\}$  (т.е. для потомка  $ch_{2,1}$  берем среднюю часть из родителя  $ch_{1,2}$ , а для потомка  $ch_{2,2}$  берем среднюю часть из родителя  $ch_{1,1}$ ). Затем расставляем оставшиеся позиции от соответствующих хромосом (в  $ch_{2,1}$  из  $ch_{1,1}$ , а для  $ch_{2,2}$  из  $ch_{1,2}$ ) слева направо до возникновения конфликта (номер вершины повторяется в уже сформированной части хромосомы). Если произошел конфликт, то записываем не конфликтующий номер, а номер из соседней хромосомы в той позиции, в которой был найден конфликт в текущей хромосоме. Так продолжается до полного разрешения конфликтов. Получаем:  $ch_{2,1} = \{1,2,6,4 \mid 9,8,7,3 \mid 9,10,1\}$  и  $ch_{2,2} = \{1,4,3,2 \mid 5,6,7,8 \mid 10,9,1\}$ .

### **Упорядоченное скрещивание.**

Для двух родительских хромосом  $ch_{1,1} = \{1,2,3,4,5,6,7,8,9,10,1\}$  и  $ch_{1,2} = \{1,4,6,2,9,8,7,3,10,5,1\}$  случайным образом находим две точки разрыва:  $p_1 = 4$  и  $p_2 = 8$  ( $1 \leq p_1 \leq p_2 \leq n$ ). Считаем, что разрыв произошел следующим образом:  $ch_{1,1} = \{1,2,3,4 \mid 5,6,7,8 \mid 9,10,1\}$  и  $ch_{1,2} = \{1,4,6,2 \mid 9,8,7,3 \mid 10,5,1\}$ . Напомним, что позиции 1 и 11 в операции не участвуют, как будто реальная хромосома расположена между этими позициями. При формировании потомков  $ch_{2,1} = \{1, x, x, x \mid x, x, x, x \mid x, x, 1\}$  и  $ch_{2,2} = \{1, x, x, x \mid x, x, x, x \mid x, x, 1\}$ , вначале производим копирование частей находящихся между точками разрыва в соот-



ветствующие потомки:  $ch_{2,1} = \{1, x, x, x \mid 5, 6, 7, 8 \mid x, x, 1\}$  и  $ch_{2,2} = \{1, x, x, x \mid 9, 8, 7, 3 \mid x, x, 1\}$  (т.е. для потомка  $ch_{2,1}$  берем среднюю часть из родителя  $ch_{1,1}$ , а для потомка  $ch_{2,2}$  берем среднюю часть из родителя  $ch_{1,2}$ ). Затем, начиная с позиции, следующей за второй точкой разрыва (в данном примере – позиция 9) расставляем последовательно справа налево в пустые позиции (т.е. в позиции 9, 10, 2, 3, 4) значения номеров вершин из соответствующих позиций перекрестных родителей (в  $ch_{2,1}$  из  $ch_{1,2}$ , а в  $ch_{2,2}$  из  $ch_{1,1}$ ), пропуская те из них, которые вызывают конфликт с уже расставленными номерами. Получаем:  $ch_{2,1} = \{1, 2, 9, 3 \mid 5, 6, 7, 8 \mid 10, 4, 1\}$  и  $ch_{2,2} = \{1, 4, 5, 6 \mid 9, 8, 7, 3 \mid 10, 2, 1\}$ .

### **Циклическое скрещивание.**

Пусть имеются следующие родительские хромосомы  $ch_{1,1} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1\}$  и  $ch_{1,2} = \{1, 4, 6, 2, 9, 8, 7, 3, 10, 5, 1\}$ . Формирование потомка  $ch_{2,1} = \{1, x, x, x, x, x, x, x, x, x, 1\}$  идет по шагам. Сначала в самую левую незанятую позицию ставится элемент из самой левой позиции родителя  $ch_{1,1}$ , который не вызывал бы конфликт с уже проставленными в  $ch_{2,1}$  элементами. Затем в самую правую незанятую позицию ставится элемент из самой правой позиции родителя  $ch_{1,2}$ , который не вызывал бы конфликт с уже проставленными в  $ch_{2,1}$  элементами. Так продолжается до тех пор, пока не будут заняты все позиции в  $ch_{2,1}$  (оба цикла встретятся посередине). Получаем:  $ch_{2,1} = \{1, 2, 3, 4, 6, 9, 8, 7, 10, 5, 1\}$ . Формирование потомка  $ch_{2,2} = \{1, x, x, x, x, x, x, x, x, x, 1\}$  идет по шагам. Сначала в самую левую незанятую позицию ставится элемент из самой левой позиции родителя  $ch_{1,2}$ , который не вызывал бы конфликт с уже проставленными в  $ch_{2,2}$  элементами. Затем в самую правую незанятую позицию ставится элемент из самой правой позиции родителя  $ch_{1,1}$ , который не вызывал бы конфликт с уже проставленными в  $ch_{2,2}$  элементами. Так продолжается до тех пор, пока не будут заняты все позиции в  $ch_{2,2}$  (оба цикла встретятся посередине). Получаем:  $ch_{2,2} = \{1, 4, 6, 2, 7, 3, 5, 8, 9, 10, 1\}$ .

Таб.1. Варианты заданий на курсовой проект

<b>вариант</b>	<b>частично-отображаемое скрещивание</b>	<b>упорядоченное скрещивание</b>	<b>циклическое скрещивание</b>
01	M1*		
02		M1*	
03			M1*
04	M2*		
05		M2*	
06			M2*
07	M3*		
08		M3*	
09			M3**
10	M4*		
11		M4*	
12			M4*
13	M5*		
14		M5*	
15			M5*
16	M6*		
17		M6*	
18			M6*
19	M7*		
20		M7*	
21			M7*
22	M8*		
23		M8*	
24			M8*
25	M9*		
26		M9*	
27			M9*
28	M10*		
29		M10*	
30			M10*

\* Матрицы путей:

**M1**

№	1	2	3	4	5	6	7	8	9	10
1	0	3	5	7	1	4	6	9	2	5
2	3	0	3	5	7	3	2	1	5	7
3	5	3	0	4	5	7	4	1	2	6
4	7	5	4	0	5	7	2	4	3	9
5	1	7	5	5	0	7	2	2	1	8
6	4	3	7	7	7	0	1	6	9	3
7	6	2	4	2	2	1	0	5	2	1
8	9	1	1	4	2	6	5	0	5	2
9	2	5	2	3	1	9	2	5	0	4
10	5	7	6	9	8	3	1	2	4	0

**M2**

№	1	2	3	4	5	6	7	8	9	10
1	0	4	5	8	1	3	6	2	2	9
2	4	0	2	5	7	3	2	1	5	7
3	5	2	0	4	5	7	4	1	2	6
4	8	5	4	0	5	7	2	4	3	9
5	1	7	5	5	0	8	2	2	1	8
6	3	3	7	7	8	0	1	6	9	3
7	6	2	4	2	2	1	0	5	2	1
8	2	1	1	4	2	6	5	0	5	2
9	2	5	2	3	1	9	2	5	0	9
10	9	7	6	9	8	3	1	2	9	0

**M3**

№	1	2	3	4	5	6	7	8	9	10
1	0	1	8	8	2	3	6	2	2	8
2	1	0	1	5	7	3	2	1	5	7
3	8	1	0	2	5	7	4	1	2	6
4	8	5	2	0	1	7	2	4	3	9
5	2	7	5	1	0	9	2	2	1	8
6	3	3	7	7	9	0	9	6	9	3
7	6	2	4	2	2	9	0	7	2	1
8	2	1	1	4	2	6	7	0	5	2
9	2	5	2	3	1	9	2	5	0	2
10	8	7	6	9	8	3	1	2	2	0

**M4**

№	1	2	3	4	5	6	7	8	9	10
1	0	9	1	2	1	4	6	1	2	9
2	9	0	2	5	7	3	2	1	5	7
3	1	2	0	4	5	7	4	1	2	6
4	2	5	4	0	5	7	2	4	3	9
5	1	7	5	5	0	8	2	2	1	8
6	4	3	7	7	8	0	1	6	9	3
7	6	2	4	2	2	1	0	5	2	1
8	1	1	1	4	2	6	5	0	5	1
9	2	5	2	3	1	9	2	5	0	9
10	9	7	6	9	8	3	1	1	9	0

**M5**

№	1	2	3	4	5	6	7	8	9	10
1	0	6	7	2	1	4	7	8	2	9
2	6	0	2	5	7	3	2	1	5	7
3	7	2	0	3	5	7	4	1	2	6
4	2	5	3	0	5	9	2	4	3	9
5	1	7	5	5	0	8	2	2	1	8
6	4	3	7	9	8	0	2	6	9	3
7	7	2	4	2	2	2	0	5	2	1
8	8	1	1	4	2	6	5	0	5	1
9	2	5	2	3	1	9	2	5	0	9
10	9	7	6	9	8	3	1	1	9	0

**M6**

№	1	2	3	4	5	6	7	8	9	10
1	0	5	2	2	1	4	6	2	1	7
2	5	0	1	5	7	2	2	1	5	7
3	2	1	0	4	5	7	4	1	2	6
4	2	5	4	0	5	7	2	4	3	9
5	1	7	5	5	0	8	2	2	1	8
6	4	2	7	7	8	0	1	6	9	3
7	6	2	4	2	2	1	0	5	2	1
8	2	1	1	4	2	6	5	0	5	1
9	1	5	2	3	1	9	2	5	0	9
10	7	7	6	9	8	3	1	1	9	0

**M7**

№	1	2	3	4	5	6	7	8	9	10
1	0	4	2	3	8	4	5	2	8	9
2	4	0	2	5	7	3	2	1	5	7
3	2	2	0	3	5	7	4	1	2	6
4	3	5	3	0	1	7	3	4	3	9
5	8	7	5	1	0	3	2	2	1	8
6	4	3	7	7	3	0	1	6	9	3
7	5	2	4	3	2	1	0	7	2	1
8	2	1	1	4	2	6	7	0	5	1
9	8	5	2	3	1	9	2	5	0	9
10	9	7	6	9	8	3	1	1	9	0

**M8**

№	1	2	3	4	5	6	7	8	9	10
1	0	1	2	4	5	4	6	2	2	9
2	1	0	4	5	7	3	2	1	5	7
3	2	4	0	4	5	7	4	1	5	6
4	4	5	4	0	5	9	2	4	6	9
5	5	7	5	5	0	8	2	2	1	8
6	4	3	7	9	8	0	2	7	9	4
7	6	2	4	2	2	2	0	5	2	1
8	2	1	1	4	2	7	5	0	5	4
9	2	5	5	6	1	9	2	5	0	5
10	9	7	6	9	8	4	1	4	5	0

**M9**

№	1	2	3	4	5	6	7	8	9	10
1	0	5	1	7	5	9	1	9	2	9
2	5	0	3	5	7	3	2	1	5	7
3	1	3	0	7	5	7	4	1	2	6
4	7	5	7	0	5	1	2	4	3	9
5	5	7	5	5	0	8	2	2	1	8
6	9	3	7	1	8	0	2	6	8	3
7	1	2	4	2	2	2	0	4	2	1
8	9	1	1	4	2	6	4	0	4	2
9	2	5	2	3	1	8	2	4	0	8
10	9	7	6	9	8	3	1	2	8	0

**M10**

№	1	2	3	4	5	6	7	8	9	10
1	0	3	2	1	7	8	9	1	2	9
2	3	0	1	5	7	3	2	1	5	7
3	2	1	0	1	5	7	4	1	2	5
4	1	5	1	0	2	8	4	4	3	9
5	7	7	5	2	0	9	2	2	1	8
6	8	3	7	8	9	0	1	3	9	3
7	9	2	4	4	2	1	0	5	2	1
8	1	1	1	4	2	3	5	0	5	3
9	2	5	2	3	1	9	2	5	0	9
10	9	7	5	9	8	3	1	3	9	0

## Литература

1. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. – М.: Горячая линия-Телеком, 2006. – 452 с.
2. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы. – М.: ФИЗМАТЛИТ, 2006. – 320 с.
3. Путькина Л. В., Пискунова Т. Г. Интеллектуальные информационные системы. – Санкт-Петербург.: СПбГУП, 2008. – 228 с.
4. Андрейчиков А. В., Андрейчикова О. Н. Интеллектуальные информационные системы. – М.: Финансы и статистика, 2006. – 424 с.
5. Гаскаров Д. В. Интеллектуальные информационные системы. – М.: Высшая школа, 2003. – 432 с.
6. Круглов В.В., Дли М.И. Интеллектуальные информационные системы: Компьютерная поддержка систем нечеткой логики и нечеткого вывода: Учебное пособие для вузов. – М., 2002 г. – 256 с.
7. Материалы: Генетические алгоритмы [Электронный ресурс]. Режим доступа: [http://www.gotai.net/documents-genetic\\_algorithms.aspx](http://www.gotai.net/documents-genetic_algorithms.aspx), свободный. – Загл. с экрана.
8. Генетические алгоритмы [Электронный ресурс]. Режим доступа: <http://algolist.manual.ru/ai/ga>, свободный. – Загл. с экрана.
9. Искусственный интеллект/Блог/Хабрахабр [Электронный ресурс]. Режим доступа: [http://habrahabr.ru/blogs/artificial\\_intelligence/](http://habrahabr.ru/blogs/artificial_intelligence/), свободный. – Загл. с экрана.
10. Генетические алгоритмы – математический аппарат [Электронный ресурс]. Режим доступа: [http://www.basegroup.ru/library/optimization/ga\\_math/](http://www.basegroup.ru/library/optimization/ga_math/), свободный. – Загл. с экрана.
11. Генетические алгоритмы: почему они работают? когда их применять? – Журнал «Компьютерра» [Электронный ресурс]. Режим доступа: <http://www.computerra.ru/offline/1999/289/2523/>, свободный. – Загл. с экрана.