CSC373H1 Summer 2014 Assignment 3

Names: John Armstrong, Henry Ku

SNs\CDF username: 993114492\g2jarmst, 998551348\g2kuhenr

Question #	Score
1	
2	
3	
4	
5	
6	
7	
8	
Total	

Acknowledgements:

"We declare that we have not used any outside help in completing this assignment."

Name: John Armstrong, Henry Ku Date: July 18, 2014

Q1. Construction:

To construct a family of graphs where G(V, E) has exponential number of minimum cuts between source and terminal, we being with n being the number of vertices that appear between the source and the terminal, we will also assume that the capacity on each edge is 1. If n = 0 then there are no vertices between the source and the terminal and thus V = {s, t} and E = $\emptyset$, Clearly, a single minimum-cut exists since we can only s $\in$ S and t $\in$ T, no edges exist to place a capacity on, so indeed $2^{|V|-2} = 2^0 = 1$.

For arbitrary n, n $\geqslant$ 1, and n = |V| - 2, then G(V, E) would be constructed such that V = {s, t, v_1, v_2, ..., v_n} and E = {(s, v_1), (v_1, t), (s, v_2), (v_2, t), ..., (s, v_n), (v_n, t)} (i.e. the only edges connected to a vertex are those that have s as their startpoint, or t as their endpoint). This arrangement is chosen because every cut in G(V, E) will be a minimum cut. By the arrangement of G(V, E), and since all capacities are the same. Whether a vertex v is or is not in the set S will not affect the summation of outgoing edge capacities, either the capacity from (v, t) or from (s, v), respectively, will be used in computing the minimum cut. Therefore we have a maximum number of minimum cuts in G.

Proof of exponential minimum cuts:

From the description above, in our construction of G(V, E), for any $v_i \in$ V the vertex can be contained within or outside the set that describes a cut in G and the cut is still a minimum cut. If we think of each minimum-cut as being described by a binary string for n = i, i $\geqslant$ 1, (q_{v_1}, q_{v_2}, ..., q_{v_n}), such that:

$$q_{v_i} = \begin{cases} 1 & \text{if } v_i \in \text{minimum cut set} \\ 0 & \text{if } v_i \notin \text{minimum cut set} \end{cases}$$

Base: Let n = 1, and let the B_1 be the set of binary strings describing the minimum cuts in G(V, E). Then B_1 contains two binary strings such that B_1 = {0, 1}. Indeed, $|B_1| = 2 = 2^n = 2^{|V|-2}$.

IH: Let n = i, i > 1, and let B_i be the set of binary strings describing the minimum cuts in G(V, E), such that V = {s, t, v_1, v_2, ..., v_i} and E = {(s, v_1), (v_1, t), (s, v_2), (v_2, t), ..., (s, v_i), (v_i, t)}. Suppose that $|B_i| = 2^i = 2^{|V|-2}$.

IS: Let n = i + 1, it follows from the construction description that V' = {s, t, v_1, v_2, ..., v_i, v_{i+1}} and E' = {(s, v_1), (v_1, t), (s, v_2), (v_2, t), ..., (s, v_i), (v_i, t), (s, v_{i+1}), (v_{i+1}, t)}. Clearly, the addition of v_{i+1} extends G(V, E) to G(V', E'). The minimum cuts in G(V', E') are composed from the minimum cuts in G(V, E) except that v_{i+1} is either inside or outside these previous cuts. Thus, to compute the set of minimum cuts as binary strings, B_{i+1}, simply take each binary string in B_i and append a 0 to be a minimum cut where v_{i+1} is outside the set, and then again take each binary string in B_i and append a 1 to be a minimum cut where v_{i+1} is inside the set. Thus, $|B_{i+1}| = 2 * (|B_i|) = 2 * 2^i = 2^{i+1}$.

By proof of induction this construction will have exponential minimum cuts between the source and the terminal. ■

Q2. Claim:

Function f, such that f(S) is the number of edges (u, v) with $u \in S$, $v \in V\backslash S$, is submodular.

Proof:

Using the fact provided in the question, all that is necessary to show that f is submodular is to show that for any two subsets $A, B \subseteq V$, $f(A) + f(B) \geqslant f(A \cup B) + f(A \cap B)$.

Consider the set of edges in G(V, E), $E = \{e_1, e_2, ..., e_n\}$. If we think of f(S) as an iterative algorithm that goes through each $e_i \in E$ and checks whether the startpoint of $e_i \in S$ and whether the endpoint of $e_i \in V\backslash S$. Then we have $f_{e_i}(S)$:

$$f_{e_i}(S) = \begin{cases} 1 & \text{if startpoint of } e_i \text{ in S and endpoint of } e_i \text{ in } V\backslash S \\ 0 & \text{otherwise} \end{cases}$$

And so, $f(S) = \sum_{i=1}^{n} f_{e_i}(S)$. Suppose (u, v) is an arbitrary edge in E, also suppose A, B are some arbitrary subsets of V. We wish to show in all cases, however vertices u and v appear or do not appear in A and/or B, that $f_{(u,v)}(A) + f_{(u,v)}(B) \geqslant f_{(u,v)}(A \cup B) + f_{(u,v)}(A \cap B)$:

<u>Case 1: $u \notin A$ and $u \notin B$:</u> Clearly, $u \notin (A \cup B)$ and $u \notin (A \cap B)$, thus, $f_{(u,v)}(A) = f_{(u,v)}(B) = f_{(u,v)}(A \cup B) = f_{(u,v)}(A \cap B) = 0$. So, $f_{(u,v)}(A) + f_{(u,v)}(B) \geqslant f_{(u,v)}(A \cup B) + f_{(u,v)}(A \cap B)$ holds.

<u>Case 2: $u \in A$ and $u \notin B$ (W.L.O.G. swtiching B for A in this case and all subcases the claim holds):</u>

<u>Case 2.1: $v \in A$:</u> Clearly, $u, v \in (A \cup B)$ and $u \notin (A \cap B)$, it follows that $f_{(u,v)}(A) = f_{(u,v)}(B) = f_{(u,v)}(A \cup B) = f_{(u,v)}(A \cap B) = 0$. So, $f_{(u,v)}(A) + f_{(u,v)}(B) \geqslant f_{(u,v)}(A \cup B) + f_{(u,v)}(A \cap B)$ holds.

<u>Case 2.2: $v \notin A$ and $v \in B$:</u> So, $f_{(u,v)}(A) = 1$ and $f_{(u,v)}(B) = 0$, but since $u, v \in (A \cup B)$ and $u \notin (A \cap B)$, then $f_{(u,v)}(A \cup B) = f_{(u,v)}(A \cap B) = 0$. So, $f_{(u,v)}(A) + f_{(u,v)}(B) \geqslant f_{(u,v)}(A \cup B) + f_{(u,v)}(A \cap B)$ holds.

<u>Case 2.3: $v \notin A$ and $v \notin B$:</u> Again, $f_{(u,v)}(A) = 1$ and $f_{(u,v)}(B) = 0$. Also, $u \in (A \cup B)$ and $v \notin (A \cup B)$, then $f_{(u,v)}(A \cup B) = 1$. Again, $u \notin (A \cap B)$, then $f_{(u,v)}(A \cap B) = 0$. So, $f_{(u,v)}(A) + f_{(u,v)}(B) \geqslant f_{(u,v)}(A \cup B) + f_{(u,v)}(A \cap B)$ holds.

<u>Case 3: $u \in A$ and $u \in B$ (W.L.O.G. swtiching B for A in this case and all subcases the claim holds):</u>

<u>Case 3.1: $v \in A$ and $v \in B$:</u> Since $u, v \in A$ and B, then $f_{(u,v)}(A) = f_{(u,v)}(B) = 0$. Also, $u, v \in (A \cup B)$ and $(A \cap B)$, then $f_{(u,v)}(A \cup B) = f_{(u,v)}(A \cap B) = 0$. So, $f_{(u,v)}(A) + f_{(u,v)}(B) \geqslant f_{(u,v)}(A \cup B) + f_{(u,v)}(A \cap B)$ holds.

<u>Case 3.2: $v \notin A$ and $v \in B$:</u> It follows that $f_{(u,v)}(A) = 1$ and $f_{(u,v)}(B) = 0$. Since $u, v \in (A \cup B)$ then $f_{(u,v)}(A \cup B) = 0$, but $u \in (A \cap B)$ and $v \notin (A \cap B)$, thus $f_{(u,v)}(A \cap B) = 1$. Again, $f_{(u,v)}(A) + f_{(u,v)}(B) \geqslant f_{(u,v)}(A \cup B) + f_{(u,v)}(A \cap B)$ holds.

<u>Case 3.3: $v \notin A$ and $v \notin B$:</u> Clearly, $f_{(u,v)}(A) = 1$ and $f_{(u,v)}(B) = 1$. Also, $u \in (A \cup B)$ and $(A \cap B)$, but $v \notin (A \cup B)$ and $(A \cap B)$, then $f_{(u,v)}(A \cup B) = f_{(u,v)}(A \cap B) = 1$. Then, $f_{(u,v)}(A) + f_{(u,v)}(B) \geqslant f_{(u,v)}(A \cup B) + f_{(u,v)}(A \cap B)$ holds.

In all cases in which the vertices of some edge $(u, v) \in E$ might appear in $A, B \subseteq V$ it has been shown that $f_{(u,v)}(A) + f_{(u,v)}(B) \geqslant f_{(u,v)}(A \cup B) + f_{(u,v)}(A \cap B)$ holds. It must follow then that $\sum_{i=1}^{n} f_{e_i}(A) + \sum_{i=1}^{n} f_{e_i}(B) \geqslant \sum_{i=1}^{n} f_{e_i}(A \cup B) + \sum_{i=1}^{n} f_{e_i}(A \cap B) \rightarrow f(A) + f(B) \geqslant f(A \cup B) + f(A \cap B)$. Clearly, function f is submodular by the equivalence of submodular functions. ■

Q3. Consider the restructuring of LP formulation of maximum flow into standard form:

maximize: $\vec{c} \cdot \vec{x}$

subject to: A$\vec{x} \leqslant \vec{b}$, $\vec{x} \geqslant 0$.

Suppose this LP operates on G(V, E) such that paths P = {p_1, ..., p_n} and E = {e_1, ..., e_m}.

Let $\vec{c}$ = [1, ..., 1], such that $|\vec{c}|$ = n.

Let $\vec{x}$ = [x_1, ..., x_n], such that $x_i = f_{p_i}$, maximum flow along path p_i.

Let A be and m x n matrix and let A be defined as follows:

$$A[i,j] = \begin{cases} 1 & \text{edge } e_i \text{ appears on path } p_j \\ 0 & \text{edge } e_i \text{ does not appears on path } p_j \end{cases}$$

Let $\vec{b}$ = [c(e_1), ..., c(e_m)], such that $|\vec{b}|$ = m.

Using the restructured LP, we construct the dual as follows:

minimize: $\vec{b} \cdot \vec{y}$

subject to: $A^t\vec{y} \geqslant \vec{c}$, $\vec{y} \geqslant 0$.

Let $\vec{y}$ = [y_1, ..., y_m], y_i is a value associated with e_i.

Again, we define $\vec{c}$ and $\vec{b}$ as the same in the original construction.

Now, A^t is an n x m matrix, and so A^t is defined as follows:

$$A^t[i,j] = \begin{cases} 1 & \text{in path } p_i \text{ the edge } e_j \text{ appears} \\ 0 & \text{in path } p_i \text{ the edge } e_j \text{ does not appear} \end{cases}$$

The conclusion for the dual LP is that the objective function and constraints represent a construction of a max-flow min-cut, such that when the dual LP is complete its solution will equal the max-flow calculated by the original LP. It can now be stated:

$$y_i = \begin{cases} 1 & \text{edge } e_i \text{ connects sets S and T*} \\ 0 & \text{edge } e_i \text{ does not connect sets S and T*} \end{cases}$$

*clearly S and T represent the set of vertices separated by the minimum cut of G(V, E).

Then the objective function represents the summation of the capacity of those edges where y_i = 1, and the summation will be the maximum flow in the network G(V, E). The n path constraints, $A^t\vec{y} \geqslant \vec{c}$, represent the fact that on each path at least one edge connects S to T. Thus, for path p_i, $\sum_{j=1}^{m} a_{ij}y_j \geqslant 1$.

Q4.

Q5.

Q6.

Q7.

Q8.