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

Лабораторная работа №4
по курсу «Методы оптимизации»

«Численное решение задач многокритериальной оптимизации»

Выполнил:
студент ИУ9-111
Выборнов А. И.

Руководитель:
Каганов Ю. Т.

Москва 2016

1. Методы свертки критериев (метод «идеальной точки»)

1.1. Постановка задачи

Найти минимум:

$$\begin{cases} f_1(x) = 2x_1^2 + x_1x_2 + x_2^2 - 6x_1 - 5x_2 \to \min, \\ f_1(x) = 3(x_1-2)^2 + 2(x_2-5)^2 \to \min, \\ g_1(x) = x_1 + x_2 \le 6, \\ g_2(x) = 3x_2 - 2x_1 \le 10. \end{cases}$$

1.2. Решение на языке программирования python

```
from scipy import optimize
from numpy import linalg

r = 1e-2
eps = 1e-9
C = 4

f1 = lambda (x1, x2): 2*x1**2 + x1*x2 + x2**2 - 6*x1 - 5*x2
f2 = lambda (x1, x2): 3*(x1-2)**2 + 2*(x2-5)**2
f = [f1, f2]

g1 = lambda (x1, x2): x1 + x2 - 6
g2 = lambda (x1, x2): 3*x2 - 2*x1 - 10
g = [g1, g2]

P = lambda x, r: r * sum([max( 0, gi(x) )**2 for gi in g]) / 2.0

F = lambda x, r: sum([wi * (fi( x ) - fki) for wi, fi, fki in zip(weight, f, ideal)]) + P(x, r)

def ideal_point( x ):
    x1 = tuple(optimize.minimize(f1, x).x)
    x2 = tuple(optimize.minimize(f2, x).x)
    return (f1(x1), f2(x2))

def get_weight():
    from random import randint
    a = randint(1, 10)

    c, v = linalg.eig([[ 1.0, 1.0 / a ], [ a*1.0, 1.0 ]])
    z = zip(tuple(c), [tuple(vi) for vi in v])

    return min(z)[1]

x = (100,100)
ideal = ideal_point(x)
weight = get_weight()

while True:
    x = tuple(optimize.minimize(lambda x: F(x, r), x).x)
    print 'x = %s, penalty: %s, ' % (list(x), P(x, r))
    if not P(x, r) <= eps:
        r *= C
    else:
        break

print 'result: %s' % list(x)
print 'f1(result) = %s' % f1(x)
```

```
print 'f2(result) = %s' % f2(x)
```

1.3. Результат работы

При значениях $\varepsilon = 10^{-9}, r^0 = 0.01, C = 4, x_0 = [100, 100]$ нашли точку $[1.4067797288803341, 3.9322034626828555]$, которая является точкой решения задачи многокритериальной оптимизации.

$$f_1([1.4067797288803341, 3.9322034626828555]) = -3.14966908245,$$

$$f_2([1.4067797288803341, 3.9322034626828555]) = 3.33610976041.$$