# Homework 1: Computational Economics
Spring, 2017
Due: February 6, 2017, 11:59 PM

*Homework is individual work. You may discuss it with classmates, but sharing the homework problems or solutions (any solutions, not just your own) with anyone, at any time (including after the class has completed) is viewed to be in violation of the honor code.*

**Homework submission:** Homeworks must be submitted through OAK. All source code should be zipped and uploaded, as described below. For all code that needs to be compiled, include a Makefile, and run make as a part of your script (described below) to compile it. All other solutions should be submitted as a single pdf document named as follows: "lastname_hw1_solutions.pdf" (I recommend using LaTeX, as it makes typing up math relatively easy, and you can compile your document into a pdf by using the pdflatex command).

**Implementation Options:** You can implement the algorithms below using any of the following: python, octave, Java. If you wish to use any other language, please consult the instructor, but the following will definitely *not* be allowed: Matlab (use octave instead, which is open source and freely available for any system), Mathematica, Maple. Also, be sure that your code can compile/run on a UNIX system as is.

**Late submission policy:** Any submission between 12:00AM and 11:59 PM on February 7, 2017 will receive 50% of the credit. Any submission that is later than that will receive 0 points.

## 1 Theoretical Analysis [40 points]

Prove the following results:

a) Consider a linear system $Ax = b$ and let $\hat{A} = [A, b]$ be the augmented matrix. Prove that adding a multiple of one row to another in $\hat{A}$ does not change the set of solutions of $Ax = b$.

b) Let $A$ be a square matrix with $m$ rows and columns. Prove (from first principles) that a linear system $Ax = b$ has a solution for every $b$ iff $rank(A) = m$. (Hint: consider the reduced row echelon form.) Do it in three steps:

1. Prove that $rank(A) \leq m$.
2. Prove that if $rank(A) = m$, then $Ax = b$ has a solution for every $b$.
3. Prove that if $rank(A) < m$, then there exists a $b$ such that $Ax = b$ has no solution.

c) Let $\{S_i\}$ be a (possibly infinite) family of convex sets. Prove that the set $S = \cap_i S_i$ is convex. Give an example where $R, T$ are convex, but $S = R \cup T$ is not.

d) Write the dual formulation of the following LP:

$$\begin{aligned}
&\min_{x,y,z} 2x + 3y + 2z \\
&\text{subject to :} \\
&x - 4y + z \geq 1 \\
&z \geq 0.
\end{aligned}$$

e) Write the dual formulation of the following LP:

$$\begin{aligned}
&\max_{x,y,z,d} d \\
&\text{subject to :} \\
&d \leq 2x + y \\
&d \leq z \\
&x + y + z \leq 5.
\end{aligned}$$

# 2 Production Problem [40 points]

Consider the following optimization problem that comes up in production planning. Suppose that you can produce $m$ goods (which are indexed by $i$) using $n$ raw materials (inputs, indexed by $j$). You would like to do this so as to make the most money (i.e., maximize revenue). You can sell each good $i$ produced at a market price $p_i$. Thus, if you decide to produce $x_i$ units of each good $i$, your revenue is

$$R = \sum_{i=1}^{m} p_i x_i.$$

In order to produce a unit of good $i$, you need to use $a_{ij}$ units of raw material $j$. However, you only have $b_j \geq 0$ units of each raw material $j$.

1. Formulate *Problem 2* as a linear program. Write your formulation and submit to OAK as a part of your homework 1 solutions.

2. Using the data ($p_i$'s, $a_{ij}$'s, $b_j$'s) for this problem on OAK described in file "data_production.txt", compute the optimal solution and corresponding revenue for *Problem 2* using CPLEX. You must submit all your code, and include a shell script "runcplex.sh" that can run on a UNIX system which has CPLEX available on the path and output a file, "xcplex.txt" specifying the solution $x$, which is a column vector of real values with exactly 2 decimal places. You can choose to either use AMPL to specify your model and execute CPLEX, or to use a language of your choice (from the options listed above) using CPLEX API. Zip all your solution files and source code, name this zip file "solnProblem2.zip", and submit through OAK.

# 3 Implement the Simplex Method [40 points]

Implement the simplex method described in class in a language of your choice (from the options listed above). Use the data on OAK in files "A.txt", "b.txt", and "c.txt" as input data to your simplex routine. Provide a shell script, "runsimplex.sh", which can be executed on a UNIX system which would read these files as input parameters (for matrix $A$, vector $b$ in the inequality $Ax \leq b$, and cost vector $c$, respectively), and output a file "xsimplex.txt" specifying the solution $x$, which is a column vector of real values with exactly 2 decimal places. In your implementation, assume that there is also a non-negativity constraint on $x$, so that the problem has the form

$$\begin{aligned}
&\min_x c^T x \\
&\text{subject to :} \\
&Ax \leq b \\
&x \geq 0.
\end{aligned}$$

Zip all your source code, name this zip file "solnProblem3.zip", and submit through OAK.

# 4 Implement the KKT Method for Solving Equality Constrained Convex QPs [40 points]

Implement the method for solving equality constrained convex quadratic programs using the KKT system described in class. Use the data on OAK in files "A_QP.txt", "d_QP.txt", "M_QP.txt", and "b_QP.txt" as inputs to your routine. These correspond to matrix $A$, vector $d$, matrix $M$, and vector $b$ in the following formulation:

$$\begin{aligned}
&\min_x x^T Ax + d^T x \\
&\text{subject to :} \\
&Mx = b.
\end{aligned}$$

Provide a shell script, "runkkt.sh", which can be executed on a UNIX system which would read these files as input parameters and output a file "xkkt.txt" specifying the solution $x$, which is a column vector of real values with exactly 2 decimal places. Zip all your source code, name this zip file "solnProblem4.zip", and submit through OAK.

# 5 Monopolist Production Problem [40 points]

Consider the production problem above again, but suppose that the production amount actually impacts the market price. Specifically, if $x_i$ units of good $i$ are produced, suppose that the market price is

$$p_i = P_i - r_i x_i,$$

where $P_i, r_i > 0$.

a) Formulate this problem as a quadratic program.

b) Prove that the resulting quadratic program is convex.

c) Solve this problem using CPLEX. The data for this problem is described in file "data_monopoly.txt" on OAK. Create a script "runqpcplex.sh" to run the solver and produce a file "xQPcplex.txt" which is the solution column vector $x$ to 2 decimal places. Zip your source code into "solnProblem5.zip".