AMS/MBA 546 (Spring, 2008) Estie Arkin

Network Flows - Final, sketch of solutions

Average 85.

1). Consider a set of n numbers $a_1, a_2, \ldots, a_n$ arranged in non-decreasing order of their values ($a_i \leq a_{i+1}$). We wish to partition these numbers into "clusters" so that (1) each cluster contains at least p numbers; (2) each cluster contains consecutive numbers from the list $a_1, a_2, \ldots, a_n$; and (3) the sum of the squared deviation of the numbers from their cluster mean is as small as possible.

Let $\bar{a}(S) = (\sum_{i \in S} a_i)/|S|$ denote the mean of a set S of numbers forming a cluster. For a number a_k in the cluster, its squared deviation from the cluster mean is $(a_k - \bar{a}(S))^2$.

Describe how to formulate this problem as a shortest path problem. Make sure to clearly define the nodes and arcs of your graph, the costs on the arcs, and what are s and t.

Construct nodes $0, 1, 2, \ldots, n$ where $s = 0$ and $t = n$. We build arcs (i, j) if $j \geq i + p$. The cost of an arc (i, j) is the cost of a cluster $i+1, i+2, \ldots j$ which is $\sum_{k=i+1}^{j} (a_k - \bar{a}(S))^2$, where $\bar{a}(S) = (\sum_{k=i+1}^{j} a_k)/(j - i)$.

2). (a). Given a tree on $n > 2$ nodes, with exactly 2 nodes of degree equal to 1, show that all other nodes of this tree must have degree equal to 2.

Let d_i be the degree of node i. We have $1 + 1 + d_3 + d_4 + \cdots d_n = 2(n-1)$, since we know that the sum of the node degrees is 2 times the number of edges in any graph, and the number of edges in a tree with n nodes is $n - 1$. So $\sum_3^n d_i = 2n - 4$, and $d_i \geq 2$ for $i = 3, ..., n$. So $2(n-2) \geq 2n - 4$. Since we have equality, this implies that $d_i = 2$ for $i = 3, ..., n$.

(b). Let G be a graph on n nodes, such that both G and its complement are trees. What is n. (Make sure to justify your answer!)

If G and its complement are both trees then the number of edges in each is $n - 1$, and the number of edges in the two graphs combined is $\binom{n}{2}$ (as the two graphs together form a complete graph. $2(n-1) = n(n-1)/2$ implies that $n = 4$, or $n = 1$.

3). We are given a directed graph $G = (N, A)$ with arc costs c_{ij}, and we know that no negative cost cycles exist. Define f_{ij} to be the maximum amount we can decrease the cost of arc (i, j) without creating any negative cycles, assuming that all other arc costs remain the same. Describe an efficient algorithm for determining f_{ij} for all arcs $(i, j) \in A$.

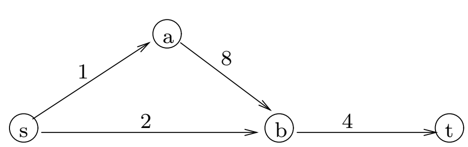
Using one of the algorithms from class, such as Floyd Warshall, find d_{ij} length of shortest path from i to j, in time $O(n^3)$. Now $f_{ij} = d_{ji} + c_{ij}$.

Note: An algorithm that starts at $f_{ij} = 0$ and then increases 1 by 1 until the first time a negative cycle is detected will work, but the running time is not polynomial.

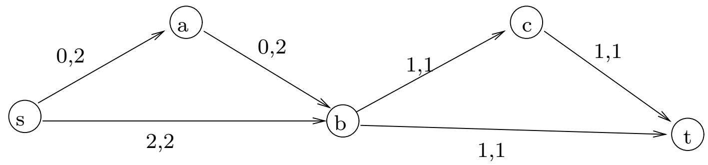
4). We are given a directed graph $G = (N, E)$, a source node s and sink node t with (nonnegative) capacities on the arcs u_{ij}. We consider the MAX FLOW problem on this graph. A *most vital arc* is defined as an arc whose deletion causes the largest decrease in the maximum flow value v. A *least vital arc* is defined as an arc whose deletion causes the smallest decrease in the maximum flow value v.

Prove or give a counterexample:

(a). A most vital arc is an arc with the maximum value of u_{ij}. False. In the example below (b, t) is most vital, but u_{ab} is largest.



(b). A most vital arc is an arc with the maximum value of x_{ij}. False. In the example below, the flow (first number next to each arc) on arc (s, b) is largest, but removing that arc will not change the max flow, so it is least vital, not most vital!



(c). Any arc with the $x_{ij} = 0$ in any maximum flow is a least vital arc. True, deleting this arc causes no change in the max flow, so it is least vital.

(d). Any arc that does belongs to some minimum cut cannot be a least vital arc. False. See example in part (a), where arc (s, a) is least vital (as is arc (a, b)), but arc (s, a) is in the only min cut, which contains arcs (s, a) and (s, b).

5). We are given a connected graph $G = (V, E)$, with n nodes and nonnegative costs on the edges. Define a 1-TREE to be a connected subset of n edges $E' \subset E$ that spans V. (In other words, a 1-TREE is a spanning tree of G plus one more edge.)

(a). Describe an efficient (polynomial time) algorithm to find a MINIMUM COST 1-TREE. Don't forget to **prove** that the output of your algorithm is indeed a MINIMUM COST 1-TREE! You may use any of the algorithms described in class, and you do not have to reprove their correctness.

The algorithm is simple, find a MST by any of the algorithms (Prim, Kruskal etc) and then add the min cost edge not in the tree already. Clearly, the algorithm produces a 1-tree. To show that this 1-tree is optimal, let T^* be the MST and (p, q) the additional edge added to it by our algorithm, and let $S^* = T^* \cup (p, q)$ be the 1-tree obtained by our algorithm. Let S be some other arbitrary 1-tree ($S^* \neq S$). S contains a cycle (since it is a 1-tree) and at least one of the edges of this cycle does not belong to T^*, since T^* is a tree. Let one such edge be (i, j), and define $T = S \setminus (i, j)$. By definition, T is a tree. If $(p, q) = (i, j)$ then we are done, since $c(T^*) \leq c(T)$ (T^* was the MST) and so $c(S^*) = c(T^*) + c(p, q) \leq c(T) + c(p, q) = c(T) + c(i, j) = c(S)$. The remaining case is when $(p, q) \neq (i, j)$, by the choice of (p, q) we have that $c(p, q) \leq c(i, j)$ and so $(c(S^*) = c(T^*) + c(p, q) \leq c(T) + c(i, j) = c(S)$. Thus, T^* has smallest cost among all 1-trees.

Note: It is not obvious that an MST plus the smallest weight edge yields an optimal 1-tree, this has to be proven. Specifically, you should show that using some other tree (perhaps worse than MST) and then some edge cannot yield a better solution.

(b). Prove that the cost of a MINIMUM COST 1-TREE is less than or equal to the cost of any TRAVELING SALESMAN TOUR on G.

A TSP tour is a special case of a 1-tree, since it contains n edges and spans all the nodes. Therefore the optimal 1-tree has cost that is less than or equal a TSP tour.

6). The RURAL POSTMAN PROBLEM is: Given a graph $G = (V, E)$ with non negative costs on the edges, and a subset of the edges $E' \subset E$. The postman must to find a closed walk including all edges of E' at minimum total cost.

Describe an approximation algorithm for the RURAL POSTMAN. You should clearly describe an algorithm that runs in polynomial time, then prove that the closed walk you obtain has length at most some constant times the length of the optimal closed walk, for all instances of the problem.

Build a new graph with connected components of E' as nodes, and edges between them whose length is given by the shortest path length between the components. This can be found say by running any all pairs shortest path algorithm, and then checking for every pair of nodes, one in each component, what is the shortest path. Note that edges in this new graph represent paths in the original graph.

In this new graph, find an MST, and let F be the set of edges of G obtained by "expaning" the MST edges back into paths. Now double $F \cup E'$ to get an Eulerian graph, and use that closed walk. This gives an approximation factor 2, since one can argue that the cost of F plus the cost of E' is at most the cost of an optimal rural postman. Using the same idea as in Christofides, we can get a 3/2 approximation algorithm.