Problem Sheet on Network Optimisation

---

1. Consider an Internet router with N input ports and N output ports. Each input port maintains a separate queue for packets intended for each output port. Let Q_{ij} denote the number of packets at input port i waiting to be delivered to output port j. Time is slotted and, in each time slot, the router can schedule a *matching* between the input and output ports, namely a 1-1 map of input to output ports. For example, if $N = 3$, a possible matching is $\{(1, 3), (2, 2), (3, 1)\}$, while $\{(1, 3), (2, 3), (3, 1)\}$ is not a matching. More formally, denoting the input and output ports by the sets $\mathcal{I}$ and $\mathcal{O}$, a matching is a map $\sigma : \mathcal{I} \to \mathcal{O}$ such that $\{\sigma(i) = \sigma(j)\} \iff \{i = j\}, 1 \leq i, j \leq N$.

   One scheduling algorithm that is known to have good properties is the MaxWeight algorithm, which schedules a matching of maximum weight in each time slot. In other words, in each time slot, it seeks to find a matching that achieves

   $$\max_{\sigma} \sum_{i=1}^{N} Q_{i,\sigma(i)}.$$

   Suppose you are given an algorithm for the transshipment problem that takes as input an incidence matrix A, a vector of node demands b and a vector of edge costs c, and outputs an optimal integer-valued solution (if one exists, and declares the problem infeasible or bounded otherwise). Explain in detail how you would use this algorithm to solve the MaxWeight matching problem. What could go wrong if the algorithm is guaranteed to output an optimal solution, but not necessarily an integer-valued solution?

2. **König's theorem**: If, in a set of n boys and n girls, every girl knows exactly k boys and every boy knows exactly k girls, for some $k \geq 1$, then it is possible to match the boys and girls in such a way that the boy and girl in each match know each other.

   Give a proof of the above theorem. *Hint*: Turn the problem into a transshipment problem by associating a unit supply or demand with each boy or girl node. Guess an initial feasible solution, which doesn't have to be a tree solution, to show that the problem is feasible. Invoke the Integrality Theorem.

3. An $n \times n$ matrix X is called a doubly stochastic matrix if all its elements are non-negative, and each row sums to 1, and each column sums to 1. Note that the matrix need not be symmetric. (If only the rows are required to sum to 1, the matrix is called stochastic.) An $n \times n$ matrix P is called a permutation matrix if the elements take values in $\{0, 1\}$, each row has exactly one 1, and each column has exactly one 1. Prove the following:

   Theorem: Let X be an $n \times n$ doubly stochastic matrix. Then there is a permutation matrix P such that $p_{ij} = 0$ whenever $x_{ij} = 0$.

   Hint: Construct a transshipment problem where a unit demand is associated with each row and a unit supply with each column. Show the problem is feasible and invoke the Integrality Theorem.

Remark: The theorem above is a key ingredient in proving a deep result in linear algebra called the Birkhoff-von Neumann Theorem, which states that every doubly stochatic matrix X can be written as a convex combination of permutation matrices $P_1, P_2, \dots, P_m$. In other words, there exist non-negative constants $\alpha_1, \alpha_2, \dots, \alpha_m$ summing to 1, such that $X = \sum_{i=1}^{m} \alpha_i P_i$. If you are interested in the proof, see Theorem 20.4 in V. Chvatal, "Linear Programming", W.H.Freeman and Co., New York, 1983.

4. A hidden Markov model consists of the following ingredients. There is a Markov chain X_n, $n = 0, 1, 2, \dots$ on a finite state space $\mathcal{X}$, with transition probabilities $p(i, j), i, j \in S$; what this means is that n denotes times, X_n denotes the state at time n, and if the Markov chain is in state i at time n, then at time $n+1$ it is in state j with probability $p(i, j)$, irrespective of its past history. Moreover, if the Markov chain undergoes a transition from state i at time n to state j at time $n + 1$, it outputs a symbol $Y_{n+1} = y$, from a finite output alphabet $\mathcal{Y}$, with probability $\pi(i, j; y)$. The outputs $Y_1, Y_2, Y_3, \dots$ are visible, while the states $X_0, X_1, X_2, \dots$ are hidden. The model parameters $p(\cdot, \cdot)$ and $\pi(\cdot, \cdot; \cdot)$ are assumed to be known.

   Suppose X_0 is known, say $X_0 = u \in \mathcal{X}$. The goal is to infer X_n having observed $Y_1, Y_2, \dots Y_{n-1}$. The maximum likelihood estimator (MLE) of X_n given $Y_1, Y_2, \dots Y_{n-1}$ is defined as any solution x of

$$\max_{x : u = X_0, X_1, \dots, X_n = x} \prod_{k=0}^{n-1} p(X_k, X_{k+1}) \pi(X_k, X_{k+1}; Y_{k+1}).$$

   Explain how to reduce this problem to a shortest path problem. How many vertices does the graph in the shortest path problem have, as a function of n, $|\mathcal{X}|$ and $|\mathcal{Y}|$?

   Remark: A special case of this problem arises in convolutional codes for error correction. The Viterbi algorithm for decoding convolutional codes also solves the estimation problem for hidden Markov models. Both these are examples of dynamic programming problems, which you will study.