

Задание 3(8). ЯЗЫК ПАСКАЛЬ+АССЕМБЛЕРА. ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ.

3.1. ПОСТАНОВКА ЗАДАЧИ

Требуется найти экстремум функции на заданном интервале методом генетического алгоритма[1].

3.2. ВАРИАНТЫ ЗАДАНИЯ

Во всех вариантах точность, с которой необходимо найти решение равно 1/16384. Размер начальной популяции равен N=10. Начальная популяция формируется случайным образом. Количество скрещиваемых особей и вероятность мутации особи задаются пользователем.

Функции на отрезке [0;4):

- 1) $(x - 2)(x - 2.5)(x - 3)(x - 3.5)(1 - e^{x-1.5}) \ln(x + 0.5)$
- 2) $(x - 2.1)(x - 1.5)(x - 2.4)(x - 0.33)(1 - e^{x-3.5}) \cos(x)$
- 3) $(x - 2)(x - 0.5)(x - 0.25)(x - 1.5) \sin\left(\frac{x}{5}\right)$
- 4) $(x - 1)^5(x - 0.05)(x - 3)(x - 3.5)(1 - e^{x-3.95}) \ln(x + 0.22)$
- 5) $(x - 3)(x - 2)(x - 0.01)^4(x - 3.99)^4(1 - e^{x-1.5}) \sin\left(\frac{x}{3} + 0.2\right)$
- 6) $x(x - 1.1)^5(x - 1.2)^4(x - 1.3)^3 \cos(x + 100)$
- 7) $x \cdot \sin(x + 5) \cos(x - 6) \sin(x + 7) \cos(x - 8) \sin\left(\frac{x}{3}\right)$
- 8) $x(x - 2)(x - 2.75)e^{\frac{x}{10}} \cos\left(\frac{x}{10}\right)(2 - 3^{x-2})$

В функциях 1,4-8 необходимо найти **max** , а в функциях 2,3 необходимо найти **min**

Критерий останова:

- 1) Выполнение алгоритмом априорно заданного числа итераций.
- 2) Выполнение алгоритмом априорно заданного числа итераций без улучшения целевой функции.
- 3) Достижение некоторого априорно заданного значения целевой функции.

Виды селекции:

- 1) Случайная схема
- 2) Схема пропорционального отбора

- 3) При помощи рулетки
- 4) Турнирная
- 5) Отбор усечением

Виды скрещивания:

- 1) Одноточечное
- 2) Двухточечное
- 3) Универсальное
- 4) Однородное

Виды мутации:

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

3.3. ТРЕБОВАНИЯ К ПРОГРАММЕ

1. Программа должна работать в двух режимах:

- тестовый;
- 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 точка, значение функции в этой точке и количество итераций, за которые результат был достигнут.

2. Все шаги алгоритма должны быть реализованы в виде отдельных процедур.

3.4. ЛИТЕРАТУРА

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.

3.5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Классический генетический алгоритм состоит из следующих шагов [1, 2, 3]:

1. Формирование начальной популяции решений;
2. Вычисление целевой функции для каждого решения популяции;
3. Если критерий останова не достигнут, перейти к шагу 4, иначе завершить работу алгоритма;
4. Выполнение операции селекции;
5. Выполнение операции скрещивания;
6. Выполнение операции мутации и перейти к шагу 2.

Общая схема работы генетических алгоритмов представлена на рисунке 1 [1].

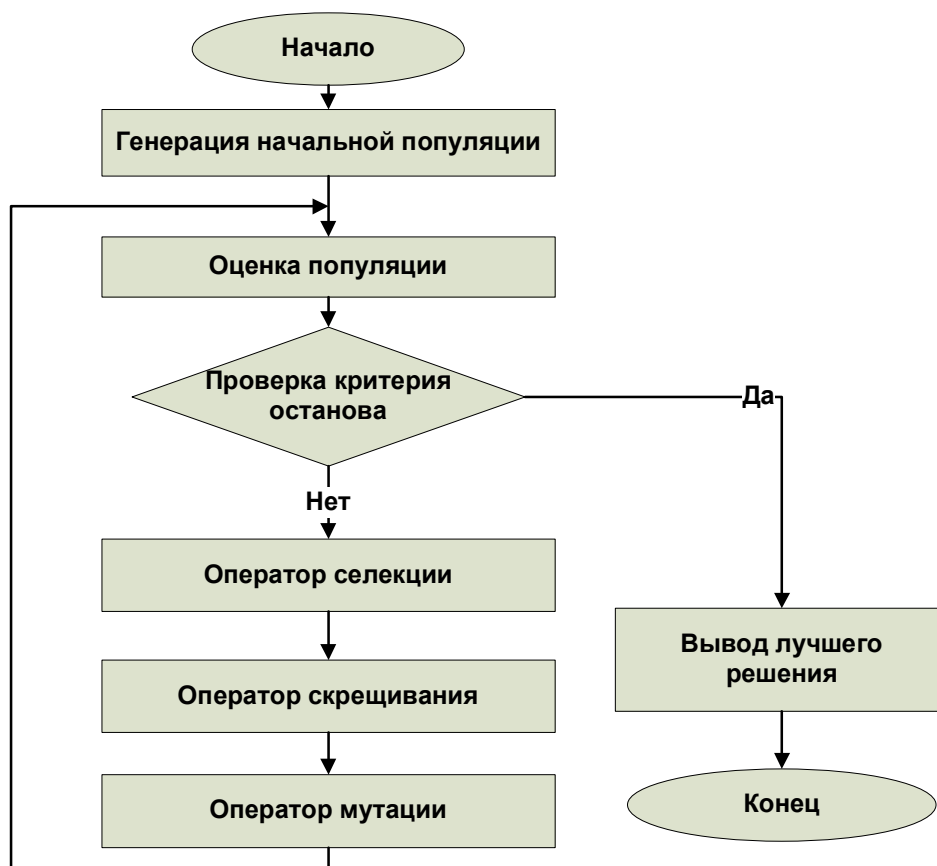


Рис. 1 Общая схема работы генетического алгоритма.

Рассмотрим более подробно каждый из этапов работы ГА и варианты используемых генетических операторов.

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

Целевая функция позволяет оценить степень приспособленности данной особи в популяции и характеризует качество получаемого решения. В данной задаче целевая функция – это значение многочлена 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.

Критерий останова. В качестве критерия останова генетического алгоритма могут выступать следующие условия:

- Выполнение алгоритмом априорно заданного числа итераций.
- Выполнение алгоритмом априорно заданного числа итераций без улучшения целевой функции.
- Достижение некоторого априорно заданного значения целевой функции.

Оператор 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.

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

- Случайная схема

В данной схеме отбора пары для скрещивания выбираются случайным образом.

- Схема пропорционального отбора.

В данной схеме отбора вычисляется значение целевой функции для каждого решения F_i и определяется среднее значение целевой функции в популяции F_{cp} . Затем для каждого решения вычисляется отношение F_i / F_{cp} . Например, если отношение равно 2.36, то данное решение имеет двойной шанс на скрещивание, и будет иметь вероятность равную 0.36 третьего скрещивания. Если же приспособленность равна 0.54, то решение примет участие в единственном скрещивании с вероятностью 0.54.

- Схема отбора на основе рулетки.

Каждому решению выделяется сектор рулетки $2\pi \cdot F_i / \sum_{j=1}^N F_j$

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

- Турнирный отбор.

Схему турнирного отбора можно описать следующим образом: из популяции, содержащей N решений, выбирается случайным образом 2 решения и между выбранными решениями проводится турнир. Победившее решение используется для скрещивания.

- Отбор усечением. Данная стратегия использует отсортированную по возрастанию популяцию. Число решений для скрещивания выбирается в соответствии с порогом $T \in [0; 1]$. Порог

определяет, какая доля особей, начиная с самой первой (самой приспособленной) будет принимать участие в отборе. Порог можно задавать числом, большим единицы, тогда он будет равен числу решений из текущей популяции, допущенных к отбору.

Оператор скрещивания:

Оператор скрещивания используется для передачи родительских признаков потомкам. Пары для скрещивания выбираются на основе одной из схем селекции, описанных выше. 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. Процесс повторяется с новыми родителями для создания второго потомства. Новая маска скрещивания случайно генерируется для каждой пары родителей.

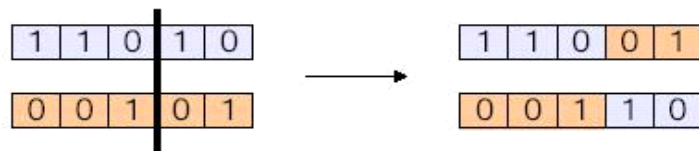
После операции скрещивания новые решения занимают места своих родителей в популяции.

Оператор мутации

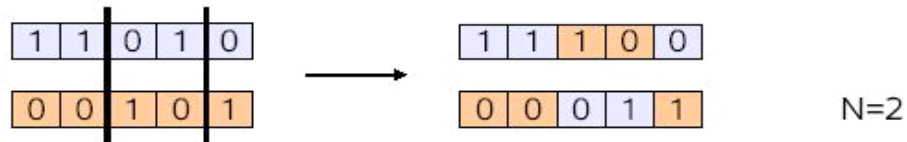
Оператор мутации используется для внесения в решение некоторых новых признаков. Некоторые варианты реализации операции мутации представлены на рисунке 3. Все варианты изменяют биты битовой строки с некоторой вероятностью.

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

Одноточечное скрещивание



N-точечное скрещивание



Универсальное скрещивание

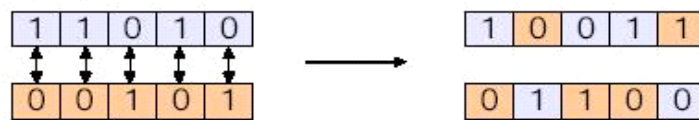
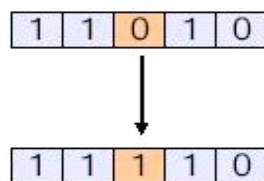


Рис. 2 Варианты оператора скрещивания.



Каждый бит изменяется с вероятностью P



Рисунок 3. Варианты операции мутации