# ISyE/CS/Math 728: Integer Optimization
# Trees and Submodular Polyhedra
# Total Dual integrality

Trees and Submodular Polyhedra

UW-Madison

Spring 2021

# Outline

- Maximum weight trees/forests
- Generalization of formulation: Submodular polyhedra $\Rightarrow$ Proof that max weight forest polyhedron is integral
- Total Dual Integrality

# Outline

Maximum weight trees

Submodular function and polyhedra

# Maximum Weight Tree Problem

**Tree?**

Consider a graph $G = (V, E)$

- A subgraph $G' = (V, E')$ of $G$ is a forest if it contains no cycles
- A subgraph $G' = (V, E')$ is a tree if it is a forest and is connected.

**Maximum weight forest (tree) problem:**

Given a graph $G = (V, E)$ and edge weights $c_e$ for $e \in E$, find a maximum subgraph that is a forest (tree).

- Common variant: *Minimum spanning tree* – find a tree with minimum cost

# Notation and some definitions

Let $G = (V, E)$ be an undirected graph:

- For $S \subseteq V$, let $E(S) = \big\{e = \{i, j\} \in E : i \in S, j \in S\big\}$

We'll need these definitions later.

- $G$ is connected if there exists a path between every pair of nodes in $V$
- Given a graph $G = (V, E)$, its connected components, $(V_i, E(V_i)), i = 1, \ldots, k$, are disjoint subgraphs of $G$ such that $V = \bigcup_{i=1}^k V_i$, $E = \bigcup_{i=1}^k E(V_i)$, and $(V_i, E(V_i))$ are connected for $i = 1, \ldots, k$

# Is it an integer programming problem?

**Maximum weight forest (tree) problem:**
Given a graph $G = (V, E)$ and edge weights $c_e$ for $e \in E$, find a maximum subgraph that is a forest (tree).

**Theorem**
A graph $G = (V, E')$ is a tree if and only if $E'$ has no cycles and $|E'| = |V| - 1$.

- Now we can formulate them as an integer programming problems (as in your homework!)

# Maximum weight forest formulation

The LP relaxation of this IP formulation is an integral polyhedron

- This is *far* from obvious (but we'll prove it later!)
- It is sufficient to solve the LP relaxation

The separation problem for these inequalities can be solved in polynomial-time

- So, the optimization problem can be solved in polynomial-time by the ellipsoid algorithm

But it can be solved *much more easily!*

# Solution: Be greedy!

**Kruskal's Algorithm (Greedy algorithm)**
*Initialize*: Start with $T^0 = \emptyset$, Order edges by nonincreasing weight: $c_1 \geq c_2 \geq \cdots \geq c_m$, where $c_t$ is cost of edge $e_t$
*Iteration t*: If $T^{t-1} \cup \{e_t\}$ contains no cycle, set $T^t = T^{t-1} \cup \{e_t\}$. Otherwise set $T^t = T^{t-1}$. If $|T^t| = n - 1$, Stop.

- Stop if $c_t \leq 0$ to obtain maximum weight forest
- Proof of correctness: Special case of later result
- It is polynomial time: $O(m \log m)$
  - First sort the edges
  - Then step through the edges in decreasing order – key work here is determining whether adding an edge creates a cycle – can be done in $\log(n)$ time
- That's quite a bit better than the ellipsoid algorithm!

# Outline

Maximum weight trees

Submodular function and polyhedra

# Submodular set function

Notation:

- Given a ground set $N$, we let $2^N$ denote all possible subsets of $N$

**Definition**

A set function $f : 2^N \to \mathbb{R}$ is submodular if:

$$f(A) + f(B) \geq f(A \cap B) + f(A \cup B) \quad \text{for all } A, B \subseteq N$$

**Definition**

A set function $f : 2^N \to \mathbb{R}$ is nondecreasing if:

$$f(A) \leq f(B) \quad \text{for all } A, B \text{ with } A \subset B \subseteq N$$

# Submodular set function

**Equivalent Definition**

A set function $f : 2^N \to \mathbb{R}$ is submodular if for all $A \subseteq B$ and $i \in N \setminus B$

$$f(A \cup \{i\}) - f(A) \geq f(B \cup \{i\}) - f(B)$$

# Examples of a submodular function

Given a graph $(V, E)$:

- Let $N = E$, for $A \subseteq E$ let $(V_1, E(V_1)), \ldots, (V_k, E(V_k))$ denote the connected components of the graph induced by the edges $A$, and define

$$f^G(A) = \sum_{i=1}^{K} (|V_i| - 1)$$

- Claim: $f^G$ is submodular

Other examples

- Coverage function: $f(S) = |\bigcup_{i \in S} E_i|$ for $S \subseteq \{1, \ldots, n\}$ where $E_i \subseteq \Omega$, $i = 1, \ldots, n$
- Shannon entropy $h(S)$ where $S$ is index set of subset of random variables from $X_1, \ldots, X_n$
- Cut function in a graph $G = (V, E)$: $f(S) = |\delta(S)|$, for $S \subseteq V$

# Submodular polyhedra

**Definition**

Given a submodular and nondecreasing function $f$ with $f(\emptyset) = 0$, the submodular polyhedron associated with $f$ is the set:

$$P(f) := \{x \in \mathbb{R}^n_+ : \sum_{j \in Q} x_j \leq f(Q) \text{ for } Q \subseteq N\}.$$

- Such a set is also sometimes called a *polymatroid*

**Claim**

Given a connected graph $G = (V, E)$, the associated maximum weight forest polyhedron is equivalent to $P(f^G)$.

# Optimization over submodular polyhedra

Given a nondecreasing submodular function $f$ with $f(\emptyset) = 0$, and cost vector $c \in \mathbb{R}^m$, where $m = |N|$, we are often interested in the following linear optimization problem:

$$\max\{cx : x \in P(f)\} \tag{1}$$

**Greedy algorithm**

(i) Order the variables so that $c_1 \geq c_2 \geq \cdots \geq c_r > 0 \geq c_{r+1} \geq \cdots \geq c_m$.

(ii) Set $\bar{x}_i = f(S^i) - f(S^{i-1})$ for $i = 1, \ldots, r$ and $\bar{x}_j = 0$ for $j > r$, where $S^i = \{1, \ldots, i\}$ for $i = 1, \ldots, r$ and $S^0 = \emptyset$.

**Theorem**

The greedy algorithm solves (1).

# Integrality of the maximum weight forest polyhedron

**Corollary**

The maximum weight forest inequalities, $x \geq 0$ and

$$\sum_{e \in E(S)} x_e \leq |S| - 1, \quad \text{for } S \subseteq N$$

define an integral polyhedron.

Proof:

- For any extreme point $x$ of a polyhedron, there exists a cost vector $c \in \mathbb{Z}^m$ such that $x$ is the unique optimal solution to $\max\{cx : x \in P(f^G)\}$
- Greedy algorithm constructs an optimal solution with either $x_i = 0$, or $x_i = f(S^i) - f(S^{i-1})$
- $f(S)$ is integer for all $S \subseteq E$, and hence $x \in \mathbb{Z}^m$

# Another condition for integral polyhedra

We showed polyhedron defined by max forest problem formulation is integral

- Example of another way to prove a polyhedron is integral

For a polyhedron $P$ and $c \in \mathbb{R}^n$, define the LP:

$$z_{LP} = \max\{cx : x \in P\} \tag{LP}$$

**Theorem**

Let $P$ be a pointed polyhedron. Then, the following are equivalent:

1. $P$ is integral (i.e., $P = \text{conv}(P \cap \mathbb{Z}^n)$).
2. (LP) has an integral optimal solution for all $c \in \mathbb{R}^n$ for which it has an optimal solution.
3. (LP) has an integral optimal solution for all $c \in \mathbb{Z}^n$ for which it has an optimal solution.
4. $z_{LP}$ is integral for every $c \in \mathbb{Z}^n$ for which (LP) has an optimal solution.

# Application: TDI

**Definition**
A set of linear inequalities $Ax \leq b$ is called Totally Dual Integral (TDI) if, for all $c \in \mathbb{Z}^n$ for which the linear program $\max\{cx : Ax \leq b\}$ has a finite optimal value, the dual linear program

$$\min\{yb : yA = c, y \geq 0\}$$

has an optimal solution with $y$ integral.

- If $A$ is TU, then $Ax \leq b$ is TDI
- Inequalities defining a submodular polyhedron are TDI.

**Theorem**
If $Ax \leq b$ is TDI, $b$ is an integer vector, and $P = \{x \in \mathbb{R}^N : Ax \leq b\}$ is pointed, then $P$ is an integral polyhedron.

# TDI: An inequality system property

Careful!

- TDI is a property of inequality description $Ax \leq b$, *not* of the polyhedron $P = \{x : Ax \leq b\}$

**Proposition**

Every polyhedron $P$ can be represented by a TDI linear inequality system.

- Integral polyhedra are precisely those for which there is a TDI representation with the right-hand side $b$ integral