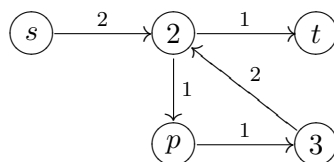


Network Optimization (Fall 2008) – Solutions to Midterm Exam

- There are four directed cycles. $W_1 = 1-3-2-1$, $W_2 = 1-3-4-2-1$, $W_3 = 3-4-5-3$, and $W_4 = 4-5-6$. The network can be made acyclic by deleting arcs $(1, 3)$ and $(4, 5)$, and $\text{ord} = [6\ 5\ 4\ 3\ 1\ 2]$ is a corresponding topological order.
- $\mathcal{P} = \{P_1 = 6-4-3-1, \Delta(P_1) = 3\}$ and $\mathcal{W} = \{W_1 = 6-4-5-6, \Delta(W_1) = 4; W_2 = 4-3-1-2-4, \Delta(W_2) = 2; W_3 = 2-4-3-2, \Delta(W_3) = 1; W_4 = 2-4-5-3-2, \Delta(W_4) = 1\}$. The decomposition is not unique. For instance, $P_2 = 6-4-5-3-1$ is another path from a supply node to a demand node.
- $d = [3\ 0\ 0\ -5\ -2\ -3]$. The shortest path tree is the path 3-1-2-5-6-4.
- Replace the original arc (i, j) with two parallel arcs. The first arc has flow bounds $[l_{ij}, l'_{ij}]$ and cost c_{ij}^1 . The second arc has flow bounds $[0, u_{ij} - l'_{ij}]$ and cost c_{ij}^2 . We need $c_{ij}^1 \leq c_{ij}^2$ for this transformation to work. Under this condition, the first arc is saturated before any flow goes through the second arc. If parallel arcs are not desirable, we can add a new intermediate node to split one of the two arcs.
- Find the shortest path from node s to node p , and then the shortest path from node p to node t . The shortest s - t walk is the concatenation of the two shortest paths. The complexity of this method is the same as that of the shortest path algorithm used – $O(n^2)$ if Dijkstra's algorithm is implemented. This walk need not always be a path, as illustrated below.



- $\ell_S(i) \leq \ell_{\text{sp}}(i) \quad \forall i \in N$. Note that $\ell_S = \ell_{\text{bfs}}$. The number of arcs in the shortest path from s to i can never be smaller than the number of arcs in the BFS tree.
 - If we strictly follow the definition of a cycle as given in the book (AMO), then we cannot say anything decisive about d_{\min} , and the best choice will be iv. According to the definition given in the book, we can form a cycle with just *one* arc (i, j) , by taking the arc and its reversal together. If we avoid this trivial case, G contains a cycle if $d_{\min} \geq 2$. The result for the existence of a *cycle* (not necessarily directed) in a directed graph is the same as that for the existence of a cycle in an undirected graph. With $d_{\min} \geq 2$, we can enter and leave every node at least once, and hence a cycle must exist. Having just two nodes with degrees 2 or more does not guarantee the existence of a cycle, unless these two nodes are connected to each other by more than one arc.

7. (a) FALSE. $n^{\log \log n} = \Omega(n \log n)$, but the O result does not hold. Taking logarithms of the functions, we get

$$\log \log n \log n \geq \log n + \log \log n.$$

- (b) **FALSE**. We need to find *all* the s - t shortest paths. A most vital arc must be part of one such path. For each s - t shortest path, we remove each arc and recalculate the s - t shortest path. The arc whose removal causes the largest increase in the s - t shortest path distance is a most vital arc. **In the worst case, all arcs can be part of some s - t shortest path, and hence we will have to solve m shortest path problems.** We can use the FIFO label correcting algorithm to solve the original shortest path instance, followed by Dial's implementation for each of the remaining shortest path runs, using reduced costs based on the optimal distance labels from the first run. Since we are only removing an arc in each subsequent run, the shortest path distance(s) can only increase (or stay unchanged), hence justifying the use of the reduced costs. The overall complexity is $O(mn + (m)(m + nC)) = O(m^2 + mnC)$.

8. We can formulate this problem as a circulation problem. The node set is $N = \{1, \dots, t\} \cup \{s, t\}$. The parameters listed for each arc (i, j) in the network are $[c_{ij}, u_{ij}]$. All lower bounds are set to zero.

