Network Flows - Final, sketch of solutions

Average 80, median 88, high 97, low 30.

1). Prove: A graph $G = (V, E)$ has a vertex cover which is also an independent set if and only if G is a bipartite graph. (Definitions: $C \subset V$ is a vertex cover if for every edge $(i, j) \in E$ either $i \in C$ or $j \in C$ or both. $I \subset V$ is an independent set if for every edge $(i, j) \in E$ at most one of the nodes i, j is in I.)

Proof: If G is a bipartite graph, with partition X, Y, then we choose $C = X$. Clearly C is a vertex cover (since every edge has one endpoint in X) and C is also an independent set, as there is no edge between two nodes of X.

For the other direction, let C be a vertex cover which is also an independet set. Define a partition on the nodes of G given by $X = C$ and $Y = V \setminus C$. We claim that all edges have exactly one endpoint in X the other endpoint in Y. Since C is a vertex cover, for every $(i, j) \in E$, either $i \in C$ or $j \in C$. Could both i and j be in C? No, because C is also an independent set, and so either $i \in Y$ or $j \in Y$.

2). Consider a *min cost flow problem* on $G = (N, A)$ with capacities u_{ij} supply/demand $b(i)$ and costs c_{ij}:

$$\begin{array}{lllll} \min & z = \sum_{(i,j)\in A} c_{ij}x_{ij} & & \\ \text{s.t.} & \sum_j x_{ij} - \sum_j x_{ji} & = b(i) & \text{for all } i \in N \\ & 0 \le x_{ij} & \le u_{ij} & \text{for all } (i,j) \in A \end{array}$$

(a). Let (k, l) be an arc of minimum cost in the graph. Is it possible that no min cost flow for this has $x_{kl} > 0$? (One obvious such example is a graph for which all the $b(i)$ are equal to zero, so to make the problem non boring, you should assume that some of the $b(i)$ are *not* equal to zero.)

Yes, it is. Consider the graph on 4 nodes p, q, k, l With supply $b(p) = 2$ and $b(k) = 1$ and demands $b(l) = -1$ and $b(q) = -2$. Arc costs are $c_{pl} = c_{kq} = 1$, $c_{pq} = 5$ and $c_{kl} = 0$. All other arcs costs are infinity. The only feasible solution that does not have infinite cost, and is therefore the optimal solution is $x_{kq} = x_{pl} = x_{pq} = 1$, $x_{pq} = 0$.

(b). Let (p, q) be an arc of maximum cost in the graph.Is it possible that in every min cost flow on this graph $x_{pq} > 0$?

Yes, same example.

(c). Suppose some $u_{ij} = \infty$ and some $c_{ij} < 0$. Show that the optimal solution z is finite, $z > -\infty$ if and only if the arcs with capacities $u_{ij} = \infty$ do not contain a negative cost cycle.

Clearly, if the graph contains a negative cost cycle using arcs of infinite capacity, then $z = -\infty$, since we can find any feasible solution, and then send an inifinite amount of flow along the negative cycle.

For the other direction, suppose $z = -\infty$, let x^* be an unbounded solution. We know that any solution can be decomposed into flows on paths and cycles. Since $z = -\infty$ there must be a cycle along which the flow is infinite, and therefore its cost must be negative, or else one could decrease the flow, without increasing the cost.

3). In set S of nk ($k \neq 0$) balls each ball has one of n distinct colors and has one of n distinct diameters. Furthermore, each color is represented by exactly k balls and each diameter is represented by exactly k balls.

(a). We wish to select the largest set of balls such that every diameter and color is represented at most once. Show how to formulate the problem as a maximum matching problem.

Create a graph $G = (X \cup Y, E)$ where X consists of n vertices representing the different colors of the balls and Y consists of n vertices representing the different diameters of the balls. From each vertex in X, draw k edges, one corresponding to each element of the set of balls of that color, to the vertex in Y corresponding to the diameter of that ball.

(b). Show that in fact n such balls can be selected (i.e., n balls such that every diameter and color is represented exactly once). Hint: Use Hall's Theorem.

Each node has degree k. We follow a problem from exam 2006, and show that G, satisfies Hall's condition, namely that for every subset $X \subset N_1$ we have $|X| \leq |\Gamma(X)|$. To show this, consider the subgraph on nodes $X \cup \Gamma(X)$, and denote by F the set of edges in this subgraph. Since each node in X has degree k in G and also in this subgraph, we have that $|F| = k|X|$. Also, each node in $\Gamma(X)$ has degree k in G so it has degree $\leq k$ in this subgraph, so we have $|F| \leq k|\Gamma(X)|$. Thus $k|X| \leq k|\Gamma(X)|$. Since $k > 0$ we can divide by k and get Hall's condition, which implies that there is a matching in which all the nodes of N_1 are matched.

Finally, observes that since G is k regular, we have $k|N_1| = |E| = k|N_2|$ and so $|N_1| = |N_2|$. Thus if all the nodes of N_1 are matched, then so are all the nodes of N_2.

Each edge represents a ball, and the n edges in the perfect matching represent n balls corresponding between them to every color and diameter.

4). The Clustering TSP problem is: Given a graph $G = (V, E)$ with non negative costs on the edges satisfying the triangle inequality. The nodes are partitioned into "clusters" $V = V_1 \cup V_2 \cup ...V_k$, $V_i \cap V_j = \emptyset$ for $i \neq j$. The salesman wants to visit all nodes at minimum total cost, with the additional restriction that nodes of a cluster be traversed consecutively.

(a). Prove that the Clustering TSP problem is NP-hard. The problem is NP-hard since if the number of clusters is one, this is the usual TSP. In other words, we reduce TSP to cluster TSP by taking the same graph as given in the TSP instance, and setting $k = 1$.

(b). Describe an approximation algorithm for the clustering TSP. You should clearly describe an algorithm that runs in polynomial time, then prove that the tour you obtain has length at most some constant times the length of the optimal clustering TSP tour, for all instances of the problem.

1. Pick a representative point from each V_i arbitrarily. Call it r_i. Find an approximate tour on the nodes r_i. call this tour *conn*
2. Find an approximate TSP tour on each G_i which is the subgraph of G with nodes V_i. Call the edges used here apx_i. Do this by finding MST_i on each G_i and doubling it.
3. Combine *conn* and apx_i into *apx* a clustered tour on all the nodes: Walk around the tour *conn*, whenever you reach a new r_i make a detour by going around apx_i and then continue to the next vertex of *conn*.

To prove a bound note that $opt \geq \sum_i MST_i$. We also know that $apx_i \leq 2opt_i$. Next we know that $opt \geq optconn$ where *optconn* is the length of an optimal tour on the representative points. (This is because *opt* must visit all r_i's as well as the other vertices.) By Chrsitofides $1.5optconn \geq conn$. Finally,

$$apx \leq conn + \sum apx_i \leq 1.5optconn + 2 \sum MST_i \leq 1.5opt + 2opt = 3.5opt$$

So, $apx/opt \leq 3.5$.

Comment: several people claimed a better bound (such as 2) but I don't know how to make that work.

5). An *orientation* of an undirected graph $G = (V, E)$ is a directed graph $D = (V, A)$, such that A contains exactly one of (u, v) and (v, u) for every edge $\{u, v\} \in E$. Prove that $\chi(G) \leq p+1$ if and only if there exists an orientation D of G in which the longest directed path has length at most p arcs.

For the first direction: Consider an optimal vertex colouring with $\chi(G)$ colours. Call the colours 1,2,..., $\chi(G)$. Orient the edges of G such that each edge goes from the smaller number colour to the larger number colour. Since there are no edges with the same colour at both endpoints, we get that every directed path can have at most $\chi(G) - 1$ arcs, and so $\chi(G) \leq p + 1$ for this orientation.

For the second direction, consider an orientation G' of G with longest path of length at most p. Consider a maximal acyclic directed subgraph $G'' = (V'', A'')$ of G' (i.e., G'' does not contain any directed cycles, but no arc of G' can be added to G'' without introducing a directed cycle). Note that $V'' = V$ otherwise an arc can be added to or from a node in $V'' \setminus V$ without creating directed cycles. Colour each vertex v by a colour corresponding to the length of the longest directed path in G'' ending in v. Clearly, along any directed path, colours strictly increase, and so the number of colours used is at most $p + 1$. Now consider any arc $(u, v) \in G'$. There must be a directed path in G'' connecting u and v, since either $(u, v) \in G''$, or adding (u, v) to G'' creates a directed cycle. Since the colours of any two nodes along a directed path must be distinct, we get that the colours of u and v are distinct.

6). Given a directed graph $G = (N, A)$ with non negative lengths on the arcs c_{ij} and two specified nodes $s, t \in N$. A *vital arc* is an arc whose removal from the graph causes the length of the shortest path from s to t to increase. A *most vital arc* is a vital arc whose removal results in the largest increase in the shortest path length from s to t. For parts (a),(b),(c) prove or give a counterexample. Part (d) asks for an algorithm.

(a). A most vital arc is an arc with maximum value of c_{ij} on some shortest path from s to t. False: Consider the graph on 3 nodes s, a, t with $c_{sa} = 1$, $c_{at} = 1$, and $c_{st} = 2$. There are 2 shortest paths from s to t each of length 2. Removing the arc (s, t) of maximum c_{ij} does not increase the shortest path length.

(b). An arc that does not belong to any shortest path from s to t cannot be a most vital arc. True: If such an arc is removed, the shortest path remains intact, and therefor the removed arc is not vital.

(c). A graph might contain several most vital arcs. True: Consider the example in (a), except $c_{st} = 3$. Now both arcs (s, a) and (a, b) are most vital, as removing either one will increase the length of the shortest path from 2 to 3.

(d). Describe an algorithm for determining a most vital arc in a graph, or showing that none exists. What is the running time of your algorithm?

First, find the length of the shortest path length from s to t, call it d. Now, for each arc (i, j) in the graph, find the length of the shortest path in $G \setminus (i, j)$ from s to t, call is $d_{(i,j)}$. Let the vitality of arc (i, j), $V_{(i,j)} = d_{(i,j)} - d$. if all $V_{(i,j)} = 0$ the graph has no vital arc (such as in part (a).). Otherwise the most vital arcs are those with maximum $V_{(i,j)}$. The running time can be improved by only compution $d_{(i,j)}$ and $V_{(i,j)}$ for the arcs on the shortest path from s to t, of which there are at most $n - 1$ instead of m.