Network Optimization (Fall 2008) – Brief Solutions to Homework 3

1. Add node 3-2 with $b(\text{3-2}) = u_{32} = 20$, arc $(\text{3-2}, 2)$ with $c_{\text{3-2},2} = c_{32} = -1$, arc $(\text{3-2}, 3)$ with $c_{\text{3-2},3} = 0$. Change $b(3)$ to $5 - u_{32} = -15$. Then add node 2-4 with $b(\text{2-4}) = u_{24} = 10$, arc $(\text{2-4}, 4)$ with $c_{\text{2-4},4} = c_{24} = -2$, and arc $(\text{2-4}, 2)$ with $c_{\text{2-4},2} = 0$. Change $b(2)$ to $-15 - u_{24} = -25$.

2. Reverse arc $(3, 2)$, set $c_{23} = -c_{32} = 1,\ u_{23} = u_{32} = 20$. Also, reverse arc $(2, 4)$, set $c_{42} = -c_{24} = 2,\ u_{42} = u_{24} = 10$. Change $b(3)$ to $b(3) - u_{32} = -15,\ b(2)$ to $b(2) + u_{32} - u_{24} = -5$, and $b(4)$ to $b(4) + u_{24} = 0$.

3. Delete arc (j, i), add a dummy transshipment node i-j, arc $(j, i\text{-}j)$ with cost c_{ji} and capacity u_{ji}, and arc $(i\text{-}j, i)$ with zero cost and infinite capacity.

4. Since $l_{ij} = u_{ij}$, we can fix $x_{ij} = l_{ij}$. Hence the arc (i, j) can be removed from further consideration (as the flow through it is fixed). Remove the arc (i, j), change $b(i)$ to $b(i) - l_{ij}$, and change $b(j)$ to $b(j) + l_{ij}$. Note that since the value of x_{ij} is fixed, its contribution to the objective function (of $c_{ij}x_{ij}$) is also fixed, and hence will not affect the optimal solution (and so, we can ignore c_{ij}).

5. The bounds can be implemented by splitting node i into three nodes with two arcs in between – add node i', arc (i', i), node i'', and arc (i, i''). Replace any arc (j, i) coming into node i by arc (j, i') with the same cost and capacity. Similarly, replace any arc (i, j) going out of node i by arc (i'', j) with the same cost and capacity. Set $l_{i'i} = \delta'$, $u_{i'i} = \gamma'$ and $l_{ii''} = \delta$, $u_{ii''} = \gamma$.

6. Let $N = N_1 \cup N_2 \cup N_3$ where

   - nodes in N_1 are supply nodes (i.e., $b(i_1) > 0\ \forall i_1 \in N_1$) for which the flow balance constraints are given as "$\leq$",
   - nodes in N_2 are demand nodes (i.e., $b(i_2) < 0\ \forall i_2 \in N_2$) for which the flow balance constraints are given as "$\geq$", and
   - nodes in N_3 have the flow balance constraints given as equations (these could be supply, demand, or transshipment nodes).

   We can convert the inequalities for the nodes in $N_1 \cup N_2$ to equations as follows.

   $$\sum_{(i_1,j)\in A} x_{i_1 j} - \sum_{(j,i_1)\in A} x_{j i_1} + s_{i_1} = b(i_1) \quad \forall i_1 \in N_1. \tag{1}$$

   $$\sum_{(i_2,j)\in A} x_{i_2 j} - \sum_{(j,i_2)\in A} x_{j i_2} - e_{i_2} = b(i_2) \quad \forall i_2 \in N_2. \tag{2}$$

   Here, s_{i_1} is the slack variable which models the supply from node i_1 which is *not used up*. Similarly, e_{i_2} is the excess variable which models the *unsatisfied* demand in node i_2. Both these sets of variables have to be non-negative. The flow balance equations for the nodes in N_3 have the standard form:

   $$\sum_{(i_3,j)\in A} x_{i_3 j} - \sum_{(j,i_3)\in A} x_{j i_3} = b(i_3) \quad \forall i_3 \in N_3. \tag{3}$$

Adding up the equations (1),(2), and (3) for *all* nodes, we get

$$\sum_{i_1 \in N_1} s_{i_1} - \sum_{i_2 \in N_2} e_{i_2} = \sum_{i_1 \in N_1} b(i_1) + \sum_{i_2 \in N_2} b(i_2) + \sum_{i_3 \in N_3} b(i_3),$$

which can be written equivalently as

$$\sum_{i_2 \in N_2} e_{i_2} - \sum_{i_1 \in N_1} s_{i_1} = -\sum_{i \in N} b(i). \tag{4}$$

The original problem can be cast as a standard min-cost flow problem after modifying the network as follows. Add a new node, say k. For each node $i_1 \in N_i$, add arc (i_1, k) with $l_{i_1 k} = 0$, $u_{i_1 k} = b(i_1)$, and $c_{i_1 k} = 0$. For each node $i_2 \in N_2$, add arc (k, i_2) with $l_{k i_2} = 0$, $u_{k i_2} = -b(i_2)$, and $c_{k i_2} = 0$. Set $b(k) = -\sum_{i \in N} b(i)$.

7. Note that the definition of $\Omega(\,)$ as given in class is slightly different from the one used in this problem. The definition given in class is (given in the book *Introduction to Algorithms* by Cormen et al.)

$$\Omega(g(n)) = \{f(n) \,:\, \exists\, c' > 0, n_0' \geq 0 \text{ with } f(n) \geq c' g(n) \;\forall\; n \geq n_0'\}. \tag{5}$$

The book gives the following definition:

$$\Omega(g(n)) = \{f(n) \,:\, \exists\, c' > 0, n_0' \geq 0 \text{ with } f(n) \geq c' g(n) \textbf{ for infinitely many } n \geq n_0'\}. \tag{6}$$

The conclusions could be different for some of the instances under the two different definitions. Further, the book uses the notation "$f(n)$ *is* $O(g(n))$", while, with the definition given in class, it is more natural to use the notation "$f(n) \in O(g(n))$", i.e., $f(n)$ belongs to (or, is an element of) the set $O(g(n))$.

(a) $f(n) \notin O(g(n))$. If we choose $c' = 1$ and $n_0' = 1$, there are infinitely many $n \geq n_0'$ such that $f(n) \geq c' g(n)$ (all even n). Hence $f(n) \in \Omega(g(n))$ according to definition (6). At the same time, $f(n) \notin \Omega(g(n))$ according to the definition (5).

(b) Since 2 is the only even prime number, $f(n) \leq g(n) \;\forall\; n \geq 3$. Hence, $f(n) \in O(g(n))$. Also, $f(n) \geq g(n)$ for all even n. Thus, $f(n) \in \Omega(g(n))$ according to definition (6), and hence $f(n) \in \Theta(g(n))$ as well. At the same time, $f(n) \ngeq g(n)$ for all odd primes n. Hence, $f(n) \notin \Omega(g(n))$ according to definition (5).

(c) $f(n) \in \Theta(g(n))$. Choose $c = 4$, $c' = 1$, and $n_0 = 2$.