CMPUT 675: Approximation Algorithms **Fall 2014**

Lecture 16 (Oct 10): Multicut in Trees

Lecturer: Zachary Friggstad *Scribe: Nikos Fasarakis-Hilliard*

16.1 Relaxed Complementary Slackness

In previous lectures, the concept of *duality* was introduced for linear programs (LPs). A general form of a LP along with its dual is shown bellow:

$$\begin{aligned} \text{minimize:} \quad & \mathbf{c}^T\mathbf{x} && \textbf{(Primal)} \\ \text{subject to:} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b}, && (16.1) \\ & \mathbf{x} \geq \mathbf{0}. && (16.2) \end{aligned}$$

$$\begin{aligned} \text{maximize:} \quad & \mathbf{b}^T\mathbf{y} && \textbf{(Dual)} \\ \text{subject to:} \quad & \mathbf{A}^T\mathbf{y} \leq \mathbf{c}, && (16.3) \\ & \mathbf{y} \geq \mathbf{0}. && (16.4) \end{aligned}$$

We also saw a property of optimal primal & dual LP solutions called *complementary slackness*. It provides a connection between the optimal solutions of both the primal and the dual problem. Complementary slackness can also be utilized in the context of approximation algorithms.

Theorem 1 *(Relaxed Complementary Slackness): Suppose $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ are feasible primal and dual solutions and suppose α and β are values such that*

(1) $\bar{x}_j = 0$ or $\sum_i \mathbf{A}_{ij}\bar{y}_i \geq c_j/\alpha$, for each j,

(2) $\bar{y}_i = 0$ or $\sum_j \mathbf{A}_{ij}\bar{x}_j \geq \beta b_i$, for each i,

then $\mathbf{c}^T\bar{\mathbf{x}} \leq \alpha\beta \cdot OPT_{LP}$ and $\mathbf{b}^T\bar{\mathbf{y}} \geq OPT_{LP}/\alpha\beta$.

Proof.

$$\begin{aligned} \mathrm{OPT_{LP}} &\leq \mathbf{c}^T\bar{\mathbf{x}} = \sum_j c_j\bar{x}_j && \text{(Since } \bar{\mathbf{x}} \text{ is feasible)} \\ &\leq \sum_j \big(\alpha\sum_i \mathbf{A}_{ij}\bar{y}_i\big)\bar{x}_j && \text{(Condition (1) and } \bar{\mathbf{x}} \geq \mathbf{0}) \\ &= \alpha\sum_i \big(\sum_j \mathbf{A}_{ij}\bar{x}_j\big)\bar{y}_i \\ &\leq \alpha\sum_i \beta b_i\bar{y}_i && \text{(Condition (2) and } \bar{\mathbf{y}} \geq \mathbf{0}) \\ &= \alpha\beta \cdot \mathbf{b}^T\bar{\mathbf{y}} \\ &\leq \alpha\beta \cdot \mathrm{OPT_{LP}} && \text{(By weak duality and feasibility of } \mathbf{y}) \end{aligned}$$

■

16.2 Multicut in Trees

In the MUTLICUT problem *in trees*, we are given a tree $T = (V, E)$, edge costs $c_e \geq 0$ for each edge $e \in E$, and k pairs of vertices $(s_1, t_1),\ (s_2, t_2),\ \ldots (s_k, t_k)$. The goal is to find the cheapest $F \subseteq E$ such that all (s_i, t_i) pairs are disconnected in $(V, E - F)$ (i.e no $s_i - t_i$ path in $(V, E - F)$). We studied this problem in general graphs in lecture 13.

As in lecture 13, we introduce variable x_e which indicates whether or not edge $e \in E$ is cut in the solution. Let P_i denote the set of edges in the unique path between vertices s_i and t_i in T. The following is a valid LP relaxation:

$$\begin{aligned} \text{minimize: } & \sum_{e \in E} c_e x_e && \textbf{(MT-Primal)} \\ \text{subject to: } & \sum_{e \in P_i} x_e \geq 1,\ 1 \leq i \leq k, && (16.5) \\ & \mathbf{x} \geq \mathbf{0}, && (16.6) \end{aligned}$$

with corresponding dual

$$\begin{aligned} \text{maximize: } & \sum_{i=1}^{k} f_i && \textbf{(MT-Dual)} \\ \text{subject to: } & \sum_{i : e \in P_i} f_i \leq c_e,\ \text{for each edge } e \in E, && (16.7) \\ & \mathbf{f} \geq \mathbf{0}. && (16.8) \end{aligned}$$

Suppose T is rooted at an arbitrary vertex (Fig. 16.1). For each $1 \leq i \leq k$, let $v(i)$ be the deepest common ancestor of s_i, t_i. The following algorithm is a 2-approximation for the MUTLICUT problem on trees.

Algorithm 1 MUTLICUT on Trees

$F \leftarrow \emptyset$
$\bar{\mathbf{f}} \leftarrow \mathbf{0}$
STEP 1*: Initialization Phase*
for i in decreasing order of the depth $v(i)$ **do**
  if $F \cap P_i = \emptyset$ **then**
    raise f_i until some dual constraint goes tight.
    add all $e \in P_i$ whose dual constraint goes tight to F.
  end if
end for
STEP 2*: Pruning Phase*
for each $e \in F$ in *reverse* order of when it was added to F **do**
  if $F - \{e\}$ is feasible **then**
    $F \leftarrow F - \{e\}$.
  end if
end for
return F

Theorem 2 *cost(F)* $\leq 2OPT_{LP}$.

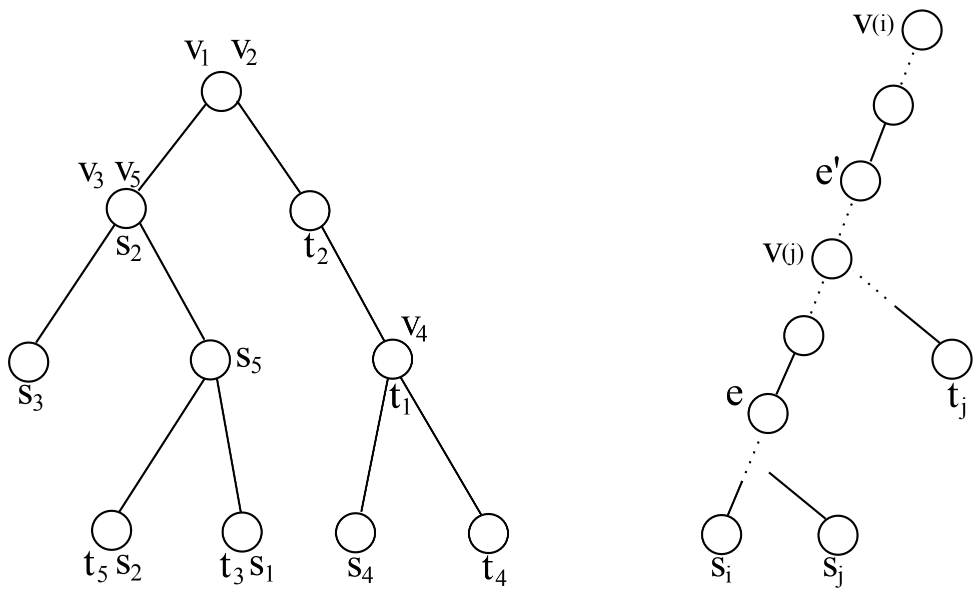

Figure 16.1: Left: Tree $T = (V, E)$ with 5 s_i, t_i pairs. Note that $s_i = t_j$ is allowed for some $i \neq j$. The deepest common ancestor of s_i, t_i, denoted as v_i, is also depicted. Right: Graphical illustration of proof of Theorem 2.

Proof. Let $\bar{\mathbf{x}}$ be the integer solution

$$\bar{x}_e = \begin{cases} 1, & e \in F, \\ 0, & e \notin F. \end{cases} \tag{16.9}$$

for the set F returned by the algorithm.

Note: Due to step 1 of the algorithm, $\bar{x}_i = 1$ if $\sum_{i:e\in P_i} \bar{f}_i = c_e$, so the relaxed complementary slackness condition (*1*) holds with $\alpha = 1$. All that is left to show is that

$$\bar{f}_i > 0 \Rightarrow |P_i \cap F| \leq 2, \tag{16.10}$$

i.e., $\sum_{e\in P_i} x_e \leq 2$. If so, the relaxed complementary slackness condition (*2*) holds with $\beta = 2$.

Claim 1 *For each i such that $f_i > 0$, there is at most one edge in F on the s_i-$v(i)$ path and and most one edge on the $v(i)$-t_i path.*

To see this, suppose the s_i-$v(i)$ path has two edges $e, e' \in F$ and that e lies bellow e' (see Fig. 16.1-right). Since $F - \{e\}$ is not feasible when it is considered in step 2 of the algorithm (or else we could have removed it), there is some $1 \leq j \leq k$ such that $|P_j \cap F| = \{e\}$.

Note: $v(j)$ is deeper than e' and e is deeper than $v(j)$ since $e \in P_j$ but $e' \notin P_j$.

Since $\bar{f}_i > 0$ and $v(j)$ is deeper that $v(i)$, e was not in F just after the iteration in step 1 of the algorithm that considered j. So, some other $\bar{e} \in P_j$ was in F after iteration j. Therefore, e is added to F *after* $\bar{e}$.

The pruning considered e before $\bar{e}$ due to reverse order processing. But this contradicts the fact that e was the only edge in $|P_j \cap F|$ at this time.

Therefore Claim 1 holds and so does (16.10), concluding the proof. ∎