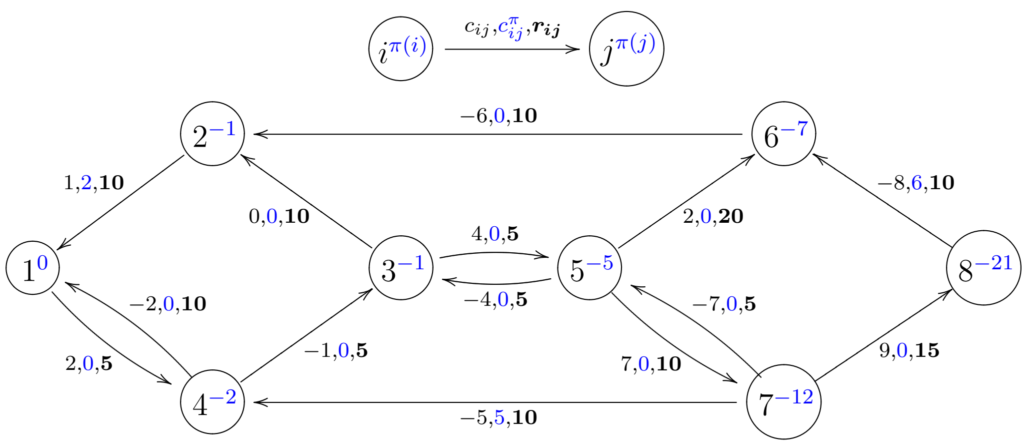
Network Optimization (Fall 2008) – Brief Solutions to Homework 11

1. (a) The cycles in $G(x^*)$ are W_1 = 1-4-3-2-1, W_2 = 3-5-7-4-3, W_3 = 1-4-3-5-6-2-1, and W_4 = 1-4-3-5-7-8-6-2-1. As demonstrated on the residual network below, all of these cycles have non-negative total costs: $c(W_1) = 2, c(W_2) = 5, c(W_3) = 2$, and $c(W_4) = 8$.

   (b) Using the c_{ij} values given, we can find the shortest path distances from node 1 to all other nodes in $G(x^*)$. These distance labels are d = [0, 1, 1, 2, 5, 7, 12, 21] (given for nodes 1-8 in that order). Taking $\pi = -d$ as the node potentials, we can verify that the reduced cost optimality conditions are satisfied ($c^\pi_{ij} \geq 0$).



   (c) The given set of node potentials π along with the optimal flow x^* do satisfy the complementary slackness conditions. Note that the complementary slackness conditions are given for the original network G, and not for the residual network $G(x^*)$. For the original network, we get $c^\pi_{21} = 2$, $c^\pi_{45} = -5$, $c^\pi_{68} = -6$, and all other $c^\pi_{ij} = 0$. Indeed, $x^*_{21} = 0$, $x^*_{47} = 10 = u_{47}$, and $x^*_{68} = 10 = u_{68}$. Further, for arcs $(1, 4)$, $(3, 5)$, and $(5, 7)$, we have $0 < x^*_{ij} < u_{ij}$, and hence the corresponding c^π_{ij} values must be zero, which they are.

   Notice that $c^\pi_{ij} = \mathbf{0}$ does *not* force any restriction on the x^*_{ij} value as part of the complementary slackness optimality conditions.

2. Let (x^*, π) be the optimal flow and node potential for the original network. Consider the same reduced costs c^π_{ij} for the new network. Notice that $u'_{ij} \geq u_{ij}\ \forall\ (i,j)\ \in A$. It is straightforward to check that the complementary slackness optimality conditions are satisfied by (x^*, π) on the new network as well.

3. Let $S, D \subset N$ be the sets of supply and demand nodes. The mathematical programming formulation for the minimum cost flow problem with surplus is given below.

$$\begin{array}{lll} \min & z = \sum_{(i,j)\in A} c_{ij}x_{ij} & \\ \text{s.t.} & \sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} & \left\{ \begin{array}{l} \leq b(i),\ i \in S; \\ = b(i),\ i \in D; \end{array} \right. \\ & 0 \leq x_{ij} \leq u_{ij}, & \forall (i,j) \in A. \end{array}$$

Let $b(i)$ be increased by 1 for some $i \in S$. Then the current optimal solution will also be optimal for the modified problem, with node i potentially holding one unit of excess flow. The flow balance constraints will all be satisfied. As such, the optimal value of z cannot increase. It can potentially decrease if all of the original supply available at node i was already used up (i.e., no surplus at node i), and it is cheaper to satisfy some demand from node i rather than another supply node. In such a case, a different optimal solution may give a smaller value for z.

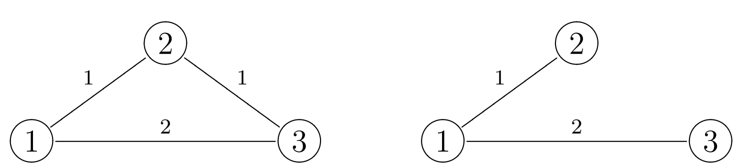
When increasing $b(i)$ to $b(i) + 1$ for some $i \in D$, assume that the value of z indeed decreases, and we get a new optimal solution x'. Since all demand of i is being met, we can use flow decomposition to find a path from some supply node $s \in S$ to i in the original network carrying at least 1 unit of flow. By sending one unit of flow back along this path, we can get a solution $x^{''}$, which would be feasible for the original problem (before changing $b(i)$). At the same time, since all $c_{ij} \geq 0$, it must be the case that $cx^{''} < cx$, where x was an optimal solution for the original problem. This result contradicts the optimality of x.

4. (a) The successive shortest path algorithm augments one unit of flow along 1-2-4-6, 1-3-4-6, 1-3-4-2-5-6, and 1-3-5-6, in that order.

   (b) The augmentations occur along the cycles 1-2-4-6-1, 1-3-4-6-1, 1-3-4-2-5-6-1, and 1-3-5-6-1, in that order. These cycles are the same as the paths identified in Part (a), with the arc 6-1 added to complete the cycle.

   (c) We can give the arguments for the case of the min-cost flow problem with a single supply node s and a single demand node t (of course, $b(s) = -b(t)$), with all costs being non-negative. We can apply network transformations to modify the generic min-cost flow problem to this setting. Let this network be named G. Obtain G' by adding arc (s,t) with a large cost, and $u_{st} = b(s)$. Start the successive shortest path algorithm on G with the initial feasible zero flow. Start the negative cycle canceling algorithm on G' with the initial feasible flow set as $x'_{st} = b(s)$, and $x'_{ij} = 0$ otherwise. If P is the shortest s-t path in G, then $P \cup \{(t,s)\}$ will be the most negative directed cycle in the $G'(x')$. Since the reduced costs remain non-negative after each augmentation, there will be no negative cycle in the original part (i.e., $G(x)$-part) of $G'(x')$. In other words, every negative cycle in $G'(x')$ will be of the form $P \cup \{(t,s)\}$, where P is an s-t path in $G(x)$. Hence, the most negative cycle in $G'(x')$ will indeed correspond to the shortest s-t path in $G(x)$.

5. Consider an MST T in which (p,q) is a non-tree arc. Take the unique path connecting p and q in T. By path optimality conditions, at least one arc in this path should have the same cost as (p,q), as c_{pq} is the smallest cost. If not, T will violate the path optimality condition, and hence will not be an MST. If there is another arc (i,j) in this path with $c_{ij} = c_{pq}$, then replacing (p,q) in T with (i,j) will give another MST T', which includes (p,q). Not every MST needs to include (p,q). In the graph show below, arc $(2,3)$ is not part of the MST shown.



6. (a) Apply Kruskal's algorithm with cost of red arcs set to 1 and cost of blue arcs set to 2.

   (b) Let T' and T'' be the spanning trees with k' and k'' red arcs, respectively, with $k'' > k'$. There should be *at least* $k'' - k'$ *blue* arcs in T' that are not there in T''. Consider one such blue arc (i, j), and the cut generated by deleting it from T'. There has to be at least one arc (k, l) in this cut that is in T'', but not in T'. This arc could be red or blue. If it is red, replacing (i, j) with (k, l) gives a spanning tree T with one more red arc than T'. If it is blue, then T will have the same number of red (and blue) arcs, but will have one less blue arc that is in T' and not in T''. We can repeat this process, and we are guaranteed to find a tree T with one more red arc than T'.

7. The result follows from cut optimality conditions for an MST. Let T be an MST, and let (i, j) be an arc with the largest c_{ij} in T. Consider the cut $[S, \bar{S}]$ obtained by deleting (i, j) from T. By the cut optimality conditions, $c_{ij} \leq c_{kl}$ for any arc $(k, l) \in [S, \bar{S}]$. This condition implies that T is also a bottleneck spanning tree. If not, we could replace (i, j) with (k, l) to get a tree with a lower bottleneck cost.

   The converse result need not hold. Both spanning trees shown are bottleneck spanning trees, but only the second tree is an MST.