CMPUT 675: Topics in Algorithms and Combinatorial Optimization (Fall 2009)

Lecture 18&19: Iterative relaxation

Lecturer: Mohammad R. Salavatipour **Scriber:** Zhong Li
Date: Nov 5, 2009

1 Spanning Tree Polytope

Last lecture we started studying the following LP for the problem of finding minimum spanning tree in a given graph $G = (V, E)$:

$$\begin{array}{rl} \min & \sum w_e x_e \\ \text{s.t.} & x(E(S)) \leq |S| - 1, \forall S \subset V \\ & x(E(V)) = |V| - 1 \\ & x_e \geq 0. \end{array}$$

We call this LP_{MST}. Our goal is to prove that:

Theorem 1.1 *LP_{MST} is integral, i.e. every bfs of this LP is integral.*

We know that any bfs is uniquely determined by n linearly independent tight constraints (where n is number of variables of the LP). Since we have exponentially many constraints in this LP, a bfs may be satisfying many of them with equality (i.e. being tight). We want a "good" set of linearly independent tight constraints defining it. The notion of "good" here will be clear soon. First, observe that in any bfs, we can safely delete any edge $e \in E$ with $x_e = 0$ from the graph. So we can assume that every edge of G has $x_e > 0$. Our goal is to show that there are at most $|v| - 1$ linearly independent tight constraints which implies that there are at most $|v| - 1$ non-zero variables. Since for any set S with size $|S| = 2$, the condition $x(E(S)) \leq |S| - 1$ implies the value of that edge must be at most 1 and because $x(E(v)) = |v| - 1$ we get that all the $|v| - 1$ non-zero variables must have value exactly 1, i.e. the bfs is integral.

1.1 Uncrossing Technique and Laminar Families of Tight Sets

Here we introduce a technique by which we can prove the existence of a special structure for bfs.

Definition 1.2 *Two sets X, Y over a ground set U are called crossing if $X \cap Y \neq \emptyset$, $X - Y \neq \emptyset$, and $Y - X \neq \emptyset$. A family of sets is called laminar if no two sets in the family cross.*

From this definition, it is easy to see that if two sets in a laminar family have non-empty intersection then one is a subset of the other. If we define a graph T associated with a laminar family $\mathcal{F}$ of sets, in which we have a node u_S in T for every sets $S \in \mathcal{F}$, u_S is called parent of $v_{S'}$ if S is the smallest set of $\mathcal{F}$ that contains S' then it is easy to see that T is a forest (doesn't have any cycles).

The next lemma shows that if every set of a laminar family has size at least 2 then there are at most $n - 1$ sets in the laminar family.

Lemma 1.3 *Let U be a ground set of size n and $\mathcal{L}$ be a laminar family of non-crossing sets without sets of size 1 (singletons). Then $|\mathcal{L}| \leq n - 1$.*

Proof: We can prove this by induction. We say $S \in \mathcal{L}$ is maximal if there is no set $S' \in \mathcal{L}$ with $S \subset S'$. Let $S_1, S_2, \ldots, S_m$ be maximal sets in $\mathcal{L}$. Since they don't cross, this gives a partitioning of U. As for the base case, if $|U| = 2$, since we don't have singletons, we have only one set S_i. For the induction step, using induction hypothesis, the number of sets of $\mathcal{L}$ contained in S_i is at most $|S_i| - 1$. Therefore, $|\mathcal{L}| \leq \sum_{i=1}^{m}(|S_i| - 1) = (\sum_{i=1}^{m} |S_i|) - m = n - m \leq n - 1$. ■

Uncrossing, is a powerful technique in combinatorial optimization. The following observation can be proved by noting that every type of edge (as in Figure ??) contributes the same amount to each side of the equation below:

Observation 1.4 *For any two sets of vertices $X, Y \subseteq V$. Then*

$$x(E(X)) + x(E(Y)) = x(E(X \cup Y)) + x(E(X \cap Y)) - x(\delta(X, Y))$$

where $\delta(X, Y)$ denotes the edges between X and Y.

An immediate corollary is:

Corollary 1.5 *For any two sets $X, Y \subseteq V$: $x(E(X)) + x(E(Y)) \leq x(E(X \cup Y)) + x(E(X \cap Y))$.*

Let x be a bfs of the LP_{MST} with $x_e > 0$ for all edges $e \in E$. Let $\mathcal{F} = \{S | x(E(S)) = |S| - 1\}$ be the family of all tight constraints of the LP. For each set S we use $\chi(E(S))$ to denote the characteristic vector of $E(S)$ of size $|E|$:

$$\chi(E(S)) = \begin{cases} 1 & if\ e \in E(S) \\ 0 & o.w. \end{cases}$$

Lemma 1.6 *If $S, T \in \mathcal{F}$ and $S \cap T \neq \emptyset$, then both $S \cap T$ and $S \cup T$ are in $\mathcal{F}$. Furthermore, $\chi(E(S)) + \chi(E(T)) = \chi(E(S \cup T)) + \chi(E(S \cap T))$.*

Proof:

$$\begin{aligned} |S| - 1 + |T| - 1 &= x(E(S)) + x(E(T)) && \text{since } S, T \in \mathcal{F} \\ &\leq x(E(S \cup T)) + x(E(S \cap T)) && \text{by Corollary 1.5} \\ &\leq |S \cap T| - 1 + |S \cup T| - 1 && \text{since these are constraints in the LP} \\ &= |S| - 1 + |T| - 1 \end{aligned}$$

So all inequalities must hold with equality, and in particular

$$x(E(S \cap T)) + x(E(S \cup T)) = |S \cap T| - 1 + |S \cup T| - 1.$$

Thus, we must have both $S \cap T$ and $S \cup T$ be tight constraints. Therefore, $S \cap T$ and $S \cup T$ are in $\mathcal{F}$. Also we must have $x(\delta(S, T)) = 0$ which implies $\chi(E(S)) + \chi(E(T)) = \chi(E(S \cap T)) + \chi(E(S \cup T))$. ■

We use $span(\mathcal{F})$ to denote the vector space of those sets $S \in F$ i.e. the vector space of $\{\chi(E(S)) : S \in \mathcal{F}\}$.

Lemma 1.7 *If $\mathcal{L}$ is a maximal laminar subfamily of $\mathcal{F}$, then $span(\mathcal{L}) = span(\mathcal{F})$.*

Proof: Suppose $\mathcal{L}$ is a maximal laminar family of $\mathcal{F}$, but $span(\mathcal{L}) \subset span(\mathcal{F})$. For each $S \notin \mathcal{L}$ define $intersect(S, \mathcal{L}) = |\{T \in \mathcal{L}|$ S and T intersect$\}|$. There must be a set $S \in \mathcal{F}$ with $\chi(E(S)) \notin span(\mathcal{L})$. Choose such a set S with smallest value of $intersect(S, \mathcal{L})$. Observe that $intersect(S, \mathcal{L}) \geq 1$ or else $\mathcal{L} \cup \{S\}$ is a larger laminar family. Let T be one of the sets in $\mathcal{L}$ that S intersects. We will prove the following Proposition shortly.

Proposition 1.8

$$intersection(S \cap T,\ \mathcal{L}) < intersection(S,\ \mathcal{L})$$

$$intersection(S \cup T,\ \mathcal{L}) < intersection(S,\ \mathcal{L})$$

For now assume this proposition is true. Applying Lemma 1.6 to S and T, we get both $S \cap T$ and $S \cup T$ are in $\mathcal{F}$. So using this proposition and by minimality of $intersect(S, \mathcal{L})$, both $S \cap T$ and $S \cup T$ are in $span(\mathcal{L})$. On the other hand, $\chi(E(S)) + \chi(E(T)) = \chi(E(S \cap T)) + \chi(E(S \cup T))$. Since $\chi(E(S \cap T))$ and $\chi(E(S \cup T))$ are in $span(\mathcal{L})$ and $T \in \mathcal{L}$, we must have $\chi(E(S)) \in span(\mathcal{L})$, a contradiction.

So it only remains to prove the above proposition.
For a set $R \in \mathcal{L}$ with $R \neq T$, R does not intersect T (since $\mathcal{L}$ is a laminar family). So whenever R intersects $S \cap T$ or $S \cup T$, R also intersects S. ■

Thus, we obtain the following:

Lemma 1.9 *Let x be a bfs of the LP_{MST} with $x_e > 0$ for all $e \in E$ and let $\mathcal{F} = \{S | x(E(S)) = |S| - 1\}$. Then there is a laminar family $\mathcal{L} \subseteq \mathcal{F}$ such that:*

1. *vectors of $\{\chi(E(S)) | S \in \mathcal{L}\}$ are linearly independent, and*
2. *$span(\mathcal{L}) = span(\mathcal{F})$*
3. *$|\mathcal{L}| = |E|$*

1.2 Iterative Algorithm

Here we describe an iterative algorithm to obtain a tree T from a bfs of the LP_{MST}; this is done by picking edges with value 1 in the LP iteratively:

Iterative MST algorithm

> $F \leftarrow \emptyset$
> **while** $V(G) \neq \emptyset$ **do**
>   Find a bfs x of LP_{MST} and remove any edge e with $x_e = 0$
>   Find a vertex v with degree 1, say $e = uv$; then $G \leftarrow G - \{v\}$ and $F \leftarrow F \cup \{e\}$

Lemma 1.10 *For any bfs x with $x_e > 0$ for all edges, there is a vertex v with $deg(v) = 1$.*

Proof: Suppose all nodes have $d(v) \geq 2$ in support E (i.e. edges that are left). Then $|E| = \frac{1}{2}\sum_{v \in G} deg(v) \geq |V|$. Since there is no edges with $x_e = 0$, each tight constraint is of the form $x(E(S)) = |S| - 1$. By Lemma 1.9, there is a laminar family $\mathcal{L}$ with $|\mathcal{L}| = |E| \geq |V|$; but by Lemma 1.3, $|\mathcal{L}| \leq |V| - 1$, this is a contradiction. ■

We can also modify the last step in the following way:

"Find an edge e with $x_e = 1$ and then $G \leftarrow G/e$ and $F \leftarrow F \cup \{e\}$"

where G/e means the graph obtained contracting edge e. Then we can prove the following lemma instead of Lemma 1.10:

Lemma 1.11 *For any bfs x with $x_e > 0$ for all edges, there is an edge e with $x_e = 1$.*

Proof: By Lemma 1.9, there are $|\mathcal{L}|$ linearly independent tight constraints of the form $x(E(S)) = |S| - 1$, and $|\mathcal{L}| = |E|$. We derive a contradiction by a counting argument. Assign one token for each edge e to the smallest set in $\mathcal{L}$ that contains both endpoints of e. So there are a total of $|E|$ tokens. We show we can collect 1 token for each set and still have some extra tokens, which is a clear contradiction.

Let $S \in \mathcal{L}$ be a set have children $R_1, \ldots, R_k$ (in the laminar family). Since these are all tight sets, we have:

$$\begin{aligned} x(E(S)) &= |S| - 1 \\ x(E(R_i)) &= |R_i| - 1 \quad \text{for all } 1 \leq i \leq k \end{aligned}$$

Subtracting the sides we get:

$$x(E(S)) - \sum_i x(E(R_i)) = |S| - \sum_i |R_i| + k - 1$$

Let $A = E(S) \setminus \bigcup_i E(R_i)$. Then $x(E(A)) = |S| - \sum_i |R_i| + k - 1$. Set S gets exactly one token for each edge in A. If $A = \emptyset$ then $\chi(E(S)) = \sum_i \chi(E(R_i))$ which contradicts linear independence of $\mathcal{L}$. Also we cannot have $|A| = 1$ since $x(E(A))$ is an integer (as the right-hand side is sum/difference of sizes of a number of sets) and each x_e is assumed to be fractional. Therefore, S gets at least two tokens! ■

Using Lemma 1.10 or 1.11 it is easy to show that the iterative algorithms find a MST.

Theorem 1.12 *The iterative algorithm (and its alternate form) find a MST.*

Proof: It only remains to show that the result is a spanning tree and this is done by induction on the number of iterations. Consider the first form of the algorithm. If we find a vertex of degree 1, say $deg(v) = 1$ then the edge incident to it must have $x_e = 1$ (since $x(\delta(v)) \geq 1$ is a constraint). Thus each edge added to F in either form of the algorithm has value 1. When e is added to F and v is removed from G note that for any spanning tree T' of $G - \{v\}$, we can build a spanning tree T of G by defining $T = T' \cup \{e\}$. So it is sufficient that we find a spanning tree in $G' = G - \{v\}$. Note that the restriction of x to $E(G')$, call it x_{res}, is a feasible solution to the LP for G'. So by induction, we find a tree F' for G' of cost at most optimum value of the LP for G'. Thus $c(F') \leq c \cdot x_r es$ and $C(F) = C(F') + c_e$. Thus

$$c(F) \leq c \cdot x_{res} + c_e = c \cdot x$$

since $x_e = 1$. ■

2 Min-cost Arborescence

We can use the same technique to show that the standard LP for the minimum cost r-arborescence problem is integral. Here we are given a digraph $D = (V, A)$ and with root r, a cost function $c : A \rightarrow R^{\geq 0}$. Our goal is to find a minimum cost r-arborescence. It is easy to see that the following is an LP relaxation:

$$
\begin{array}{ll}
\min & \sum c_a x_a \\
\text{s.t.} & x(\delta^{in}(S)) \geq 1 \quad \forall S \subset V - r \\
& x(\delta^{in}(v)) = 1 \quad \forall v \in V - r \\
& x(\delta^{in}(r)) = 0 \\
& x_a \geq 0.
\end{array}
$$

We leave it as an exercise to prove the following lemma:

Lemma 2.1 *Let x be a bfs of the above LP, and assume all edges a have $x_a > 0$. There is a laminar family $\mathcal{L}$ such that:*

1. *x is the unique solution to the linear system $\{x(\delta^{in}(S)) = 1 : S \in \mathcal{L}\}$*
2. *The vectors $\{\chi(\delta^{in}(S)) : S \in \mathcal{L}\}$ are linearly independent, and*
3. *$|\mathcal{L}| = |A|$*

Then an algorithm similar to the algorithm of MST in which in each rounds picks an edge with $x_a = 1$ finds a minimum cost r-arborescence (details are an exercise).

3 Minimum Cost Bounded Degree Spanning Tree

In this section we show how iterative algorithms can solve even more general problems (although approximately). Here we consider the problem of bounded degree spanning trees. Given a graph $G = (V, E)$ and a bound k, suppose we want to find a spanning tree with maximum degree at most k. This is NP-complete since with $k = 2$, it is the Hamiltonian path problem.

Theorem 3.1 (Furer & Raghavarchi '90) *There is a polynomial time algorithm that finds a spanning tree of maximum degree a most $k + 1$ (if there is one of with maximum degree at most k).*

As a more general case, suppose each edge of the graph has some given cost c_e. Also, each vertex v has a given bound B_v and our goal is to find a minimum cost spanning tree with degree bounded by B_v's.

Theorem 3.2 (Singh & Lan '07) *There is a polynomial time algorithm that finds a spanning tree of cost at most opt and degree in which every vertex v has degree at most $B_v + 1$.*

In this lecture and the next lecture, we prove this theorem. For that end, we first formulate the problem as an integer program and consider the LP relaxation. The following LP is the relaxation for an even more general form of the problem in which we have degree bounds B_v for a subset $W \subseteq V$ of vertices. We call the following LP, LP_{BDMST}.

$$
\begin{array}{rl}
 & min \sum c_e x_e \\
\forall S \subseteq V & X(E(S)) \leq |S| - 1 \\
 & X(E(V)) = |V| - 1 \\
v \in W & X(\delta(v)) \leq B_v \\
 & X_e \geq 0
\end{array}
$$

For ease of exposition, we first prove a weaker version of the above theorem. We show that the following algorithm finds a tree whose cost is at most optimum and degree of every vertex v is bounded by at most $B_v + 2$.

Additive +2 approximation for BDMST

> $F \leftarrow \emptyset$
> **while** $V(G) \neq \emptyset$ **do**
>   Find a bfs x of LP_{BDMST} and remove any edge e with $x_e = 0$
>   If there is a vertex $v \in V$ with at most one edge $e = uv$ incident to v then
> $F \leftarrow F \cup \{e\}$
> $G \leftarrow G - \{v\}$
> $W \leftarrow W - \{v\}$
> $B_u \leftarrow B_u - 1$
>   If there is a vertex $v \in W$ with $deg_E(v) \leq 3$ then
> $W \leftarrow W - \{v\}$
> Return F

So at each iteration, if there is an edge e that is the only edge incident to v, and we show that we must have $x_e \geq 1$ and we pick this edge. Thus the cost we pay for an edge is not more than what the optimum pays. We argue that if there is no such vertex v with an edge of value 1 then there is a vertex $v \in W$ with $d(v) \leq 3$; so at each iteration we make progress in one of the two steps. Note that if there is a vertex v with $d(v) \leq 3$ and we remove this constraint since $B_v \geq 1$ in the worst case we will have picked all the at most 3 edges incident with v in our final solution, so the degree bound will be at most 3 which is at most $B_v + 2$.

Let $\mathcal{F} = \{S \subseteq V : x(E(S)) = |S| - 1\}$ be the set of tight set constraints. Then the following lemma can be proved similar to lemma 1.9 by applying uncrossing to sets in $\mathcal{F}$:

Lemma 3.3 *Let x be a bfs of LP_{BDMSP} with $x_e > 0$ for all edges. There is a $T \subseteq W$ with $x(\delta(v)) = B_v$ for each $v \in T$, and a laminar family $\mathcal{L} \subseteq \mathcal{F}$ such that*

1. *vectors $\{\chi(E(S)) : S \in \mathcal{L}\} \cup \{\chi(\delta(v)) : v \in T\}$ are linearly independent*
2. *vector space of $span(\mathcal{L}) \cup \{\chi(\delta(v)) : v \in T\} = span(\mathcal{F})$*
3. *$|\mathcal{L}| + |T| = |E|$*

By the argument given earlier, it is thus sufficient to prove the following lemma:

Lemma 3.4 (Main lemma) *Let x be a bfs of LP_{BDMSP} with $x_e > 0$ for all edges. There is a vertex v with $degree(v) = 1$ or there is a vertex $v \in W$ with $degree(v) \leq 3$.*

Proof: By way of contradiction, suppose that none of these holds, i.e. each vertex $v \in V$ has $degree(v) \geq 2$ and each $v \in W$ has $degree(v) \geq 4$. Thus, with $|V| = n$:

$$|E| = \frac{1}{2} \sum_{v \in V} degree(v) \geq \frac{1}{2}(4|W| + 2(n - |W|)) = n + |W|. \tag{1}$$

By the previous lemma, there is a laminar family $\mathcal{L} \subseteq \mathcal{F}$ and set $T \subseteq W$ with $|\mathcal{L}| + |T| = |E|$. Since $\mathcal{L}$ has sets of size at least 2, $|\mathcal{L}| \leq n - 1$. Thus $|E| = |\mathcal{L}| + |T| \leq n - 1 + |T| \leq n - 1 + |W|$ which contradicts inequality (1). ■

Theorem 3.5 *The iterative algorithm given above returns a tree T with cost at most optimum and $d(v) \leq B_v + 2$ for each $v \in W$.*

Proof: If there is a node v with $degree(v) = 1$ then since $x(\delta(v)) \geq 1$ is a valid constraint (obtained by subtracting $x(E(V - v)) \leq |V| - 2$ from $x(E(V)) = |V| - 1$), we must have $x_e \geq 1$. So we pay no more than what the LP pays at each step we pick an edge. Also, the remaining variables define a feasible solution for the residual LP, so inductively, the cost of T is at most the cost of the LP solution. As for the degree bounds, let B'_v be the current residual degree bound for a vertex v. It is easy to see that since we always pick full edges and update the degree bounds, if $v \in W$ then $deg_F(v) + B'_v = B_v$. Now when v is removed from W (because it has $deg(v) \leq 3$) then $deg_T(v) \leq deg_F(v) + 3 \leq B_v - B'_v + 3 \leq B_v + 2$, since $B'_v \geq 1$. ■

3.1 Additive +1 approximation algorithm

In this Section we prove Theorem 3.2. We start from a bfs and show that at each iteration we can either find an edge e with $x_e = 1$ (and so pick it) or there is a vertex $v \in W$ with $deg(v) \leq B_v + 1$ and we relax the constraint. The following equivalent algorithm is easier to analyze. We start from a bfs x with $x_e > 0$, for all $e \in E$. We iteratively find a vertex $v \in W$ with $deg(v) \leq B_v + 1$ and remove v from W. At the end we have the LP without any degree constraints, so it is the same LP as for MST and is thus integral.

Additive +1 approximation for BDMST

> **while** $V(G) \neq \emptyset$ **do**
>
> > Find a bfs x of LP_{BDMST} and remove any edge e with $x_e = 0$
> >
> > Let $v \in W$ be a node with $deg(v) \leq B_v + 1$.
> >
> > $W \leftarrow W - \{v\}$
>
> Return all edges with $x_e = 1$.

It is easy to see that if at each iteration we find a vertex $v \in W$ with $deg(v) \leq B_v + 1$ then the degree of v at the final solution is no more than $B_v + 1$ once we remove that constraint from the LP. Also, x as it is, is feasible for the more relaxed LP. Therefore, the value of the solution for the residual (relaxed) LP is no more than opt. This implies that at the end we have a tree with cost at most optimum and degree bounds are violated by no more than +1. Thus we only have to show that at each iteration of the algorithm we can find such a vertex $v \in W$ to remove from W. Note that from Lemma 3.3 we can find the laminar family $\mathcal{L} \subseteq \mathcal{F}$ and tight degree nodes $T \subseteq W$ such that $|\mathcal{L}| + |T| = |E|$ and the corresponding constraints are linearly independent. We prove the following Lemma:

Lemma 3.6 *Let x be a bfs of the LP_{BDMST} with $x_e > 0$ for all edges and $\mathcal{L}$ and T be as in Lemma 3.3. Then if $T \neq \emptyset$ then there is a node $v \in W$ such that $degree(v) \leq B_v + 1$.*

We will see the proof of this lemma next lecture and that completes the proof of Theorem 3.2.