МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Э. БАУМАНА

Факультет информатики и систем управления

Кафедра теоретической информатики и компьютерных технологий

Численное решение задач многокритериальной оптимизации. Метод свёртки критериев (метод «идеальной точки»)

Исполнитель: Ю.А. Волкова

Группа: ИУ9-111

25 Декабря, 2016

1 Постановка задачи

Даны дважды непрерывно дифференцируемые целевые функции, представленные в виде вектор-функции $F(x) = [f_1(x), f_2(x) \dots f_l(x)]^T$ и функции ограничений $g_j(x) = 0, j = 1 \dots m$ и $g_j(x) \leq 0, j = m+1 \dots p$, определяющие множество допустимых решений X.

Требуется найти недоминируемые (эффективные) на множестве X, т.е. такое множество решений $X^e \in X$, что

$$F(X^e) = \inf_{x \in X} [f_1(x), f_2(x) \dots f_l(x)]^T$$

$$\begin{cases} g_j(x) = 0, j = 1 \dots m; m < n \\ g_j(x) \leq 0, j = m+1 \dots p \end{cases}$$

Где $f(x), g(x)$:

$$f_1(x) = x_1^2 + x_2^2 \to min$$

$$f_2(x) = 4(x_1 - 5)^2 + 2(x_2 - 6)^4 \to min$$

$$g_1(x) = -x_1 + x_2 + 1 \leq 0$$

$$g_2(x) = x_1 + x_2 - 2 \leq 0$$

2 Код программы

```
from lsgnonlin import lsgnonlin
from numpy import linalg as la
from random import randint


r = 0.01
eps = 0.0001
C = 4


def f1( x ):
    return x[0]**2 + x[1]**2
```

```python
def grad1( x ):
    return [ 2 * x[0], 2 * x[1] ]

def H1( x ):
    return [[ 2, 0 ], [0, 2]]

def f2( x ):
    x1 = x[0]
    x2 = x[1]

    return 4 * ( x1 - 5 )**2 + 2 * ( x2 - 6 )**4

def grad2( x ):
    return [ 8 * x[0] - 40, 8 * x[1]**3 - 144 * x[1]**2 + 864 * x[1] - 1728 ]

def H2( x ):
    return [[8, 0],[0, 24 * x[1]**2 - 288 * x[1] + 864]]

def g1( x ):
    return -x[0] + x[1] + 1

def g2( x ):
    return x[0] + x[1] - 2

f = [ f1, f2 ]
g = [ g1, g2 ]
fk= [ 0, 0 ]
w = [ 0, 0 ]

def F( x ):
    return sum([ wi * ( fi( x ) - fki ) for wi, fi, fki in zip( w, f, fk ) ]) +
```

```
def P( x ):
    return r * sum([ max( 0, gi(x) )**2 for gi in g ]) / 2

def grad( x ):
    g11 = 0 if g1(x) <= 0 else r * ( x[0] - x[1] - 1 )
    g12 = 0 if g1(x) <= 0 else r * ( x[1] - x[0] + 1 )
    g21 = 0 if g2(x) <= 0 else r * ( x[0] + x[1] - 2 )

    return [ w[0] * grad1(x)[0] + w[1] * grad2(x)[0] + g11 + g21,\
             w[0] * grad1(x)[1] + w[1] * grad2(x)[1] + g12 + g21 ]

def H( x ):
    g11 = 0 if g1(x) <= 0 else r
    g22 = 0 if g2(x) <= 0 else r

    return [[ w[0] * 2 + w[1] * 8 + g11 + g22, 0 + 0 - g11 + g22 ],\
            [ 0 + 0 - g11 + g22, w[0] * 2 + w[1] * H2(x)[1][1] + g11 + g22 ]]

def countfk( x ):
    k, fk1 = lsgnonlin( f1, grad1, H1, x, 10**4, 0 )
    k, fk2 = lsgnonlin( f2, grad2, H2, x, 10**4, 0 )

    return [ fk1, fk2 ]

def countw():
    a = randint(1, 10)

    print a

    c, v = la.eig([[ 1, 1 / a ], [ a, 1 ]])
```

```
    c = [ i for i in c ]
    v = [ [ i for i in vi ] for vi in v ]

    z = zip( c, v )
    print z

    return min( z )[1]

def penalty( x ):
    global r

    xk, Fxk = lsgnonlin( F, grad, H, x, 10**4, 0 )

    if abs( P( xk ) ) <= eps:
        return xk, f1( xk ), f2( xk )

    r *= C
    return penalty( xk )

def main():
    global fk, w

    x = [0, 0]

    fk = countfk( x )
    w  = countw()
    print 'fk:', fk, 'w', w

    print penalty( x )
```

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

3 Результат работы программы

Для поиска безусловного минимума используется программа, написанная для лабораторной работы 2.2.

При $w = [0.70710678118654746, 0.70710678118654746]$ получена точка
$x = [1.5000000032155494, 0.50000017106720973]$.
Значение функций в этой точке:

$$f_1(x) = 2.5000001807138874$$
$$f_2(x) = 1879.1247722195185$$

.