

MA/AMA 514 :

Networks & Combinatorial Optimization

Today: §1.4, §2.1, §2.2

From last time :

Given a connected graph $G = (V, E)$
and length function $l: E \rightarrow \mathbb{R}$, want to
find minimum length spanning tree :

$$l(T) = \sum_{e \in T} l(e).$$

Thm 1.11 Suppose

- F is a greedy forest of G
- U is the vertex set of a conn. component of F
- $e \in \delta(U)$ has min. length in $\delta(U)$

then $F \cup \{e\}$ is a greedy forest of G .

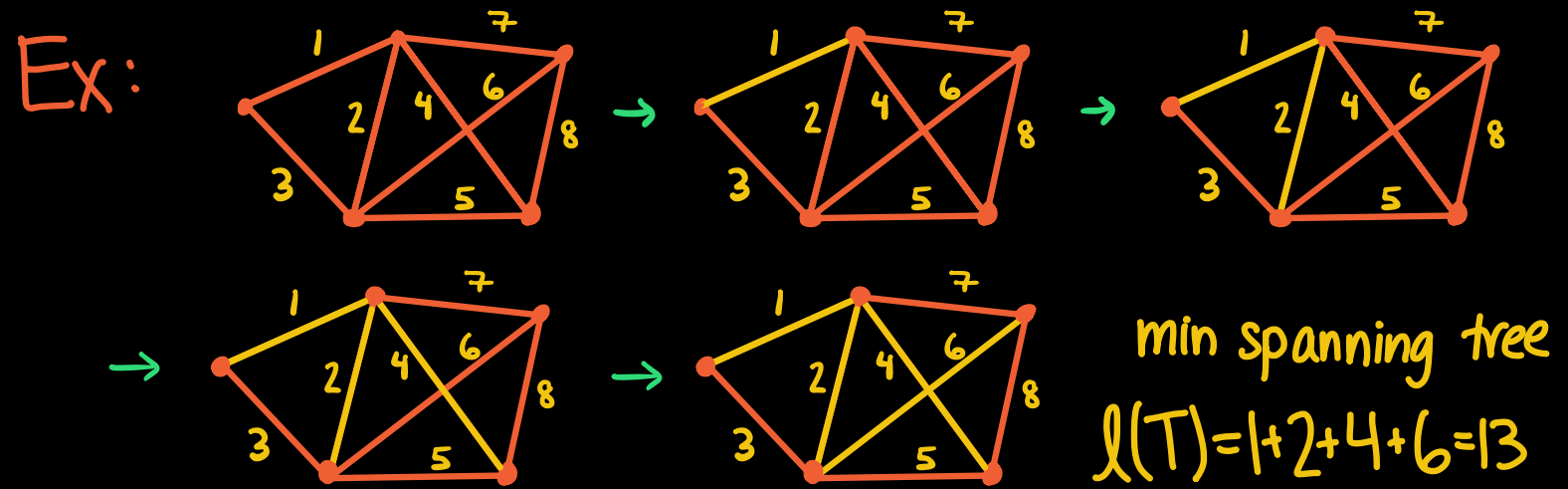
Kruskal's Algorithm (1956)

- 1) Sort edges s.t. $l(e_1) \leq l(e_2) \leq \dots \leq l(e_m)$ ($|E| = m$)
- 2) Set $F_0 = (V, \emptyset)$

3) For $i=1, \dots, m$, if $F_{i-1} \cup \{e_i\}$ acyclic, $F_i = F_{i-1} \cup \{e_i\}$
 otherwise $F_i = F_{i-1}$.

Output $T = F_m$.

Run time:
 $O(|E| \log(|E|))$
 sorting edges!



(Proof of correctness) Start F_0 is a greedy forest
 If $F_{i-1} \cup \{e_i\}$ is acyclic, e_i connects two connected components U, U' of $F_{i-1} \Rightarrow e_i \in \delta(U)$.

If $l(f) < l(e_i)$, end pts of f are connected in F_{i-1} .

$\Rightarrow e_i$ has min length among $\delta(U)$.

$\Rightarrow$ By Thm 1.1, F_i is a greedy forest.

Q: How do we implement step (3)?

One way is to keep track of the connected components of F_i . $F_{i-1} \cup \{e_i\}$ acyclic $\Leftrightarrow$
 endpts of e_i lie in different conn. comp.

This is an example of a greedy algorithm.
Greedy Algorithms

"Greedy algorithms" don't always find the optimal solution! Formally..

Problem: Given a finite set X , collection $\mathcal{B}$ of subsets of X , and weights $w: X \rightarrow \mathbb{R}$, find the minimum weight element of $\mathcal{B}$, where

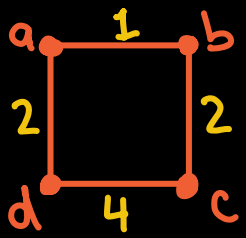
$$w(S) = \sum_{x \in S} w(x).$$

Greedy Algorithm Start: $S = \emptyset$

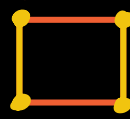
While $S \notin \mathcal{B}$, choose $x \in X \setminus S$ of minimum weight so that $S \cup \{x\} \subseteq B$ for some $B \in \mathcal{B}$ and update $S \rightarrow S \cup \{x\}$.

If there is no such x , stop and output S .

Ex (failure! Find the minimum length spanning set of $G=(V,E)$, $l:E \rightarrow \mathbb{R}$



Greedy alg: choose shortest edge (a,b)
try to add to it...

Only optimal solution  doesn't use ab

Structures for which the greedy alg. always finds the optimal solution are called matroids.
(§10)

Another general tool for solving combinatorial optimization problems is linear programming!

Linear Programming : Convexity

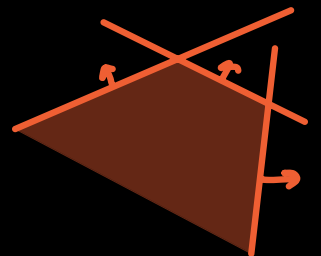
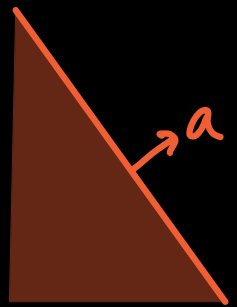
A closed halfspace is a set of the form $\{x \in \mathbb{R}^n : a^T x \leq b\}$

where $a \in \mathbb{R}^n$, $b \in \mathbb{R}$

A polyhedron is the intersection of finitely-many halfspaces, i.e.

$$P = \{x \in \mathbb{R}^n : a_1^T x \leq b_1, \dots, a_m^T x \leq b_m\}$$

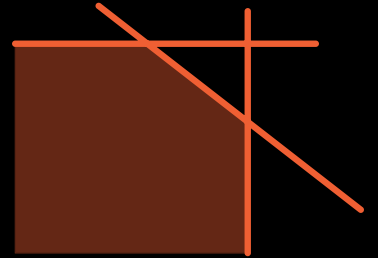
where $a_1, \dots, a_m \in \mathbb{R}^n$, $b_1, \dots, b_m \in \mathbb{R}$



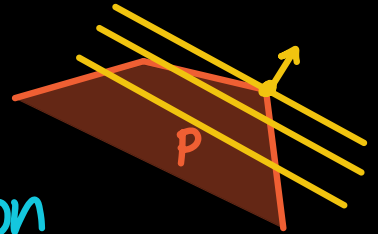
We write this as $P = \{x \in \mathbb{R}^n : Ax \leq b\}$
where

$$A = \begin{bmatrix} -a_1 & - \\ \vdots & \\ -a_m & - \end{bmatrix} \in \mathbb{R}^{m \times n} \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \in \mathbb{R}^m$$

Ex: $Ax = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_1 + x_2 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} = b$



Linear programming



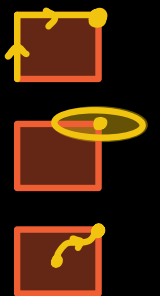
The problem of maximizing a linear function over a polyhedron is known as a linear program (LP)

One standard form: $\max \{c^T x : Ax \leq b\}$
where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$

Polyhedron $P = \{x : Ax \leq b\} \subseteq \mathbb{R}^n$, defined by m inequalities

There are many methods for solving LP's:

- Simplex method (Dantzig, 1951)
- ellipsoid method (Khachiyan, 1979)
- interior point methods (Karmarkar, 1984)

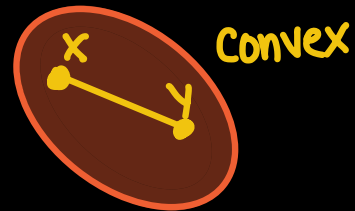


first polynomial time alg. for solving LP's

What points attain the optimal value?

Convexity

A set $C \subseteq \mathbb{R}^n$ is convex if for every $x, y \in C$, $\lambda \in [0, 1]$, $\lambda x + (1 - \lambda)y \in C$. That is, C contains the line segment between any two points in C .



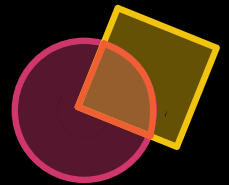
Remark: The intersection of convex sets is convex.

$C_i \subseteq \mathbb{R}^n$ convex for all $i \in I \Rightarrow \bigcap_{i \in I} C_i$ convex

$$x, y \in \bigcap_i C_i \Rightarrow x, y \in C_i \quad \forall i \in I$$

$$\Rightarrow \lambda x + (1 - \lambda)y \in C_i \quad \forall i \in I$$

$$\Rightarrow \lambda x + (1 - \lambda)y \in \bigcap_{i \in I} C_i$$



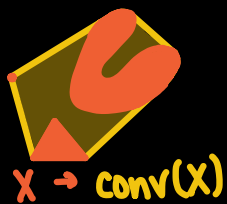
The convex hull of a set $X \subseteq \mathbb{R}^n$, denoted $\text{conv}(X)$ is the smallest convex set containing X , i.e.

$$\text{conv}(X) = \bigcap_{\substack{C \text{ convex} \\ X \subseteq C}} C$$

Alternatively,

$$\text{conv}(X) = \left\{ \sum_{i=1}^m \lambda_i x_i : x_1, \dots, x_m \in X, \lambda_i \geq 0, \sum_{i=1}^m \lambda_i = 1 \right\}$$

← a "convex combination" of $x_1, \dots, x_m$



The convex hull of a finite set is a polytope