МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНЖЕНЕРНО-ТЕХНОЛОГИЧЕСКАЯ АКАДЕМИЯ

Институт компьютерных технологий и информационной безопасности

Кафедра систем автоматизированного проектирования

Отчет по лабораторной работе № 6

на тему: «Изучение основных операторов мутаций»

по курсу «Методы машинного обучения»

Выполнили:   
студенты гр. КТбо4-4

Батагов С.С.\_\_\_\_\_\_\_\_\_\_

Москаленко М.В.\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

Проверил:

Доцент

Лебедев О.Б.\_\_\_\_\_\_\_\_\_

(подпись, дата)

Таганрог 2022

**СОДЕРЖАНИЕ**

[**1 ЦЕЛЬ РАБОТЫ 3**](#_Toc119191423)

[**2 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 3**](#_Toc119191424)

[**3 ПРАКТИЧЕСКАЯ ЧАСТЬ 9**](#_Toc119191426)

[**4 ВЫВОД 18**](#_Toc119191433)

[**ПРИЛОЖЕНИЕ А 19**](#_Toc119191434)

# ЦЕЛЬ РАБОТЫ

Ознакомиться с алгоритмом и стратегиями выполнения операции мутации. Научиться выполнять оператор мутации с различными вероятностями.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Классическая генетика к началу

После процесса воспроизводства происходят мутации (mutation). Данный оператор необходим для «выбивания» популяции из локального экстремума и препятствует преждевременной сходимости. Это достигается за

счет того, что изменяется случайно выбранный ген в хромосоме (Рисунок 1).

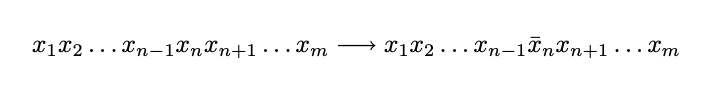


Рисунок 1 — Мутация в точке xn

Так же как и кроссинговер, мутации могут проводиться не только по

одной случайной точке. Можно выбирать для изменения несколько точек в

хромосоме, причем их число также может быть случайным. Используют и

мутации с изменением сразу некоторой группы подряд идущих точек.

Вероятность мутации pm (как правило, pm << 1) может являться или

фиксированным случайным числом на отрезке [0; 1], или функцией от

какой-либо характеристики решаемой задачи. Например, можно положить

вероятность мутирования генов, обратно пропорциональную числу всех генов в особи (размерности).

Оптимальное значение вероятности мутации обсуждается в разных статьях. Так, например, мутация с фиксированной вероятностью приводит к

хорошим результатам для широкого класса тестовых функций (например,

для унимодальных функций). Для мультимодальных функций применяют само адаптирующуюся оценку вероятности. Ниже приведены основные

варианты мутирования.

**Мутация для вещественных особей** (Real valued mutation). Для мутации особей с вещественными числами необходимо определить величину шага мутации — число, на которое изменится значение гена при мутировании (Рисунок 2).

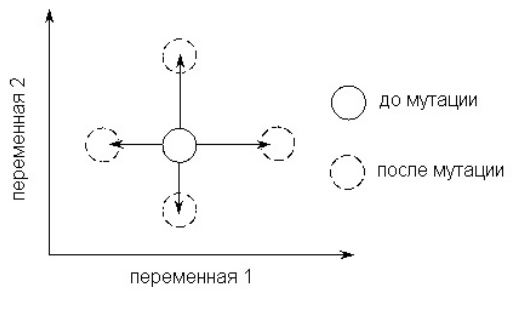
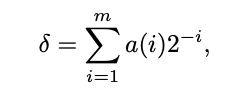


Рисунок 2 — Мутация для особей с вещественными генами

Обычно определение шага мутации представляет некоторую трудность. Оптимальный размер шага должен меняться в течение всего процесса поиска. Наиболее пригодны маленькие шаги, но иногда большие шаги могут привести к ускорению процесса. Гены могут мутировать согласно следующему правилу:

новая переменная = старая переменная ± α · δ,

где знаки + или − выбираются с равной вероятностью, α = 0,5 ×поисковое пространство (интервал изменения данной переменной),



a(i) = 1 с вероятностью 1/m , в противном случае a(i) = 0, m — параметр.

Новая особь, получившаяся при такой мутации, в большинстве случаев не намного отличается от старой. Это связано с тем, что вероятность маленького шага мутации выше, чем вероятность большого шага. При m = 20,

данный алгоритм мутации пригоден для локализации оптимума с точностью .

**Двоичная мутация (Binary mutation)**. Для особей, кодированных двоичным кодом или кодом Грея, мутация заключается в случайном инвертировании гена (0 заменяется 1 и наоборот). Эффект мутации зависит от примененного способа кодирования генов. Так, в одних задачах при мутации наилучший эффект достигается в случае, когда особи закодированы кодом Грея, а в других — с помощью двоичного кода.

**Плотность мутации (Density mutation)**. Стратегия мутации с использованием понятия плотности заключается в мутировании каждого гена потомка с заданной вероятностью. Таким образом, кроме вероятности применения мутации к самому потомку используется еще вероятность применения мутации к каждому его гену, величину которой выбирают с таким расчетом, чтобы в среднем мутировало от 1 до 10 % генов.

**Другие виды мутаций.** Пусть особь t представлена следующей последовательностью генов ti : t = t1, . . . , tk. Тогда можно применить следующие операторы мутации:

1. Присоединение случайного гена из совокупности всевозможных значений генов к концу последовательности: t → t1, . . . , tk, s.
2. Вставка случайного гена из совокупности всевозможных значений генов в случайно выбранную позицию в последовательности: t → t1, . . . , ti−1, s, ti , . . . , tk.
3. Удаление случайно выбранного гена из последовательности: t → t1, . . . , ti−1, ti+1, . . . , tk.
4. Обмен местами в последовательности двух соседей одного случайно выбранного гена: t → t1, . . . , ti+1, s, ti−1, . . . , tk.

Следует заметить, что мутация 1 является частным случаем мутации 2. Для особей с фиксированным размером (количество генов в последовательности) возможно применение в чистом виде только 4 мутации, а 1 и 2 мутации должны применяться в сочетании с мутацией 3.

# 3 ПРАКТИЧЕСКАЯ ЧАСТЬ

Полный код рассматриваемой программы представлен в приложении А.

Для генерации популяции, содержащей хромосомы для кроссинговера, создадим класс Chromos, который будет генерировать хромосомы числового типа (Рисунок 3).

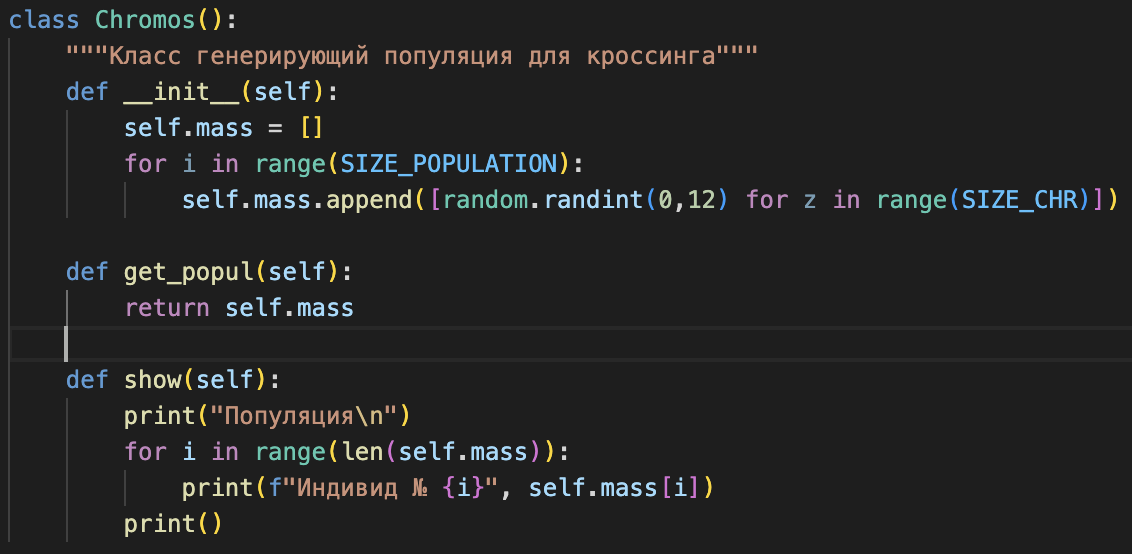


Рисунок 3 — Класс для генерации популяции

Рассмотрим кратко основные операторы мутации (ОМ). Простейшим ОМ является одноточечный. При его реализации случайно выбирают ген в родительской хромосоме и, обменивая его на рядом расположенный ген, получают хромосому потомка (Рисунок 4).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4 — Функция одноточечной мутации

На вход функции поступает список популяции, после создаётся разделительная точка. Далее элементы с указанными индексами меняются местами (Рисунок 5).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 — Результат выполнения функции

Двухточечная мутация схожа с одноточечной, однако вторая разделительная точка, в итоге меняется с первой (Рисунок 6).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6 — Алгоритм двухточечной мутации

Код данной функции схож с предыдущей, однако было добавлена переменная второй разделительной точки (Рисунок 7).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7 — Результат двухточечной мутации

Инвертированный участок при нечетной длине хромосомы включает центральный ген (перецентрическая инверсия) или не включает его при четной длине хромосомы (парацентрическая) (Рисунок 8).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8 — Алгоритм мутации инверсия

Данный алгоритм состоит из одного цикла, внутри которого создаются 2 переменные, присваивающие значение разделительных точек. Далее в массив res добавляются мутированные хромосомы, то есть происходит конкатенация массивов, то есть срезов список и массива tmp, содержащий инвертированный порядок элементов (Рисунок 9).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 — Результат выполнения функции

Мутация делеции, т.е. удаление внутреннего участка хромосомы (Рисунок 10).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10 — Алгоритм делеции

На вход данной функции поступает популяция, то есть массив хромосом. Далее в цикле создаются 2 переменные, отвечающие за разделительные точки, а после в массив res добавляется сложение двух срезов списков (Рисунок 11).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 11 — Результат выполнения функции

Дупликация. При этом один из участков хромосомы представляется более одного раза (Рисунок 12).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12 — Код функции дупликации

Код данной функции имеет несколько различий в отличии от предыдущий функций. Так же в цикле создаются 2 точки разделения, далее во временную переменную tmp присваивается промежуток, которые две точки образуют. Далее в массив res добавляются итоговые хромосомы, полученные в результате сложения срезов и переменной tmp (Рисунок 13).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 13 — Результат работы функции

Межхромосомные перестройки часто называют транслокациями. При этом участок хромосомы перемещается (транслоцируется) на другое место хромосомы. Выделяют следующие типы транслокаций.

Реципрокные – взаимный обмен участками негомологичных хромосом. В отличие от кроссинговера, при транслокации происходит обмен участков хромосом различной длины (Рисунок 14).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 14 — Код функции

На вход функции подаётся список хромосом популяции. Далее в цикле создаются 2 разделительные переменные, а также две переменные, хранящие срез каждой хромосомы. После в массив res добавляются мутированные хромосомы. Стоит отметить, что в цикле используются сразу пара хромосом (Рисунок 15).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 15 — Результат выполнения функции

# 4 ВЫВОД

В данной лабораторной работе были реализованы разные вариации алгоритма мутации, применяемого в простом генетическом алгоритме. Также была изучена теоретическая часть данных алгоритмов для дальнейшего использования в следующих лабораторных работах.

# ПРИЛОЖЕНИЕ А

**Исходный код программы**

Ниже будет представлен исходный код программы, которая была разработана в рамках данной лабораторной работы.

import random

SIZE\_POPULATION = 6

SIZE\_CHR = 7

class Chromos():

"""Класс генерирующий популяция для кроссинга"""

def \_\_init\_\_(self):

self.mass = []

for i in range(SIZE\_POPULATION):

self.mass.append([random.randint(0,12) for z in range(SIZE\_CHR)])

def get\_popul(self):

return self.mass

def show(self):

print("Популяция\n")

for i in range(len(self.mass)):

print(f"Индивид № {i}", self.mass[i])

print()

def oneDote(ls):

"""Простая мутация (одноточечная)"""

print("Одноточечная мутация")

for i in range(len(ls)):

cut\_point = random.randint(0, len(ls[0])-2)

ls[i][cut\_point],ls[i][cut\_point+1] = ls[i][cut\_point+1], ls[i][cut\_point]

show(ls)

def twoDote(ls):

"""Простая мутация (двухточечная)"""

print("Двухточечная мутация")

for i in range(len(ls)):

cut\_point1 = random.randint(0, len(ls[0])-1)

cut\_point2 = random.randint(0, len(ls[0])-1)

ls[i][cut\_point1],ls[i][cut\_point2] = ls[i][cut\_point2], ls[i][cut\_point1]

show(ls)

def invers(ls):

"""Мутация инверсии"""

print("Мутация инверсии")

res = []

for i in range(len(ls)):

point1 = random.randint(1, len(ls)-2)

point2 = random.randint(point1+2, len(ls))

tmp = ls[i][point1:point2]

res.append(ls[i][:point1] + tmp[-1:-len(tmp)-1:-1] + ls[i][point2:])

show(res)

def delic(ls):

"""Метод делеции, удаление части хромосомы """

print("Делиция")

res = []

for i in range(len(ls)):

point1 = random.randint(1, len(ls)-1)

point2 = random.randint(point1+1, len(ls))

res.append(ls[i][:point1] + ls[i][point2:])

show(res)

def doubl(ls):

"""Мутация дубликации"""

print("Мутация дупликации")

res = []

for i in range(len(ls)):

point1 = random.randint(1, len(ls)-1)

point2 = random.randint(point1+1, len(ls))

tmp = ls[i][point1:point2]

res.append(ls[i][:point1] + tmp + tmp + ls[i][point2:])

show(res)

def transloc(ls):

"""Мутация транслокации"""

print("Мутация транслокации")

res = []

for i in range(0, len(ls), 2):

point1 = random.randint(1, len(ls)-1)

point2 = random.randint(1, len(ls)-1)

hr1 = ls[i][point1:]

hr2 = ls[i+1][point2:]

res.append(ls[i][:point1] + hr2)

res.append(ls[i+1][:point2] + hr1)

show(res)

def show(ls):

"""Вспомогательеая функция вывода"""

print()

for i in range(len(ls)):

print(f"Хромосома {i}", ls[i])

print()

def main():

gen = Chromos()

gen.show()

var = gen.get\_popul()

oneDote(var)

twoDote(var)

invers(var)

delic(var)

doubl(var)

transloc(var)

if \_\_name\_\_ == "\_\_main\_\_":

main()