CSCI 2510 - Problem Set 2

Shay Mozes
shay@cs.brown.edu

Problem 4.2

Consider the greedy algorithm for the multiway cut problem that starts with $G_o = G$, and in the ℓ^{th} iteration computes the minimum s_i-s_j cut for all pairs $s_i, s_j \in S$ that are connected in $G_{\ell-1}$. It then obtains G_ℓ by removing the lightest of these cuts from $G_{\ell-1}$. The algorithm repeats until all pairs are disconnected. The output of the algorithm is the set of edges it removed.

We first discuss some properties of the algorithm and then use them to obtain the approximation guarantee.

Lemma 1. *The algorithm terminates after $k-1$ iterations.*

Proof. Let $n(G_\ell)$ denote the number of unique connected components in G_ℓ that contain at least one $s \in S$. Initially the graph is connected, so $n(G_0) = 1$. If s_i and s_j are in the same connected component, then removing any s_i-s_j cut strictly increases n. Since the cut removed is minimal, n increases by exactly 1. Hence after $k-1$ iterations, $n(G_{k-1}) = k$, so all the pairs (s_i, s_j) are disconnected and the algorithm terminates.

□

Consider the $k-1$ pairs $(s_{i_1}, s_{j_1}), (s_{i_2}, s_{j_2}), \ldots, (s_{i_{k-1}}, s_{j_{k-1}})$ that correspond to the cuts removed by the algorithm. For $0 \leq \ell < k$, Let H_ℓ be the graph on S with (undirected) edges $\{(s_{i_1}, s_{j_1}), \ldots, (s_{i_\ell}, s_{j_\ell})\}$. If terminals $u, v \in S$ are connected in H_ℓ, then u and v are not connected in G_ℓ. This implies that none of the H_ℓs contains a cycle, since the algorithm only considers pairs of connected terminals. In particular H_{k-1} is acyclic, so it is a forest (actually it is a tree, but we do not need the stronger property).

Now, back to the approximation guarantee. Consider the optimal solution OPT. For $1 \leq i \leq k$, let A_i be the cut in OPT that separates s_i from $S \setminus s_i$. Thus $A = \cup_i A_i$. Since each edge in A appears in exactly two of the cuts A_i, we have $2w(OPT) = \sum_i w(A_i)$. Let C_i be the minimal weight set of edges separating s_i from $S \setminus s_i$. Clearly, $w(C_i) \leq w(A_i)$. So we have that $2w(OPT) \geq \sum_i w(C_i)$. Assume, without loss of generality, that $k = argmax_i w(C_i)$, so that $2\left(1 - \frac{1}{k}\right) w(OPT) \geq \sum_{i=1}^{k-1} w(C_i)$.

Root the forest H_{k-1} so that s_k is a root in the forest. This induced directions pointing away from the root on the edges. Each edge $s_i \rightarrow s_j$ of the forest corresponds to the a minimal (s_i, s_j) cut of some G_ℓ removed by the algorithm. Denote this cut by γ_j. This notation is well defined since each vertex in the forest has indegree at most one. Note that γ_j is a minimal cut in some G_ℓ and that G_ℓ is a subgraph of the G. Therefore, $w(\gamma_j) \leq C_j$ since C_j is a cut in G that separates s_j from *all* other terminals of S. Thus, the weight of the multiway cut produced by the greedy algorithm is:

$$\begin{aligned}
\sum_{(s_i \to s_j) \in H_{k-1}} w(\gamma_j) &\leq \sum_{(s_i \to s_j) \in H_{k-1}} w(C_j) \\
&\leq \sum_{j=1}^{k-1} w(C_j) \\
&\leq 2\left(1 - \frac{1}{k}\right) w(OPT),
\end{aligned}$$

where the second inequality follows from the fact that H_{k-1} is a rooted forest, so the indegree of each vertex is at most 1, and the indegree of s_k, which is a root of this forest is 0. We have thus established the approximation guarantee of the algorithm.
Remark: the tight example for the algorithm discussed in Vazirani's book for the multiway cut problem (example 4.5) is also tight for this algorithm.