CPSC 4850/5850/7850 – Algorithms in OR
Assignment 2, total 50(70) pts **Due Feb 25**

Instructions: hand in your written or typed answers in class on the due date shown. If you are taking the class as undergraduate, problems marked *(GRAD)* are optional.

Problem 1) **[10 pts]** Consider the LP in standard form,

$$\begin{aligned} \zeta &= \min c^T x, \\ Ax &= b \\ x &\geq 0, \end{aligned}$$

whose constraint, cost, and right hand side coefficients are given in a table with the following layout.

$$\begin{array}{c|l} A & b \\ \hline c^T & \zeta, \text{ minimize} \end{array}$$

Use the problem whose table is given below to solve exercises 1a-1b.

$$\begin{array}{ccccccc|l} \xi_1 & \xi_2 & \xi_3 & \xi_4 & \xi_5 & \xi_6 & \xi_7 & b \\ \hline 1 & 2 & -2 & 1 & 0 & 3 & 1 & -5 \\ -2 & 1 & -1 & 2 & 1 & 1 & 2 & 1 \\ 1 & 0 & 2 & 2 & 2 & 2 & -2 & 8 \\ \hline 10 & 0 & 9 & -2 & -1 & -1 & -10 & \zeta, \text{ minimize} \end{array} \qquad \xi_i \geq 0 \text{ for all } i.$$

(a) Write the dual of the problem and the complementary slackness conditions.
(b) Using complementary slackness, check whether $y^T = (-2, -3, 1)$ is an optimal dual solution. If it is, derive an optimal solution to the primal.

Problem 2) **[10 pts]** Consider a balanced transportation problem with 4 sources and 6 destinations whose cost matrix c_{ij} and availabilities α_i and β_j are given in the following table.

$$\begin{array}{c|cccccc|c} & & & & & & & \alpha_i \\ \hline & 10 & 2 & 9 & 1 & 11 & 12 & 13 \\ & 12 & 9 & 3 & 11 & 4 & 15 & 31 \\ c_{ij} & 3 & 7 & 10 & 9 & 6 & 6 & 51 \\ & 12 & 9 & 11 & 3 & 5 & 18 & 21 \\ \hline \beta_j & 17 & 4 & 16 & 13 & 54 & 12 & \end{array}$$

Find an initial basic feasible solution using Vogel's method. Test whether the solution you found is optimal. If not, compute a new basic feasible solution by performing a pivot operation (θ-loop). Test whether the second solution obtained is optimal. If not, give an estimate on how far from optimal (in terms of cost) your second solution is.

Problem 3) **[10 pts]** Solve the assignment problem with the following cost matrix, where the entries are the costs of assigning a worker to a job and the objective is to minimize the total cost of the assignment. Provide enough details to show how you calculate the main steps of the primal-dual algorithm, but you should find the best primal solution for a given dual, by inspection (no need to go through the steps of finding augmenting paths).

$$\begin{pmatrix} 5 & 3 & 7 & 3 & 4 \\ 5 & 6 & 12 & 7 & 8 \\ 2 & 8 & 3 & 4 & 5 \\ 9 & 6 & 10 & 5 & 6 \\ 3 & 2 & 1 & 4 & 5 \end{pmatrix}$$

Problem 4) **[20 pts]** Write an Octave function called *pivot* which performs a full pivot operation on a matrix. The function has three parameters (A, r, c) and two return values, $[M, stat]$. The parameters are: A the matrix, r the row pivot, and c the column pivot. The return values are: M the matrix after the pivot operation, and *stat* which is true if pivoting succeeded and false otherwise. A typical error condition is to pivot when $A(r, c)$ is zero. A full pivot operation means that column c of the resulting matrix is zero everywhere except at row r where it is 1. *Hint: use the column notation to change entire row of matrices.*

What to submit:

(i) A printout of your function, appropriately commented.

(ii) Generate a 5 by 5 random matrix. Perform one pivot operation on a column/row of your choice. Display the resulting matrix. For all these, submit a printout of your session captured with the *diary* command.

Problem 5) **[20 pts]** (GRAD) Using the pivot operation that you implemented for Problem 4, write another function called *myinverse* which returns the inverse of a square matrix given as parameter (see Section 3.3.8 in the text). Test your function by comparing its result with Octave's own matrix inverse operation *inv.*

What to submit:

(i) A printout of your function, appropriately commented.

(ii) Generate a 5 by 5 random matrix. Display the inverse returned by your function and then the inverse returned by Octave's *inv* function.

(iii) Download from the course webpage, in the section for Assignment 2, the three matrix files and the Matlab/Octave routine for reading these files. Type "*help mmread*" to get instructions on the function. The matrices and the code was downloaded from the Matrix Market database where you can find more information (`http://math.nist.gov/MatrixMarket/`).

(iv) Compute the inverse for the three matrices above, first using your routine and then using Octave's *inv* method. To compare the results, do NOT display the matrices, they are too large. Instead, select 4 arbitrary entries which you will show for comparison. Make sure you measure the time spent in computation for each function call, using *etime* and *clock* functions. Submit a printout of your session captured with the *diary* command.