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

Решение задач многоэкстремальной оптимизации на основе популяционных алгоритмов. Метод дифференциальной эволюции (вариант генетического алгоритма)

Исполнитель: Ю.А. Волкова
Группа: ИУ9-111

25 Декабря, 2016

1 Постановка задачи

Дана модифицированная функция Розенброка:

$$f(x_1, x_2) = a(x_1^2 - x_2)^2 + b(1 - x_1)^2 - cx_2^3 x_1$$

Множество допустимых решений:

$$[x_i^{min}, x_i^{max}] = [-2, 2]$$

Исследовать характер решений в зависимости от параметров a, b, c при значениях: $a = (50, 100, 250); b = (5, 8, 10); c = (0.5, 1.0, 2.0)$.

Требуется найти глобальный экстремум $X^* \in X^e$ – (X^e – множество локальных экстремумов) на множестве D.

2 Код программы

```
from random import uniform as rand, randint, random, choice

#a, b, c = 50, 5, 0.5
#a, b, c = 100, 8, 1.0
a, b, c = 250, 10, 2.0

Np   = 100
F    = 100
CR   = random()
M    = 1000
xmin = -2
xmax = 2
n    = 2

def f( x ):
```

```
    return a * ( x[0]**2 - x[1] )**2 + b * ( 1 - x[0] )**2 - c * x[0] * x[1]**3

def addX( xs1, xs2 ):
    return [ x1 + x2 for x1, x2 in zip( xs1, xs2 ) ]

def subX( xs1, xs2 ):
    return [ x1 - x2 for x1, x2 in zip( xs1, xs2 ) ]

def mulX( xs, k ):
    return [ x * k for x in xs ]

def init():
    return [ [ rand( xmin, xmax ) for i in range(n) ] for j in range( Np ) ]

def choose3( X, Xt ):
    Xa = choice(X)
    Xb = choice(X)
    Xc = choice(X)

    while Xa[0] == Xt[0] and Xa[1] == Xt[1]:
        Xa = choice(X)

    while Xb[0] == Xt[0] and Xb[1] == Xt[1] or \
          Xb[0] == Xa[0] and Xb[1] == Xa[1]:
        Xb = choice(X)

    while Xc[0] == Xt[0] and Xc[1] == Xt[1] or \
          Xc[0] == Xa[0] and Xc[1] == Xa[1] or \
          Xc[0] == Xb[0] and Xc[1] == Xb[1]:
        Xc = choice(X)
```

```
    return Xa, Xb, Xc

def getXcl( X, Xt ):
    Xa, Xb, Xc = choose3( X, Xt )

    Xcl = addX( Xc, mulX( subX( Xa, Xb ), F ) )

    for i in range( n ):
        if Xcl[i] < xmin or Xcl[i] > xmax:
            Xcl[i] = rand( xmin, xmax )

    return Xcl

def getXs( Xcl, Xt ):
    Xs = [0] * n

    for i in range( n - 1 ):

        U = random()
        Xs[i] = Xcl[i] if U <= CR else Xt[i]

    Xs[ -1 ] = Xcl[-1]

    return Xs

def calcF( Xs, Xt ):
    return Xs if f( Xs ) < f( Xt ) else Xt

def main():
    X = init()
```

```
    for i in range(M):
        X = [ calcF( getXs( getXcl( X, Xt ), Xt ), Xt ) for Xt in X ]

    Xf = [ ( f(x), x ) for x in X ]

    print min( Xf )

if __name__ == '__main__':
    main()
```

3 Результат работы программы

Результат работы программы представлен в таблице в зависимости от параметров:

a	b	c	$[x_1, x_2]$	f
50	5	0.5	[1.4068860283340956, 1.9988157589133708]	-4.7707842184979095
100	8	1.0	[1.4049365385116972, 1.9981178246862883]	-9.837092457332599
250	10	2.0	[1.4220618355043535, 1.999313468375218]	-20.81657024214862