COMP 360 - Fall 2015 - Midterm

1. Consider a flow network G. Prove or Disprove each one of the following statements.

   - (10 points) If e has the lowest capacity among all edges of G, then e is in a minimum cut.
   - (10 points) If the capacity of e is strictly greater than the capacity of every other edge of G, then e cannot be in a minimum cut.

   Solution: Both statements are FALSE. The flow network with five nodes s, v, u_1, u_2, t and capacities $c_{sv} = 3$ and $c_{vu_1} = c_{vu_2} = c_{u_1t} = c_{u_2t} = 2$ is a counterexample to both statements.

2. (10 points) State the complementary slackness theorem.

   Solution: If in the *optimal solution* a constraint in the primal linear program has slack then the corresponding dual variable is equal to zero in every optimal solution to the dual. Equivalently, if in the optimal solution, a dual varaible is not equal to zero, then there is no slack in the corresponding constraint in the primal.

3. (20 points) Write the dual of the following linear program without converting it to standard form:

$$\begin{array}{ll} \max & x_1 - 5x_2 \\ \text{s.t.} & x_1 - x_2 + x_3 \leq 8 \\ & x_1 + 2x_2 = 4 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

   Solution:

$$\begin{array}{ll} \min & 8y_1 + 4y_2 \\ \text{s.t.} & y_1 + y_2 \geq 1 \\ & -y_1 + 2y_2 \geq -5 \\ & y_1 \geq 0 \end{array}$$

4. (20 points) Suppose that $S_1, \ldots, S_m$ are subsets of $\{1, \ldots, n\}$. We want to assign non-negative weights $w_1, \ldots, w_m$ to the sets $S_1, \ldots, S_m$, respectively, such that for every element $j \in \{1, \ldots, n\}$ the total weight of the sets that contain j is *at least* 1. Formulate this problem as a linear program assuming that we want to minimize the total weight (i.e. the sum of all w_i's).

   Solution:

$$\begin{array}{lll} \min & \sum_{i=1}^{m} w_i & \\ \text{s.t.} & \sum_{i:j \in S_i} w_i \geq 1 & \forall j \in \{1, \ldots, n\} \\ & w_i \geq 0 & \forall i \in \{1, \ldots, m\} \end{array}$$

5. (20 points) Write the dual of your linear program for the previous problem. Explain what the dual program is solving.

Solution:

$$\begin{array}{lll} \max & \sum_{i=1}^{n} x_j & \\ \text{s.t.} & \sum_{j \in S_i} x_j \leq 1 & \forall i \in \{1, \ldots, m\} \\ & x_j \geq 0 & \forall i \in \{1, \ldots, n\} \end{array}$$

It assigns non-negative weights to the vertices and tries to maximize the total weight assuming that the total weight of the elements in each set $S_1, \ldots, S_m$ is at most 1.

6. As we have seen in the class, a flow network can have multiple minimum cuts. Here we want to find a *minimum cut* (A, B) that in addition to being a minimum cut, it also has the smallest number of edges crossing from A to B. Consider the following algorithm:

- Increase the capacity of every edge by 1 (simultaneously).
- Run the Ford-Fulkerson algorithm to find a minimum cut (A, B) in the new network.
- Output (A, B).

Note that if in the original network *capacity*$(A, B) = k$, after increasing the capacities of all the edges by 1, the capacity of (A, B) will increase to $k + r$ where r is the number of edges crossing from A to B. So between the two different minimum cuts (A, B) and (A', B') (in the original network), the one that has a smaller number of edges crossing the cut will have the smaller capacity in the new network.

(a) (10 points) Show that despite the above explanation, the algorithm is incorrect. In other words give an example to show that the algorithm can output a cut that is not a minimum cut with the smallest number of edges crossing the cut.

Solution: Consider the network with 6 nodes s, v, u_1, u_2, u_3, t and edges with capacities $c_{sv} = 7$ and $c_{vu_1} = c_{vu_2} = c_{vu_3} = c_{u_1t} = c_{u_2t} = c_{u_3t} = 2$. If we increase all the capacities by 1, in the new network the minimum cut will be $(\{s\}, \{v, u_1, u_2, u_3, t\})$. Note that this is not even a minimum-cut in the original network.

(b) (Bonus 20 points) Can you fix the algorithm?

Solution: Replace the capacity c_e of every edge e by $2mc_e + 1$ where m is the number of edges. Now if in the original network *capacity*$(A, B) = k$, after changing the capacities of the edges, the capacity of (A, B) will increase to $2mk + r$ where r is the number of edges crossing from A to B. Since $r \leq m < 2m$, a minimum cut in the new network has to be a minmum cut in the original network too (anything with capacity larger than k will become too large). Furthermore as in the suggested algorithm, between the two different minimum cuts (A, B) and (A', B') (in the original network), the one that has a smaller number of edges crossing the cut will have the smaller capacity in the new network.