CO453: Network Design – Winter 2007

Instructor: Chaitanya Swamy

Solutions to Assignment 3

Throughout $G = (V, E)$ with $|V| = n$, $|E| = m$. A function $f : 2^V \mapsto \{0, 1\}$ is called *proper* if (i) $f(V) = 0$; (ii) for any $S \subseteq V$, $f(S) = f(V \setminus S)$; and (iii) if A, B are non-empty disjoint subsets of V, then $f(A \cup B) = 1$ implies that $f(A) = 1$ or $f(B) = 1$.

Q1: Consider the graph C_n, which denotes a cycle on n vertices. The terminal set T consists of all n nodes, so this is in fact a minimum spanning tree instance. All edge costs are 1. Setting $x_e = \frac{1}{2}$ yields a feasible solution to the LP relaxation (in fact this is the optimal solution). However every integer solution has cost $n - 1$, since a spanning tree must have $n - 1$ edges. So the ratio of costs is $2\big(1 - \frac{1}{n}\big) = 2\big(1 - \frac{1}{|T|}\big)$.

Q2: Clearly, $|T|$ must be even for the problem to have a feasible solution (if we had a feasible solution $F \subseteq E$ when $|T|$ is odd, then $\sum_v \delta_F(v) = 2|F|$ would be odd). By the same reasoning, observe that if F is a feasible T-join, then for any set S containing an odd number of nodes of T (called terminals), F must contain an edge from $\delta(S)$ (otherwise $\sum_{v \in S} \delta_F(v) = 2|\{(u, v) \in F : u, v \in S\}|$ would be odd). Therefore, we model the T-join problem by the 0-1 function $f(S) = 1$ if $|S \cap T|$ is odd and 0 otherwise. We have shown that any T-join yields a feasible solution to the network connectivity problem with this 0-1 function; thus, the value of the LP-optimum is a lower bound on the cost of the optimal T-join. But to apply the 2-approximation algorithm for 0-1 proper functions done in class, we need to show two other things: (a) f is proper; and (b) a solution returned by the primal-dual algorithm for the problem with the 0-1 proper function f is indeed a feasible T-join.

We first show that f is proper. Property (i) clearly holds. If $f(S) = 1$, then $V \setminus S$ also contains an odd number of terminals (since $|T|$ is even), so $f(V \setminus S) = 1$. Finally, if $S = A \cup B$ with $A, B \neq \emptyset$, $A \cap B = \emptyset$ and $f(S) = 1$, then since $|(A \cup B) \cap T| = |A \cap T| + |B \cap T|$, one of these two terms must be odd; hence, $f(A) = 1$ or $f(B) = 1$. To show (b), we will prove that every *minimal solution* F to the network connectivity problem with function f (i.e., for every $e \in F$, $F \setminus \{e\}$ is infeasible for the network connectivity problem) is a feasible T-join. Observe that the solution returned by the primal-dual algorithm is minimal (because of the reverse-delete step), so this proves (b).

Consider any component C of F, so C contains an even number of terminals. Since F is minimal, it is acyclic, so the edges in C form a tree. Consider first a terminal $t \in C$. Rooting the tree at t, let $F_1, \ldots, F_k$ be the trees hanging off t, where k is the degree of t (in F), with edges $e_1, \ldots, e_k$ of F joining these trees to t respectively. If k is even, then there is some F_i that contains an even number of terminals. This follows since the total number of terminals in $F_1, \ldots, F_k$ is odd, so there must be an odd number of F_js containing an odd number of terminals. Deleting edge e_i yields a smaller feasible solution to the network connectivity problem (since every component of $F \setminus \{e_i\}$ has an even number of terminals, and therefore has $f(.)$-value 0). This contradicts the minimality of F. Now consider a non-terminal $v \in C$. Rooting the tree at v, let $G_1, \ldots, G_\ell$ be the trees hanging off v, where ℓ is the degree of v, which are connected to v using edges $g_1, \ldots, g_\ell$. If ℓ is odd, then by a similar parity argument, there must be some G_i with an even number of terminals; deleting edge g_i yields a smaller feasible solution, giving a contradiction. Thus, every terminal must have odd degree, and every non-terminal even degree in a minimal solution F.

Q3:

$$\text{(P)} \quad \min \sum_{e\in E} c_e x_e \qquad\qquad \text{(D)} \quad \max \sum_S f(S) y_S$$

$$\text{(1)} \quad \text{s.t.} \sum_{e\in\delta(S)} x_e \geq f(S) \quad \forall S \subseteq V, \qquad\qquad \text{(2)} \quad \text{s.t.} \sum_{S\subseteq V: e\in\delta(S)} y_S \leq c_e \quad \forall e \in E,$$

$$x_e \geq 0 \quad \forall e \in E. \qquad\qquad y_S \geq 0 \quad \forall S \subseteq V.$$

(a) Given a set $F \subseteq E$, suppose that there is a violated set S. Clearly S must be a union of components $C_1, \ldots, C_k$ of F. Since $f(S) = 1$, by the downwards-monotone property of f, we get that $f(C_i) = 1$ for every $i = 1, \ldots, k$, so each C_i is a minimal violated set. Thus, if a set is a minimal violated set, then it must be a component of F. This immediately also implies that if F is such that each of its components has $f(.)$-value 0, then F is feasible. We will use this in the proof of part (b).

(b) From the description of the primal-dual algorithm, it is clear that F', the set of edges added, is acyclic (since we always add an edge on the boundary of an MVS). Thus F is also acyclic, and hence the graph H obtained by contracting components of F'' is also acyclic. Every node $v \in H$ corresponds to some component S_v of F'', and $\delta_F(S_v) = \delta_H(v)$. Thus, since the MVSs are a subcollection of the components of F'' (by part (a)), we need to show that $\sum_{v\in H: f(S_v)=1} \delta_H(v) \leq 2|\{v \in H : f(S_v) = 1\}|$. We will prove this by showing that in every non-singleton component C of H, there is at most one leaf v with $f(S_v) = 0$ (singleton components contribute 0 to both sides of the above inequality). This yields the above inequality, since

$$\begin{aligned}
\sum_{v\in H: f(S_v)=1} \delta(v) &= \sum_{\text{components } C \text{ of } H} \Big(\sum_{v\in C} \delta(v) - \sum_{v\in C: f(S_v)=0} \delta(v)\Big) \\
&\leq \sum_{\text{components } C \text{ of } H} \Big(2|C| - 2 - \big(2|\{v \in C : f(S_v) = 0\}| - 1\big)\Big) \\
&\leq \sum_{\text{components } C \text{ of } H} \big(2|\{v \in C : f(S_v) = 1\}| - 1\big) \;\leq\; 2|\{v \in H : f(S_v) = 1\}|.
\end{aligned}$$

To prove the claim, consider a component C of H, and suppose u and v are two distinct leaves of C such that $f(S_u) = f(S_v) = 0$. Let e be the edge of F incident on v. Let $S = \bigcup_{w\in C} S_w$ be the subset of vertices of G corresponding to C. Since $S_u \subseteq S \setminus S_v$ and $f(S_u) = 0$, we must have $f(S \setminus S_v) = 0 = f(S_v)$. This implies that $F \setminus \{e\}$ is a feasible solution, since all components of $F \setminus \{e\}$ have $f(.)$-value 0. So reverse delete should have deleted e, and this yields a contradiction. Thus, there cannot be two such leaves u and v whose corresponding subsets in V have $f(.)$-value 0.

(c) Let us divide the execution of the algorithm into stages, each stage is a maximal time-interval where the collection of MVSs does not change. Suppose the algorithm runs for k stages, so at the end of stage k there are no violated sets. Let $\mathcal{V}_i$ denote the collection of MVSs in stage i, and Δ_i be the length of stage i. So in stage i, we raise the dual variables of all sets $S \in \mathcal{V}_i$ by Δ_i. Clearly, $y_S = \sum_{i: S\in\mathcal{V}_i} \Delta_i$, and for every edge $e \in F$, we have $c_e = \sum_{S: f(S)=1, e\in\delta(S)} y_S$ by the design of the algorithm. Putting everything together and using part (b), we get

$$\begin{aligned}
\sum_{e\in F} c_e &= \sum_{e\in F}\Big(\sum_{S:f(S)=1,\ e\in\delta(S)} y_S\Big) = \sum_{S:f(S)=1} y_S \delta_F(S) = \sum_{S:f(S)=1}\Big(\sum_{i:S\in\mathcal{V}_i}\Delta_i\Big)\delta_F(S) \\
&= \sum_{i=1}^{k}\Delta_i\Big(\sum_{S\in\mathcal{V}_i}\delta_F(S)\Big) \\
&\le 2\sum_{i=1}^{k}\Delta_i\Big(\sum_{S\in\mathcal{V}_i}1\Big) \qquad \text{(by part (b))} \\
&= 2\sum_{S:f(S)=1}\Big(\sum_{i:S\in\mathcal{V}_i}\Delta_i\Big) = 2\sum_{S:f(S)=1} y_S = 2\sum_{S} f(S)y_S \le 2\cdot OPT.
\end{aligned}$$