Intro to LO, Lecture 9

Martin Böhm
University of Wrocław, Winter 2023/2024

Spanning tree polytope

D(Spanning tree): Given a graph $G = (V, E)$ with n vertices, a subgraph T is a spanning tree if T is acyclic and $V(T) = V(G)$.

Clm: Given weights on the edges, the problem of finding a spanning tree of minimum total weight is solvable in polynomial time using discrete algorithms, such as the Kruskal or Jarník algorithms.

One equivalent characteristic of a spanning tree:

L: A subgraph is a spanning tree if and only if it is acyclic and it has $|V| - 1$ edges.

$$\begin{array}{rl} \min & \sum w_e x_e \\ \text{s.t.} \forall S \subseteq V, |S| \geq 2: & x(E(S)) \leq |S| - 1, \\ & x(E(V)) = |V| - 1 \\ & x_e \geq 0. \end{array}$$

Fact: This linear program does not have a totally unimodular matrix.

We have two problems with this linear program:

1. (Postponed until next week.) Is the linear program even solvable in polynomial time, since it has exponentially many constraints?
2. Do vertices of this linear program correspond to spanning trees?

Our goal for today: Given any basic feasible solution of the LP above, prove that this corresponds to an integer solution (a spanning tree).

O: Given any basic feasible solution, we can remove all edges e from the graph G where the value $x_e = 0$. The solution remains of the same value and remains feasible even for the smaller graph. The number of vertices remains the same.

(For the purposes of the proof, we could also consider removing all edges with $x_e = 1$, as they are integral, but we do not do this.)

Laminar families of sets

D(Crossing): Two sets X, Y over a ground set U are called crossing if $X \cap Y \neq \emptyset, X - Y \neq \emptyset$, and $Y - X \neq \emptyset$.

D(Laminarity): A family of sets is called laminar if no two sets in the family cross.

L: If $\mathcal{F}$ is a laminar family on n elements without singletons (sets of size 1). Then, $|\mathcal{F}| \leq n - 1$.

P: For the proof, we imagine we return singletons to the laminar family below their last occurence in the family.

A laminar family is essentially a tree decomposition. Any tree with at most n leaves has at most $n - 1$ internal vertices.

Vector space of edges in a graph

Given a graph with n vertices and m edges, we can sometimes think of it as a vector space with a long vector, namely of m coordinates. Then, we can encode any subset of edges as a vector in $\{0, 1\}^m$.

For the mapping from subset of edges to the vectors, we will use the *characteristic vector* of a subset of edges $F \subseteq E$:

$$\chi(F)_e = \begin{cases} 1 & \text{if } e \in F, \\ 0 & \text{o.w.} \end{cases}$$

In this lecture, we will also wish to define the edges associated with a subset of vertices, namely for a subset $S \subseteq V$:

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

O: For any two sets of vertices $X, Y \subseteq V$, we have:

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

and

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

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

O: For any two sets of vertices $X, Y \subseteq V$, we have:

$$\chi(E(X)) + \chi(E(Y)) \leq \chi(E(X \cup Y)) + \chi(E(X \cap Y))$$

and

$$x(E(X)) + x(E(Y)) \leq x(E(X \cup Y)) + x(E(X \cap Y))$$

with equality holding if and only if $\delta(X, Y) = \emptyset$.

Uncrossing technique

We start with the proof of integrality of the basic feasible solution. For a given b.f.s., we define the *tight subsets* as the subsets of vertices for which $x(E(S)) = |S| - 1$, meaning the constraints in the linear program are tight. Note that V, the full set of vertices, is tight, and that no subset of size 1 is tight by definition (they are not listed as equations).

D(Family $\mathcal{F}$): For a given basic feasible solution, let $\mathcal{F}$ be the family (set) of all subsets $S \subseteq V$ which are tight.

The size of $\mathcal{F}$ could be exponentially large, so we wish to find a small representative that covers the same *basis* (in the linear-algebraic sense). Some *laminar* subfamily would make sense, and to get it, we use the uncrossing technique:

L(Uncrossing): 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,

$$\chi(E(S)) + \chi(E(T)) = \chi(E(S \cup T)) + \chi(E(S \cap T)).$$

Sidenote: We can use the above lemma algorithmically:

1. Take any $S, T \in \mathcal{F}$.
2. If $S \subseteq T$ or $T \subseteq S$ or $S \cap T = \emptyset$, continue.
3. Else, remove T from the family but add $S \cap T$ and $S \cup T$ into the family.

Ignoring the sidenote, suppose we have any maximal laminar subfamily $\mathcal{L}$ of $\mathcal{F}$. We wish to prove that all other tight sets, i.e., the ones in $\mathcal{F} \setminus \mathcal{L}$, are linear equalities that can be derived from the equalities in $\mathcal{L}$.

Formally:

D(Span): We use $\text{span}(\mathcal{F})$ to denote the vector space of those sets $S \in F$ i.e. the linear subspace generated by the set $\{\chi(E(S)) : S \in \mathcal{F}\}$.

L(Laminar family is sufficient): If $\mathcal{L}$ is a maximal laminar subfamily of $\mathcal{F}$, then $\text{span}(\mathcal{L}) = \text{span}(\mathcal{F})$.

D(Crossing size with a laminar family): We define

$$\text{cross}(S, \mathcal{L}) = |\{T \in \mathcal{L} \mid S \text{ crosses } T\}|.$$

Proof steps: If the span of $\mathcal{L}$ is smaller, there are tight sets of vertices S' whose edges are not in $\text{span}(\mathcal{L})$. Among them, we find the set S with the minimum value of $\text{cross}(S, \mathcal{L})$.

Suppose that T is any set in $\mathcal{L}$ that crosses S. We prove the following:

Clm: For S and T as above, we have:

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

and

$$\text{cross}(S \cup T, \mathcal{L}) < \text{cross}(S, \mathcal{L}).$$

If this claim is true, we can prove (as a contradiction) that S is actually also in $\text{span}(\mathcal{L})$, because T, $T \cup S$ and $T \cap S$ are there, and the Uncrossing lemma claims that S is there also.

Proof of the claim: There is no set in $\mathcal{L}$ that crosses any of the two sets $S \cap T$ or $S \cup T$ without also crossing S itself. Furthermore, T itself is counted with $+1$ in the expression $\text{cross}(S, \mathcal{L})$ but T does not cross $S \cap T$ or $S \cup T$ – as $S \cap T \subseteq L$ and $T \subseteq S \cup T$. This makes the inequality strict.

Exercises

Exercise one — Prove the following equivalence about trees:

Given a graph G, a subgraph $T \subseteq G$ is a spanning tree if and only if it is a minimal (in number of edges) subgraph such that for any subset $S \subseteq V(G)$ that is nonempty and also $S \neq V$, there exists at least one edge from S to $V \setminus S$ that lies in T.

Exercise two — Inspired by the above exercise, we might be tempted to create the following linear program for spanning trees:

$$\begin{array}{rl} \min & \sum x_e \\ \forall S \subsetneq V, |S| \geq 1: & \sum_{uv \in E(G), u \in S, v \in V \setminus S} x_{uv} \geq 1, \\ & x_e \geq 0. \end{array}$$

Can you find some graph showing that this linear program may have an optimum that is smaller than any spanning tree solution?

Exercise three — Suppose we investigate the spanning tree polytope for the following graph. It has four vertices, v_1, v_2, v_3, v_4 and five edges, namely

$$\{v_1, v_3\}, \{v_1, v_4\}, \{v_2, v_3\}, \{v_2, v_4\}, \{v_3, v_4\}.$$

1. Write the full left-hand side matrix A of the spanning tree polytope for this instance.
2. Select the right square submatrix to show that A is not totally unimodular.

D(Interval form**):** Imagine one column vector v with values in $\{0, 1\}$. We say that v is in *interval form* if v has all the 1-values one after each other in one continuous interval (possibly of length zero). For example, a vector $(0, 0, 0, 0)$ has interval form, and so do $(0, 1, 1, 0)$ or $(1, 1, 0, 0)$. The vector $(1, 1, 0, 1)$ does not.

A matrix M is in interval form if all of its columns are in interval form.

Exercise four — Consider a *laminar* family $\mathcal{F}$ of subsets of V (so for any $F \in \mathcal{F}, F \subseteq V$). Suppose that each of these subsets $F \in \mathcal{F}$ corresponds to a tight inequality for the spanning tree polytope, with respect to some basic feasible solution x. In other words, for every $F \in \mathcal{F}$, if we take the sum of fractional values of x on edges inside $E(F)$, which we also denote as $x(E(F))$, then $x(E(F)) = |F| - 1$.

Now, consider a matrix A' corresponding to the left-hand sides of the tight inequalities of this laminar family. Prove that this matrix A' is in interval form, as defined above.

Two problems remaining from last time:

Exercise five — The following set of ideas pushes the running time for the Dinitz algorithm with unit capacities from $O(nm)$ to $O(m^{3/2})$. This only helps for sparse graphs, but following this idea further (which we will not do today) actually leads to an improved analysis of $O(n^{2/3}m)$.

1. Suppose that we are in the k-th iteration of the "outer loop" of the Dinitz algorithm. Thus, the shortest s-t-path in the residual network is of length at least k. Can you prove that there exists a directed s-t-cut in the residual network of size $\leq m/k$?
2. If we find a cut of size m/k, how many iterations of the "outer loop" are remaining?
3. How can we conclude that the running time is $O(m^{3/2})$?

Exercise six — Suppose that all capacities are integral and bounded by some constant C (and the graph is not a multigraph). Can you improve the analysis of the Dinitz algorithm (without changing it) to show that the overall runtime is $O(Cn^2 + mn)$?