CSC 373 - Algorithm Design, Analysis, and Complexity **Summer 2014**

Assignment 3: Due Friday July 25, Midnight

Please follow the instructions provided on the course website to submit your assignment. You may submit the assignments in pairs. Also, if you use **any** sources (textbooks, online notes, friends) please cite them for your own safety.
You can use those data-structures and algorithms discussed in CSC263 (e.g. merge-sort, heaps, etc.) and in the lectures by stating their name. You do not need to provide any explanation or pseudo-code for their implementation. You can also use their running time without proving them: for example, if you are using the merge-sort in your algorithm you can simply state that merge-sorts running time is $\mathcal{O}(n \log n)$.

Every time you are asked to design an efficient algorithm, you should provide both a short high level explanation of how your algorithms works in plain English, and the pseudo-code of your algorithm similar to what we've seen in class. State the running time of your algorithm with a brief argument supporting your claim. You must prove that your algorithm finds an optimal solution!

1. Show how to construct a family of graphs where $G(V, E)$ has an exponential number (in $|V|$) of minimum cuts between the source and the terminal.

   Solution:

   For $n = 3$, let G be the path of length 2 between s and t. For $n > 3$, every new vertex with be connected to s and t only and no other vertex, thus creating an additional path of length 2 between s and t.

   For n in general, to separate s and t, we must cut on edge along every edge-disjoint path between them. There are $n - 2$ vertices between s and t and thus $n - 2$ paths between them, each of length 2. So we have $n - 2$ binary choices, thus 2^{n-2} different minimum cuts.

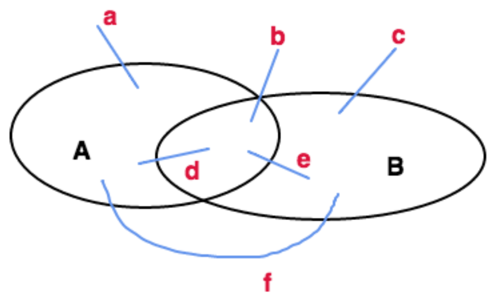
2. A function $f : 2^{|V|} \to R$ is **submodular** if and only if for any two subsets $A, B \subseteq V$, we have:

   $$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$$

   Let $G(V, E)$ be a graph, $S \subseteq V$, and let $f(S)$ be the number of edges (u, v) with $u \in S$, $v \in V \backslash S$. Show that f is submodular.

   Solution:

   Let f denote the cut function of G. For any $A \subseteq V$, $f(A) = \{\# \text{ of edges } (u, v) \text{ with } u \in A, v \in V \backslash A\}$. Now consider any two arbitrary subsets $A, B \subseteq V$, and examine the possible "type" of edges going through them. The figure below depicts all possible configurations:



That is the edge labelled **a** represents all edges $e(u,v)$ with $u \in A, v \in V\backslash(A \cup B)$, the edge **e** represents the edge (u,v) with $u \in A, v \in B$ etc..

Now it suffices to compute $f(A) + f(B)$ and compare it to $f(A \cup B) + f(A \cap B)$.

$$f(A) = a + b + e + f$$
$$f(B) = b + c + d + f$$
$$f(A \cup B) = a + b + c$$
$$f(A \cap B) = d + e + b$$

We therefore get:

$$f(A) + f(B) = a + 2b + c + d + e + 2f \quad (1)$$
$$f(A \cup B) + f(A \cap B) = a + 2b + c + d + e \quad (2)$$

It follows from (1) and (2) that $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ since all the edges has positive capacity ($2f \geq 0$).

3. Consider the following LP formulation of maximum flow:

$$\text{maximize} \sum f_P$$
$$\text{s.t.} \sum_{e(u,v) \in P} f_P \leq c(u,v) \qquad \forall (u,v) \in E$$
$$f_P \geq 0$$

In this formulation, we consider the paths that were used to send the flow. f_P denotes the amount of flow sent from s to t along path P. Give the dual of this LP and briefly explain your objective function and constraints.

Solution:

$$\text{minimize} \sum_{e(u,v) \in E} c(u,v) y_{u,v}$$
$$\text{s.t.} \sum_{e(u,v) \in P} y_{u,v} \geq 1 \qquad \forall P$$
$$y_{u,v} \geq 0 \qquad \forall e(u,v) \in E$$

We introduce a dual variable $y_{u,v}$ for every edge $e(u,v) \in E$, where $y_{u,v}$ is 1 if the edge (u,v) is in the cut, or 0 otherwise. Therefore the objective function minimizes the total sum of the capacities along the edges in the cut. The constraints ensure that we are indeed constructing a proper s, t cut; that is, along every path P, there is at least one edge separating s from t.

4. Let P be a polytope represented by a set of inequalities $Ax \leq b$. Give an LP that computes the center of the largest ball you can fit inside P.

   Solution:
   For any ball $B(x_c, r)$ of center x_c and radius r, B fits inside the polytope P if and only if x_c satisfies the $Ax_c \leq b$ constraints and is at distance at most r from the boundary of every hyperplane defined by every constraint $a_i x = b_i$. We know that the closest distance of a point to a plane is its orthogonal projection to the that plane; and we also know that for every halfspace described by a constraint $a_i x \leq b_i$, the vector a_i is orthogonal to the hyperplane defined by $a_i x = b_i$. Therefore it suffices to compute the normals of each a_i and verify that the radius in that direction doesn't violate any constraint. More formally, for every vector a_i, let $\tilde{a_i}$ be the normalized vector of a_i:

   $$\tilde{a_i} = \frac{a_i}{||a_i||}$$

   Given the center x_c, we want to be able to translate x_c by $r\tilde{a_i}$ for every $i \in [1...m]$ (where m is the number of constraints) and still be inside the feasible region of the polytope. In other words, we want to satisfy:

   $$a_i(x_c + r\tilde{a_i}) \leq b_i \text{ For all } i \in [1...m]$$

   Since we want the largest ball, we need maximize the radius, and thus our LP is:

   $$\begin{aligned} \text{maximize } & \sum r \\ \text{subj. to } & a_i(x_c + r\tilde{a_i}) \leq b_i \qquad \forall \text{i} \in [1...m] \\ & r \geq 0 \end{aligned}$$

5. Consider two polyhedra defined as $P_1 = \{x | A_1 x \leq b_1\}$ and $P_2 = \{x | A_2 x \leq b_2\}$. Show that if P_1 and P_2 do not intersect, then there exists $s, t \geq 0$ where $sA_1 + tA_2 = 0$ but $sb_1 + tb_2 < 0$.

   Solution:
   If P_1 and P_2 don't intersect then no linear program with the following constraints is feasible:

   $$\begin{aligned} \text{maximize } & c \cdot x \\ \text{subject to } & A_1 x \leq b_1 \\ & A_2 x \leq b_2 \\ & x \geq 0 \end{aligned}$$

   In particular, we chose $c = 0$ and the LP above would still not feasible. We rewrite the LP as follows:

   $$\begin{aligned} \text{maximize } & 0 \cdot x \\ \text{subject to } & Ax \leq b \\ & x \geq 0 \end{aligned}$$

   where $A = [A_1 \ A_2]^T, b = [b_1 \ b_2]^T$. Taking the asymmetric dual, we introduce a new variable y:

   $$\begin{aligned} \text{minimize } & y \cdot b = y \cdot [b_1 b_2]^T \\ \text{subject to } & yA = 0 = y \cdot [A_1 A_2]^T \\ & y \geq 0 \end{aligned}$$

Let $y = [s\ t]$, and rewrite the above LP as:

$$\begin{aligned} \text{minimize } & sb_1 + tb_2 \\ \text{subject to } & sA_1 + tA_2 = 0 \\ & s, t \geq 0 \end{aligned}$$

This dual is clearly feasible, $(0,0)$ is a feasible solution with objective value 0. However, since the primal is not feasible, the dual must be unbounded and thus there exists s, t such that $sA_1 + tA_2 = 0$ and $sb_1 + tb_2 < 0$.

6. Let $G(V, E, c)$ be an edge weighted graph with $c : E \to \mathbb{R}$. The **minimum mean cycle** problem asks for a cycle $\mathcal{C}(V', E')$ which minimizes the ratio:

$$\frac{\sum\limits_{e \in E'} c(e)}{|E'|}$$

Consider the following LP:

$$\begin{aligned} \text{minimize} \sum c(u,v) f(u,v) & \\ \text{s.t.} \sum_v f(u,v) - f(v,u) = 0 & \qquad (\forall u) \\ \sum f(u,v) = 1 & \\ f(u,v) \geq 0 & \end{aligned}$$

Show that this LP captures the minimum mean cycle problem. Give its dual, and show why your dual formulation also captures minimum mean cycles.

Solution:

The first constraint captures the flow conservation. Since we don't have a specified source or sink in G where the flow conservation is not preserved, we conclude that the value of f is 0 on any edge not in a cycle. Notice in particular that no vertex v not on cycle can have flow value greater than 0, as that would violate flow conservation on a different vertex.

Consider the cycle decomposition of G. For every cycle $\mathcal{C}$, the value of flow on every every edge of $\mathcal{C}$ must be identical. Why? Because if it isn't, we would violate the first constraint. Therefore the amount of flow on every cycle $\mathcal{C}$ is just $f_{\mathcal{C}} \cdot |\mathcal{C}|$ where $|\mathcal{C}|$ denotes the length (# of edges) of the cycle.

The second constraint tells us that the sum of the flow over all edges, and thus over the edges of cycles only, is one. In other words, if C is the set of cycles we obtained from the cycle decomposition of G, then $\sum\limits_{\mathcal{C} \in C} |\mathcal{C}| f_{\mathcal{C}} = 1$. However, since we're trying to minimize the objective function, we would just allocate the entire flow to the cycle with minimum mean. Let $\tilde{C}$ be such a cycle, then $\sum\limits_{\mathcal{C} \in C} |\mathcal{C}| f_{\mathcal{C}} = |\tilde{C}| f_{\tilde{C}} = 1$ which imples $f(u,v) = \frac{1}{|E_{\tilde{C}}|}$.

Summing over all the edges of $\tilde{C}$, we get precisely the minimum mean as defined above.

Let n and m denote the size of V and E respectively. To construct the dual, we first convert the primal into standard form.

$$\begin{aligned} \text{minimize } & c \cdot x \\ \text{s.t. } & Ax = b \\ & x \geq 0 \end{aligned}$$

Where:

(a) $c = [c_1, c_2, ..., c_m]$ and c_i is the capacity of edge i.

(b) $x = [f_1, ..., f_m]$ and f_i represents the f value of edge i.

(c) A is an $(n+1)$ by m matrix that captures the flow conservation of the network. In particular, we let $A[i,j] = 1$ if vertex i is an outvertex of the edge j, and $A[i,j] = -1$ if vertex i is invertex of the edge j. Finally $A[n+1,j] = 1$ for all $j \in [1...m]$ to capture the second constraint of the primal.

(d) $b = [0_1, 0_2, ..., 0_n, 1]$ to satisfy the flow conservation and capture the last constraint.

Given the primal, we introduce a new variable y, and construct the dual as follows:

$$\begin{aligned} \text{maximize } & y \cdot b \\ \text{s.t. } & yA \leq c \\ & y \geq 0 \end{aligned}$$

Given the dimension of b, we know that y is a vector of length $(n+1)$. Let y_u denote the entry of y for vertex u, and μ denote the $(n+1)^{th}$ entry of y.

By the construction of A, we know that a single constraint of the dual is of the form $y_y - y_v + \mu \leq c(u,v)$. Why? Every column in A has exactly three nonzero values. In particular, it has a 1 and a -1 that represent the invertex and outvertex of that edge, plus the last entry (1) used in the primal to capture the second constraint. Without loss of generality, suppose vertex u is the outvertex of the edge (u,v) and v the invertex; we thus get $y_u - y_v + \mu \leq c(u,v)$.

Now consider the objective function $\max y \cdot b$: Since $b = [0_1, 0_2, ..., 0_n, 1]$ and $y[n+1] = \mu$, it follows that the objective function is just $\max \mu$.

Putting everything together, we get the following the dual:

$$\begin{aligned} \text{maximize } & \mu \\ \text{s.t. } & y_u - y_v + \mu \leq c(u,v), \forall (u,v) \in E \\ & y \geq 0 \end{aligned}$$

We can rewrite the constraint of the LP above as follows: $(c(u,v) - \mu) + y_v - y_u \geq 0$. This is a constraint with respect to the edges of G. Notice therefore that if we sum over all the edge of a cycle $C(V', E')$ in G, the y_i values cancel out and we end up with:

$$\sum_{e(u,v) \in E'} (c(u,v) - \mu) \geq 0 \tag{3}$$

Consider the optimal value of the dual. If μ is optimal, then there must exist a cycle $\tilde{C}$ whose sum as described in (3) is 0. Since otherwise we can still increase μ by a small ϵ that still satisfies the constraint but contradict the maximality (and thus optimality) of μ. Let $\tilde{C}(C', E')$ be a cycle whose sum is 0 and let , then:

$$\begin{aligned} \sum_{e(u,v) \in E'} (c(u,v) - \mu) = 0 \implies & \sum_{e(u,v) \in E'} c(u,v) = |E'| \cdot \mu \\ \implies & \mu = \frac{\sum_{e(u,v) \in E'} c(u,v)}{|E'|} \end{aligned}$$

7. Alice is deciding how much organic milk and how much conventional milk to order each week. Conventional milk costs Alice \$1 per litre and she sells it at \$2 per litre; organic milk costs Alice \$1.50 per litre and she sells it at \$3 per litre. However, the milk company, CowCo, will only sell a litre of organic milk for each two litres or more of conventional milk that Alice buys. Furthermore, CowCo will not sell Alice more than 3,000 litres per week. Alice knows that she can sell however much milk she has. Formulate a linear program for deciding how much organic and how much conventional milk to buy, so as to maximize Alice's prot.

   Solution:
   Let x and y denote the number of litres of conventional and organic milk respectively. Alice makes twice as much per litre for every type, i.e. \$1 for conventional milk per litre and \$1.5 for organic milk. Since we want to maximize Alice's profit, the objective function is:

$$\text{maximize } x + 1.5y$$

   Since CowCo would not sell more than 3000 litres per week to Alice, we know that $x + y \leq 3000$. And because Alice can only buy one litre of organic milk for every 2 litres of conventional milk, then $y \leq \frac{x}{2} \implies 2y - x \leq 0$. Putting it all together we:

$$\begin{aligned} \text{maximize } & x + 1.5y \\ \text{subject to } & x + y \leq 3000 \\ & 2y - x \leq 0 \\ & x, y \geq 0 \end{aligned}$$

8. Let $\mathcal{M} = \{x_1, x_2, ..., x_n\}$ be a set of marbles. You have m friends, each with a set $S_i \subseteq \mathcal{M}$ of marbles. The marbles minister asked you whether it's possible to construct a set $X \subseteq \mathcal{M}$ of size $|X| \leq k$ such that $X \cap S_i \neq \emptyset$ for all $S_1, S_2, ..., S_m$. Show that this problem is NP-complete.

   Solution:
   Notice that this problem is precisely the HITTING SET problem: Given a family of sets $\{S_1, S_2, .., S_m\}$ and an integer k, the HITTING SET decision problem asks whether there exists a set X with k or fewer elements such that X intersects all sets in the family.

   To show that this problem is NP-complete, we first show that HITTING SET $\in NP$, then show that HITTING SET is NP-hard by giving a reduction from VERTEX COVER to HITTING SET.

   1. It is easy to see that HITTING SET is in NP, since checking the intersection of two sets takes at most $\mathcal{O}(n^2)$. The verifier takes $\mathcal{M}, \{S_1, ..., S_m\}, k$ and X and checks whether $|X| \leq k$ and whether $X \cap S_i \neq \emptyset$ for $i \in [1...m]$.

   2. To show that HITTING SET is NP-hard, we give a reduction from VERTEX COVER.

   Let $G(V, E), k'$ be an instance of VERTEX COVER. We construct an instance of HITTING SET as follows: For every edge $e(u, v) \in E$, we construct a set $S_e = \{u, v\}$. In total we have $|E|$ sets, and we set $k = k'$. We claim that G has a vertex cover of size at k' iff $\{S_1, S_2, ..., S_m\}$ has a hitting set X of size at most $k = k'$.

   Proof. ($\rightarrow$) Suppose G has a vertex cover VC of size at most k'. Then for any edge $e(u, v)$ either $u \in VC$ or $v \in VC$; in other words for any set $\{u, v\}$, $\{u, v\} \cap VC \neq \emptyset$ which is equivalent to saying for any set S_e, $S_e \cap VC \neq \emptyset$. Thus $X = VC$ is a hitting set of size $k = k'$.

($\leftarrow$) Conversely, Suppose X is a hitting of size at most $k = k'$ for the collection of sets $\{S_e | e \in E\}$. Then $\forall e \in E, S_e \cap X \neq \emptyset$, which again is equivalent to saying $\forall e \in E, e(u, v) \cap X \neq \emptyset$, i.e. every edge e is covered by an element if X. Thus X is a vertex cover for G of size size $k = k'$. $\square$