

*УДК 681.3*

*Антон Крижановский  
Татьяна Крижановская*

**ГЕНЕТИЧЕСКИЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧ  
ПОИСКА ОПТИМАЛЬНЫХ ПУТЕЙ**

*У статті представлені теоретичні дослідження особливостей роботи генетичних алгоритмів та можливість їх реалізації для вирішення широкого спектру завдань.*

*В статье представлены теоретические исследования особенностей работы генетических алгоритмов и возможность их реализации для решения широкого спектра задач.*

*The paper presents theoretical investigations of the features of genetic algorithms and their feasibility for a wide range of applications.*

**Ключевые слова:** генетический алгоритм, задача оптимизации, моделирование.

Генетические Алгоритмы (ГА) – адаптивные методы поиска, которые в последнее время часто используются для решения задач функциональной оптимизации. Они основаны на генетических процессах биологических организмов: биологические популяции развиваются в течении нескольких поколений, подчиняясь законам естественного отбора и по принципу «выживает наиболее приспособленный», открытому Чарльзом Дарвином. Используя этот процесс, генетические алгоритмы способны «развивать» решения реальных задач, если те соответствующим образом внесены в программу. ГА могут быть эффективно применяться для решения задач проектирования транспортных маршрутов, управления технологическими процессами производства, оптимизации загрузки процессоров в многопроцессорных система.

Основателем генетических алгоритмов считается Джон Холланд (John Henry Holland), основные подходы к построению ГА изложены в труде «Адаптация в естественных и искусственных системах» (1975).

В отличие от эволюции, происходящей в природе, ГА только моделируют те процессы в популяциях, которые являются существенными для развития. Точный ответ на вопрос: какие биологические процессы существенны для развития, и какие нет? – все еще открыт для исследователей. Задача формализуется таким образом, чтобы её решение могло быть закодировано в виде вектора («генотипа») генов, где каждый ген может быть битом, числом или другим объектом. В классических реализациях ГА предполагается, что генотип имеет фиксированную длину.

© Крижановский А.В., Крижановская Т.В., 2012

## МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ І ТЕОРЕТИЧНА МЕХАНІКА

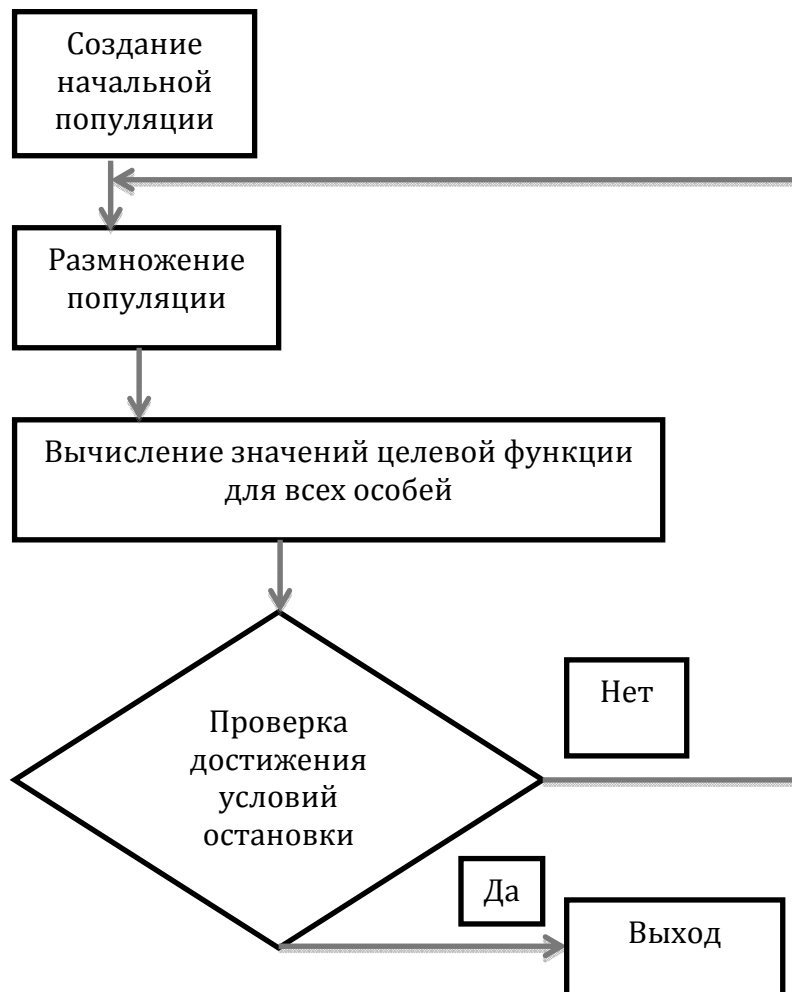
Однако существуют вариации ГА, свободные от этого ограничения.

Некоторым, обычно случайным образом создаётся множество генотипов начальной популяции. Они оцениваются с использованием «функции приспособленности», в результате чего с каждым генотипом ассоциируется определённое значение («приспособленность»), которое определяет, насколько хорошо фенотип, им описываемый, решает поставленную задачу.

Из полученного множества решений («поколения») с учётом значения «приспособленности» выбираются решения (обычно лучшие особи имеют большую вероятность быть выбранными), к которым применяются «генетические операторы» (в большинстве случаев «скрещивание» и «мутация»), результатом чего является получение новых решений. Для них также вычисляется значение приспособленности, и затем производится отбор («селекция») лучших решений в следующее поколение. Этот набор действий повторяется итеративно, так моделируется «эволюционный процесс», продолжающийся несколько жизненных циклов (*поколений*), пока не будет выполнен критерий останова алгоритма. Таким критерием может быть:

- нахождение глобального, либо субоптимального решения;
- исчерпание числа поколений, отпущенных на эволюцию;
- исчерпание времени, отпущенного на эволюцию.

**Этапы генетического алгоритма:**



## МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ І ТЕОРЕТИЧНА МЕХАНІКА

---

В последние годы реализовано много вариантов ГА, и в настоящее время под термином «генетические алгоритмы» скрывается не одна модель, а достаточно широкий класс алгоритмов, подчас мало похожих друг на друга. Исследователи экспериментировали с различными типами представлений, операторов кроссовера и мутации, специальных операторов, и различных подходов к воспроизводству и отбору.

Наиболее популярное приложение ГА – оптимизация многопараметрических функций. Многие реальные задачи могут быть сформулированы как поиск оптимального значения, где значение – сложная функция от нескольких параметров. Представляет интерес нахождение значения параметров, при которых достигается наилучшее точное значение функции. В других случаях точный оптимум не требуется, так как решением может считаться любое значение, которое лучше некоторой заданной величины. В этом случае генетические алгоритмы – часто наиболее приемлемый метод для поиска «хороших» значений. ГА использовалась в сотнях прикладных программ, включая проектирование самолетов, настройку параметров алгоритмов и поиск устойчивых состояний систем нелинейных дифференциальных уравнений.

Рассмотрим использование ГА на примере задачи минимизации полиномиальных представлений булевых функций (задача кратчайшего пути с большим количеством входных переменных и сложностью эффективного прогнозирования влияния переменных на результат в связи с их нелинейным влиянием).

Любую булеву функцию можно представить в виде полинома:

$$f(x_1, x_2, x_3) = x_1 \cdot x_2 \cdot x_3 \oplus x_1 \cdot x_2 \cdot x_3 \oplus x_1 \cdot x_2 \cdot x_3$$

Для решения задачи минимизации полиномиального представления булевых функций был реализован ГА с использованием языка C++.

**Начальные параметры алгоритма:** Функция, сложность которой определяется; Количество итераций, которое должен пройти алгоритм; количество мутаций, которое произойдет за время выполнения алгоритма; количество генов, которые претерпевают мутацию; количество инверсий, которое произойдет за время выполнения алгоритма; количество генов, которые претерпевают инверсию; размер популяции; тип генерирования начальной популяции; тип кроссовера; тип закона распределения при осуществлении выборки особей. Как и в классическом ГА, в виде функций было реализовано несколько генетических операторов. Реализованный алгоритм выполняет следующие стадии:

1. Инициализация начальной популяции: на первом шаге генерируется популяция предков, которая даст в результате выполнения операторов селекции и скрещивания новое потомство. Популяция представляет собой динамический двумерный массив типа `unsigned int` (один элемент – 32 бита) и, в зависимости от длины функции, состоит из нескольких элементов одной записи массива. Это позволяет легко работать с функциями любой длины. Функция `Gen_ind()` принимает несколько параметров, главным из которых является тип генерирования первоначальной популяции:

а. Конкретная хромосома представлена набором случайных генов, представляющих собой «0» или «1»;

б. Каждый элемент хромосомы принимает случайное значение в диапазоне от 0 до  $2^{32}-1$  – максимальное значение, которое может принять переменная типа `unsigned int`.

## МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ І ТЕОРЕТИЧНА МЕХАНІКА

---

с. Также как и в п.2, только каждый элемент хромосомы разбивается на 8 частей по 4 бита.

d. Случайный выбор из первых 3-х вариантов.

2. Следующим этапом происходит оценивание хромосом. В качестве функции приспособленности выступает сложность булевых функций. Для достижения этой реализована функция AbsMin(), которая возвращает точное значение сложности любой функции.

Оператор селекции Selection() возвращает двух представителей из массива случайным образом. Для этого используются следующий закон распределения вероятностей. Равномерный – вероятность выбора того или иного представителя одинакова. Случайная величина имеет непрерывное равномерное распределение на отрезке [a,b], где её плотность  $f(x)$  имеет вид:

$$f(x) = \begin{cases} \frac{1}{b-a}, & x \in [a, b] \\ 0, & x \notin [a, b] \end{cases}.$$

В зависимости от количества мутаций, которое задается, как параметр ГА, следующим  $\square$ этапом выступает оператор мутации (функция mutation()). Далее последовательно идут несколько операторов ГА. Оператор инверсии (функция Inv()) – зеркально отображает часть генов в хромосоме конкретного представителя. Основной параметр – число генов, которые будут зеркально отображены. Сортировка (функция bubbleSort()) – сортирует представителей по возрастанию функции приспособленности, в качестве которой выступает сложность булевой функции. Завершающим этапом каждой итерации является формирование новой популяции, т.к. и родительская популяция и популяция потомков отсортированы, т.е. упорядочены от «лучших» к «худшим», поэтому в новую родительскую популяцию извлекается половина представителей родительской популяции и половина популяции потомков. Тем самым, качественные решения задачи – потомки, не выпадают в ходе дальнейшего выполнения алгоритма. После выполнения всех этих этапов алгоритм возвращается к п. 2 и продолжается то число раз, которое пользователь задал при запуске данной программы.

Сферами применения выше изложенного ГА являются оптимизация функций, игровые стратегии, оптимизация запросов в базах данных, теория приближений и моделирование процессов жизнедеятельности.

### ЛИТЕРАТУРА

1. *J. H. Holland*. Adaptation in natural and artificial systems. University of Michigan Press, 1975.– 183 с.
2. *Емельянов В. В., Курейчик В. В., Курейчик В. М.* Теория и практика эволюционного моделирования. – М: Физматлит, 2003. – 427с.
3. *R.P.Clement & A.Wren*, «Genetic Algorithms and Bus-Driver Scheduling». Presented at the 6th International Conference for Computer-Aided Transport Scheduling, Lisbon, Portugal, 1993.
4. *Еремеев А.В.* Разработка и анализ генетических и гибридных алгоритмов для решения задач дискретной оптимизации. Дисс. канд. физ.-мат. наук. Омск, 2000. – 119 с.
5. Genetic Operators, D.Whitley and K. Mathias, Parallel Problem Solving from Nature-PPSN R. Mainer and B Manderick, North Holland-Elsevier, 1992.