**Н И Т У « М И С и С »**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ОТЧЁТ**

по

**ЛАБОРАТОРНОЙ РАБОТЕ**

**«Генетические алгоритмы»**

Учебная дисциплина «Нейронные сети»

**Группа:** БПМ-16-2

**Студент:** Новицкий Дмитрий

**Преподаватель:** доц., к.т.н. Курочкин И. И.

**Отметка:**

**Дата защиты:**

**2019 г.**

Оглавление

[Постановка задачи 3](#_Toc25445203)

[Основное условие 3](#_Toc25445204)

[Загрузка файлов 3](#_Toc25445205)

[Описание работы программы 4](#_Toc25445206)

[Переменные класса «Genetic\_algoritm» 4](#_Toc25445207)

[Алгоритм работы класса «Genetic\_algoritm» 4](#_Toc25445208)

[Результаты работы программы 7](#_Toc25445209)

[Функция Растригина 7](#_Toc25445210)

[Функция Экли 7](#_Toc25445211)

[Функция сферы 8](#_Toc25445212)

[Функция Розенброка 8](#_Toc25445213)

[Функция Била 9](#_Toc25445214)

[Функция Гольдман-Прайса 9](#_Toc25445215)

[Функция Бута 10](#_Toc25445216)

[Функция Букина 10](#_Toc25445217)

[Функция Леви 11](#_Toc25445218)

# Постановка задачи

## Основное условие

1. Реализовать классический генетический алгоритм (ГА) и его модификацию по вариантам (мой вариант – Island Model).
2. Подобрать параметры модификации ГА, так чтобы решалась задача поиска глобального минимума на функциях с точностью до .
3. Провести сравнительный анализ классического ГА и его модификации с целью демонстрации преимуществ модификации ГА. При сравнительном анализе варьировать параметры классического ГА и модификации для демонстрации подбора субоптимальных («хороших») значений параметров.

## **Загрузка файлов**

1. Работающее приложение.
2. Текстовый файл со сценариями запуска (к примеру: myapp.exe dataset1 results).
3. Подробный отчет по ЛР с приведенными результатами и пояснениями.

# Описание работы программы

Данная программа была написана на языке C# без использования сторонних библиотек, которые позволяют решить задачу «из коробки». Для решения данного задания был создан класс «Genetic\_algoritm», реализующий необходимые функции и методы данного алгоритма.

## Переменные класса «Genetic\_algoritm»

Рассмотрим переменные данного класса:

* count\_of\_individuals – количество особей в популяции;
* min\_x – минимальное значение переменной x;
* max\_x – максимальное значение переменной x;
* min\_y – минимальное значение переменной y;
* max\_y – максимальное значение переменной y;
* mutation\_rate – вероятность мутации для каждой особи;
* mutation\_x – начальное значение мутации по y;
* mutation\_y – начальное значение мутации по x;
* period\_change\_mutation\_values – количество эпох работы алгоритма, через которое меняется значение параметра для пределов мутации;
* mean\_change\_mutation\_values – параметр для изменения пределов мутации;
* death\_rate – доля от всей популяции, которая погибнет на итерации;
* count\_of\_eras – количество эпох работы генетического алгоритма;
* x\_population – список со значениями переменной x на определённой итерации;
* y\_population – список со значениями переменной y на определённой итерации;
* f – список со значениями функции от переменных x и y на определённой итерации;
* min\_value – лучшее (минимальное) значение из всех эпох обучения;
* era\_min\_value – эпоха, в которой было достигнуто минимальное значение;
* min\_value\_x – значение переменной x, для которой было получено наименьшее значение функции из всех эпох обучения;
* min\_value\_y – значение переменной y, для которой было получено наименьшее значение функции из всех эпох обучения;
* mas\_min\_value – список с минимальными значениями функции на каждой эпохе обучения.

## Алгоритм работы класса «Genetic\_algoritm»

Изначально, при создании экземпляра класса инициализируются основные параметры алгоритма:

* count\_of\_individuals;
* count\_of\_eras;
* mutation\_rate;
* death\_rate;
* period\_change\_mutation\_values;
* mean\_change\_mutation\_values;

Также дополнительно можно задать следующие параметры:

* min\_x;
* max\_x;
* min\_y;
* max\_y;

После чего начинается процесс обучения генетического алгоритма. За это отвечает функция «Learning\_function».

Изначально вычисляются значения x и y по одному из нескольких следующих алгоритмов:

* случайная генерация значений – значения переменных x и y генерируются случайно в промежутке и соответственно;
* детерминированная генерация значений через определённый промежуток – значения генерируются в промежутке и соответственно, но с заданным шагом и соответственно, где n – это количество особей в популяции.

В зависимости от поставленной задачи один из представленных алгоритмов задания начальных значений для переменных x и y может быть более предпочтителен, чем другой.

Далее начинается цикл обучения генетического алгоритма, критерием остановки которого будет являться достижение заданного количества эпох обучения.

В цикле последовательно будут выполняться следующие методы:

* Calculation\_population\_values – метод, вычисляющий значения заданной функции приспособленности в зависимости от значений x и y.
* Get\_population\_min\_value – нахождение минимального значения функции на данной эпохе обучения и запись данного значения в массив mas\_min\_value. Если минимальное значение функции в данной эпохе наименьшее за весь период обучения, то данное значение записывается в переменную min\_value. Также записываются эпоха, в которой было получено наименьшее значение функции и значения переменных x и y, при которых было достигнуто данное значение.
* Death – метод, исключающий (int)(count\_of\_individuals \* death\_rate) значений, которые дают наибольшее значение функции на данной эпохе обучения.
* Селекция (отбор) особей для дальнейшего размножения может происходить несколькими способами:
  + Случайным образом (метод «Random\_selection») – родители выбираются случайным образом из всех выживших особей и со случайной вероятностью (в сумме равной единице) наследуются значения переменных x и y от обоих родителей.
  + С использованием алгоритма сигмы-отсечения.

В зависимости от поставленной задачи возможно применять один из вышеперечисленных методов. Отбор происходит до того момента, когда количество особей не станет равно параметру count\_of\_individuals (начальному числу особей).

* Mutation – мутация (int)(count\_of\_individuals \* mutation\_rate) количества особей в популяции. Выбор особи из популяции для мутации происходит случайным образом.
* Изменение параметров мутации. Через каждые period\_change\_mutation\_values максимальное значение мутации увеличивается в mean\_change\_mutation\_values раз ().

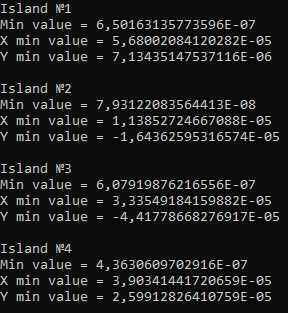
# Результаты работы программы

Зададим следующие параметры программы:

* Количество островов – 4.
* Общее количество эпох – 1000.
* Период переселения особей – 100 эпох.
* Количество переселяющихся особей – 25.
* Переселение особей осуществляется по кругу (с первого острова на второй, со второго на третий и т. д.).
* mutation\_rate = 0.1.
* death\_rate = 0.05.
* period\_change\_mutation\_values = 10.
* mean\_change\_mutation\_values = 0.99.

## Функция Растригина

Результат работы программы будет следующим:

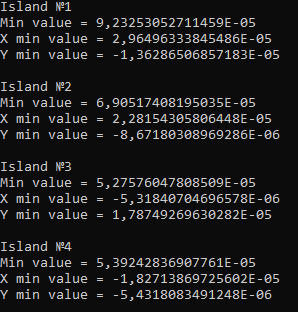


*Рис. 1. Результат работы программы.*

Точное значение глобального минимума данной функции равно f(0, 0) = 0.

## Функция Экли

Результат работы программы будет следующим:

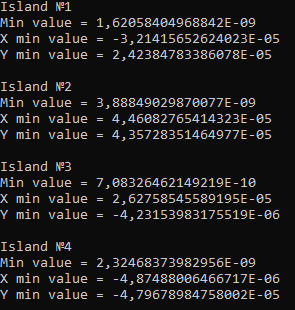


*Рис. 2. Результат работы программы.*

Точное значение глобального минимума данной функции равно f(0, 0) = 0.

## Функция сферы

Результат работы программы будет следующим:

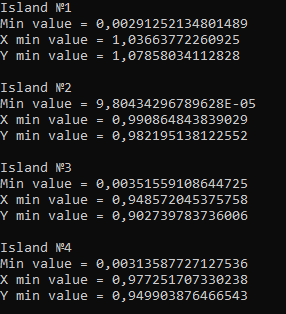


*Рис. 3. Результат работы программы.*

Точное значение глобального минимума данной функции равно f(0, 0) = 0.

## Функция Розенброка

Результат работы программы будет следующим:

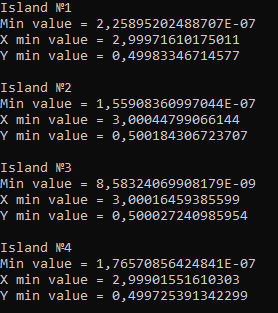


*Рис. 4. Результат работы программы.*

Точное значение глобального минимума данной функции равно f(1, 1) = 0.

## Функция Била

Результат работы программы будет следующим:

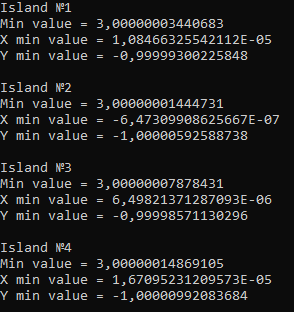


*Рис. 5. Результат работы программы.*

Точное значение глобального минимума данной функции равно f(3, 0,5) = 0.

## Функция Гольдман-Прайса

Результат работы программы будет следующим:

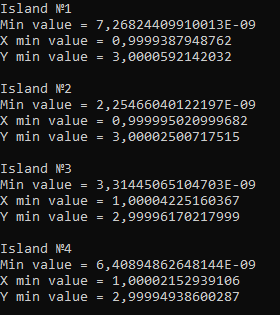


*Рис. 6. Результат работы программы.*

Точное значение глобального минимума данной функции равно f(0, -1) = 3.

## Функция Бута

Результат работы программы будет следующим:

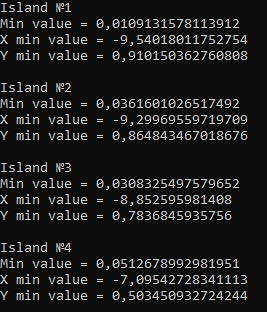


*Рис. 7. Результат работы программы.*

Точное значение глобального минимума данной функции равно f(1, 3) = 0.

## Функция Букина

Результат работы программы будет следующим:

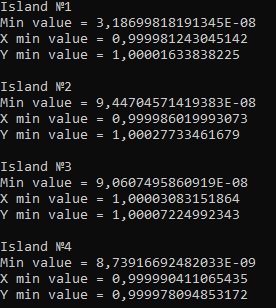


*Рис. 8. Результат работы программы.*

Точное значение глобального минимума данной функции равно f(-10, 1) = 0.

## Функция Леви

Результат работы программы будет следующим:



*Рис. 9. Результат работы программы.*

Точное значение глобального минимума данной функции равно f(1, 1) = 0.