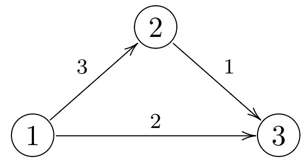
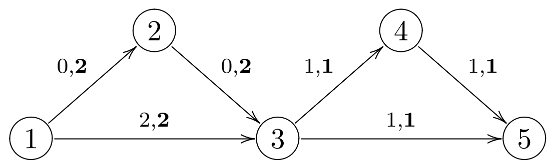
Network Optimization (Fall 2008) – Brief Solutions to Homework 10

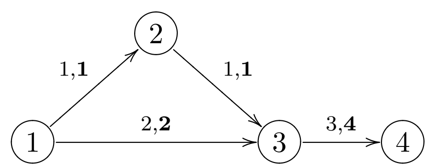
1. (a) FALSE. In the counterexample, numbers given are u_{ij}'s.



   (b) FALSE. Values shown are $x_{ij}, \boldsymbol{u_{ij}}$ pairs. Deleting $(1,3)$ with maximum arc flow does not change the max flow value.

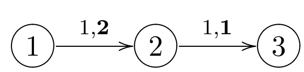


   (c) FALSE. $[\{1\}, \{2,3,4\}]$ is a min-cut here. Deleting arc $(1,3)$ decreases the max flow value only by 2, while deleting arc $(3,4)$ decreases it by 3. Thus, $(3,4)$ is the most vital arc, but it is not part of any min cut.

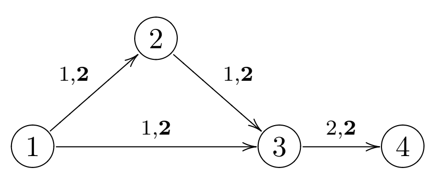


   (d) FALSE. In the network shown above (for part (c)), the most vital arc $(3,4)$ is not part of any min cut.

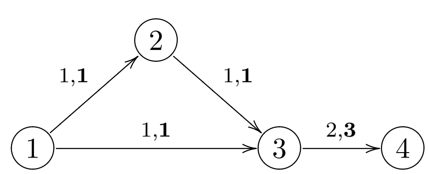
   (e) TRUE. Both arcs are most vital arcs in this network:



2. (a) FALSE. Values shown are $x_{ij}, \boldsymbol{u_{ij}}$ pairs.



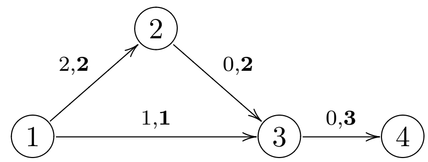
   (b) FALSE. Increasing all u_{ij}'s by 2 increases the max flow value by 3.



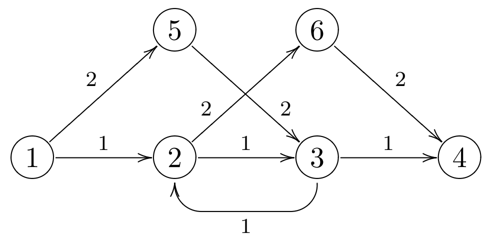
   (c) TRUE. The maximum increase in the flow into sink node t is $\sum_{i\,\text{active}} e(i)$. Since v^* is the max flow, we should have $v + \sum_{i\,\text{active}} e(i) \le v^*$.

(d) TRUE. The difference between a preflow x_p and an optimal flow x can be decomposed into at most $m+n$ flows along paths and cycles.

(e) FALSE. After pushing from node 2 to 3, $e_{\max}$ increases to 3.



(f) FALSE. After augmenting 1 unit along 1-2-3-4, we can augment 2 units along 1-5-3-2-6-4. Values shown are u_{ij}'s.



(g) FALSE. Consider a graph where the actual $d(s) = 3$. The distance labels given by $d(i) = 0 \ \forall i \neq s$ and $d(s) = 2$ is a lower bound on the actual distance labels, but is not valid.

3. The complexity of FIFO preflow push algorithm cannot be improved from $O(n^3)$. The excess scaling algorithm could potentially be sped up by changing the scaling factor to 2α (instead of 2) - so, set $\triangle = \triangle/2\alpha$ at the end of each scaling phase. Hence the number of scaling phases becomes $O(\log K)$, and the overall complexity will be $O(mn + n^2 \log K)$, instead of $O(mn + n^2 \log \alpha K)$. Notice that when $\alpha = 1$, we get the default setting.

   Equivalently, we can approach this problem using the idea of scaling all capacities to start with. We had seen earlier (AMO problem 6.34 (e), in Hw9) that a minimum cut remains minimal when all capacities are scaled by a constant $\lambda > 0$. Hence we could scale all arc capacities by α, and work with this scaled network instead of the original one. For this modified network, the upper bound on arc capacities becomes $U_\alpha = U/\alpha$, and using this value in the complexity bound for the default case gives the improved complexity bound stated above. The complexity of the FIFO preflow-push algorithm does not depend on U.

4. k saturating pushes and $3k$ non-saturating pushes. These counts do not include the k saturating pushes that are part of the pre-processing. We can solve the problem much quicker by **not** labeling $d(s) = n = 2k+2$ to start with. $d(s) = 4$ (exact distance label for node s) will be sufficient.

5. Consider a cut $[S, \bar{S}]$ of the original graph G with capacity u, and having k arcs. The capacity of the same cut in the modified graph G' will be $u' = mu + k$. Note that $k \leq m$. Consider any two cuts of the original graph G, say $[S_1, \bar{S_1}]$ and $[S_2, \bar{S_2}]$, with capacities u_1, u_2, and number of arcs k_1, k_2, respectively. Also, let the capacities of the same cuts in G' be u_1' and u_2'. If $u_2 > u_1$, then

$$u_2' = mu_2 + k_2 > mu_1 + k_1 = u_1', \quad \text{as } k_1 - k_2 \leq m - 1, \text{whatever } k_1 \text{and } k_2 \text{ are.}$$

   On the other hand, if $u_1 = u_2$ and $k_1 < k_2$, then we get $u_1' = mu_1 + k_1 < mu_1 + k_2 = mu_2 + k2 = u_2'$. Hence a min cut in G' will be a min cut in G with the smallest number of arcs.

6. Check the course web page for MATLAB functions.