CS599: Convex and Combinatorial Optimization
Fall 2013
Lecture 17: Combinatorial Problems as Linear Programs III

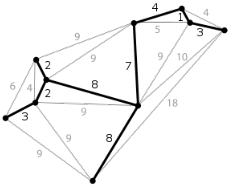
Instructor: Shaddin Dughmi

Announcements

- Today: Spanning Trees and Flows
  - Flexibility awarded by polyhedral perspective

Outline

The Minimum Cost Spanning Tree Problem



Given a connected undirected graph $G = (V, E)$, and costs c_e on edges e, find a minimum cost spanning tree of G.

- Spanning Tree: an acyclic set of edges connecting every pair of nodes
- When graph is disconnected, can search for min-cost spanning forest instead
- We use n and m to denote $|V|$ and $|E|$, respectively.

Kruskal's Algorithm

The minimum spanning tree problem can be solved efficiently by a simple greedy algorithm

Kruskal's algorithm

1. $T \leftarrow \emptyset$
2. Sort edges in increasing order of cost
3. For each edge e in order
    - if $T \bigcup e$ is acyclic, add e to T.

Kruskal's Algorithm

The minimum spanning tree problem can be solved efficiently by a simple greedy algorithm

Kruskal's algorithm

1. $T \leftarrow \emptyset$
2. Sort edges in increasing order of cost
3. For each edge e in order
   - if $T \bigcup e$ is acyclic, add e to T.

- Proof of correctness is via a simple exchange argument.
- Generalizes to Matroids

MST Linear Program

MST LP

$$
\begin{array}{lll}
\text{minimize} & \sum_{e \in E} c_e x_e & \\
\text{subject to} & \sum\limits_{e \in E} x_e = n - 1 & \\
& \sum\limits_{e \subseteq X} x_e \leq |X| - 1, & \text{for } X \subset V. \\
& x_e \geq 0, & \text{for } e \in E.
\end{array}
$$

MST Linear Program

MST LP

$$\begin{array}{lll} \text{minimize} & \sum_{e\in E} c_e x_e & \\ \text{subject to} & \sum\limits_{e\in E} x_e = n-1 & \\ & \sum\limits_{e\subseteq X} x_e \le |X|-1, & \text{for } X \subset V. \\ & x_e \ge 0, & \text{for } e \in E. \end{array}$$

Theorem

The feasible region of the above LP is the convex hull of spanning trees.

MST Linear Program

MST LP

$$
\begin{array}{lll}
\text{minimize} & \sum_{e\in E} c_e x_e & \\
\text{subject to} & \sum\limits_{e\in E} x_e = n-1 & \\
 & \sum\limits_{e\subseteq X} x_e \leq |X|-1, & \text{for } X \subset V. \\
 & x_e \geq 0, & \text{for } e \in E.
\end{array}
$$

Theorem

The feasible region of the above LP is the convex hull of spanning trees.

- Proof by finding a dual solution with cost matching the output of Kruskal's algorithm.

MST Linear Program

MST LP

$$\begin{array}{lll} \text{minimize} & \sum_{e\in E} c_e x_e & \\ \text{subject to} & \sum\limits_{e\in E} x_e = n-1 & \\ & \sum\limits_{e\subseteq X} x_e \leq |X|-1, & \text{for } X \subset V. \\ & x_e \geq 0, & \text{for } e \in E. \end{array}$$

Theorem

The feasible region of the above LP is the convex hull of spanning trees.

- Proof by finding a dual solution with cost matching the output of Kruskal's algorithm.
- Generalizes to Matroids

MST Linear Program

MST LP

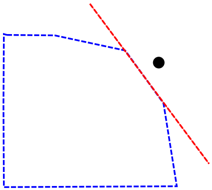
$$
\begin{aligned}
\text{minimize} \quad & \textstyle\sum_{e \in E} c_e x_e \\
\text{subject to} \quad & \sum_{e \in E} x_e = n - 1 \\
& \sum_{e \subseteq X} x_e \le |X| - 1, \quad \text{for } X \subset V. \\
& x_e \ge 0, \qquad\qquad\quad\ \text{for } e \in E.
\end{aligned}
$$

Theorem

The feasible region of the above LP is the convex hull of spanning trees.

- Proof by finding a dual solution with cost matching the output of Kruskal's algorithm.
- Generalizes to Matroids
- Note: this LP has an exponential (in n) number of constraints

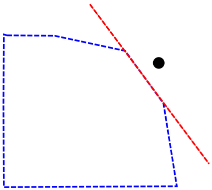
Solving the MST Linear Program



Definition

A separation oracle for a linear program with feasible set $\mathcal{P} \subseteq \mathbb{R}^m$ is an algorithm which takes as input $x \in \mathbb{R}^m$, and either certifies that $x \in \mathcal{P}$ or identifies a violated constraint.

Solving the MST Linear Program



Definition

A separation oracle for a linear program with feasible set $\mathcal{P} \subseteq \mathbb{R}^m$ is an algorithm which takes as input $x \in \mathbb{R}^m$, and either certifies that $x \in \mathcal{P}$ or identifies a violated constraint.

Theorem

A linear program with a polynomial number of variables is solvable in polynomial time if and only if it admits a polynomial time separation oracle.

Follows from the ellipsoid method, which we will see next week.

Solving the MST Linear Program

Primal LP

$$\begin{array}{lll} \text{minimize} & \sum_{e \in E} c_e x_e & \\ \text{subject to} & \sum_{e \subseteq X} x_e \leq |X| - 1, & \text{for } X \subset V. \\ & \sum_{e \in E} x_e = n - 1 & \\ & x_e \geq 0, & \text{for } e \in E. \end{array}$$

- Given $x \in \mathbb{R}^m$, separation oracle must find a violated constraint if one exists

Solving the MST Linear Program

Primal LP

$$
\begin{array}{lll}
\text{minimize} & \sum_{e \in E} c_e x_e & \\
\text{subject to} & \sum\limits_{e \subseteq X} x_e \leq |X| - 1, & \text{for } X \subset V. \\
& \sum\limits_{e \in E} x_e = n - 1 & \\
& x_e \geq 0, & \text{for } e \in E.
\end{array}
$$

- Given $x \in \mathbb{R}^m$, separation oracle must find a violated constraint if one exists
- Reduces to finding $X \subset V$ with $\sum_{e \subseteq X} x_e > |X| - 1$, if one exists
  - Equivalently $\frac{1+\sum_{e \subseteq X} x_e}{|X|} > 1$

Solving the MST Linear Program

Primal LP

$$
\begin{array}{lll}
\text{minimize} & \sum_{e \in E} c_e x_e & \\
\text{subject to} & \sum\limits_{e \subseteq X} x_e \leq |X| - 1, & \text{for } X \subset V. \\
& \sum\limits_{e \in E} x_e = n - 1 & \\
& x_e \geq 0, & \text{for } e \in E.
\end{array}
$$

- Given $x \in \mathbb{R}^m$, separation oracle must find a violated constraint if one exists
- Reduces to finding $X \subset V$ with $\sum_{e \subseteq X} x_e > |X| - 1$, if one exists
  - Equivalently $\frac{1 + \sum_{e \subseteq X} x_e}{|X|} > 1$
- In turn, this reduces to maximizing $\frac{1 + \sum_{e \subseteq X} x_e}{|X|}$ over X

Solving the MST Linear Program

Primal LP

$$\begin{array}{lll} \text{minimize} & \sum_{e \in E} c_e x_e & \\ \text{subject to} & \sum\limits_{e \subseteq X} x_e \leq |X| - 1, & \text{for } X \subset V. \\ & \sum\limits_{e \in E} x_e = n - 1 & \\ & x_e \geq 0, & \text{for } e \in E. \end{array}$$

- Given $x \in \mathbb{R}^m$, separation oracle must find a violated constraint if one exists
- Reduces to finding $X \subset V$ with $\sum_{e \subseteq X} x_e > |X| - 1$, if one exists
  - Equivalently $\frac{1 + \sum_{e \subseteq X} x_e}{|X|} > 1$
- In turn, this reduces to maximizing $\frac{1 + \sum_{e \subseteq X} x_e}{|X|}$ over X

We will see how to do this efficiently later in the class, since $\frac{1 + \sum_{e \subseteq X} x_e}{|X|}$ is a supermodular function of the set X.

Application of Fractional Spanning Trees

- The LP formulation of spanning trees has many applications
- We will look at one contrived yet simple application that shows the flexibility enabled by polyhedral formulation

Fault-Tolerant MST

- Your tree is an overlay network on the internet used to transmit data
- A hacker is looking to attack your tree, by knocking off one of the edges of the graph
- You can foil the hacker by choosing a random tree
- The hacker knows the algorithm you use, but not your random coins

Fault-tolerant MST LP

$$
\begin{array}{lll}
\text{minimize} & \sum_{e \in E} c_e x_e & \\
\text{subject to} & \sum_{e \subseteq X} x_e \leq |X| - 1, & \text{for } X \subset V. \\
& \sum_{e \in E} x_e = n - 1 & \\
& x_e \leq p, & \text{for } e \in E. \\
& x_e \geq 0, & \text{for } e \in E.
\end{array}
$$

- Above LP can be solved efficiently
- Can interpret resulting fractional spanning tree x as a recipe for a probability distribution over trees T
  - $e \in T$ with probability x_e
  - Since $x_e \leq p$, no edge is in the tree with probability more than p.

Fault-tolerant MST LP

$$
\begin{array}{lll}
\text{minimize} & \sum_{e \in E} c_e x_e & \\
\text{subject to} & \sum_{e \subseteq X} x_e \le |X| - 1, & \text{for } X \subset V. \\
& \sum_{e \in E} x_e = n - 1 & \\
& x_e \le p, & \text{for } e \in E. \\
& x_e \ge 0, & \text{for } e \in E.
\end{array}
$$

- Such a probability distribution exists!

Fault-tolerant MST LP

$$\begin{array}{lll} \text{minimize} & \sum_{e \in E} c_e x_e & \\ \text{subject to} & \sum_{e \subseteq X} x_e \leq |X| - 1, & \text{for } X \subset V. \\ & \sum_{e \in E} x_e = n - 1 & \\ & x_e \leq p, & \text{for } e \in E. \\ & x_e \geq 0, & \text{for } e \in E. \end{array}$$

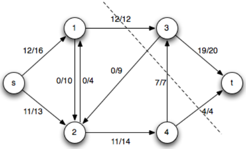
- Such a probability distribution exists!
  - x is in the (original) MST polytope
  - Caratheodory's theorem: x is a convex combination of $m + 1$ vertices of MST polytope
  - By integrality of MST polytope: x is the "expectation" of a probability distribution over spanning trees.

Fault-tolerant MST LP

$$
\begin{array}{lll}
\text{minimize} & \sum_{e\in E} c_e x_e & \\
\text{subject to} & \sum_{e\subseteq X} x_e \leq |X| - 1, & \text{for } X \subset V. \\
& \sum_{e\in E} x_e = n - 1 & \\
& x_e \leq p, & \text{for } e \in E. \\
& x_e \geq 0, & \text{for } e \in E.
\end{array}
$$

- Such a probability distribution exists!
  - x is in the (original) MST polytope
  - Caratheodory's theorem: x is a convex combination of $m+1$ vertices of MST polytope
  - By integrality of MST polytope: x is the “expectation” of a probability distribution over spanning trees.
- Consequence of Ellipsoid algorithm: can compute such a decomposition of x efficiently!

Outline



The Maximum Flow Problem

Given a directed graph $G = (V, E)$ with capacities u_e on edges e, a source node s, and a sink node t, find a maximum flow from s to t respecting the capacities.

$$
\begin{array}{lll}
\text{maximize} & \sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e & \\
\text{subject to} & \sum_{e \in \delta^-(v)} x_e = \sum_{e \in \delta^+(v)} x_e, & \text{for } v \in V \setminus \{s, t\}. \\
& x_e \leq u_e, & \text{for } e \in E. \\
& x_e \geq 0, & \text{for } e \in E.
\end{array}
$$

Can be computed either by solving the LP, or by a combinatorial algorithm such as Ford Fulkerson.

Primal LP

$$\max \sum_{e\in\delta^+(s)} x_e - \sum_{e\in\delta^-(s)} x_e$$

s.t.

$$\sum_{e\in\delta^-(v)} x_e = \sum_{e\in\delta^+(v)} x_e, \qquad \forall v \in V \setminus \{s,t\}$$

$$x_e \le u_e, \qquad \forall e \in E.$$

$$x_e \ge 0, \qquad \forall e \in E.$$

Dual LP (Simplified)

$$\min \sum_{e\in E} u_e z_e$$

s.t.

$$y_v - y_u \le z_e, \qquad \forall e = (u,v) \in E.$$

$$y_s = 0$$

$$y_t = 1$$

$$z_e \ge 0, \qquad \forall e \in E.$$

- Dual solution describes fraction z_e of each edge to fractionally cut

Primal LP

$$\max \sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e$$

s.t.

$$\sum_{e \in \delta^-(v)} x_e = \sum_{e \in \delta^+(v)} x_e, \qquad \forall v \in V \setminus \{s,t\}$$

$$x_e \le u_e, \qquad \forall e \in E.$$

$$x_e \ge 0, \qquad \forall e \in E.$$

Dual LP (Simplified)

$$\min \sum_{e \in E} u_e z_e$$

s.t.

$$y_v - y_u \le z_e, \qquad \forall e = (u,v) \in E.$$

$$y_s = 0$$

$$y_t = 1$$

$$z_e \ge 0, \qquad \forall e \in E.$$

- Dual solution describes fraction z_e of each edge to fractionally cut
- Dual constraints require that at least 1 edge is cut on every path from s to t.
  - $\sum_{(u,v) \in P} z_{uv} \ge \sum_{(u,v) \in P} y_v - y_u = y_t - y_s = 1$

Primal LP

$$\max \sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e$$

s.t.

$$\sum_{e \in \delta^-(v)} x_e = \sum_{e \in \delta^+(v)} x_e, \qquad \forall v \in V \setminus \{s,t\}$$

$$x_e \le u_e, \qquad \forall e \in E.$$

$$x_e \ge 0, \qquad \forall e \in E.$$

Dual LP (Simplified)

min $\sum_{e \in E} u_e z_e$

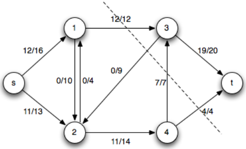
s.t.

$$y_v - y_u \le z_e, \qquad \forall e = (u,v) \in E.$$

$$y_s = 0$$

$$y_t = 1$$

$$z_e \ge 0, \qquad \forall e \in E.$$



- Every integral $s - t$ cut is feasible.

Primal LP

$$\max \sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e$$

s.t.

$$\sum_{e \in \delta^-(v)} x_e = \sum_{e \in \delta^+(v)} x_e, \qquad \forall v \in V \setminus \{s, t\}$$

$$x_e \leq u_e, \qquad \forall e \in E.$$

$$x_e \geq 0, \qquad \forall e \in E.$$

Dual LP (Simplified)

$$\min \sum_{e \in E} u_e z_e$$

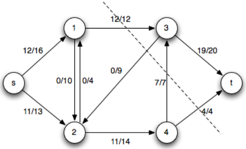
s.t.

$$y_v - y_u \leq z_e, \qquad \forall e = (u, v) \in E.$$

$$y_s = 0$$

$$y_t = 1$$

$$z_e \geq 0, \qquad \forall e \in E.$$



- Every integral $s - t$ cut is feasible.
- By weak duality: max flow $\leq$ minimum cut

Primal LP

$$\max \sum_{e\in\delta^+(s)} x_e - \sum_{e\in\delta^-(s)} x_e$$

s.t.

$$\sum_{e\in\delta^-(v)} x_e = \sum_{e\in\delta^+(v)} x_e, \qquad \forall v \in V \setminus \{s,t\}$$

$$x_e \le u_e, \qquad \forall e \in E.$$

$$x_e \ge 0, \qquad \forall e \in E.$$

Dual LP (Simplified)

$$\min \sum_{e\in E} u_e z_e$$

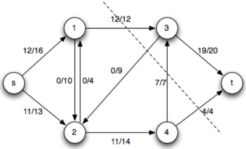
s.t.

$$y_v - y_u \le z_e, \qquad \forall e = (u,v) \in E.$$

$$y_s = 0$$

$$y_t = 1$$

$$z_e \ge 0, \qquad \forall e \in E.$$



- Every integral $s-t$ cut is feasible.
- By weak duality: max flow $\le$ minimum cut
- Ford-Fulkerson shows that max flow $=$ min cut
  - i.e. dual has integer optimal

Primal LP

$$\max \sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e$$

s.t.

$$\sum_{e \in \delta^-(v)} x_e = \sum_{e \in \delta^+(v)} x_e, \qquad \forall v \in V \setminus \{s,t\}$$

$$x_e \le u_e, \qquad \forall e \in E.$$

$$x_e \ge 0, \qquad \forall e \in E.$$

Dual LP (Simplified)

$$\min \sum_{e \in E} u_e z_e$$

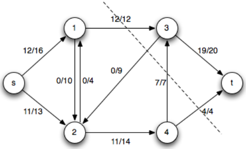
s.t.

$$y_v - y_u \le z_e, \qquad \forall e = (u,v) \in E.$$

$$y_s = 0$$

$$y_t = 1$$

$$z_e \ge 0, \qquad \forall e \in E.$$



- Every integral $s-t$ cut is feasible.
- By weak duality: max flow $\le$ minimum cut
- Ford-Fulkerson shows that max flow $=$ min cut
  - i.e. dual has integer optimal
- Ford-Fulkerson also shows that there is an integral optimal flow when capacities are integer.

Generalizations of Max Flow

$$\max \sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e$$

s.t.

$$\sum_{e \in \delta^-(v)} x_e = \sum_{e \in \delta^+(v)} x_e, \qquad \forall v \in V \setminus \{s, t\}.$$

$$x_e \leq u_e, \qquad \forall e \in E.$$

$$x_e \geq 0, \qquad \forall e \in E.$$

Writing as an LP shows that many generalizations are also tractable

Generalizations of Max Flow

$$\begin{aligned}
&\text{max} \sum_{e\in\delta^+(s)} x_e - \sum_{e\in\delta^-(s)} x_e \\
&\text{s.t.} \\
&\sum_{e\in\delta^-(v)} x_e = \sum_{e\in\delta^+(v)} x_e, && \forall v \in V \setminus \{s,t\}. \\
&x_e \le u_e, && \forall e \in E. \\
&x_e \ge 0, && \forall e \in E.
\end{aligned}$$

Writing as an LP shows that many generalizations are also tractable

- Lower and upper bound constraints on flow: $\ell_e \le x_e \le u_e$

Generalizations of Max Flow

$$\max \sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e$$

s.t.

$$\sum_{e \in \delta^-(v)} x_e = \sum_{e \in \delta^+(v)} x_e, \qquad \forall v \in V \setminus \{s, t\}\,.$$

$$x_e \le u_e, \qquad \forall e \in E.$$

$$x_e \ge 0, \qquad \forall e \in E.$$

Writing as an LP shows that many generalizations are also tractable

- Lower and upper bound constraints on flow: $\ell_e \le x_e \le u_e$
- minimum cost flow of a certain amount r
  - Objective $\min \sum_e c_e x_e$
  - Additional constraint: $\sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e = r$

Generalizations of Max Flow

$$\max \sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e$$

s.t.

$$\sum_{e \in \delta^-(v)} x_e = \sum_{e \in \delta^+(v)} x_e, \qquad \forall v \in V \setminus \{s,t\}.$$

$$x_e \leq u_e, \qquad \forall e \in E.$$

$$x_e \geq 0, \qquad \forall e \in E.$$

Writing as an LP shows that many generalizations are also tractable

- Lower and upper bound constraints on flow: $\ell_e \leq x_e \leq u_e$
- minimum cost flow of a certain amount r
    - Objective $\min \sum_e c_e x_e$
    - Additional constraint: $\sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e = r$
- Multiple commodities sharing the network

Generalizations of Max Flow

$$
\begin{aligned}
&\text{max} \sum_{e\in\delta^+(s)} x_e - \sum_{e\in\delta^-(s)} x_e \\
&\text{s.t.} \\
&\sum_{e\in\delta^-(v)} x_e = \sum_{e\in\delta^+(v)} x_e, \qquad && \forall v \in V \setminus \{s,t\}. \\
&x_e \leq u_e, && \forall e \in E. \\
&x_e \geq 0, && \forall e \in E.
\end{aligned}
$$

Writing as an LP shows that many generalizations are also tractable

- Lower and upper bound constraints on flow: $\ell_e \leq x_e \leq u_e$
- minimum cost flow of a certain amount r
  - Objective $\min \sum_e c_e x_e$
  - Additional constraint: $\sum_{e\in\delta^+(s)} x_e - \sum_{e\in\delta^-(s)} x_e = r$
- Multiple commodities sharing the network
- …