Compulsory Project 2
MAT-INF3100 Linear optimization, Spring 2016

- **NOTE.** Send the project by *Thursday April 14th, 15:00 o'clock*, to Torkel Haufmann (torkelah@math.uio.no) in a PDF file that you name "username.pdf" (your username!). Also attach your computer code, either as a listing in the PDF file or (preferably) as separate files. Moreover, you should read the general information about compulsory projects at the course web page.

Problem 1

1a)

Consider the linear programming problem

$$\max c^T x, \quad \text{subject to } Ax \leq b,\ x \geq 0, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $x, c \in \mathbb{R}^n$. State the weak and strong duality theorems. Moreover, prove the weak duality theorem.

1b)

Let $x \in \mathbb{R}^n$ be primal feasible and $y \in \mathbb{R}^m$ dual feasible. Denote by $w \in \mathbb{R}^m$ the corresponding primal slack variables and $z \in \mathbb{R}^n$ the corresponding dual slack variables. Suppose x is optimal for the primal problem and y is optimal for the dual problem. State and prove the complementary slackness equations.

1c)

Consider the linear programming problem

$$\begin{array}{lrcrcl} \text{maximize} & 3x_1 & + & 2x_2 & & \\ \text{subject to} & & & & & \\ & 2x_1 & + & x_2 & \leq & 4, \\ & 2x_1 & + & 3x_2 & \leq & 6, \\ & & & x_1, x_2 & \geq & 0. \end{array} \tag{2}$$

Show that $x^\star = (3/2, 1)$ is primal feasible and $y^\star = (5/4, 1/4)$ is dual feasible. Moreover, show that $x^\star$ is an optimal solution of (2).

Problem 2

2a)

Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. Consider the linear programming problem

$$\max 0^T x, \quad \text{subject to } Ax = b,\ x \geq 0, \tag{P}$$

where 0 denotes the m-dimensional null vector. Show that the dual of (P) is

$$\begin{aligned} &\min b^T y, \\ &\text{subject to } A^T y \geq 0. \end{aligned} \tag{D}$$

2b)

Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. Farkas' Lemma states that exactly one of the following two systems has a solution:

$$Ax = b, \quad x \geq 0. \tag{S1}$$

$$A^T y \geq 0, \quad b^T y < 0. \tag{S2}$$

Use the weak/strong duality theorems for (P) and (D) to prove Farkas' Lemma.

Problem 3

Consider the following linear programming problem on matrix form:

$$\max c^T x, \quad \text{subject to } Ax \leq b,\ x \geq 0. \tag{3}$$

In this exercise you will program two algorithms for solving this problem.

The simplex method on matrix form. Detailed in Section 6.2 of the book. The function should be called `simplex`, take A, b, and c as arguments, and return the optimal solution x.

The path-following method. Presented in Figure 18.1 in the book (you will need to consult Chapters 17–19 for more details). The function should be called `interior`, take A, b, and c as arguments (and perhaps a tolerance ϵ), and return the optimal solution x.

To simplify somewhat we will assume $b \geq 0$, so you do not have to implement a two-phase method. Your solution should contain the following:

- The code for both algorithms, preferably as separate files.
- A test run using the data files accompanying this assignment – print out a line of information for each iteration (or every few iterations).
- A brief discussion and comparison of your results with the two algorithms.

You may choose whatever language you like, but we recommend Python or MATLAB.

Some tips:

- The optimal solution value for the test problem is ≈ 321.4047.
- When solving a linear system $Vx = d$ do not use the `inv(V)` command (or the corresponding command in your language of choice) – rather, use the backslash operator: `x = V \ d`.
- The major part of the work for the interior-point algorithm is solving the *KKT-system* to find the steps Δx, Δy, Δz, and Δw. There are various ways to do this, but we suggest the *normal equations in primal form*, see the start of Section 19.9 in the book (equation (19.9)).
- If you are using MATLAB, you can get the A, b, and c matrices from the `.mat` file using the command `load('matinf3100oblig2.mat')`.
- In Python, assuming you have imported numpy as `np`, you can use the following commands to read the data from the text files:

```
c = np.loadtxt('c.txt', delimiter = ',')
A = np.loadtxt('A.txt', delimiter = ',')
b = np.loadtxt('b.txt', delimiter = ',')
```

- The test problem is quite large, so it might be a good idea to develop against a simpler problem at first see for example pages 96–101 in the book by Vanderbei (2nd ed.).