

Министерство науки и высшего образования
Федеральное государственное бюджетное образовательное учреждение высшего
образования
Югорский государственный университет

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №10
по дисциплине «Методы оптимизации»

Выполнил

Студент группы 11626

_____ Панчишин И. Р.

«___» _____ 2019 г.

Принял

Доцент ИЦЭ

_____ Самарин В. А.

«___» _____ 2019 г.

Ханты-Мансийск, 2019

Цель

Изучить численные методы поиска условного экстремума.

Задачи

1. Ознакомиться с методами штрафных функций.
2. Написать программную реализацию методов и применить ее для нахождения минимума.

Ход работы

Исходный код на языке *Octave*, используемый для иллюстрации поиска минимума, приведен в листинге ниже. Там можно встретить функции, которые были реализованы и рассмотрены в предыдущих лабораторных работах.

```
1  % Copyright © 2019 Panchishin Ivan
2
3  addpath('./code')
4
5  set(0, defaultaxesfontsize, 14);
6  set(0, defaulttextfontsize, 14);
7
8
9  % исходные данные
10 F = @(X) X(1)^2 + X(2)^2 - X(1)*X(2) - X(1) - X(2);
11 G = {
12     @(X) -X(1) + 3,
13     @(X) -X(2) - 1,
14     @(X) X(1)*3/2 + X(2) - 5
15 };
16
17
18 % контур
19 levels = [-2 -1 0 1 2 5 10 15 20 25 30];
20
21 [XX1, XX2, YY] = fungrid(F, -2, 5, 50);
22 c = contour(XX1, XX2, YY, levels, b--, DisplayName, Целевая функция);
23 hold on
24 grid on
25
26 [XX1, XX2, YY] = fungrid(@(X) penalty(F, G, X, 1), -2, 5, 50);
27 contour(XX1, XX2, YY, levels, r, DisplayName, Функция штрафов);
28
29 % ограничивающая область
30 plot([3, 3, 4, 3], [-1, 1, -1, -1], k);
31
32
33 % применение метода
34 R = [1 2 5 10 20 50 100 1000 10000 100000 1000000];
35
36 % внешняя точка
37 Hist = [4, 4.5];
38 gradparams = struct(a, -10, b, 10);
39 for r = R
40     [X0, ym] = graddesc(@(X) penalty(F, G, X, r), Hist(end, :), 0.1, coord, gradparams);
41     Hist = [Hist; X0];
```

```

42 end
43 plot(Hist(:, 1), Hist(:, 2), g*-, DisplayName, Путь внешней точки);
44
45 % внутренняя
46 Hist = [3.75, -0.95];
47 gradparams = struct(h, 1);
48 for r = [50, 100, 1000, 10000]
49     [Xm, ym] = graddesc(@(X) penalty(F, G, X, r, interior), Hist(end, :), 0.5, primal, gradparams)
50     Hist = [Hist; X0];
51 end
52 plot(Hist(:, 1), Hist(:, 2), m*-, DisplayName, Путь внутренней точки);
53
54
55 title(Метод штрафных функций);
56 xlabel(x1);
57 ylabel(x2);
58 legend
59
60 pause

```

Исходный код штрафных функций:

```

1  % Copyright © 2019 Panchishin Ivan
2
3  % метод штрафных функций
4  % penalty method
5
6  % G - массив штрафных функций (отрицательное значение в ограничивающей области)
7  % r - строгость штрафа
8  % type - метод внешней (interior) или внутренней (exterior) точки
9
10 function p = penalty(F, G, X, r, type)
11     Gres = [];
12     for i = 1:length(G)
13         Gres = [Gres, G{i}(X)];
14     end
15
16     if (nargin == 5 && type == interior)
17         p = F(X) - 1/r * sum(1./Gres);
18         return
19     end
20
21     % exterior
22     p = F(X) + r*sum(cut(Gres) .^2);
23 end
24
25 % функция срезки
26 function res = cut(x)
27     res = (x > 0) .* x;
28 end

```

Результат поиска минимума приведен на рисунке 1. Можно заметить, что в ограничивающей области линии уровней или значения целевой и штрафной функций совпадают, что объясняется тем, что штраф не накладывается и штрафная функция вырождается в целевую. Также можно заметить, что штрафная функция имеет свой минимум, смещенный в зависимости от величины масштабного коэффициента штрафа в сторону ограничивающей области.

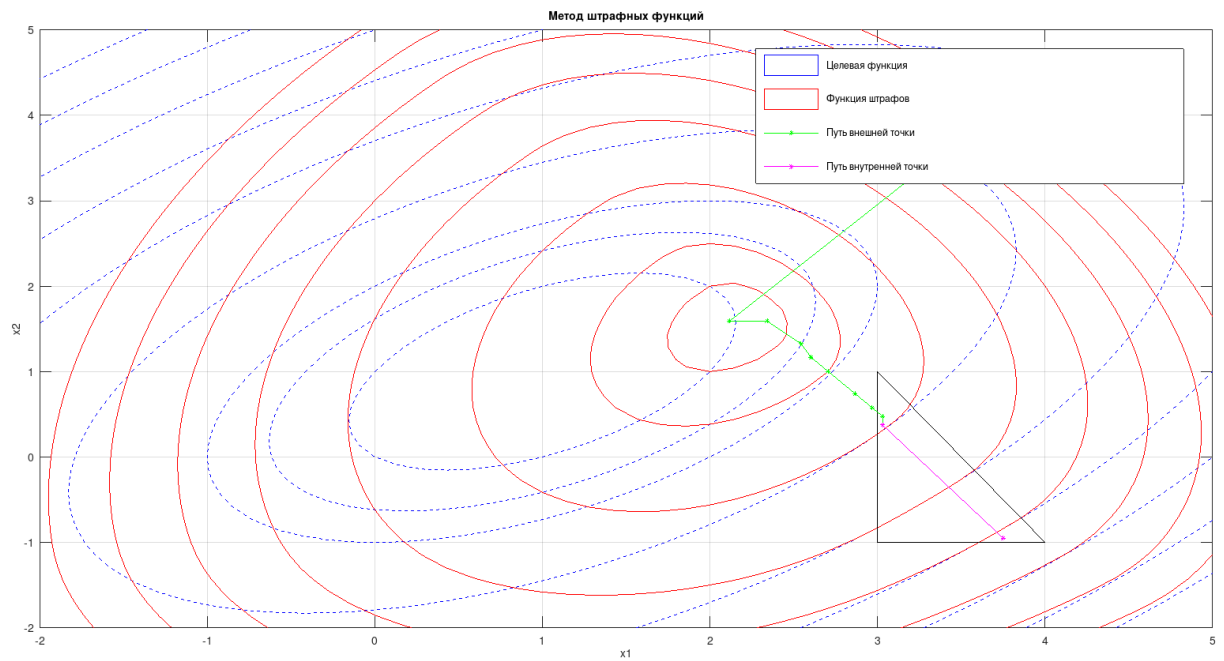


Рис. 1: Применение методов штрафных функций

Вывод

Ознакомился с методами штрафных функций, реализовал и применил их. В процессе выполнения столкнулся с проблемой "преодоления барьера" в методе внутренней точки. Так как штраф накладывается по мере приближения к границе области, важно не перепрыгнуть ее. Для этого нужно правильно подобрать отрезок поиска или шаг.