

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ Н. Э. БАУМАНА**
Факультет информатики и систем управления
Кафедра теоретической информатики и компьютерных технологий

Лабораторная работа №5
по курсу «Методы оптимизации»

«Решение задач многоэкстримальной оптимизации на основе
популяционных алгоритмов»

Выполнил:
студент ИУ9-111
Выборнов А. И.

Руководитель:
Каганов Ю. Т.

Москва 2016

1. Метод дифференциальной эволюции (вариант генетического алгоритма)

1.1. Постановка задачи

Дана функция Швеефеля:

$$f(x_1, x_2) = ax_1 \sin(b\sqrt{|x_1|}) + x_2 \sin(c\sqrt{|x_2|}),$$

определённая на множестве допустимых решений: $[x_{min}, x_{max}] = [-500, 500]$. Исследовать характер решений в зависимости от параметров a, b, c при значениях: $a = (0.5, 1.0, 2.5), b = (0.5, 0.8, 1.0), c = (0.5, 1.0, 2.0)$, а именно найти глобальный экстремум.

1.2. Решение на языке программирования python

```
from math import sin, sqrt
from random import random, uniform, randint
from copy import deepcopy

Np = 100 #population size
F = 1 #100 # weight
CR = random() # prop of sequence
M = 1000 #2000 # max num of population

#a, b, c = 0.5, 0.5, 0.5
#a, b, c = 1.0, 0.8, 1.0
a, b, c = 2.5, 1.0, 2.0

f = lambda (x1,x2): a*x1*sin(b*sqrt(abs(x1))) + x2*sin(c*sqrt(abs(x2)))
x_min = -500
x_max = 500

def normalize(x=None):
    if x < x_min or x_max < x:
        return uniform(x_min, x_max)
    return x

def getXcl(incorrect_index):
    Xabc = set()
    while not len(Xabc) == 3:
        index = randint(0, len(X)-1)
        if index != incorrect_index:
            Xabc.add(index)
    i, j, k = list(Xabc)
    Xa, Xb, Xc = X[i], X[j], X[k]

    return [normalize(Xc[0] + F*(Xa[0]-Xb[0])), normalize(Xc[1] + F*(Xa[1]-Xb[1]))]

def getXs(Xcl, Xt):
    Xs = deepcopy(Xcl)
    if random() > CR:
        Xs[0] = Xt[0]
    return Xs

X = [[normalize(), normalize()] for _ in range(Np)]

m = 0
for _ in range(M):
```

```

X_new = []
for j in range(Np):
    Xt = X[j]
    Xs = getXs(getXcl(j), Xt)

    X_new.append(Xs if f(Xs) < f(Xt) else Xt)
X = X_new

print 'a = %s, b = %s, c = %s' % (a,b,c)
print 'f = %s, x = %s' % min((f(x), x) for x in X)

```

1.3. Результат работы

Результат работы программы при разных параметрах приведён в таблице:

| a | b | c | $[x1, x2]$ | $f([x1, x2])$ |
|-----|-----|-----|---|----------------|
| 0.5 | 0.5 | 0.5 | [499.31742395249955, 499.3174262797162] | -737.257041083 |
| 1.0 | 0.8 | 1.0 | [472.6952114926131, -420.968746351129] | -888.583750801 |
| 2.5 | 1.0 | 2.0 | [-420.9687469299634, 450.6825579994172] | -1497.64060671 |