

# Задание практикума: генетический алгоритм

Дмитрий Волканов, Алексей Сальников

2022

## Содержание

1. Введение	1
2. Кодирование особи	2
3. Требования к программе	2
4. Методические указания	2
4.1. Шаги генетического алгоритма	2
4.2. Генерация начальной популяции	2
4.3. Вычисление целевой функции	4
4.4. Произведение селекции	4
4.5. Произведение скрещиваний	4
4.6. Организация мутаций	5

## 1. Введение

Требуется найти экстремум функции одной переменной на заданном интервале генетическим алгоритмом [1].

Размер начальной популяции *population\_volume* является параметром программы. Значение по умолчанию равно 30. Начальная популяция формируется случайным образом. Количество скрещиваемых особей (допустимых решений) и вероятность мутации особи задаются в параметрах программы.

Все функции заданы на отрезке  $[0, 4]$ . В функциях 1,4-8 необходимо найти *max* и *argmax*, а в функциях 2,3 необходимо найти *min* и *argmin*

Далее перечислены функции:

$$f_1(x) = (x - 2)(x - 2.5)(x - 3.5)(1 - \exp^{x-1.5}) \quad (1)$$

$$f_2(x) = (x - 2.1)(x - 1.5)(x - 2.4)(x - 0.33)(1 - \exp^{x-3.5})\cos(x) \quad (2)$$

$$f_3(x) = (x - 2)(x - 0.5)(x - 0.25)(x - 1.5)\sin\left(\frac{x}{5}\right) \quad (3)$$

$$f_4(x) = (x - 1)^5(x - 0.05)(x - 3)(x - 3.5)(1 - \exp^{x-3.95})\ln(x + 0.22) \quad (4)$$

$$f_5(x) = (x - 3)(x - 2)(x - 0.01)^4(x - 3.99)^4(1 - \exp^{x-1.5})\sin\left(\frac{x}{3} + 0.2\right) \quad (5)$$

$$f_6(x) = x(x - 1.1)^5(x - 1.2)^4(x - 1.3)^3\cos(x + 100) \quad (6)$$

$$f_7(x) = x \cdot \sin(x + 5)\cos(x - 6)\sin(x + 7)\cos(x - 8)\sin\left(\frac{x}{3}\right) \quad (7)$$

$$f_8(x) = x(x - 2)(x - 2.75)\cos\left(\frac{x}{10}\right)(2 - 3^{x-2})\exp^{\frac{x}{10}} \quad (8)$$

Критерий останова алгоритма:

1. Выполнение алгоритмом априорно заданного числа итераций *max\_iters*.
2. Выполнение алгоритмом априорно заданного числа итераций без улучшения качества популяции при заданном *quality\_epsilon* (По умолчанию  $10^{-5}$ ). *max\_valueless\_iters*.
3. Достижение некоторого априорно заданного значения целевой функции *enough\_function\_value*.

По окончании работы программы необходимо вывести какие из условий остановки алгоритма сработали.

**Виды селекции:** Случайная схема; При помощи рулетки; Отбор усечением.

**Виды скрещивания:** Одноточечное; Двухточечное; Универсальное; Однородное.

Виды мутации:

1. Изменение случайно выбранного бита
2. Перестановка случайно выбранных битов местами
3. Реверс битовой строки, начиная со случайно выбранного бита

## 2. Кодирование особи

Генетические алгоритмы работают только с дискретным пространством. Однако максимальное и минимальное значение по условию задачи приходится искать для непрерывного пространства. Поэтому необходимо придумать способ дискретизации индивида – «кодирования особи». С этой целью, весь отрезок от  $[0, 4)$  разобьём на  $2^M$  точек. Величину  $M$  - желательно подбирать так, чтобы образовавшееся число влезло в стандартные типы данных языка программирования. Соответственно для отрезка  $[a, b]$  особь *ent* в её непрерывное значение *val* можно пересчитать по простой формуле:

$$val = a + ent \frac{(b - a)}{2^M}$$

## 3. Требования к программе

Программа должна работать в двух режимах: тестовый и основной.

В тестовом режиме программа должна выводить в файл и, при желании пользователя, на экран популяцию решений, получаемую на каждом шаге работы алгоритма.

В основном режиме на экран выводится только наилучшая точка в популяции и, значение функции в этой точке, номер итерации. В конце окончательную найденную точку, значение в этой точке и число итераций, за которые результат был достигнут.

Все шаги алгоритма должны быть реализованы в виде отдельных процедур.

Параметры алгоритма должны быть собраны в некотором текстовом конфигурационном файле, где параметры могут быть перечислены в любом порядке. Формат записи для одного параметра: "*имя\_параметра = значение*". Так же в файле могут встречаться комментарии задаваемые символом решётки.

## 4. Методические указания

### 4.1. Шаги генетического алгоритма

Генетический алгоритм работает в соответствии со следующими шагами:

1. Чтение параметров
2. Формирование начальной популяции

3. Вычисление функции качества для каждой особи и сортировка популяции
4. Проверка условия остановки. Если достигнуто завершаем алгоритм.
5. Селекция
6. Формирование новой популяции: скрещивания, мутации
7. переход на шаг 3.

Общая схема работы генетических алгоритмов представлена на рисунке 1.

Рассмотрим более подробно каждый из этапов работы Генетического Алгоритма (ГА) и варианты используемых генетических операторов.

#### 4.2. Генерация начальной популяции

Генерация начальной популяции может происходить как случайным образом, так и с помощью некоторого алгоритма.

#### 4.3. Вычисление целевой функции

Целевая функция (функция качества) позволяет оценить степень приспособленности данной особи в популяции и характеризует качество получаемого решения. В данной задаче целевая функция — это значение одной из функций  $f_1(x), \dots, f_8(x)$  в точке, соответствующей данной особи. Во время генетического процесса вычисление целевой функции осуществляется над элементами всей популяции решений. Нужно отметить, что достаточно часто сложность генетических алгоритмов оценивается по количеству вычислений целевой функции.

#### 4.4. Произведение селекции

Выбор решений для следующей популяции (оператор селекции) предназначен для улучшения качества решений в новой популяции, а именно сохранение разнообразия популяции, сохранение лучших решений и удаление из нее недопустимых решений. Обычно выбираются элементы с наибольшей приспособленностью (наилучшем значением функции качества).

Селекция в любом случае должна быть устроена так, что как минимум одна лучшая особь защищена от истребления в результате селекции. В алгоритме число защищаемых верхних позиций от отбора является параметром алгоритма *preserved\_high\_positions*. Так же полезно защищать некоторое количество нижних позиций, так как в результате скрещивания с ними возможны выходы из локальных экстремумов. Параметр *preserved\_low\_positions*.

Возможны различные варианты операции селекции, основанные на разных схемах отбора:

- **Случайная схема.** В данной схеме отбора особи, попадающие в новую популяцию выбираются случайным образом. Верхняя защищенная часть (по значению функции качества) не участвует в отборе, нижняя так же.
- **Схема отбора на основе рулетки.** Каждому решению выделяется сектор рулетки

$$\text{discretion}(i) = 2\pi \cdot \frac{f_m(x_i)}{\sum_{j=1}^{\text{population\_volume}} f_m(x_j)}$$

Решение попадает в новую популяцию, если случайным образом сгенерированное число попадает в этот сектор.

- **Отбор усечением.** Число решений для сохранения в популяции выбирается в соответствии с порогом  $T \in [0; 1]$ . Порог определяет, какая доля особей, начиная с самой первой (самой приспособленной) не будет принимать участие в отборе, остальные будут удалены из популяции (кроме защищённых последних).

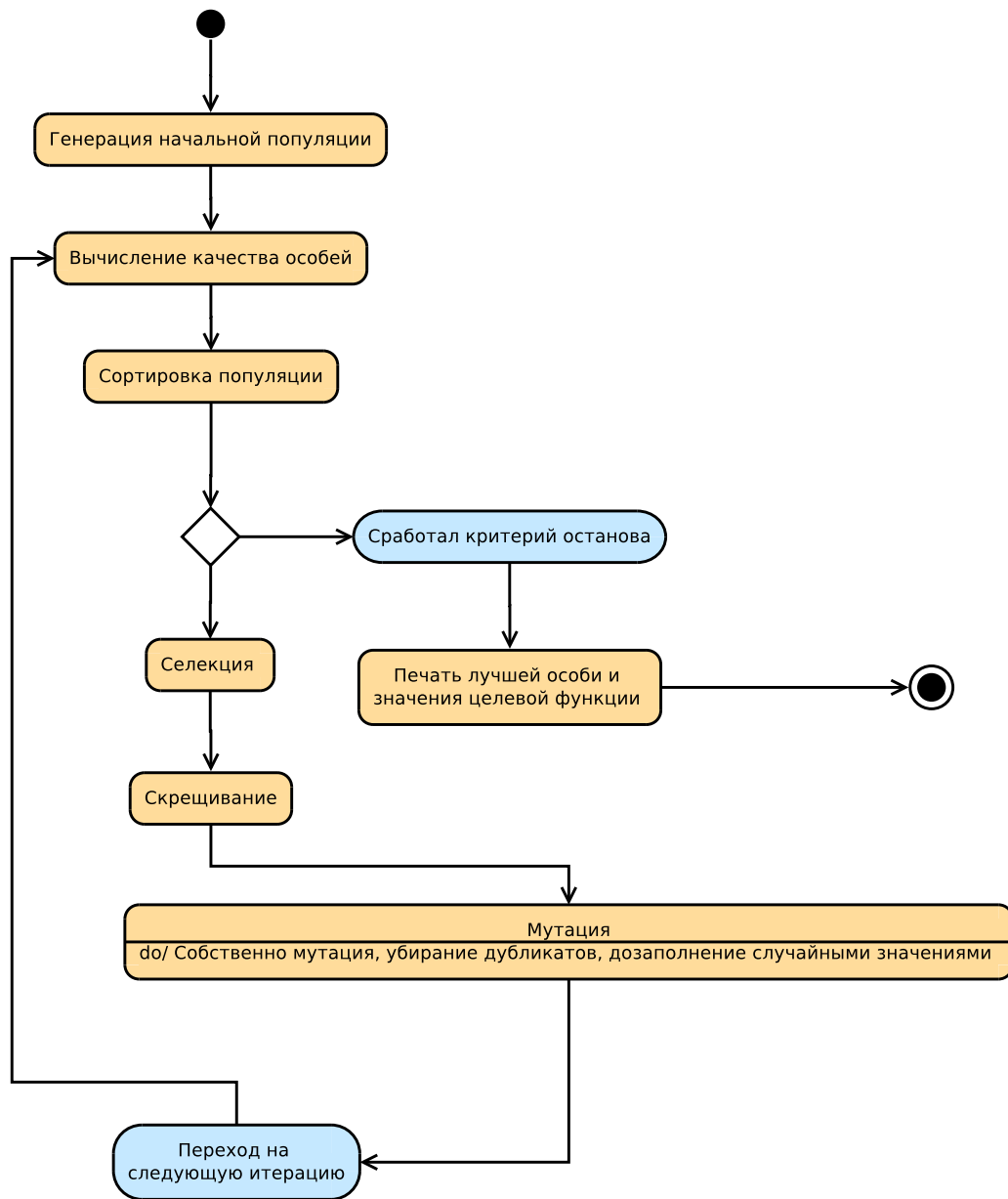


Рис. 1: Общая схема работы генетического алгоритма

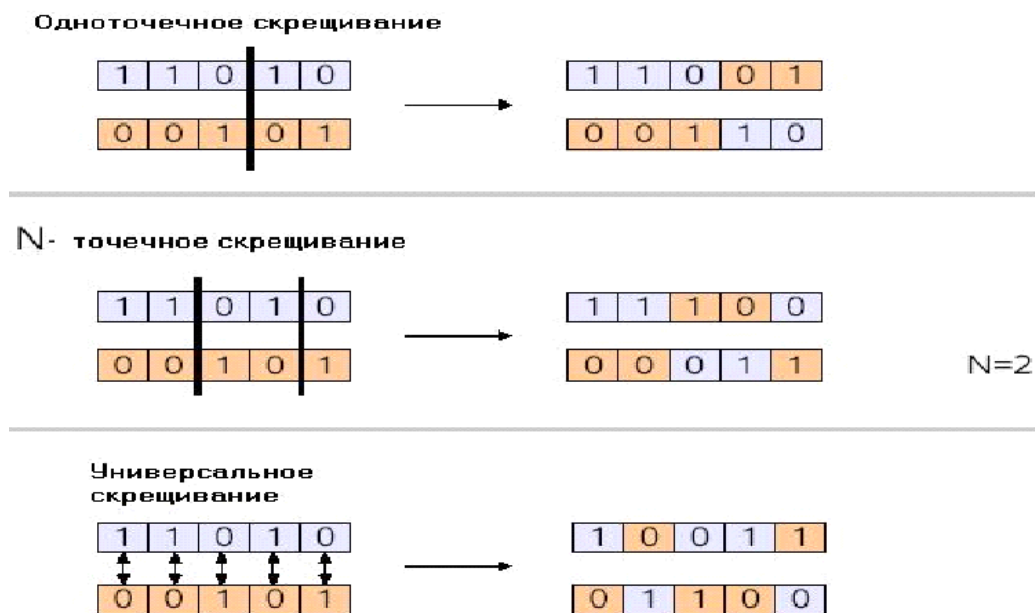


Рис. 2: Варианты оператора скрещивания.

#### 4.5. Произведение скрещиваний

Оператор скрещивания используется для передачи родительских признаков потомкам. Пары для скрещивания выбираются либо случайно, либо на основе одной из схем селекции, описанных выше.

Доля особей участвующих в скрещивании задаётся параметром *crossing\_volume*.

Возможны следующие варианты оператора скрещивания (рис. 2).

- **Одноточечное скрещивание.** Выбирается одна точка, и относительно неё решения обмениваются своими частями.
- **Двухточечное скрещивание.** Аналогично предыдущему, но точек скрещивания выбирается две.
- **Универсальное скрещивание.** С некоторой вероятностью выбирается бит либо одного, либо другого родителя.
- **Однородное скрещивание.** Каждый ген в потомстве создается посредством копирования соответствующего гена от одного или другого родителя, выбранного согласно случайно сгенерированной маске скрещивания. Если в маске скрещивания стоит 1, то ген копируется от первого родителя, если в маске стоит 0, то ген копируется от второго родителя.

После операции скрещивания новые решения остаются в популяции вместе с родителями, по которым были порождены скрещенные особи.

#### 4.6. Организация мутаций

Оператор мутации используется для внесения в решение некоторых новых признаков. Некоторые варианты реализации операции мутации представлены на рисунке 3. Все варианты изменяют биты битовой строки с некоторой вероятностью.

Доля мутирующих особей в популяции определяется параметром алгоритма *variability*.

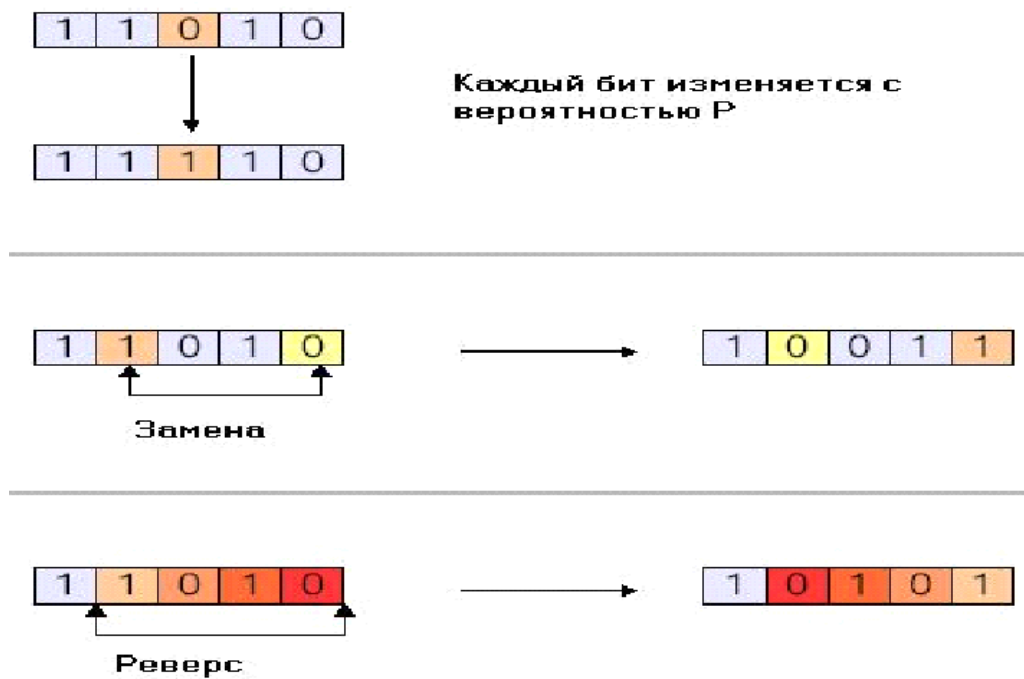


Рис. 3: Варианты операции мутации.

- Изменение случайно выбранного бита.
- Перестановка случайно выбранных битов местами.
- Реверс битовой строки, начиная со случайно выбранного бита.

## Список литературы

- [1] Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы / Под ред. В.М. Курейчика. – 2-е изд., испр. и доп. – М.: ФИЗМАТЛИТ, 2006. – 320 с. – ISBN 5-9221-0510-8.
- [2] Holland J.N. Adaptation in Natural and Artificial Systems // Ann Arbor, Michigan: Univ. of Michigan Press, 1975.
- [3] Goldberg D.E. Genetic Algorithms in Search Optimization & Machine Learning // Addison Wesley, Reading, 1989.