Assingment Project 3 and 4

Name: Tiago Moreira Trocoli da Cunha

RA: 226078

Course: Nonlinear Optimization.

In these projects, I implemented the projection gradient method and Penalty Algorithm. Both algorithms try to solve the following problem:

$$
\begin{aligned}
&min\ x_1^2 + x_2^2 + x_3^2 + x_4^2 - 2x_1 - 3x_4 \\
&s.a\ 2x_1 + x_2 + x_3 + 4x_4 = 7 \\
&\quad x_1 + x_2 + 2x_3 + x_4 = 6 \\
&\quad - x_i \leq 0, i = 1, 2, 3, 4
\end{aligned}
$$

According to Wolfram Alpha (click here), the global minima is:

$$x = (1.123, 0.65, 1.828, 0.568).$$

1) Project 3

In this part, I implemented the projection method (standard algorithm). The files descriptions are:

- projgradient.m = projection gradient method.
- fun2.m = objective function of the problem.
- maxOfAlpha.m = auxiliar function for projgradient.m.
- armijo.m = the algorithm uses armijo as a linear search.

The projection gradient method tries to solve the general problem:

$$minimize\ f(x) \quad s.a \quad a_i^T x \leq b_i, \quad i = 1, 2, 3, ..., m$$

Thus, the equality constraints becomes:

$$
\begin{aligned}
&2x_1 + x_2 + x_3 + 4x_4 \leq 7 \\
&x_1 + x_2 + 2x_3 + x_4 \leq 6
\end{aligned}
$$

The projgradient.m function needs the following inputs:

- fun2 = the objective function.
- $R = [a_1\, a_2 \, . . . \, a_m]^T$, that is, the constants of constraints.
- $b = [b_1\, b_2 . . . b_m]$, that is, the independent constants of constraints.
- the initial feasible initial point, x.

And its output is:

- The minima, xmin.
- Matrix A, in which each row is the active constraint.

- List of lambdas of active constraints.

```
R = [2 1 1 4; 1 1 2 1; -1 0 0 0; 0 -1 0 0; 0 0 -1 0; 0 0 0 -1]; %constraints
b = [7 6 0 0 0 0]; % independent terms of constraints
x = [2 2 1 0]; %initial position
[xmin,lambda,A] = projgradient(@fun2, R, b, [2 2 1 0])
```

```
xmin = 1×4
    0.9091   -0.0455   -0.0455    1.3182
lambda = 0.0909
A = 1×4
     2     1     1     4
```

The solution is $x = (0.9091, -0.0455, -0.0455, 1.3182)$. Unfortunately, it is not a feasible point. The algorithm activated only the first constraint and left out the second. Something is wrong with this algorithm, but I don't know in which part. I spent days trying to figure out the problrem without success.

2) Project 4

In this section, I implemented the "Penalty Algorithm" (pg. 289). The files description are:

- cfun.m = penalty function.
- barrierMethod.m = Penalty Algorithm.
- gradient_descent.m = the algorthm uses the gradient descent as a minimization method.

The penalty function is in file cfun.m and is:

$$p(x) = (2x_1 + x_2 + x_3 + 4x_4 - 7)^2 + (x_1 + x_2 + 2x_3 + x_4 - 6)^2 + \max\{-x_i, 0\}, i = 1, 2, 3, 4.$$

The Penalty Algorithm needs the following inputs:

- objective function.
- penalty function.
- initial position.

```
barrierMethod(@fun2, @cfun, [1 1 1 1])
```

```
ans = 1×4
    1.1233    0.6505    1.8283    0.5688
```

The solution is $x = (1.1233, 0.6505, 1.8283, 0.5688)$, that is the **minima found by Wolfram Alpha**. Thus, not only the algorithm worked, but found the global minima. The cost of penalty function (p(x)) is just:

```
cfun([1.1233 0.6505 1.8283 0.5688])
```

```
ans = 1.0000e-06
```

That is only 10^{-6}!