Department of Mathematical Sciences
CARNEGIE MELLON UNIVERSITY

OPERATIONS RESEARCH II 21-393

Homework 5: Due Monday November 7.

1. Players A and B choose integers i and j respectively from the set $\{1, 2, \ldots, n\}$ for some $n \geq 2$. Player A wins if $|i - j| = 1$. Otherwise there is no payoff. Solve the game.

   Solution: Let A_n be the matrix of the game. For example,

$$A_9 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & 1 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 & 1 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 \end{bmatrix}$$

   The matrix A_n is non-singular. These games are discussed in my notes, on the last page. Let $\mathbf{1}_n$ be the $n \times 1$ matrix $[1, 1, \ldots, 1]^T$.

   Now the solution to $A_n y = \mathbf{1}_n$ is given by $y_j = 2^{j-1}$ and is non-negative. Thus $y_1 + \cdots + y_n = 2^n - 1$. So we put $q_j = \frac{2^{j-1}}{2^n-1}$.
   The solution to $x^T A = \mathbf{1}_n^T$ is given by $x_i = 2^{n-i}$ and is non-negative. Thus $x_1 + \cdots + x_n = 2^n - 1$. So we put $p_i = \frac{2^{n-i}}{2^n-1}$.

   The vectors p, q solve the game. This follows from the notes. One can check that they solve the dual pair of linear programs associated with the game.

2. Find the optimal ordering strategy for the following inventory system. If you order an amount Q, it costs AQ^α for some $0 < \alpha < 1$ and the inventory cost is I per unit per period. The demand is λ units per period and no stock-outs are allowed.

   Solution: If we order Q units at a time then the total cost per period is

$$\frac{\lambda A Q^\alpha}{Q} + \frac{IQ}{2}.$$

The optimal choice of Q is therefore $(2\lambda A(1-\alpha))^{1/(2-\alpha)}$.

3. Show that EDD is an exact algorithm for $1 \mid r_j, pmtn \mid L_{max}$ i.e. there are n jobs with release dates $r_1, r_2, \ldots, r_n$ and due dates $d_1, d_2, \ldots, d_n$, preemption is allowed and the goal is to minimise the maximum lateness $C_j - d_j$.

   Solution: If at time t, job j with the earliest remaining due date is not being processed and job k with a later due date is, we reallocate the time spent processing job k to job j. This makes job j finish earlier and job k finish when job j did originally. Thus

$$\begin{aligned}\max\{C_j'-d_j, C_k'-d_k\} = \max\{C_j'-d_j, C_j-d_k\} &\leq \max\{C_j-d_j, C_j-d_k\} \\ &= C_j - d_j \leq \max\{C_j - d_j, C_k - d_k\}.\end{aligned}$$