Technical University of Denmark

Constrained Optimization

Course 02612

Assignment 1

Authors:

Jakub Czerny, s99999

Oskar Hint, s161559

Joachim Finn Jensen, s134052

March 24, 2017

Contents

1 Introduction **1**

2 Assignment **1**

2.1 Problem 1 - Quadratic Optimization 1

2.2 Problem 2 - Equality Constrained Quadratic Optimization 3

2.3 Problem 3 - Inequality Constrained Quadratic Programming 3

2.4 Problem 4 - Markowitz Portfolio Optimization 6

2.5 Problem 5 - Interior-Point Algorithm for Convex Quadratic Programming . . 6

3 Conclusion **6**

1 Introduction

this is some report

2 Assignment

2.1 Problem 1 - Quadratic Optimization

From the problem 1 it is seen that the system can be written as:

$$H = \begin{bmatrix} 6 & 2 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix}$$

$$g = \begin{bmatrix} -8 \\ -3 \\ -3 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

Then the general KKT system can be written as:

$$\begin{bmatrix} H & -A \\ A^T & 0 \end{bmatrix} \times \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} -g \\ b \end{bmatrix}$$

In this specific case it give:

$$\begin{bmatrix} 6 & 2 & 1 & -1 & 0 \\ 2 & 5 & 2 & 0 & -1 \\ 1 & 2 & 4 & -1 & -1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 8 \\ 3 \\ 3 \\ 3 \\ 0 \end{bmatrix}$$

The system is solve in MatLab by macking a function that solves the problem by use of the backslash command, see the code how it was done. The result shows that:

$$x = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

$$\lambda = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

To test the solver, random values for x and λ are generated, from these the H, g, A and b are determined. With these matrix and vectors the system are solved again. If the function found x and lambda there was randomly chose, the solver works. In this case the solve works as expected and gives the same values for x and lambda as there were generated.

To see the sensitivity of the problem the parameter sensitivity analyse is made. The general function is as followed.

$$F(x,\lambda;p) = \begin{bmatrix} \nabla_x f(x,p) - \nabla_x c(x,p)\lambda \\ c(x,p) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Where the implicit function theorem gives parameter sensitivities

$$[\nabla_x(p)\lambda(p)] = -[W_{xp} - \nabla_p c(x,p) \begin{bmatrix} W_{xx} & -\nabla_x c(x,p) \\ -\nabla_x c(x,p)^T & 0 \end{bmatrix}^{-1}$$

$$W_{xx} = \nabla^2_{xx} f(x,p) - \sum_{i \in \varepsilon} \lambda_i \nabla^2_{xx} c_i(x,p)$$

$$W_{xp} = \nabla^2_{xp} f(x,p) - \sum_{i \in \varepsilon} \lambda_i \nabla^2_{xp} c_i xp)$$

From this the sensitivity of this problem can be written as:

$$f(x,p) = \frac{1}{2} x^T H x + (g + \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix})^T x$$

$$c(x,p) = A^T x - b - \begin{bmatrix} p_5 \\ p_6 \end{bmatrix}$$

$$w_{xp} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$W_{xx} = H$$

$$[\nabla x(p)\lambda(p)] = -\begin{bmatrix} H & -A^T \\ -A & 0 \end{bmatrix}$$

This are added to the solver and the sensitivity are found as:

$$\nabla x(p) = \begin{bmatrix} -0.0769 & -0.0769 & 0.0769 \\ -0.0769 & -0.0769 & 0.0769 \\ 0.0769 & 0.0769 & -0.0769 \\ 0.4615 & -0.5385 & 0.5385 \\ -0.3846 & 0.6154 & 0.3846 \end{bmatrix}$$

$$\lambda(p) = \begin{bmatrix} 0.4615 & -0.3846 \\ -0.5385 & 0.6154 \\ 0.5385 & 0.3846 \\ 2.2308 & -0.6923 \\ -0.6923 & 3.0769 \end{bmatrix}$$

Why is this correct??? When having such a quadratic problem it's always also have a dual problem.

2.2 Problem 2 - Equality Constrained Quadratic Optimization

blablablablbal

2.3 Problem 3 - Inequality Constrained Quadratic Programming

From page 475 in Nocedal and Wright the following system is given.

$$\begin{aligned} \min_x q(x) &= (x_1-1)^2 + (x_2-2.5)^2 \\ s.t. x_1 - 2x_2 + 2 &>= 0, \\ -x_1 - 2x_2 + 6 &>= 0, \\ -x_1 + 2x_2 + 2 &>= 0, \\ x_1 &>= 0, \\ x_2 &>= 0. \end{aligned} \tag{1}$$

in MatLab a contour plot of this is made and seen in figure 1.

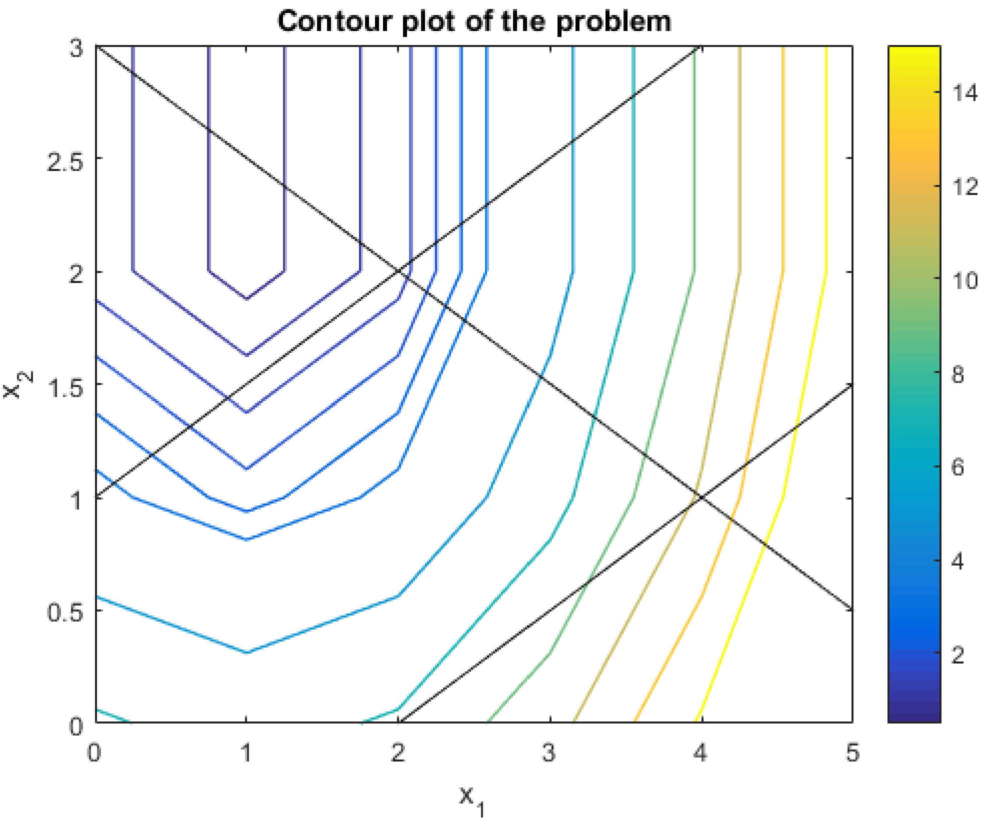


Figure 1: A contour plot of the problem.

From the constrain lines it is seen that the feasible region is a pentagon starting from the origin.
The problem can be written in the standart martix way as:

$$H = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

$$g = \begin{bmatrix} -2 \\ -5 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & -1 & -1 & 1 & 0 \\ -2 & -2 & 2 & 0 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} -2 \\ -6 \\ -2 \\ 0 \\ 0 \end{bmatrix}$$

Then the general KKT system can be written as:

$$\begin{bmatrix} H & -A \\ A^T & 0 \end{bmatrix} \times \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} -g \\ b \end{bmatrix}$$

When the system can be solve with the same QP solver as used for problem 1. That gives:

$$x = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$$

$$\lambda = \begin{bmatrix} 0 \\ 1.8 \\ -0.3 \\ -2.1 \\ 3.0 \end{bmatrix}$$

Where all the values are multiply with 10 in the power of 16.
A way to found the optimal solution is with an active set algorithm. This is applyed as the one in the Example 16.4 in N and W. The algorithm is seen in the MatLab code and the contour plot with the four different workingset are seen in figure 2 below.

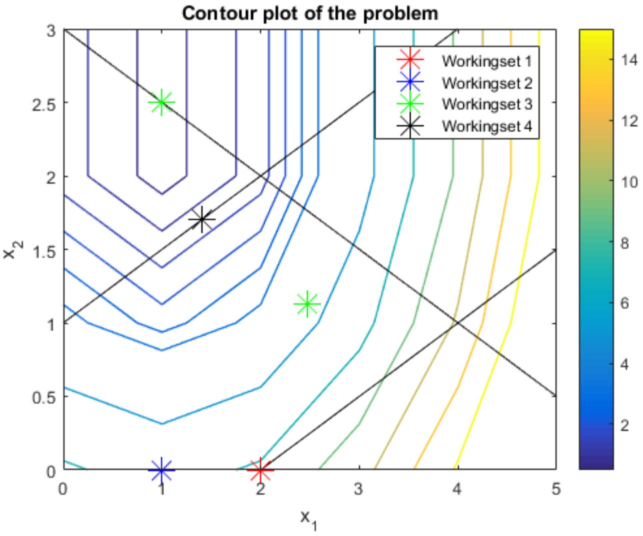
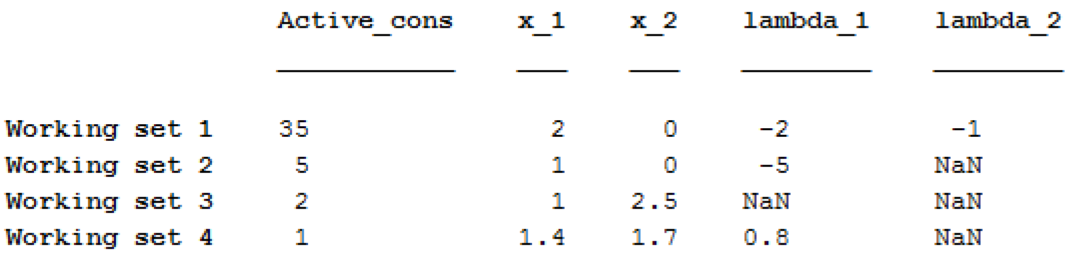


Figure 2: A contour plot of the problem, with the working set applyed.

In figure 3 below, values for the iteration are shown.



```
                  Active_cons    x_1    x_2    lambda_1    lambda_2
                  ___________    ___    ___    ________    ________

Working set 1     35               2      0      -2          -1
Working set 2      5               1      0      -5          NaN
Working set 3      2               1    2.5    NaN         NaN
Working set 4      1             1.4    1.7    0.8         NaN
```

Figure 3: Table from MatLab showing values from the four iterations.

From the figure it is seen that for the first working set, constrain 3 and 5 where active. This set found the point (2;0) at the contour plot it is seen that this point is feasible, but that it is not the optimal solution. This is also seen from the Lagrange multipliers, λ, where they both are negative which they don't are for the optimal solution.
The next step was to chose a new working set, since the first have two active constrains one of them was remove. Here the 'rule of thumbs' says that the constrain with the most negative Lagrange multipliers should be inactive. There for working set 2 have the 5th. constrain active. This iteration gives the solution (1;0) and λ = -5. Since the Lagrange multiplier still is negative, the point is not the optimal solution, this is also confirmed from the contour plot. The 3th. step is, again, to remove the constrain belong to the most negative Lagrange multiplier. In this case there are only one, this means that the third working set doesn't have any active constrains. In that case there wouldn't be any Lagrange multipliers since they 'belongs' to the constrains. The point found in is iteration was (1;2.5). Since there are no Lagrange multiplier the point is inserted in the 'original' constrains which all of the should be equal ore bigger than zero. From the first constrain it is seen that it gives -2 which mean that the constrain is not full fill and therefor not a solution to the problem. From the contour plot it is seen that the point is in the center (if it doesn't have any constrain), but it is also seen that the point is beyond one of the constrain and there for confirm that the found point is not a solution.
Finally the last working set is chosen to be the first constrain since it was this one there was not full fill. This give the point (1.4;1.7) and λ = 0.8. Since the Lagrange multiplier is positive it could be an optimal solution, there for all the constrain are controlled with the found point. It is seen that they give; 0, 1.2 and 4 for the first three constrain, and since constrain four and five are the x_1 and x_2 bigger ore equal zero they are also full fill. This mean that the found solution is valid and the Lagrange multiplier indicate an optimal solution. This is finally verified by the contour plot where it is easy seen that the point is the global minimum, and the problem is solved.

EXPLAIN THE ACTIVE SET METHODE!!

For the active set method to work, an feasible starting point is needed. A good way to found such a point is by using the MatLab function linprog. This is done and can be seen in the MatLab code. This calculate a feasible point (2.47;1.12). In figure 4 below, the found point is seen, and it is seen that it is feasible as expected.

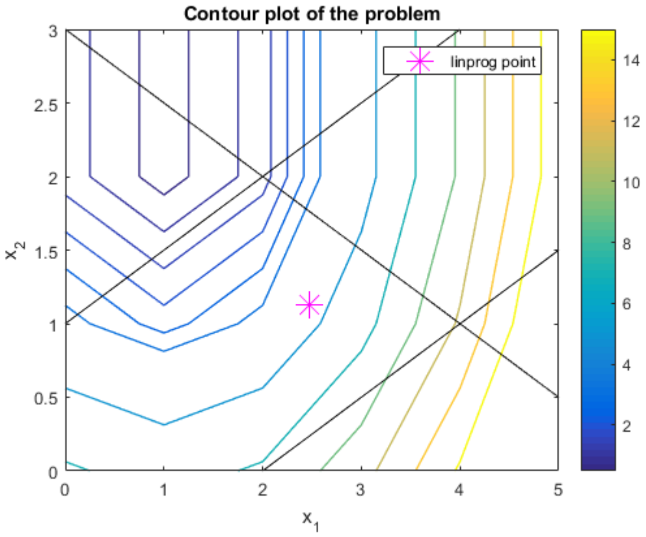


Figure 4: The contour plot with the feasible starting point found with linprog.

Finally the entire problem could have been solve using the MatLab program quadprog. This function only needs the matrix H and A, the vectors g and b, some boundaries and a feasible starting point. By inserting the feasible starting point found with linprog, and the system in the form og H,g A and b. The function found the optimal solution to be (1.4;1.7). Is is consistent with the one found from the active set algorithm, and seen in the contour plot as 'working set 4' in figure 2.

2.4 Problem 4 - Markowitz Portfolio Optimization

blablablablbal

2.5 Problem 5 - Interior-Point Algorithm for Convex Quadratic Programming

blablablablbal

3 Conclusion

Some conclusions things