

## СОВРЕМЕННЫЕ МОДИФИКАЦИИ И ОБОБЩЕНИЯ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

Скобцов Ю.А.

Донецкий национальный технический университет,  
ул.Артема 58, Донецк, Украина, 83000  
E-MAIL: *skobtsov@kita.dgtu.donetsk.ua*

Скобцов В.Ю.

Институт прикладной математики и механики НАН Украины,  
ул.Р.Люксембург 74, Донецк, Украина, 83114  
E-MAIL: *skobtsov@iamm.ac.donetsk.ua*

### Abstract

This paper presents the survey of modern promising direction in computer science and engineering-genetic algorithms. Genetic algorithms are base on the formalization of principles of natural evolution and have successful applications in different areas of modern science, especially in the problems that cannot be solved effectively with deterministic methods. In the past few decades, this direction has been deeply investigated and significant results have been reached. It is discussed the advantages and disadvantages and state-of-the-art of genetic algorithms.

### Введение

Генетические алгоритмы (ГА), основанные на формализации принципов естественной эволюции, в настоящее время успешно применяются при решении различных сложных задач и постоянно развиваются [1]. Стандартный ГА можно представить следующей последовательностью действий (Рис. 1): В настоящее время предложены многочисленные модификации и обобщения ГА, в которых: 1) разработаны различные варианты реализации каждого этапа; 2) существенно изменена структура алгоритма. Среди первой группы можно выделить различные методы отбора родителей (пропорциональный, турнирный, локальный, ранжирование и т.д.), сокращения популяции (элитарный, случайный, пропорциональный, селекционный и т.д.). Разработаны различные генетические операторы кроссинговера двоичных (одноточечный, многоточечный, однородный) и действительных значений (дискретная, промежуточная, линейная рекомбинация), мутации (однородная и неоднородная для двоичных и вещественных значений).

Среди второй группы можно отметить ГА с динамически изменяемой мощностью популяции (различные стратегии выбора срока жизни особи), адаптивные ГА с подстраиваемыми параметрами (вероятности кроссинговера и мутации), параллельные алгоритмы (модель островов, глобальная модель рабочий - хозяин, диффузионная модель), иерархические (многоуровневые) ГА.

Рассмотрим сначала различные методы реализации основных этапов классической схемы ГА.

Создание исходной популяции

Выбор родителей для процесса размножения (работает оператор отбора - репродукции)

Создание потомков выбранных родителей (работает оператор скрещивания - кроссинговера)

а-  
и)

Для одной переменной при этом отрезок  $[a, c]$ , соответствующий области решения, должен быть разбит на  $\frac{|c-a|}{\varepsilon}$  равных частей. В качестве двоичного представления используется двоичный код номера отрезка. Этот код позволяет определить соответствующее ему вещественное число, если известны границы области решения (точнее ее левый край). Отображение из двоичного представления  $(b_n b_{n-1} \dots b_0)$  ( $b_i = 0, 1$ ) в вещественное число из отрезка  $[a, c]$  соответственно выполняется в два шага.

1. Перевод двоичного числа в десятичное :  $((b_n b_{n-1} \dots b_0))_2 = \left( \sum_{i=0}^n b_i 2^i \right)_{10} = x'$
2. Вычисление соответствующего реального числа  $x = a + x' \frac{|c-a|}{2^n - 1}$ .

Естественно хромосомы  $(00 \dots 0)$  и  $(11 \dots 1)$  представляют границы отрезка  $[a, c]$  соответственно. Очевидно, при данном двоичном представлении вещественных чисел можно использовать классический простой ГА. С одной стороны это универсальный метод кодирования, с другой он не всегда эффективен в силу неадекватности данного представления для многих задач.

Например, такое представление вещественного числа, имеет существенный недостаток: расстояние между реальными числами (на числовой оси) часто не соответствует расстоянию (по Хеммингу) между их двоичными представлениями. Поэтому желательно получить двоичное представление, где близкие расстояния между хромосомами (двоичными представлениями) соответствовали близким расстояниям в проблемной области (в данном случае расстояние на числовой оси). Это можно сделать, например, с помощью кода Грея.

Решение задач комбинаторной оптимизации, как правило, требует своего метода кодирования особи. Например, для решения задачи коммивояжера предложено и исследовано множество возможных способов представления особей - от списков (посещаемых в туре) городов до двоичных матриц [3, 4]. Известно множество способов представления особей от классического двоичного вектора до нечеткого (fuzzy) кодирования. В общем случае, как правило, каждая нетривиальная задача требует для эффективного решения выбора или разработки своего способа представления особей и создания проблемно-ориентированных генетических операторов кроссинговера и мутации.

## 2. СОЗДАНИЕ ИСХОДНОЙ ПОПУЛЯЦИИ

В настоящее время наиболее известными и применяемыми на практике являются 3 способа создания исходной популяции (начального множества решений).

**Стратегия «одеяла».** — формирование полной популяции, содержащей все возможные решения. Например, для  $n$ -разрядной хромосомы мы имеем всего  $2^n$  вариантов решений, которые составляют полную популяцию.

**Стратегия «дробовика».** — генерация достаточно большого случайного подмножества решений.

**Стратегия фокусировки.** — генерация множества решений, включающих разновидности одного решения. Первый подход (стратегия «одеяла») практически не реализуем даже для задач средней размерности, вследствие больших вычислительных затрат. Заметим, что при этом подходе начальная популяция не может развиваться, т.к. в ней уже содержатся все возможные решения.

Третий способ используется тогда, когда есть предположение, что некоторое решение является вариацией известного значения. В этом случае время поиска оптимального решения существенно сокращается, т.к. алгоритм начинает работу в окрестности оптимума. Чаще всего применяют второй способ. В этом случае в результате эволюции есть возможность перейти в другие подобласти поиска и каждая из них имеет сравнительно небольшое пространство поиска. В целом эффективность ГА, качество решения и дальнейшая эволюция в значительной степени определяются структурой и качеством начальной популяции. На практике часто применяют комбинацию второго и третьего способов. Сначала определяются особи с высокими значениями целевой функции, а затем случайно формируются начальные решения в этих подобластях.

### 3. ОТБОР РОДИТЕЛЕЙ (СЕЛЕКЦИЯ)

Метод выбора родителей для последующего производства потомков с помощью генетических операторов кроссинговера и мутации оказывает существенное влияние на эффективность ГА. Оператор отбора  $S$  порождает промежуточную популяцию  $\widetilde{P}^t$  из текущей популяции  $P^t$  путем отбора и генерации новых копий особей  $P^t \rightarrow^s \widetilde{P}^t$ . При отборе конкурентно-способных особей используют целевую (fitness) функцию, как единственно доступный источник информации о качестве решения. Но различные методы отбора родителей по-разному используют эту информацию. Слишком сильное предпочтение лучшим особям ведет к чрезмерному сужению области поиска решения и чревато преждевременной сходимостью в локальных экстремумах и, наоборот, пренебрежение ими может привести к другой крайности - случайному поиску. Далее мы рассмотрим наиболее распространенные методы отбора родителей. Отбор особи  $a_i^t$  текущей популяции  $P^t$  в промежуточную популяцию  $\widetilde{P}^t$  ( $t$  - номер поколения) выполняется на основе вероятности  $P_s(a_i^t)$ , определенной для каждого индивидуума текущей популяции. Способы вычисления вероятности отбора особи  $P_s(a_i^t)$  определяют различные методы отбора [4, 5]:

**пропорциональный отбор**(метод рулетки)

$$P_s(a_i^t) = \frac{f(a_i^t)}{\sum_{j=1}^N f(a_j^t)},$$

где  $N$  - мощность популяции;

**линейное ранжирование**

$$P_s(a_i^t) = \frac{1}{N} \left( \alpha - (\alpha - \beta) \frac{i - 1}{N - 1} \right),$$

где  $1 \leq \alpha \leq 2$  выбирается случайным образом,  $\beta = 2 - \alpha$ ,

**равномерное ранжирование**

$$P_s(a_i^t) = \begin{cases} \frac{1}{\mu}, & 1 \leq i \leq \mu \\ 0, & \mu < i \leq N \end{cases},$$

где  $\mu \leq N$ - параметр метода.

Существуют также методы отбора, основанные не на вероятностном подходе. **Локальный отбор** производится среди особей, которые находятся в некоторой ограниченной среде, где определено отношение соседства на различных структурах: линейное, двумерное (4-,6-,8-связное). При отборе родителей на первом шаге производится отбор особей случайным образом, или одним из ранее рассмотренных способов. Далее, для каждой отобранной особи определяется множество локальных соседей и среди них выбирается партнер для выполнения операций скрещивания. Предыдущие методы отбора в какой-то степени имеют аналогию с эволюцией естественных популяций. **Отбор на основе усечения** не имеет аналогов в естественной эволюции и обычно используются для больших популяций ( $N > 100$ ). При этом сначала отбираемые особи упорядочиваются согласно их значениям целевой функции. Затем, в качестве родителей выбираются только лучшие особи. Далее, с равной вероятностью, среди них случайно выбирают пары, которые производят потомков. При этом методе используется параметр - порог отсечения (иногда используется термин интенсивность отбора), показывающий долю (часть популяции), которая отбирается в качестве родителей. Обычно  $10\% \leq T \leq 50\%$ .

При **турнирном отборе** из популяции случайно отбираются  $m$  особей, затем лучшая из них выбирается в качестве родителя. Далее этот процесс повторяется, пока не сформируется промежуточная популяция, в которой случайно формируются пары родителей. Параметром этой процедуры является размер тура  $m$ , который принимает значения из диапазона  $2 \leq m < N$ .

#### 4. ОПЕРАТОРЫ КРОССИНГОВЕРА

Различают операторы двоичной и вещественной рекомбинации (скрещивания).

**Двоичный кроссинговер.** При **одноточечном кроссинговере** случайно выбирается точка скрещивания с вероятностью  $P_c \approx 0.5$ , и производится обмен фрагментами хромосом после точки скрещивания, как показано в примере на рис. 2.

A:	0	1	1	1	0	1	0	1	0
B:	1	0	0	1	1	0	1	0	1
					⇕				
A':	0	1	1	1	0	0	1	0	1
B':	1	0	0	1	1	1	0	1	0

Рис. 2. Одноточечный двоичный кроссинговер

В случае **многоточечного кроссинговера** выбираются  $m$  позиций  $k_i \in 1, 2, \dots, N - 1$  и их номера сортируются по порядку. Затем родители обмениваются фрагментами, заключенными между соседними выбранными позициями и производят, таким образом, двух потомков. То есть обмен здесь идет секциями. При этом секции между первой позицией и первой точкой скрещивания не обмениваются, а далее обмен идет через одну секцию, как это показано на рис. 3.

A	0	1	1	1	0	0	1	1	0	1	0
B	1	0	1	0	1	1	0	0	1	0	1
			⇕								⇕
A':	0	1	1	0	1	1	1	1	0	1	1
B':	1	0	1	1	0	0	0	0	1	0	0

Рис. 3. Многоточечный двоичный кроссинговер

**Однородный кроссинговер.** радикально отличается от предыдущих видов. Здесь каждый ген потомка создается путем копирования соответствующего гена из первого или второго родителя. Для этого случайным образом генерируется двоичная маска кроссинговера той же длины (с тем числом бит), что у хромосом родителей. Четность бита маски показывает родителя, из которого копируется ген потомка (например, 1 соответствует первому родителю, а 0 - второму). На рис. 4 показана схема выполнения этого типа кроссинговера на примере.

Маска ОК	1	0	0	1	0	1	1	1	0	0
1-й родитель	1	0	1	0	0	0	1	1	1	0
	↓			↓		↓	↓	↓		
Потомок	1	1	0	0	0	0	1	1	1	1
		↑	↑		↑				↑	↑
2-й родитель	0	1	0	1	0	1	0	0	1	1

Рис. 4. Однородный двоичный кроссинговер

При **ограниченном кроссинговере** точки скрещивания могут вбираться только там, где значения генов у родителей различны.

**Кроссинговер действительных значений.** Данная группа операторов разработана для хромосом, представленных действительными числами [5].

**Дискретный кроссинговер.** определен над векторами, компонентами которых являются вещественные числа, и выполняется аналогично однородному кроссинговеру, определенному над двоичными векторами.

1-й родитель	12	25	5	17	39
2-й родитель	123	4	34	-5	11
1-й образец (маска)	2	2	1	1	2
2-й образец (маска)	1	2	1	2	1
1-й потомок	123	4	5	17	11
2-й потомок	12	4	5	-5	39

Рис. 5. Дискретный кроссинговер вещественных векторов

1-й родитель	12	25	5	17	39
2-й родитель	123	4	34	-5	11
Случайно выбраны следующие значения коэффициента $\alpha$					
1-й образец (маска)	0,5	1,1	0,1	0,7	0,4
2-й образец (маска)	0,1	0,8	0,5	1,1	0,3
1-й потомок	67,5	1,9	2,1	1,6	27,8
2-й потомок	23,1	8,2	19,5	-7,2	30,6

Рис. 6. Промежуточный кроссинговер

**Промежуточный кроссинговер.** применим только для особей, представленных только вещественными значениями. Здесь значения потомков строятся в окрестности или между значениями родителей. Потомок формируется следующим образом:

$$A' = A + \alpha \cdot (B - A),$$

где  $A, B$  — вещественные значения — родители;  $A'$  — вещественное значение — потомок;  $\alpha \in [-d, d + 1]$  — случайный масштабирующий множитель. В случае обычного промежуточного кроссинговера  $d = 0$  и  $\alpha \in [0, 1]$ . Для обобщенного промежуточного кроссинговера  $d > 0$ . Значение каждого потомка формируется по приведенному выражению, но со своим  $\alpha$ . В случае особи-вектора случайным образом формируется

вектор масштабирующий множителей  $\alpha = (\alpha_1, \alpha_1, \dots, \alpha_n)$  и потомок-вектор определяется покомпонентно (Рис. 6). Фактически, этот оператор заимствован из другого направления эволюционных вычислений — «эволюционные стратегии».

**Линейный кроссинговер.** аналогичен предыдущему, за исключением того, что значение масштабирующего множителя  $\alpha$  одинаково для всех компонент векторов.

## 5. Операторы мутации

После выполнения операторов кроссинговера полученные потомки с вероятностью  $P_m$  подвергаются мутации, которая может быть выполнена различными способами [5].

### Двоичная мутация

В **классической мутации** для каждой особи случайно выбирается позиция и с малой вероятностью  $P_m$  от 0,001 до 0,01 выполняется инвертирование значения в выбранной позиции.

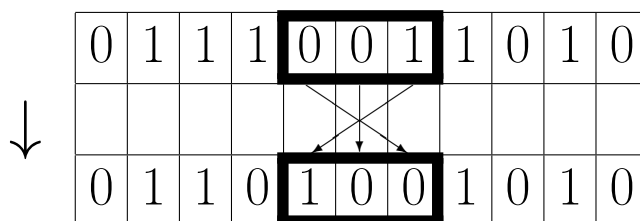


Рис. 7. Мутация-инверсия

Иногда используется **оператор инверсии**, фактически являющийся разновидностью классической мутации. При этом случайным образом выбираются 2 позиции в особи и далее производится обмен значениями генов между ними. Пример выполнения этого оператора представлен на рис. 7.

### Мутация над вещественными числами

Мутация над вещественными потомками выполняется путем сложения особи с небольшим случайным значением, которое называется шагом мутации. Выбор размера шага мутации зависит от рассматриваемой проблемы, и шаг в общем случае может изменяться в процессе решения задачи. Маленький шаг дает большую точность, но ведет к большим временным затратам. Мутация с постоянным шагом и постоянной вероятностью называется **однородной** и выполняется следующим образом

$$A_m = A \pm r \cdot \Delta,$$

где  $A$ ,  $A_m$  — значения вещественной особи до и после мутации. Часто для повышения эффективности поиска вероятность мутации и шаг изменяются в процессе



решения задачи. Мутация с постоянной вероятностью может привести как к увеличению, так и к уменьшению значения целевой функции. На этапе сходимости ГА к оптимуму целесообразно уменьшать вероятность случайной мутации. Обычно на начальном этапе  $P_m^0 = 0.05 \dots 0.01$ . А на конечном этапе вероятность мутации уменьшают. Для реализации этой процедуры иногда используют метод моделирования отжига (simulation annealing), который дает следующий закон изменения вероятности мутации:  $P_m = P_m^0 \cdot e^{-\frac{1}{t}}$ , где  $t$  — номер поколения. Аналогично может изменяться и шаг мутации.

### 6. Сокращение промежуточной популяции

Необходимой процедурой в ГА является устранение плохих решений, полученных в процессе формирования новой популяции — **редукция**. Промежуточную популяцию (репродукционную группу) составляют все особи  $t$ -го поколения, полученные в результате скрещивания и мутации. Тогда численность этой популяции можно определить следующим образом

$$R^{t+1} = r^t + r_{cr}^t + r_m^t,$$

где  $r^t$  — число особей предыдущей популяции;  $r_{cr}^t$  — численность особей, полученных путем скрещивания;  $r_m^t$  — число «мутантов». Обычно в стационарных ГА мощность популяции поддерживается постоянной  $N = |P(t)|$ . Поэтому поскольку  $R^{t+1} > N$ , то необходимо устранить неудачные решения. Для этого существуют различные методы редукции.

**Чистая замена.** В простейшем случае с помощью скрещивания и мутации генерируются столько потомков, сколько было родителей. Далее родители устраняются, а потомки формируют следующее поколение  $P(t+1)$ . При этом каждая особь живет лишь одно поколение. Такая схема часто используется в простом ГА. Однако при этом очевидно возможно, что некоторые очень хорошие решения могут быть заменены худшими, и лучшее решение будет потеряно.

**Элитарная схема.** В ней потомков генерируется меньше, чем было родителей. Далее вновь построенные потомки заменяют худших родителей согласно значениям ЦФ. Для этой схемы возможна преждевременная сходимость к локальным экстремумам.

**Равномерная случайная замена.** При таком подходе потомков также генерируется меньше, чем было родителей, далее случайно удаляется необходимое число родителей, которые заменяются новыми потомками.

**Пропорциональная редукция.** Здесь потомков генерируется больше, чем необходимо для замены. Потом заменяют заданное число родителей только лучшими потомками согласно значениям ЦФ.

**Селекционная схема.** Здесь родители и потомки выступают на равных правах, все они помещаются в одну репродукционную группу. В ней все особи этой группы ранжируются и в следующее поколение включаются только лучшие особи. Иногда применяется следующая модификация этого метода. Сначала вычисляется среднее значение ЦФ группы и в следующее поколение включаются те особи, у которых значение ЦФ больше среднего значения группы.

## 7. ГА с изменяемой мощностью популяций

В этой модификации ГА отсутствует этап и механизм селекции родителей [3]. Вместо этого вводится концепция возраста особи. Каждой особи после ее рождения на текущем этапе оценки ЦФ присваивается *life time* (срок жизни) — параметр, зависящий от ЦФ особи. Таким образом, каждая особь живет определенное число поколений и умирает по окончании заданного срока жизни. Очевидно, значение параметра *life time* влияет на размер популяции. Чем больше срок жизни, тем в большем числе поколений популяций живет данная особь, что увеличивает средний размер популяции. Структура такого ГА представлена ниже псевдокодом:

*Procedure*

*Begin*  $t = 0$

*Инициализация начальной популяции*  $p(t)$ ;

*Оценка ЦФ*  $p(t)$ ;

*Определение срока жизни особей популяции;*

*While* (*условие окончания не выполнено*)

*Begin*

$t = t + 1$ ;

*Увеличение возраста каждой особи на 1;*

*Рекомбинация*  $p(t)$ ;

*Мутация*  $p(t)$ ;

*Оценка ЦФ*  $p(t)$ ;

*Определение срока жизни особей;*

*Удаление из*  $p(t)$  *всех особей с возрастом больше срока жизни;*

*End;*

*End.*

**Стратегии определения срока жизни.** Очевидно, постоянное значение срока жизни  $L_f \geq 2$  для каждой особи ведет к экспоненциальному росту размера популяции в последующих поколениях. Для каждой особи срок жизни вычисляется индивидуально на основании ее значения ЦФ. Введем обозначения:

$\bar{f}$  — среднее значение ЦФ по популяции;  $f_{max}$  — максимальное значение ЦФ по популяции;  $f_{min}$  — минимальное значение ЦФ по популяции;  $f_{amax}$  — абсолютное

максимальное значение;  $f_{amin}$  — абсолютное минимальное значение;  $L_{max}$  — максимальный срок жизни;  $L_{min}$  — минимальный срок жизни. Существует три основных способа определения срока жизни.

**Пропорциональный метод определения срока жизни.** Срок жизни в данном случае определяется по формуле:

$$L_f = \left\{ L_{min} + \eta \cdot \frac{f[i]}{\bar{f}}, L_{max} \right\},$$

где  $\eta = \frac{1}{2} \cdot (L_{max} - L_{min})$  (используется для всех формул).

*Линейный метод определения срока жизни.*

$$L_f = L_{min} + 2 \cdot \eta \cdot \frac{f[i] - f_{amin}}{f_{amax} - f_{amin}}$$

*Билинейный метод определения срока жизни.*

$$L_f = \begin{cases} L_{min} + \eta \cdot \frac{f[i] - f_{min}}{f - f_{min}}, & \text{если } \bar{f} \geq f[i] \\ \frac{1}{2} \cdot (L_{min} + L_{max}) + \eta \cdot \frac{f[i] - \bar{f}}{f_{max} - f}, & \text{если } \bar{f} < f[i] \end{cases}$$

## 8. Параллельные ГА

Миграционная модель (модель островов) разбивает популяцию решений на множество подпопуляций (одна большая разбивается на множество мелких). Эти подпопуляции функционируют независимо друг от друга некоторое число поколений (срок изоляции). По достижению срока изоляции производится обмен между подпопуляциями. При этом задаются:

- число обмениваемых особей — степень миграции,
- метод отбора особей для миграции,
- схема миграции, которая определяет способ обмена между подпопуляциями.

Выбор особей для миграции может осуществляться несколькими методами, как и отбор родителей. Существует также несколько стандартных схем обмена отобранными особями между подпопуляциями:

- каждая подпопуляция обменивается с каждой из остальных подпопуляций;
- обмен по кольцу;
- обмен между соседними подпопуляциями.

Параллельная реализация миграционной модели показала не только увеличение скорости вычислений, но и меньшее число событий — вычислений значений ЦФ для особей по сравнению с обычными ГА (одна популяция). Поэтому даже для однопроцессорного компьютера последовательная реализация параллельного алгоритма (псевдопараллельная реализация) обычно дает лучшие результаты.

В глобальной модели «Рабочий-Хозяин» на нижнем уровне каждый «рабочий» имеет свою популяцию, каждая из которых развивается самостоятельно. На верхнем

уровне «хозяин» получает с нижнего уровня значения ЦФ из всех подпопуляций и управляет ими путем коррекции основных параметров ГА нижнего уровня таких как мощность популяции, вероятности скрещивания и мутации и т.п.

*Диффузионная модель* работает с каждой особью индивидуально и выбирает партнера для скрещивания подобно локальному отбору родителей. Таким образом, имеет место «диффузия» информации в популяции. Кроме рассмотренных выше, существуют и другие модификации ГА: адаптивные, где параметры (например,  $P_c$ ,  $P_m$ ) подстраиваются в процессе эволюции; коэволюционные, учитывающие изменение окружающей среды; нечеткие (fuzzy) ГА [6], в которых нечеткая логика используется для кодирования особей, адаптации параметров, определения останова и т.д.

#### СПИСОК ЛИТЕРАТУРЫ

1. Holland J.P. Adaption in natural and artificial systems. An arbor: the uni of Michigan press, 1975.
2. Goldberg, David E. Genetic Algorithms in Search, Optimization Machine Learning. Addison-WesleyPublishing Company, Inc., 1989.
3. Z. Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, second edition, Berlin, 1994.
4. M. Mitchel. An Introduction to Genetic Algorithms. MIT Press, 1998.
5. Genetic and evolutionary Algorithms: Principles, Methods and Algorithms. GEA Toolbox.
6. Курейчик В.М. Генетические алгоритмы. Таганрог: Изд-во ТРТУ, 1998.
7. Herrera F., Lozano M. Adaptive genetic operators based on coevolution with fuzzy behavior. IEEE Trans. On Evolutionary computation. 2000.