Department of Mathematical Sciences

CARNEGIE MELLON UNIVERSITY

OPERATIONS RESEARCH II 21-393

Homework 3: Due Wednesday October 9.

Q1 Analyse the following inventory system and derive a strategy for minimising the total cost. There are n products. Product i has demand λ_i per period and no stock-outs are allowed. The cost of making an order for Q units of a mixture of products is AQ^α where $0 < \alpha < 1$. The inventory cost is I times $\max\{L_1, L_2, \ldots, L_n\}$ per period where L_i is the average inventory level of product i in that period.

Solution: We argued in class that we order items at intervals T. Thus we order $Q_i = T\lambda_i$ units of item i each time and $L_i = Q_i/2$. Let $\lambda = \max\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$. Then the inventory cost is therefore $IT\lambda/2$. This gives us a total cost of

$$K = \frac{A\left(T\sum_{j=1}^n \lambda_i\right)^\alpha}{T} + \frac{IT\lambda}{2} = \frac{B}{T^{1-\alpha}} + \frac{IT\lambda}{2}$$

where $B = A\left(\sum_{j=1}^n \lambda_i\right)^\alpha$.

Now

$$\frac{dK}{dT} = -\frac{B(1-\alpha)}{T^{2-\alpha}} + \frac{I\lambda}{2}.$$

Therefore, the optimal value for T is given by

$$T^* = \left(\frac{2B(1-\alpha)}{I\lambda}\right)^{1/(2-\alpha)}.$$

Q2 Give an algorithm to solve the following scheduling problem. There are n jobs labelled $1, 2, \ldots, n$ that have to be processed one at a time on a single machine. There is an acyclic digraph $D = (V, A)$ such that if $(i, j) \in A$ then job j cannot be started until job i has been completed. The problem is to minimise $\max_j f_j(C_j)$ where for all j, f_j is a monotone increasing. As usual, C_j is the completion time of job j. This is distinct from its processing time p_j.

Solution: Let S be the set of jobs with no successor in D i.e. the set of sinks of D. The last job must be in S and it will complete at time $p = p_1 + p_2 + \cdots + p_n$. Let $f_k(p) = \min_{j\in S} f_j(p)$. We schedule k last and then inductively schedule the remaining jobs.

Q3 Convert the following into a standard assignment problem. We have a bipartite graph with bipartition $A = \{a_1, a_2, \ldots, a_m\}, B = \{b_1, b_2, \ldots, b_n\}$. An assignment now is a set of edges M such (i) a_i is incident to exactly r_i edges of M for $i = 1, 2, \ldots, m$ and (ii) b_j is incident to exactly s_j edges of M for $j = 1, 2, \ldots, n$. Here $\sum_i r_i = \sum_j s_j$. The cost of edge (a_i, b_j) is $c(i, j)$ and the cost of an assignment M is $\sum_{e\in M} c(e)$. The objective is to find a minimum cost assignment.

Solution We replace the vertex a_i by vertices $a_i(1), a_i(2) \ldots, a_i(r_i)$ for $i = 1, 2, \ldots, m$ and the vertex b_j by vertices $b_j(1), b_j(2), \ldots, b_j(s_j)$ for $j = 1, 2, \ldots, n$. The cost of edge $\{a_i(k), b_j(l)\}$ will be $c(a_i, b_j)$. Each solution x to the original problem can be mapped to $\prod_{i=1}^m \prod_{j=1}^n r_i!s_j!$ solutions of the expanded problem, and each of these has the same cost. Each solution to the expanded problem arises from a unique solution to the original problem.