Advanced Algorithms (Fall 2024)

Linear Programming

Lecturers: 尹一通，栗师，刘景铖

Nanjing University

Outline

1. Linear Programming
   - Introduction
   - Methods for Solving Linear Programs
2. Polytope with Polynomial Number of Facets
   - Bipartite Matching Polytope
   - Polytopes with Totally Unimodular Coefficient Matrices
3. Polytopes with Efficient Separation Oracles
   - s-t Cut Polytope
   - Spanning Tree Polytope
   - General Graph (Perfect) Matching Polytope
4. Matroid, Matroid Basis and Matroid Intersection Polytopes *
   - Preliminaries on Matroid Theory
   - Matroid Polytope
   - Matroid Basis and Matroid Intersection Polytope

Outline

1. Linear Programming
   - Introduction
   - Methods for Solving Linear Programs
2. Polytope with Polynomial Number of Facets
   - Bipartite Matching Polytope
   - Polytopes with Totally Unimodular Coefficient Matrices
3. Polytopes with Efficient Separation Oracles
   - s-t Cut Polytope
   - Spanning Tree Polytope
   - General Graph (Perfect) Matching Polytope
4. Matroid, Matroid Basis and Matroid Intersection Polytopes *
   - Preliminaries on Matroid Theory
   - Matroid Polytope
   - Matroid Basis and Matroid Intersection Polytope

Typical Combinatorial Optimizatiion Problem

Input: $[n]$: ground set

$\mathcal{S}$: feasible sets: a family of subsets of U, often implicitly given

$w_i, i \in [n]$: values/costs of elements

Output: the set $S \in \mathcal{S}$ with the minimum/maximum $w(S) := \sum_{i \in S} w_i$

Example:

- Shortest Path, Minimum Spanning Tree
- Maximum Independent Set, Maximum Matching, Knapsack Packing

Typical Combinatorial Optimizatiion Problem

Input: $[n]$: ground set

$\mathcal{S}$: feasible sets: a family of subsets of U, often implicitly given

$w_i, i \in [n]$: values/costs of elements

Output: the set $S \in \mathcal{S}$ with the minimum/maximum $w(S) := \sum_{i \in S} w_i$

Example:

- Shortest Path, Minimum Spanning Tree
- Maximum Independent Set, Maximum Matching, Knapsack Packing

- CO problem $\Longleftrightarrow$ Integer Program (IP) $\xrightarrow{\text{relax?}}$ Linear Program (LP)

Typical Combinatorial Optimizatiion Problem

Input: $[n]$: ground set

$\mathcal{S}$: feasible sets: a family of subsets of U, often implicitly given

$w_i, i \in [n]$: values/costs of elements

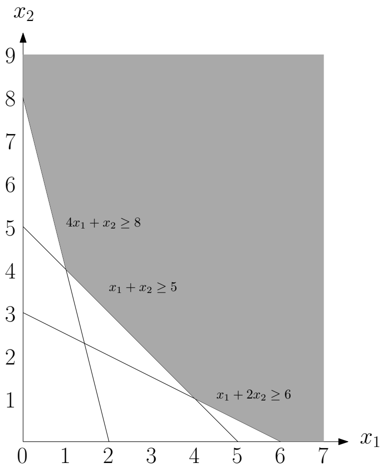
Output: the set $S \in \mathcal{S}$ with the minimum/maximum $w(S) := \sum_{i \in S} w_i$

Example:

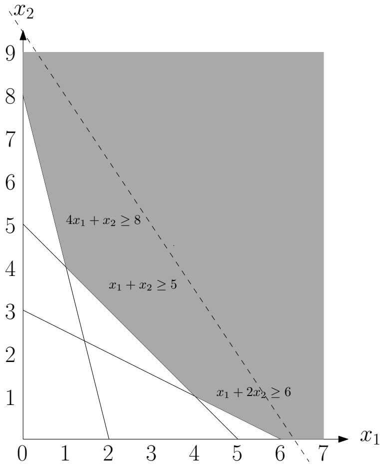
- Shortest Path, Minimum Spanning Tree
- Maximum Independent Set, Maximum Matching, Knapsack Packing

- CO problem $\Longleftrightarrow$ Integer Program (IP) $\xrightarrow{\text{relax?}}$ Linear Program (LP)
- In general: Integer programming is NP-hard; linear programming is in P

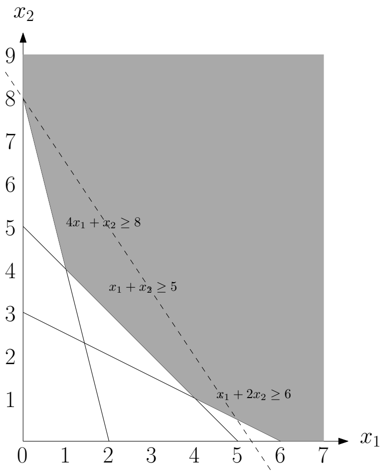
$$
\begin{aligned}
\min \quad & 7x_1 + 4x_2 \\
& x_1 + x_2 \geq 5 \\
& x_1 + 2x_2 \geq 6 \\
& 4x_1 + x_2 \geq 8 \\
& x_1, x_2 \geq 0
\end{aligned}
$$



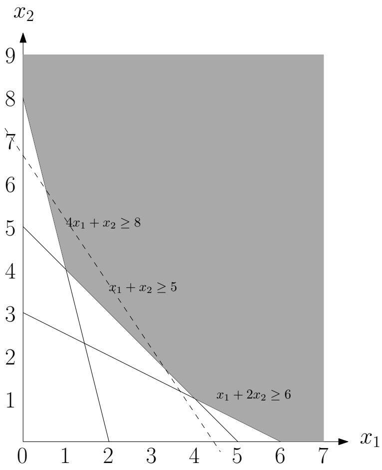
$$\begin{aligned} \min \quad & 7x_1 + 4x_2 \\ & x_1 + x_2 \geq 5 \\ & x_1 + 2x_2 \geq 6 \\ & 4x_1 + x_2 \geq 8 \\ & x_1, x_2 \geq 0 \end{aligned}$$



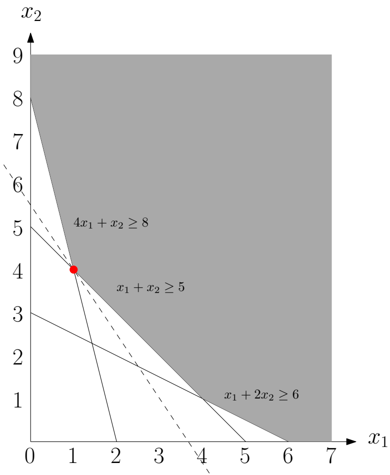
$$
\begin{aligned}
\min \quad & 7x_1 + 4x_2 \\
& x_1 + x_2 \geq 5 \\
& x_1 + 2x_2 \geq 6 \\
& 4x_1 + x_2 \geq 8 \\
& x_1, x_2 \geq 0
\end{aligned}
$$



$$\begin{aligned} \min \quad & 7x_1 + 4x_2 \\ x_1 + x_2 &\geq 5 \\ x_1 + 2x_2 &\geq 6 \\ 4x_1 + x_2 &\geq 8 \\ x_1, x_2 &\geq 0 \end{aligned}$$

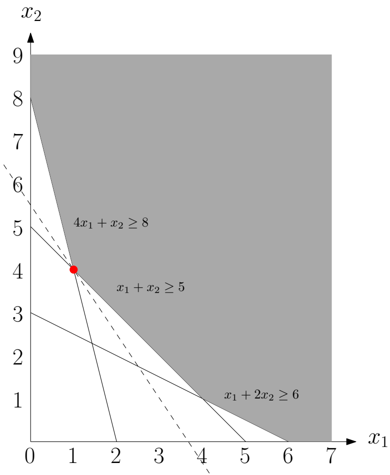


$$
\begin{aligned}
\min \quad & 7x_1 + 4x_2 \\
& x_1 + x_2 \geq 5 \\
& x_1 + 2x_2 \geq 6 \\
& 4x_1 + x_2 \geq 8 \\
& x_1, x_2 \geq 0
\end{aligned}
$$



$$\begin{aligned} \min \quad & 7x_1 + 4x_2 \\ & x_1 + x_2 \geq 5 \\ & x_1 + 2x_2 \geq 6 \\ & 4x_1 + x_2 \geq 8 \\ & x_1, x_2 \geq 0 \end{aligned}$$

- optimum solution: $x_1 = 1, x_2 = 4$
- optimum value $= 7 \times 1 + 4 \times 4 = 23$



Standard Form of Linear Programs

$$\begin{aligned} \min \quad & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n &\geq b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n &\geq b_2 \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots & \\ a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,n}x_n &\geq b_m \\ x_1, x_2, \cdots, x_n &\geq 0 \end{aligned}$$

- n: number of variables $\qquad m$: number of constraints

Standard Form of Linear Programs

$$
\begin{aligned}
\min \quad & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
& a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n \geq b_1 \\
& a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n \geq b_2 \\
& \quad \vdots \qquad\quad \vdots \qquad\quad \vdots \qquad\quad \vdots \\
& a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,n}x_n \geq b_m \\
& x_1, x_2, \cdots, x_n \geq 0
\end{aligned}
$$

- n: number of variables $\quad$ m: number of constraints
- Other considerations: $\leq$ constraints? equlities?

Standard Form of Linear Programs

$$
\begin{aligned}
\min \quad & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
& a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n \geq b_1 \\
& a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n \geq b_2 \\
& \quad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\
& a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,n}x_n \geq b_m \\
& x_1, x_2, \cdots, x_n \geq 0
\end{aligned}
$$

- n: number of variables $\qquad$ m: number of constraints
- Other considerations: $\leq$ constraints? equlities?
- variables can be negative? maximization problem?

Standard Form of Linear Programs

$$x := \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n, \qquad c := \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} \in \mathbb{R}^n,$$

$$A := \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \in \mathbb{R}^{n \times m}, \quad b := \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \in \mathbb{R}^m.$$

$$
\begin{aligned}
\min \quad & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
& a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n \geq b_1 \\
& a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n \geq b_2 \\
& \quad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\
& a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,n}x_n \geq b_m \\
& x_1, x_2, \cdots, x_n \geq 0
\end{aligned}
$$

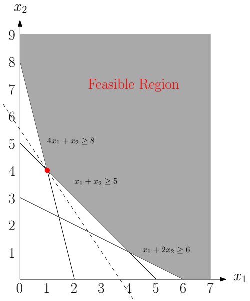
Standard Form of Linear Program

$$
\begin{aligned}
\min \quad & c^{\mathrm{T}}x \\
& Ax \geq b \\
& x \geq 0
\end{aligned}
$$

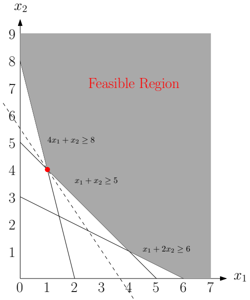
- $\geq$: coordinate-wise less than or equal to

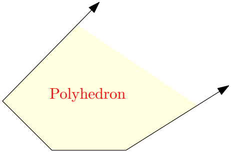
Preliminaries

- feasible region: the set of x's satisfying $Ax \geq b, x \geq 0$

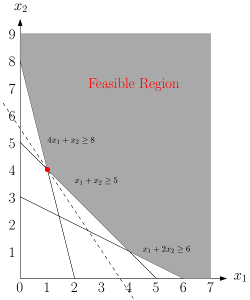


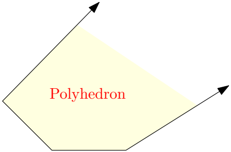
- feasible region: the set of x's satisfying $Ax \geq b, x \geq 0$
- a polyhedron is the intersection of finite number of closed half-spaces
- so, feasible region is a polyhedron

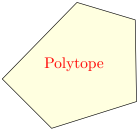




- feasible region: the set of x's satisfying $Ax \geq b, x \geq 0$
- a polyhedron is the intersection of finite number of closed half-spaces
- so, feasible region is a polyhedron
- if every coordinate has an upper and lower bound in the polyhedron, then the polyhedron is a polytope





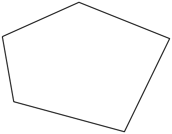


Given a polytope $\mathcal{P} \subseteq \mathbb{R}^n$:

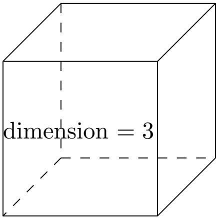
- The dimension of $\mathcal{P}$ is n minus the maximum number of linearly-independent equalities satisfied by all points in $\mathcal{P}$.

Given a polytope $\mathcal{P} \subseteq \mathbb{R}^n$:

- The dimension of $\mathcal{P}$ is n minus the maximum number of linearly-independent equalities satisfied by all points in $\mathcal{P}$.

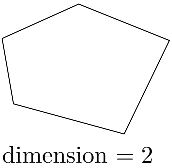


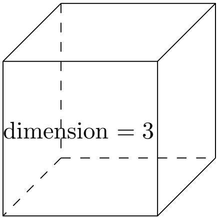
dimension $= 2$



Given a polytope $\mathcal{P} \subseteq \mathbb{R}^n$:

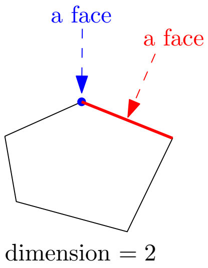
- The dimension of $\mathcal{P}$ is n minus the maximum number of linearly-independent equalities satisfied by all points in $\mathcal{P}$.
- Assume the linear inequality $a^{\mathrm{T}}x \leq b$ holds for every $x \in \mathcal{P}$, and some $x \in \mathcal{P}$ satisfies $a^{\mathrm{T}}x = b$. Then $\{x \in \mathcal{P} : a^{\mathrm{T}}x = b\}$ is said to be a face of $\mathcal{P}$.

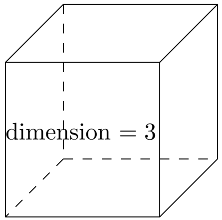




Given a polytope $\mathcal{P} \subseteq \mathbb{R}^n$:

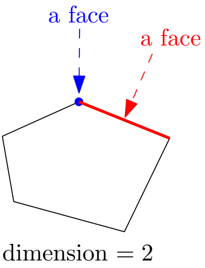
- The dimension of $\mathcal{P}$ is n minus the maximum number of linearly-independent equalities satisfied by all points in $\mathcal{P}$.
- Assume the linear inequality $a^{\mathrm{T}}x \leq b$ holds for every $x \in \mathcal{P}$, and some $x \in \mathcal{P}$ satisfies $a^{\mathrm{T}}x = b$. Then $\{x \in \mathcal{P} : a^{\mathrm{T}}x = b\}$ is said to be a face of $\mathcal{P}$.

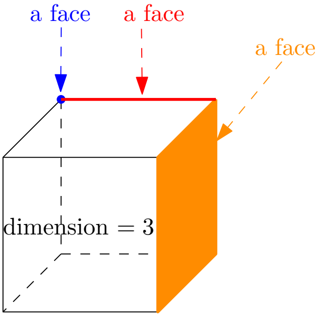




Given a polytope $\mathcal{P} \subseteq \mathbb{R}^n$:

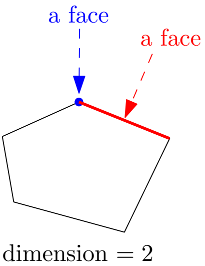
- The dimension of $\mathcal{P}$ is n minus the maximum number of linearly-independent equalities satisfied by all points in $\mathcal{P}$.
- Assume the linear inequality $a^{\mathrm{T}}x \leq b$ holds for every $x \in \mathcal{P}$, and some $x \in \mathcal{P}$ satisfies $a^{\mathrm{T}}x = b$. Then $\{x \in \mathcal{P} : a^{\mathrm{T}}x = b\}$ is said to be a face of $\mathcal{P}$.

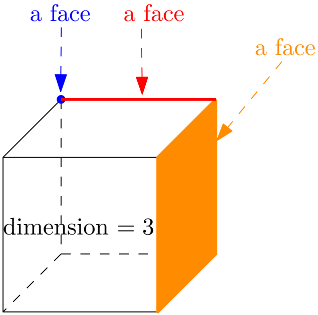




Given a polytope $\mathcal{P} \subseteq \mathbb{R}^n$:

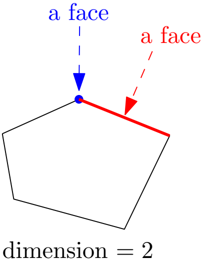
- The dimension of $\mathcal{P}$ is n minus the maximum number of linearly-independent equalities satisfied by all points in $\mathcal{P}$.
- Assume the linear inequality $a^{\mathrm{T}}x \le b$ holds for every $x \in \mathcal{P}$, and some $x \in \mathcal{P}$ satisfies $a^{\mathrm{T}}x = b$. Then $\{x \in \mathcal{P} : a^{\mathrm{T}}x = b\}$ is said to be a face of $\mathcal{P}$.
- A face of $\mathcal{P}$ is also a polytope.

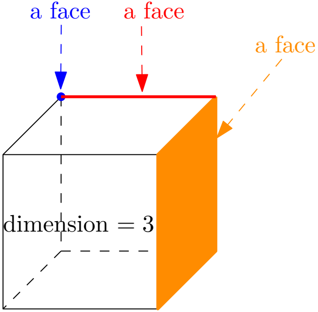




Given a polytope $\mathcal{P} \subseteq \mathbb{R}^n$:

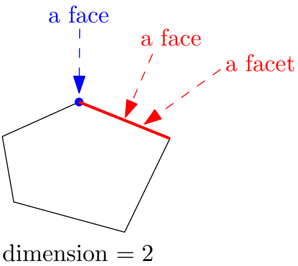
- The dimension of $\mathcal{P}$ is n minus the maximum number of linearly-independent equalities satisfied by all points in $\mathcal{P}$.
- Assume the linear inequality $a^{\mathrm{T}}x \leq b$ holds for every $x \in \mathcal{P}$, and some $x \in \mathcal{P}$ satisfies $a^{\mathrm{T}}x = b$. Then $\{x \in \mathcal{P} : a^{\mathrm{T}}x = b\}$ is said to be a face of $\mathcal{P}$.
- A face of $\mathcal{P}$ is also a polytope.
- Assume the dimension of $\mathcal{P}$ is d. Then a face of $\mathcal{P}$ of dimension $d-1$ is said to be a facet of $\mathcal{P}$.

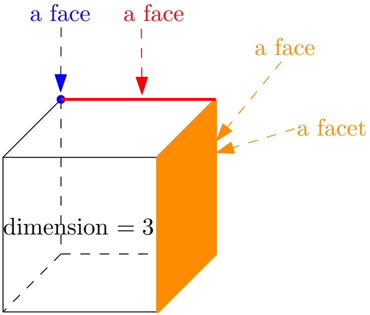




Given a polytope $\mathcal{P} \subseteq \mathbb{R}^n$:

- The dimension of $\mathcal{P}$ is n minus the maximum number of linearly-independent equalities satisfied by all points in $\mathcal{P}$.
- Assume the linear inequality $a^{\mathrm{T}}x \leq b$ holds for every $x \in \mathcal{P}$, and some $x \in \mathcal{P}$ satisfies $a^{\mathrm{T}}x = b$. Then $\{x \in \mathcal{P} : a^{\mathrm{T}}x = b\}$ is said to be a face of $\mathcal{P}$.
- A face of $\mathcal{P}$ is also a polytope.
- Assume the dimension of $\mathcal{P}$ is d. Then a face of $\mathcal{P}$ of dimension $d-1$ is said to be a facet of $\mathcal{P}$.





Preliminaries

- x is a convex combination of $\{x^{(1)}, x^{(2)}, \cdots, x^{(t)}\}$ if the following condition holds: there exist $\lambda_1, \lambda_2, \cdots, \lambda_t \in [0, 1]$ such that

$$\lambda_1 + \lambda_2 + \cdots + \lambda_t = 1, \qquad \lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \cdots + \lambda_t x^{(t)} = x$$

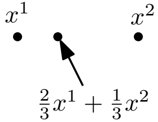
- x is a convex combination of $\{x^{(1)}, x^{(2)}, \cdots, x^{(t)}\}$ if the following condition holds: there exist $\lambda_1, \lambda_2, \cdots, \lambda_t \in [0, 1]$ such that

$$\lambda_1 + \lambda_2 + \cdots + \lambda_t = 1, \qquad \lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \cdots + \lambda_t x^{(t)} = x$$

Preliminaries

- x is a convex combination of $\{x^{(1)}, x^{(2)}, \cdots, x^{(t)}\}$ if the following condition holds: there exist $\lambda_1, \lambda_2, \cdots, \lambda_t \in [0, 1]$ such that

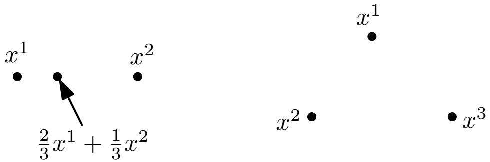
$$\lambda_1 + \lambda_2 + \cdots + \lambda_t = 1, \qquad \lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \cdots + \lambda_t x^{(t)} = x$$



Preliminaries

- x is a convex combination of $\{x^{(1)}, x^{(2)}, \cdots, x^{(t)}\}$ if the following condition holds: there exist $\lambda_1, \lambda_2, \cdots, \lambda_t \in [0, 1]$ such that

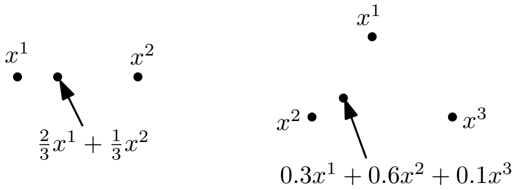
$$\lambda_1 + \lambda_2 + \cdots + \lambda_t = 1, \qquad \lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \cdots + \lambda_t x^{(t)} = x$$



Preliminaries

- x is a convex combination of $\{x^{(1)}, x^{(2)}, \cdots, x^{(t)}\}$ if the following condition holds: there exist $\lambda_1, \lambda_2, \cdots, \lambda_t \in [0, 1]$ such that

$$\lambda_1 + \lambda_2 + \cdots + \lambda_t = 1, \qquad \lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \cdots + \lambda_t x^{(t)} = x$$

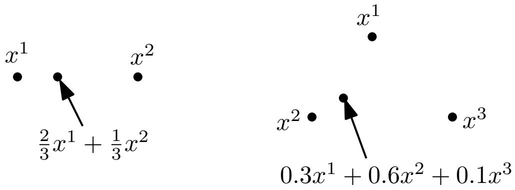


Preliminaries

- x is a convex combination of $\{x^{(1)}, x^{(2)}, \cdots, x^{(t)}\}$ if the following condition holds: there exist $\lambda_1, \lambda_2, \cdots, \lambda_t \in [0,1]$ such that

$$\lambda_1 + \lambda_2 + \cdots + \lambda_t = 1, \qquad \lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \cdots + \lambda_t x^{(t)} = x$$

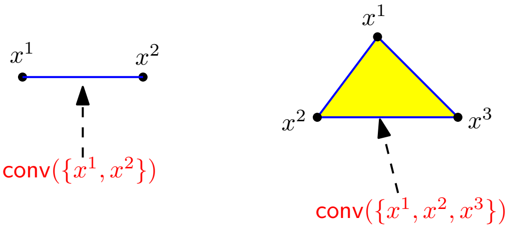
- the convex hull of a set of S of points in $\mathbb{R}^n$, denoted as $\text{conv}(S)$, is the set of convex combinations of S



- x is a convex combination of $\{x^{(1)}, x^{(2)}, \cdots, x^{(t)}\}$ if the following condition holds: there exist $\lambda_1, \lambda_2, \cdots, \lambda_t \in [0, 1]$ such that

$$\lambda_1 + \lambda_2 + \cdots + \lambda_t = 1, \qquad \lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \cdots + \lambda_t x^{(t)} = x$$

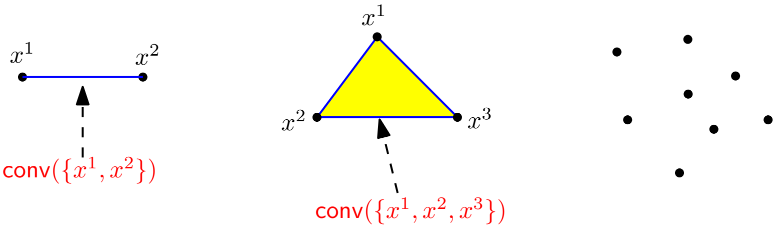
- the convex hull of a set of S of points in $\mathbb{R}^n$, denoted as conv(S), is the set of convex combinations of S



- x is a convex combination of $\{x^{(1)}, x^{(2)}, \cdots, x^{(t)}\}$ if the following condition holds: there exist $\lambda_1, \lambda_2, \cdots, \lambda_t \in [0, 1]$ such that

$$\lambda_1 + \lambda_2 + \cdots + \lambda_t = 1, \qquad \lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \cdots + \lambda_t x^{(t)} = x$$

- the convex hull of a set of S of points in $\mathbb{R}^n$, denoted as conv(S), is the set of convex combinations of S

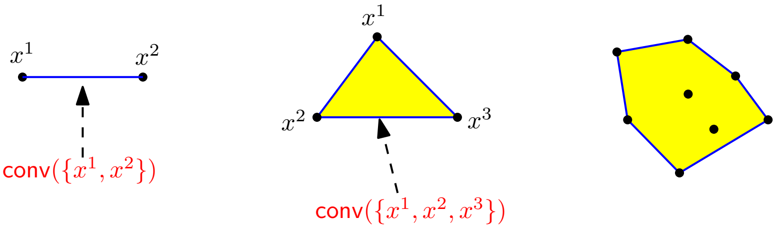


Preliminaries

- x is a convex combination of $\{x^{(1)}, x^{(2)}, \cdots, x^{(t)}\}$ if the following condition holds: there exist $\lambda_1, \lambda_2, \cdots, \lambda_t \in [0,1]$ such that

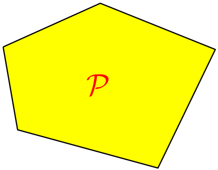
$$\lambda_1 + \lambda_2 + \cdots + \lambda_t = 1, \qquad \lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \cdots + \lambda_t x^{(t)} = x$$

- the convex hull of a set of S of points in $\mathbb{R}^n$, denoted as conv(S), is the set of convex combinations of S

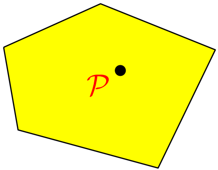


Terminology and Preliminaries

- let $\mathcal{P}$ be polytope, $x \in \mathcal{P}$. If there are no other points $x', x'' \in \mathcal{P}$ such that x is a convex combination of x' and x'', then x is called a vertex/extreme point of $\mathcal{P}$

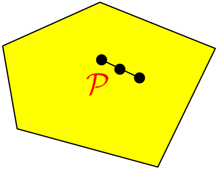


- let $\mathcal{P}$ be polytope, $x \in \mathcal{P}$. If there are no other points $x', x'' \in \mathcal{P}$ such that x is a convex combination of x' and x'', then x is called a vertex/extreme point of $\mathcal{P}$



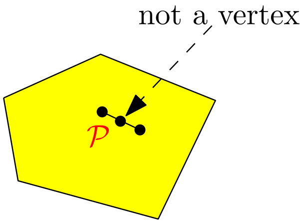
Terminology and Preliminaries

- let $\mathcal{P}$ be polytope, $x \in \mathcal{P}$. If there are no other points $x', x'' \in \mathcal{P}$ such that x is a convex combination of x' and x'', then x is called a vertex/extreme point of $\mathcal{P}$



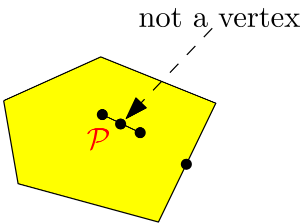
Terminology and Preliminaries

- let $\mathcal{P}$ be polytope, $x \in \mathcal{P}$. If there are no other points $x', x'' \in \mathcal{P}$ such that x is a convex combination of x' and x'', then x is called a vertex/extreme point of $\mathcal{P}$



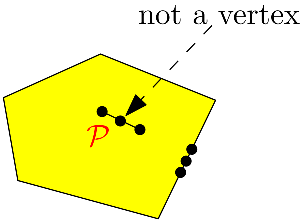
Terminology and Preliminaries

- let $\mathcal{P}$ be polytope, $x \in \mathcal{P}$. If there are no other points $x', x'' \in \mathcal{P}$ such that x is a convex combination of x' and x'', then x is called a vertex/extreme point of $\mathcal{P}$



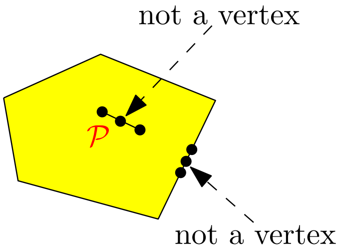
Terminology and Preliminaries

- let $\mathcal{P}$ be polytope, $x \in \mathcal{P}$. If there are no other points $x', x'' \in \mathcal{P}$ such that x is a convex combination of x' and x'', then x is called a vertex/extreme point of $\mathcal{P}$



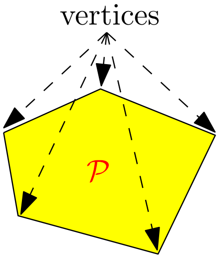
Terminology and Preliminaries

- let $\mathcal{P}$ be polytope, $x \in \mathcal{P}$. If there are no other points $x', x'' \in \mathcal{P}$ such that x is a convex combination of x' and x'', then x is called a vertex/extreme point of $\mathcal{P}$



Terminology and Preliminaries

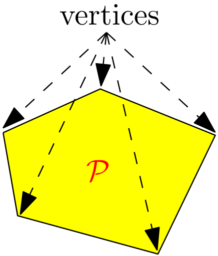
- let $\mathcal{P}$ be polytope, $x \in \mathcal{P}$. If there are no other points $x', x'' \in \mathcal{P}$ such that x is a convex combination of x' and x'', then x is called a vertex/extreme point of $\mathcal{P}$



Terminology and Preliminaries

- let $\mathcal{P}$ be polytope, $x \in \mathcal{P}$. If there are no other points $x', x'' \in \mathcal{P}$ such that x is a convex combination of x' and x'', then x is called a vertex/extreme point of $\mathcal{P}$

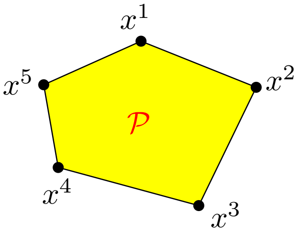
Lemma A polytope has finite number of vertices, and it is the convex hull of the vertices.



Terminology and Preliminaries

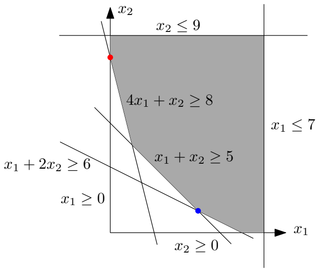
- let $\mathcal{P}$ be polytope, $x \in \mathcal{P}$. If there are no other points $x', x'' \in \mathcal{P}$ such that x is a convex combination of x' and x'', then x is called a vertex/extreme point of $\mathcal{P}$

Lemma A polytope has finite number of vertices, and it is the convex hull of the vertices.

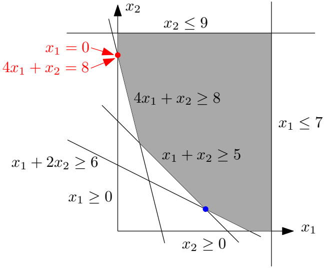


$$\mathcal{P} = \text{conv}(\{x^1, x^2, x^3, x^4, x^5\})$$

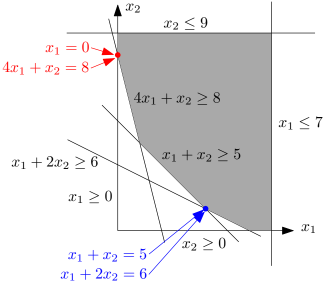
Lemma Let $x \in \mathbb{R}^n$ be a vertex of a polytope. Then, there are n constraints in the definition of the polytope, such that x is the unique solution to the linear system obtained from the n constraints by replacing inequalities to equalities.



Lemma Let $x \in \mathbb{R}^n$ be a vertex of a polytope. Then, there are n constraints in the definition of the polytope, such that x is the unique solution to the linear system obtained from the n constraints by replacing inequalities to equalities.

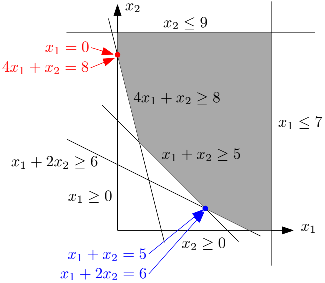


Lemma Let $x \in \mathbb{R}^n$ be a vertex of a polytope. Then, there are n constraints in the definition of the polytope, such that x is the unique solution to the linear system obtained from the n constraints by replacing inequalities to equalities.



Terminology and Preliminaries

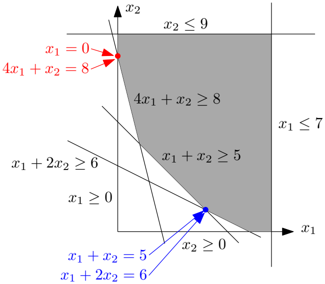
Lemma Let $x \in \mathbb{R}^n$ be a vertex of a polytope. Then, there are n constraints in the definition of the polytope, such that x is the unique solution to the linear system obtained from the n constraints by replacing inequalities to equalities.



Lemma If the feasible region of a linear program is a polytope, then the opimum value can be attained at some vertex of the polytope.

Terminology and Preliminaries

Lemma Let $x \in \mathbb{R}^n$ be a vertex of a polytope. Then, there are n constraints in the definition of the polytope, such that x is the unique solution to the linear system obtained from the n constraints by replacing inequalities to equalities.



Lemma If the feasible region of a linear program is a polytope, then the opimum value can be attained at some vertex of the polytope.

Special cases (for minimization linear programs):

- if feasible region is empty, then its value is ∞
- if the feasible region is unbounded, then its value can be $-\infty$

Outline

1. Linear Programming
   - Introduction
   - Methods for Solving Linear Programs
2. Polytope with Polynomial Number of Facets
   - Bipartite Matching Polytope
   - Polytopes with Totally Unimodular Coefficient Matrices
3. Polytopes with Efficient Separation Oracles
   - s-t Cut Polytope
   - Spanning Tree Polytope
   - General Graph (Perfect) Matching Polytope
4. Matroid, Matroid Basis and Matroid Intersection Polytopes *
   - Preliminaries on Matroid Theory
   - Matroid Polytope
   - Matroid Basis and Matroid Intersection Polytope

Algorithms for Linear Programming

algorithm	running time	practice
Simplex Method	exponential time	fast
Ellipsoid Method	polynomial time	slow
Interior Point Method	polynomial time	fast

Simplex Method

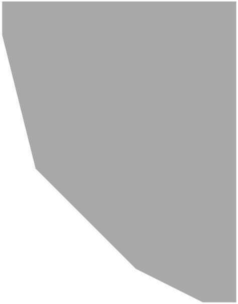
- [Dantzig, 1946]

- move from one vertex to another, so as to improve the objective
- repeat until we reach an optimum vertex

Simplex Method

- [Dantzig, 1946]

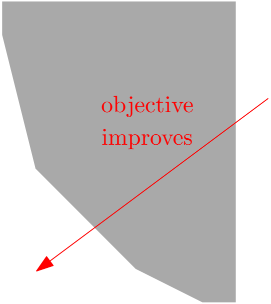
- move from one vertex to another, so as to improve the objective
- repeat until we reach an optimum vertex



Simplex Method

- [Dantzig, 1946]

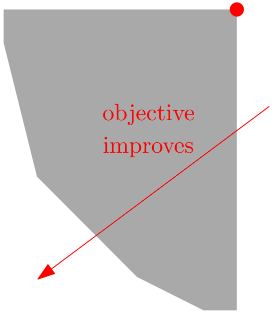
- move from one vertex to another, so as to improve the objective
- repeat until we reach an optimum vertex



Simplex Method

- [Dantzig, 1946]

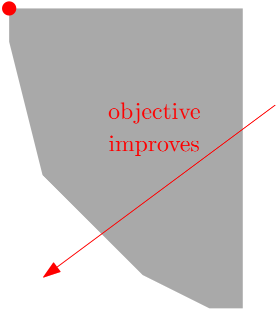
- move from one vertex to another, so as to improve the objective
- repeat until we reach an optimum vertex



Simplex Method

- [Dantzig, 1946]

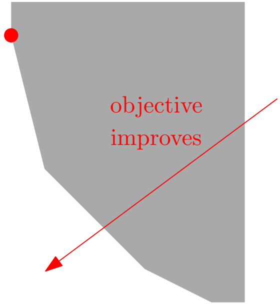
- move from one vertex to another, so as to improve the objective
- repeat until we reach an optimum vertex



Simplex Method

- [Dantzig, 1946]

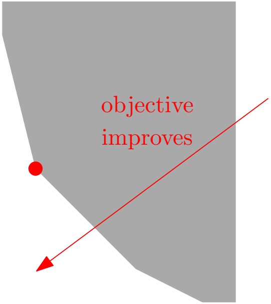
- move from one vertex to another, so as to improve the objective
- repeat until we reach an optimum vertex



Simplex Method

- [Dantzig, 1946]

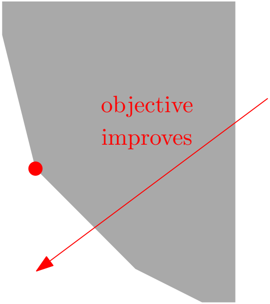
- move from one vertex to another, so as to improve the objective
- repeat until we reach an optimum vertex



Simplex Method

- [Dantzig, 1946]

- move from one vertex to another, so as to improve the objective
- repeat until we reach an optimum vertex

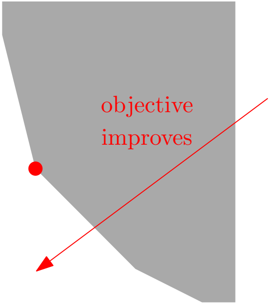


- the number of iterations might be expoentially large; but algorithm runs fast in practice

Simplex Method

- [Dantzig, 1946]

- move from one vertex to another, so as to improve the objective
- repeat until we reach an optimum vertex



- the number of iterations might be expoentially large; but algorithm runs fast in practice
- [Spielman-Teng,2002]: smoothed analysis

Interior Point Method

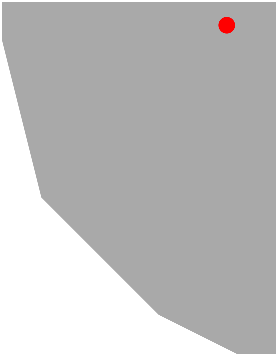
- [Karmarkar, 1984]

- keep the solution inside the polytope
- design penalty function so that the solution is not too close to the boundary
- the final solution will be arbitrarily close to the optimum solution

Interior Point Method

- [Karmarkar, 1984]

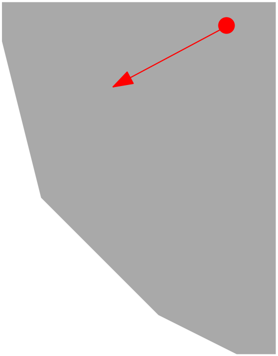
- keep the solution inside the polytope
- design penalty function so that the solution is not too close to the boundary
- the final solution will be arbitrarily close to the optimum solution



Interior Point Method

- [Karmarkar, 1984]

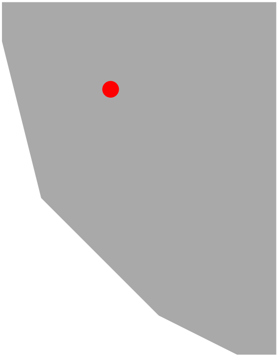
- keep the solution inside the polytope
- design penalty function so that the solution is not too close to the boundary
- the final solution will be arbitrarily close to the optimum solution



Interior Point Method

- [Karmarkar, 1984]

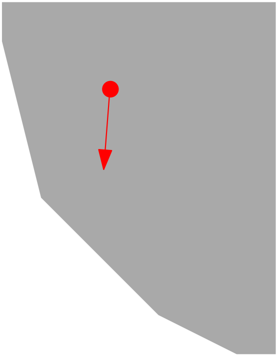
- keep the solution inside the polytope
- design penalty function so that the solution is not too close to the boundary
- the final solution will be arbitrarily close to the optimum solution



Interior Point Method

- [Karmarkar, 1984]

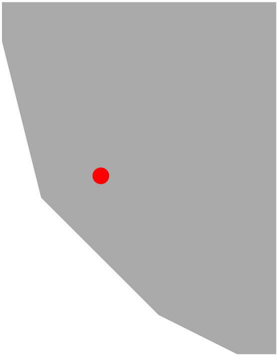
- keep the solution inside the polytope
- design penalty function so that the solution is not too close to the boundary
- the final solution will be arbitrarily close to the optimum solution



Interior Point Method

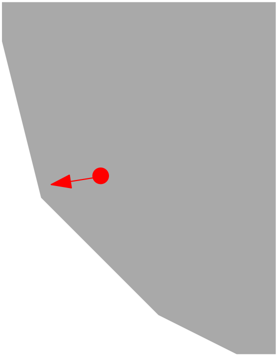
- [Karmarkar, 1984]

- keep the solution inside the polytope
- design penalty function so that the solution is not too close to the boundary
- the final solution will be arbitrarily close to the optimum solution



Interior Point Method

- [Karmarkar, 1984]

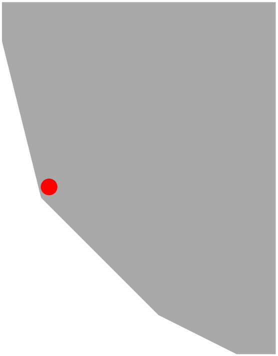


- keep the solution inside the polytope
- design penalty function so that the solution is not too close to the boundary
- the final solution will be arbitrarily close to the optimum solution

Interior Point Method

- [Karmarkar, 1984]

- keep the solution inside the polytope
- design penalty function so that the solution is not too close to the boundary
- the final solution will be arbitrarily close to the optimum solution

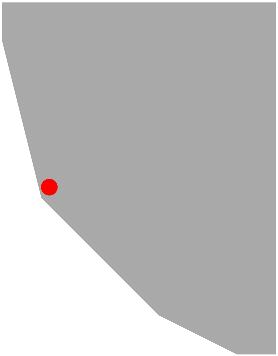


Interior Point Method

- [Karmarkar, 1984]

- keep the solution inside the polytope
- design penalty function so that the solution is not too close to the boundary
- the final solution will be arbitrarily close to the optimum solution

- polynomial time



Ellipsoid Method

- [Khachiyan, 1979]

Ellipsoid Method

- [Khachiyan, 1979]
- used to decide if the feasible region is empty or not

Ellipsoid Method

- [Khachiyan, 1979]
- used to decide if the feasible region is empty or not

- maintain an ellipsoid that contains the feasible region

Ellipsoid Method

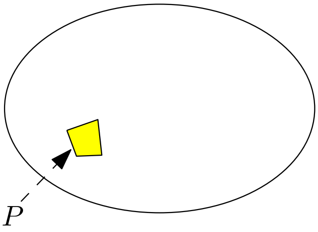
- [Khachiyan, 1979]
- used to decide if the feasible region is empty or not

- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat

Ellipsoid Method

- [Khachiyan, 1979]
- used to decide if the feasible region is empty or not

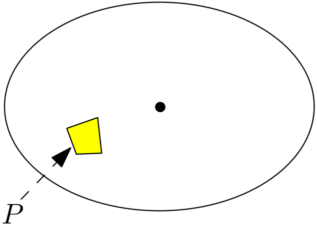
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



Ellipsoid Method

- [Khachiyan, 1979]
- used to decide if the feasible region is empty or not

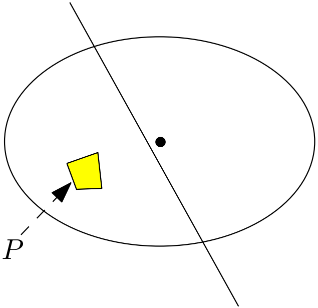
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



Ellipsoid Method

- [Khachiyan, 1979]
- used to decide if the feasible region is empty or not

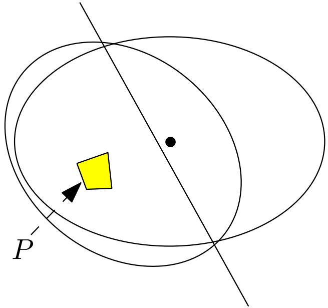
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



Ellipsoid Method

- [Khachiyan, 1979]
- used to decide if the feasible region is empty or not

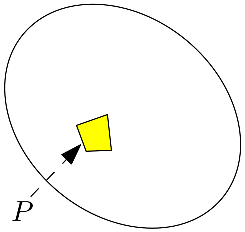
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



Ellipsoid Method

- [Khachiyan, 1979]
- used to decide if the feasible region is empty or not

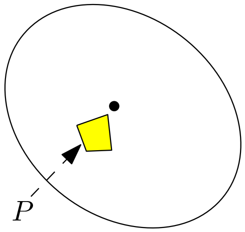
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



Ellipsoid Method

- [Khachiyan, 1979]
- used to decide if the feasible region is empty or not

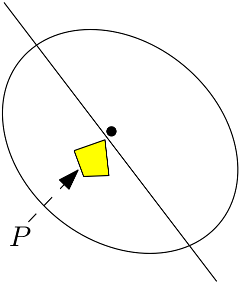
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



Ellipsoid Method

- [Khachiyan, 1979]
- used to decide if the feasible region is empty or not

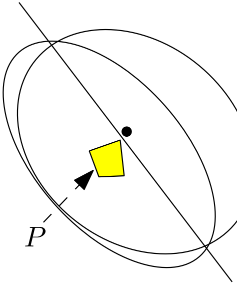
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



Ellipsoid Method

- [Khachiyan, 1979]
- used to decide if the feasible region is empty or not

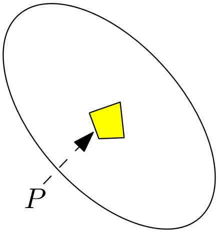
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



Ellipsoid Method

- [Khachiyan, 1979]
- used to decide if the feasible region is empty or not

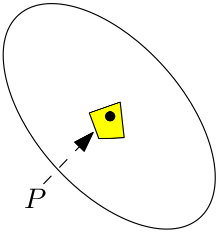
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



Ellipsoid Method

- [Khachiyan, 1979]
- used to decide if the feasible region is empty or not

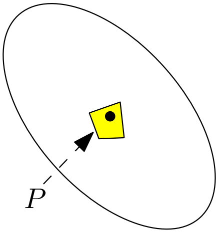
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



Ellipsoid Method

- [Khachiyan, 1979]
- used to decide if the feasible region is empty or not

- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



- polynomial time, but impractical

Q: The exact running time of these algorithms?

Q: The exact running time of these algorithms?

- it depends on many parameters: #variables, #constraints, #(non-zero coefficients), magnitude of integers
- precision issue

Q: The exact running time of these algorithms?

- it depends on many parameters: #variables, #constraints, #(non-zero coefficients), magnitude of integers
- precision issue

Open Problem

Can linear programming be solved in strongly polynomial time algorithm?

Applications of Linear Programming

- domain: computer science, mathematics, operations research, economics
- types of problems: transportation, scheduling, clustering, network routing, resource allocation, facility location

Applications of Linear Programming

- domain: computer science, mathematics, operations research, economics
- types of problems: transportation, scheduling, clustering, network routing, resource allocation, facility location

Research Directions

- polynomial time exact algorithm
- polynomial time approximation algorithm
- sub-routines for the branch-and-bound metheod for integer programming
- other algorithmic models: online algorithm, distributed algorithms, dynamic algorithms, fast algorithms

Typical Combinatorial Optimizatiion Problem

Input: $[n]$: ground set

$\mathcal{S}$: feasible sets: a family of subsets of U, often implicitly given

$w_i, i \in [n]$: values/costs of elements

Output: the set $S \in \mathcal{S}$ with the minimum/maximum $w(S) := \sum_{i \in S} w_i$

Def. For any $S \subseteq [n]$, we use $\chi^S \in \{0,1\}^{[n]}$ to denote the indicator vector for S:

$$\chi_i^S = \begin{cases} 0 & \text{if } i \notin S \\ 1 & \text{if } i \in S \end{cases}$$

polytope of interest: $\mathcal{P} = \text{conv}\left(\{\chi^S : S \in \mathcal{S}\}\right)$

Examples

Bipartite Matching Polytope

- Given bipartite graph $G = (L \cup R, E)$
- $\mathcal{P}_{\mathrm{BM}} := \mathrm{conv}\big(\{\chi^M : M \text{ is a matching in } G\}\big)$

General Matching Polytope

- Given a graph $G = (V, E)$
- $\mathcal{P}_{\mathrm{GM}} := \mathrm{conv}\left(\left\{\chi^M : M \subseteq E \text{ is a matching in } G\right\}\right)$

Spanning Tree Polytope

- Given a connected graph $G = (V, E)$
- $\mathcal{P}_{\mathrm{ST}} := \mathrm{conv}\left(\left\{\chi^T : T \subseteq E \text{ is a spanning tree of } G\right\}\right)$

Examples

Travelling Salesman Problem (TSP) Polytope

- Given the complete graph $G = (V, \binom{V}{2})$
- $\mathcal{P}_{\mathrm{TSP}} :=$ conv$(\{\chi^S, S \subseteq \binom{V}{2}$ is a TSP tour of V$\})$

polytope of interest: $\mathcal{P} = \mathsf{conv}\big(\{\chi^S : S \in \mathcal{S}\}\big)$

polytope of interest: $\mathcal{P} = \text{conv}\big(\{\chi^S : S \in \mathcal{S}\}\big)$

- Mechanic description of $\mathcal{P}$:

$$\sum_{i \in S} w_i x_i \leq \max_{S \in \mathcal{S}} \sum_{i \in S} w_i \qquad \forall w \in \mathbb{R}^{[n]}$$

polytope of interest: $\mathcal{P} = \text{conv}\big(\{\chi^S : S \in \mathcal{S}\}\big)$

- Mechanic description of $\mathcal{P}$:

$$\sum_{i \in S} w_i x_i \leq \max_{S \in \mathcal{S}} \sum_{i \in S} w_i \qquad \forall w \in \mathbb{R}^{[n]}$$

- However, the description is often useless; many constraints are redundant
- It is often interesting and important to find the facet-defining constraints; those are the constraints that can not be removed

polytope of interest: $\mathcal{P} = \text{conv}\big(\{\chi^S : S \in \mathcal{S}\}\big)$

- Mechanic description of $\mathcal{P}$:

$$\sum_{i \in S} w_i x_i \leq \max_{S \in \mathcal{S}} \sum_{i \in S} w_i \qquad \forall w \in \mathbb{R}^{[n]}$$

- However, the description is often useless; many constraints are redundant
- It is often interesting and important to find the facet-defining constraints; those are the constraints that can not be removed

1. In some cases, $\mathcal{P}$ has polynomial number of facets
2. In some cases, $\mathcal{P}$ has exponential number of facets, but has an efficient separation oracle.
3. In some cases, $\mathcal{P}$ does not have an efficient separation oracle, unless P = NP.

polytope of interest: $\mathcal{P} = \text{conv}\big(\{\chi^S : S \in \mathcal{S}\}\big)$

Def. A polytope $\mathcal{P} \subseteq [0,1]^n$ is said to be integral, if all vertices of $\mathcal{P}$ are in $\{0,1\}^n$.

Lemma For a $\mathcal{Q} \subseteq [0,1]^n$, if $\mathcal{Q} \cap \{0,1\}^n = \{\chi^S : S \in \mathcal{S}\}$ and $\mathcal{Q}$ is integral, then $\mathcal{Q} = \mathcal{P}$.

polytope of interest: $\mathcal{P} = \text{conv}\big(\{\chi^S : S \in \mathcal{S}\}\big)$

Def. A polytope $\mathcal{P} \subseteq [0,1]^n$ is said to be integral, if all vertices of $\mathcal{P}$ are in $\{0,1\}^n$.

Lemma For a $\mathcal{Q} \subseteq [0,1]^n$, if $\mathcal{Q} \cap \{0,1\}^n = \{\chi^S : S \in \mathcal{S}\}$ and $\mathcal{Q}$ is integral, then $\mathcal{Q} = \mathcal{P}$.

Proof.

- $\mathcal{P} \subseteq \mathcal{Q}$, as every vertex of $\mathcal{P}$ is χ^S for some $S \in \mathcal{S}$, and $\chi^S \in \mathcal{Q}$.
- $\mathcal{Q} \subseteq \mathcal{P}$: take some vertex x of $\mathcal{Q}$
- $\mathcal{Q}$ is integral $\implies$ x is integral $\implies$ $x = \chi^S$ for some $S \subseteq [n]$
- As $\mathcal{Q} \cap \{0,1\}^n = \{\chi^S : S \in \mathcal{S}\}$, $x = \chi^S$ for some $S \in \mathcal{S}$
- $x \in \mathcal{P}$ □

polytope of interest: $\mathcal{P} = \text{conv}\big(\{\chi^S : S \in \mathcal{S}\}\big)$

Lemma For a $\mathcal{Q} \subseteq [0,1]^n$, if $\mathcal{Q} \cap \{0,1\}^n = \{\chi^S : S \in \mathcal{S}\}$ and $\mathcal{Q}$ is integral, then $\mathcal{Q} = \mathcal{P}$.

polytope of interest: $\mathcal{P} = \text{conv}\big(\{\chi^S : S \in \mathcal{S}\}\big)$

Lemma For a $\mathcal{Q} \subseteq [0,1]^n$, if $\mathcal{Q} \cap \{0,1\}^n = \{\chi^S : S \in \mathcal{S}\}$ and $\mathcal{Q}$ is integral, then $\mathcal{Q} = \mathcal{P}$.

- Often, it is easy to guarantee $\mathcal{Q} \cap \{0,1\}^n = \{\chi^S : S \in \mathcal{S}\}$
- The linear program that defines such a $\mathcal{Q}$ is often called a LP relaxation for the problem.

polytope of interest: $\mathcal{P} = \text{conv}\big(\{\chi^S : S \in \mathcal{S}\}\big)$

Lemma For a $\mathcal{Q} \subseteq [0,1]^n$, if $\mathcal{Q} \cap \{0,1\}^n = \{\chi^S : S \in \mathcal{S}\}$ and $\mathcal{Q}$ is integral, then $\mathcal{Q} = \mathcal{P}$.

- Often, it is easy to guarantee $\mathcal{Q} \cap \{0,1\}^n = \{\chi^S : S \in \mathcal{S}\}$
- The linear program that defines such a $\mathcal{Q}$ is often called a LP relaxation for the problem.
- The harder part is often to prove that $\mathcal{Q}$ is integral.

Outline

1. Linear Programming
   - Introduction
   - Methods for Solving Linear Programs
2. Polytope with Polynomial Number of Facets
   - Bipartite Matching Polytope
   - Polytopes with Totally Unimodular Coefficient Matrices
3. Polytopes with Efficient Separation Oracles
   - s-t Cut Polytope
   - Spanning Tree Polytope
   - General Graph (Perfect) Matching Polytope
4. Matroid, Matroid Basis and Matroid Intersection Polytopes *
   - Preliminaries on Matroid Theory
   - Matroid Polytope
   - Matroid Basis and Matroid Intersection Polytope

Outline

1 Linear Programming
- Introduction
- Methods for Solving Linear Programs

2 Polytope with Polynomial Number of Facets
- Bipartite Matching Polytope
- Polytopes with Totally Unimodular Coefficient Matrices

3 Polytopes with Efficient Separation Oracles
- s-t Cut Polytope
- Spanning Tree Polytope
- General Graph (Perfect) Matching Polytope

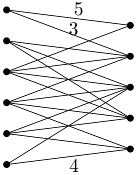
4 Matroid, Matroid Basis and Matroid Intersection Polytopes *
- Preliminaries on Matroid Theory
- Matroid Polytope
- Matroid Basis and Matroid Intersection Polytope

Bipartite Matching Polytope

Maximum Weight Bipartite Matching

Input: bipartite graph $G = (L \uplus R, E)$
edge weights $w \in \mathbb{Z}_{>0}^E$

Output: a matching $M \subseteq E$ so as to maximize $\sum_{e \in M} w_e$



Bipartite Matching Polytope

- Given bipartite graph $G = (L \cup R, E)$
- $\mathcal{P}_{\text{BM}} := \text{conv}\big(\{\chi^M : M \text{ is a matching in } G\}\big)$

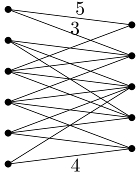
Bipartite Matching Polytope

Maximum Weight Bipartite Matching

Input: bipartite graph $G = (L \uplus R, E)$

edge weights $w \in \mathbb{Z}_{>0}^E$

Output: a matching $M \subseteq E$ so as to maximize $\sum_{e \in M} w_e$



Bipartite Matching Polytope

- Given bipartite graph $G = (L \cup R, E)$
- $\mathcal{P}_{\mathrm{BM}} := \mathrm{conv}\big(\{\chi^M : M \text{ is a matching in } G\}\big)$

Theorem $\mathcal{P}_{\mathrm{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \le 1, \forall v \in L \cup R; \qquad x_e \ge 0, \forall e \in E.$$

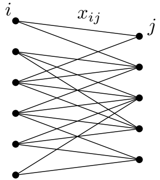
Bipartite Matching Polytope

Maximum Weight Bipartite Matching

Input: bipartite graph $G = (L \uplus R, E)$

edge weights $w \in \mathbb{Z}_{>0}^E$

Output: a matching $M \subseteq E$ so as to maximize $\sum_{e \in M} w_e$



Bipartite Matching Polytope

- Given bipartite graph $G = (L \cup R, E)$
- $\mathcal{P}_{\mathrm{BM}} := \mathrm{conv}\big(\{\chi^M : M \text{ is a matching in } G\}\big)$

Theorem $\mathcal{P}_{\mathrm{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \le 1, \forall v \in L \cup R; \qquad x_e \ge 0, \forall e \in E.$$

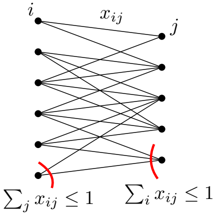
Bipartite Matching Polytope

Maximum Weight Bipartite Matching

Input: bipartite graph $G = (L \uplus R, E)$

edge weights $w \in \mathbb{Z}_{>0}^E$

Output: a matching $M \subseteq E$ so as to maximize $\sum_{e \in M} w_e$



Bipartite Matching Polytope

- Given bipartite graph $G = (L \cup R, E)$
- $\mathcal{P}_{\text{BM}} := \text{conv}\big(\{\chi^M : M \text{ is a matching in } G\}\big)$

Theorem $\mathcal{P}_{\text{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \leq 1, \forall v \in L \cup R; \qquad x_e \geq 0, \forall e \in E.$$

Theorem $\mathcal{P}_{\mathrm{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \leq 1, \forall v \in L \cup R; \qquad x_e \geq 0, \forall e \in E.$$

Proof.

Theorem $\mathcal{P}_{\mathrm{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \leq 1, \forall v \in L \cup R; \qquad x_e \geq 0, \forall e \in E.$$

Proof.

- take any x that satisfies the constraints

Theorem $\mathcal{P}_{\text{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \leq 1, \forall v \in L \cup R; \qquad x_e \geq 0, \forall e \in E.$$

Proof.

- take any x that satisfies the constraints
- prove: x non integral $\implies$ x non-vertex

Theorem $\mathcal{P}_{\text{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \leq 1, \forall v \in L \cup R; \qquad x_e \geq 0, \forall e \in E.$$

Proof.

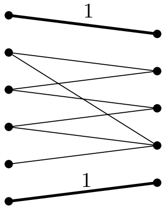
- take any x that satisfies the constraints
- prove: x non integral $\implies$ x non-vertex
- find $x', x'' \in \mathcal{P}$: $x' \neq x'', x = \frac{1}{2}(x' + x'')$

Theorem $\mathcal{P}_{\text{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \le 1, \forall v \in L \cup R; \qquad x_e \ge 0, \forall e \in E.$$

Proof.

- take any x that satisfies the constraints
- prove: x non integral $\implies$ x non-vertex
- find $x', x'' \in \mathcal{P}$: $x' \neq x'', x = \frac{1}{2}(x' + x'')$
- case 1: fractional edges contain a cycle

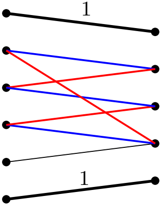


Theorem $\mathcal{P}_{\text{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \leq 1, \forall v \in L \cup R; \qquad x_e \geq 0, \forall e \in E.$$

Proof.

- take any x that satisfies the constraints
- prove: x non integral $\implies$ x non-vertex
- find $x', x'' \in \mathcal{P}$: $x' \neq x'', x = \frac{1}{2}(x' + x'')$
- case 1: fractional edges contain a cycle
  - color edges in cycle blue and red

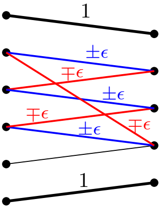


Theorem $\mathcal{P}_{\text{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \leq 1, \forall v \in L \cup R; \qquad x_e \geq 0, \forall e \in E.$$

Proof.

- take any x that satisfies the constraints
- prove: x non integral $\implies$ x non-vertex
- find $x', x'' \in \mathcal{P}$: $x' \neq x'', x = \frac{1}{2}(x' + x'')$
- case 1: fractional edges contain a cycle
  - color edges in cycle blue and red
  - x': $+\epsilon$ for blue edges, $-\epsilon$ for red edges
  - x'': $-\epsilon$ for blue edges, $+\epsilon$ for red edges

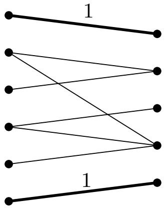


Theorem $\mathcal{P}_{\mathrm{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \le 1, \forall v \in L \cup R; \qquad x_e \ge 0, \forall e \in E.$$

Proof.

- take any x that satisfies the constraints
- prove: x non integral $\implies$ x non-vertex
- find $x', x'' \in \mathcal{P}$: $x' \neq x'', x = \frac{1}{2}(x' + x'')$
- case 1: fractional edges contain a cycle
  - color edges in cycle blue and red
  - x': $+\epsilon$ for blue edges, $-\epsilon$ for red edges
  - x'': $-\epsilon$ for blue edges, $+\epsilon$ for red edges
- case 2: fractional edges form a forest

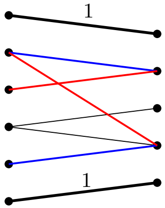


Theorem $\mathcal{P}_{\text{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \le 1, \forall v \in L \cup R; \qquad x_e \ge 0, \forall e \in E.$$

Proof.

- take any x that satisfies the constraints
- prove: x non integral $\implies$ x non-vertex
- find $x', x'' \in \mathcal{P}$: $x' \neq x'', x = \frac{1}{2}(x' + x'')$
- case 1: fractional edges contain a cycle
  - color edges in cycle blue and red
  - x': $+\epsilon$ for blue edges, $-\epsilon$ for red edges
  - x'': $-\epsilon$ for blue edges, $+\epsilon$ for red edges
- case 2: fractional edges form a forest
  - color edges in leaf-leaf path blue and red

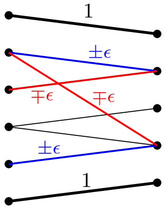


Theorem $\mathcal{P}_{\mathrm{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \le 1, \forall v \in L \cup R; \qquad x_e \ge 0, \forall e \in E.$$

Proof.

- take any x that satisfies the constraints
- prove: x non integral $\implies$ x non-vertex
- find $x', x'' \in \mathcal{P}$: $x' \neq x'', x = \frac{1}{2}(x' + x'')$
- case 1: fractional edges contain a cycle
  - color edges in cycle blue and red
  - x': $+\epsilon$ for blue edges, $-\epsilon$ for red edges
  - x'': $-\epsilon$ for blue edges, $+\epsilon$ for red edges
- case 2: fractional edges form a forest
  - color edges in leaf-leaf path blue and red
  - x': $+\epsilon$ for blue edges, $-\epsilon$ for red edges
  - x'': $-\epsilon$ for blue edges, $+\epsilon$ for red edges □



Outline

1. Linear Programming
   - Introduction
   - Methods for Solving Linear Programs
2. Polytope with Polynomial Number of Facets
   - Bipartite Matching Polytope
   - Polytopes with Totally Unimodular Coefficient Matrices
3. Polytopes with Efficient Separation Oracles
   - s-t Cut Polytope
   - Spanning Tree Polytope
   - General Graph (Perfect) Matching Polytope
4. Matroid, Matroid Basis and Matroid Intersection Polytopes *
   - Preliminaries on Matroid Theory
   - Matroid Polytope
   - Matroid Basis and Matroid Intersection Polytope

Def. A matrix $A \in \mathbb{R}^{m \times n}$ is said to be totally unimodular (TUM), if every sub-square of A has determinant in $\{-1, 0, 1\}$.

Theorem If a polytope $\mathcal{P}$ is defined by $Ax \geq b, x \geq 0$ with a totally unimodular matrix A and integral b, then $\mathcal{P}$ is integral.

Def. A matrix $A \in \mathbb{R}^{m \times n}$ is said to be totally unimodular (TUM), if every sub-square of A has determinant in $\{-1, 0, 1\}$.

Theorem If a polytope $\mathcal{P}$ is defined by $Ax \geq b, x \geq 0$ with a totally unimodular matrix A and integral b, then $\mathcal{P}$ is integral.

Proof.

- Every vertex $x \in \mathcal{P}$ is the unique solution to the linear system (after permuting coordinates): $\begin{pmatrix} A' & 0 \\ 0 & I \end{pmatrix} x = \begin{pmatrix} b' \\ 0 \end{pmatrix}$, where
  - A' is a square submatrix of A with $\det(A') = \pm 1$, b' is a sub-vector of b,
  - and the rows for b' are the same as the rows for A'.

Def. A matrix $A \in \mathbb{R}^{m \times n}$ is said to be totally unimodular (TUM), if every sub-square of A has determinant in $\{-1, 0, 1\}$.

Theorem If a polytope $\mathcal{P}$ is defined by $Ax \geq b, x \geq 0$ with a totally unimodular matrix A and integral b, then $\mathcal{P}$ is integral.

Proof.

- Every vertex $x \in \mathcal{P}$ is the unique solution to the linear system (after permuting coordinates): $\begin{pmatrix} A' & 0 \\ 0 & I \end{pmatrix} x = \begin{pmatrix} b' \\ 0 \end{pmatrix}$, where
  - A' is a square submatrix of A with $\det(A') = \pm 1$, b' is a sub-vector of b,
  - and the rows for b' are the same as the rows for A'.
- Let $x = \begin{pmatrix} x^1 \\ x^2 \end{pmatrix}$, so that $A'x^1 = b'$ and $x^2 = 0$.

Def. A matrix $A \in \mathbb{R}^{m\times n}$ is said to be totally unimodular (TUM), if every sub-square of A has determinant in $\{-1, 0, 1\}$.

Theorem If a polytope $\mathcal{P}$ is defined by $Ax \geq b, x \geq 0$ with a totally unimodular matrix A and integral b, then $\mathcal{P}$ is integral.

Proof.

- Every vertex $x \in \mathcal{P}$ is the unique solution to the linear system (after permuting coordinates): $\begin{pmatrix} A' & 0 \\ 0 & I \end{pmatrix} x = \begin{pmatrix} b' \\ 0 \end{pmatrix}$, where
  - A' is a square submatrix of A with $\det(A') = \pm 1$, b' is a sub-vector of b,
  - and the rows for b' are the same as the rows for A'.
- Let $x = \begin{pmatrix} x^1 \\ x^2 \end{pmatrix}$, so that $A'x^1 = b'$ and $x^2 = 0$.
- Cramer's rule: $x^1_i = \frac{\det(A'_i|b)}{\det(A')}$ for every $i \implies x^1_i$ is integer

  $A'_i|b$: the matrix of A' with the i-th column replaced by b □

Example for the Proof

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \geq \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

Example for the Proof

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \geq \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

The following equation system may give a vertex:

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_3 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Example for the Proof

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_3 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Example for the Proof

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_3 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Equivalently, the vertex satisfies

$$\begin{pmatrix} a_{1,2} & a_{1,3} & 0 & 0 & 0 \\ a_{3,2} & a_{3,3} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_2 \\ x_3 \\ x_1 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_3 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Lemma Let $A' \in \{0, \pm 1\}^{n \times n}$ such that every row of A' contains at most one 1 and one -1. Then $\det(A') \in \{0, \pm 1\}$.

Proof.

Lemma Let $A' \in \{0, \pm 1\}^{n \times n}$ such that every row of A' contains at most one 1 and one -1. Then $\det(A') \in \{0, \pm 1\}$.

Proof.

- wlog assume every row of A' contains one 1 and one -1

Lemma Let $A' \in \{0, \pm 1\}^{n \times n}$ such that every row of A' contains at most one 1 and one -1. Then $\det(A') \in \{0, \pm 1\}$.

Proof.

- wlog assume every row of A' contains one 1 and one -1
  - otherwise, we can reduce the matrix

Lemma Let $A' \in \{0, \pm 1\}^{n \times n}$ such that every row of A' contains at most one 1 and one -1. Then $\det(A') \in \{0, \pm 1\}$.

Proof.

- wlog assume every row of A' contains one 1 and one -1
  - otherwise, we can reduce the matrix
- treat A' as a directed graph: columns $\equiv$ vertices, rows $\equiv$ arcs

Lemma Let $A' \in \{0, \pm 1\}^{n \times n}$ such that every row of A' contains at most one 1 and one -1. Then $\det(A') \in \{0, \pm 1\}$.

Proof.

- wlog assume every row of A' contains one 1 and one -1
  - otherwise, we can reduce the matrix
- treat A' as a directed graph: columns $\equiv$ vertices, rows $\equiv$ arcs
- #edges $=$ #vertices $\implies$ underlying undirected graph contains a cycle $\implies$ det$(A') = 0$ □

Lemma Let $A' \in \{0, \pm 1\}^{n \times n}$ such that every row of A' contains at most one 1 and one -1. Then $\det(A') \in \{0, \pm 1\}$.

Proof.

- wlog assume every row of A' contains one 1 and one -1
  - otherwise, we can reduce the matrix
- treat A' as a directed graph: columns $\equiv$ vertices, rows $\equiv$ arcs
- $\#$edges $=$ $\#$vertices $\implies$ underlying undirected graph contains a cycle $\implies$ det$(A') = 0$ □

Lemma Let $A \in \{0, \pm 1\}^{m \times n}$ such that every row of A contains at most one 1 and one -1. Then A is TUM.

Lemma Let $A' \in \{0, \pm 1\}^{n \times n}$ such that every row of A' contains at most one 1 and one -1. Then $\det(A') \in \{0, \pm 1\}$.

Proof.

- wlog assume every row of A' contains one 1 and one -1
  - otherwise, we can reduce the matrix
- treat A' as a directed graph: columns $\equiv$ vertices, rows $\equiv$ arcs
- #edges $=$ #vertices $\implies$ underlying undirected graph contains a cycle $\implies$ det$(A') = 0$ □

Lemma Let $A \in \{0, \pm 1\}^{m \times n}$ such that every row of A contains at most one 1 and one -1. Then A is TUM.

Coro. In the LP for s-t network flow problem with integer capacities, every vertex solution to the LP is integral.

Example for the Proof

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$

Example for the Proof

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \color{red}{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$

Example for the Proof

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$

Example for the Proof

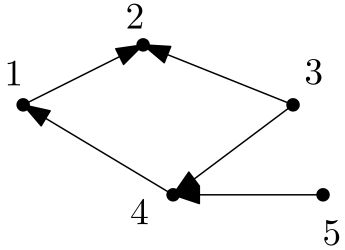
$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \color{red}{1} & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$

Example for the Proof

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 & 0 \end{pmatrix}$$

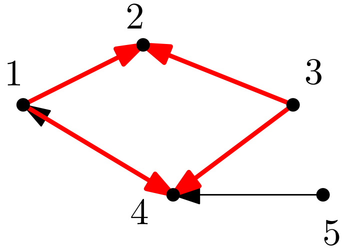
Example for the Proof

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 & 0 \end{pmatrix}$$



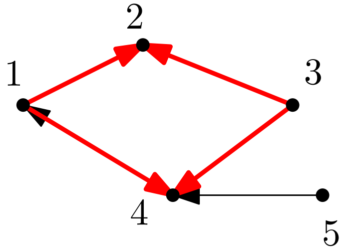
Example for the Proof

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 & 0 \end{pmatrix}$$



Example for the Proof

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 & 0 \end{pmatrix}$$



$$\begin{array}{cl} + & (1 \quad -1 \quad 0 \quad 0 \quad 0) \\ - & (0 \quad -1 \quad 1 \quad 0 \quad 0) \\ + & (0 \quad 0 \quad 1 \quad -1 \quad 0) \\ - & (1 \quad 0 \quad 0 \quad -1 \quad 0) \\ \\ = & (0 \quad 0 \quad 0 \quad 0 \quad 0) \end{array}$$

Lemma A matrix $A \in \{0,1\}^{m \times n}$ where the 1's on every row form an interval is TUM.

Proof.

Lemma A matrix $A \in \{0, 1\}^{m \times n}$ where the 1's on every row form an interval is TUM.

Proof.

- take any square submatrix A' of A,

Lemma A matrix $A \in \{0,1\}^{m\times n}$ where the 1's on every row form an interval is TUM.

Proof.

- take any square submatrix A' of A,
- the 1's on every row of A' form an interval.

Lemma A matrix $A \in \{0,1\}^{m\times n}$ where the 1's on every row form an interval is TUM.

Proof.

- take any square submatrix A' of A,
- the 1's on every row of A' form an interval.
- $A'M$ is a matrix satisfying condition of first lemma, where

$$M = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$ det$(M) = 1$.

Lemma A matrix $A \in \{0,1\}^{m \times n}$ where the 1's on every row form an interval is TUM.

Proof.

- take any square submatrix A' of A,
- the 1's on every row of A' form an interval.
- $A'M$ is a matrix satisfying condition of first lemma, where
$$M = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}. \ \det(M) = 1.$$
- $\det(A'M) \in \{0, \pm 1\} \implies \det(A') \in \{0, \pm 1\}$. □

Example for the Proof

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Example for the Proof

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Example for the Proof

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Example for the Proof

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

- $(\text{col } 1, \text{col } 2 - \text{col } 1, \text{col } 3 - \text{col } 2, \text{col } 4 - \text{col } 3, \text{col } 5 - \text{col } 4)$

Example for the Proof

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix} \Longrightarrow \begin{pmatrix} 0 & 1 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

- $(\text{col } 1, \text{col } 2 - \text{col } 1, \text{col } 3 - \text{col } 2, \text{col } 4 - \text{col } 3, \text{col } 5 - \text{col } 4)$

Example for the Proof

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix} \Longrightarrow \begin{pmatrix} 0 & 1 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

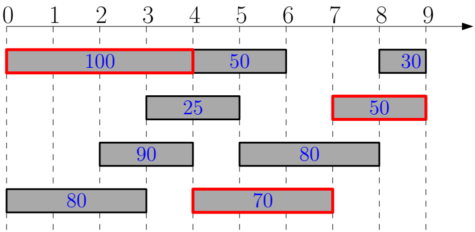
- $($col $1,$ col $2-$ col $1,$ col $3-$ col $2,$ col $4-$ col $3,$ col $5-$ col $4)$
- every row has at most one 1, at most one -1

Weighted Interval Scheduling Problem

Input: n activities, activity i starts at time s_i, finishes at time f_i, and has weight $w_i > 0$

i and j can be scheduled together iff $[s_i, f_i)$ and $[s_j, f_j)$ are disjoint

Output: maximum weight subset of jobs that can be scheduled



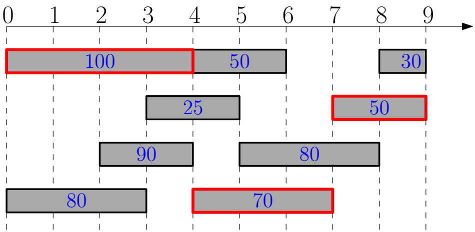
- optimum value $= 220$

Weighted Interval Scheduling Problem

Input: n activities, activity i starts at time s_i, finishes at time f_i, and has weight $w_i > 0$

i and j can be scheduled together iff $[s_i, f_i)$ and $[s_j, f_j)$ are disjoint

Output: maximum weight subset of jobs that can be scheduled



- optimum value$= 220$
- Classic Problem for Dynamic Programming

Weighted Interval Scheduling Problem

Linear Program

$$\max \sum_{j\in[n]} x_j w_j$$

$$\sum_{j\in[n]:t\in[s_j,f_j)} x_j \le 1 \qquad \forall t \in [T]$$

$$x_j \ge 0 \qquad \forall j \in [n]$$

Weighted Interval Scheduling Problem

Linear Program

$$\max \sum_{j\in[n]} x_j w_j$$

$$\sum_{j\in[n]:t\in[s_j,f_j)} x_j \le 1 \qquad \forall t \in [T]$$

$$x_j \ge 0 \qquad \forall j \in [n]$$

- The polytope is integral as the 1's in every column are consecutive.

Weighted Interval Scheduling Problem

Linear Program

$$\max \sum_{j \in [n]} x_j w_j$$

$$\sum_{j \in [n] : t \in [s_j, f_j)} x_j \leq 1 \qquad \forall t \in [T]$$

$$x_j \geq 0 \qquad \forall j \in [n]$$

- The polytope is integral as the 1's in every column are consecutive.

Lemma The edge-vertex incidence matrix A of a bipartite graph is totally-unimodular.

Proof.

Example

Lemma The edge-vertex incidence matrix A of a bipartite graph is totally-unimodular.

Proof.

- $G = (L \uplus R, E)$: the bipartite graph

Example

Lemma The edge-vertex incidence matrix A of a bipartite graph is totally-unimodular.

Proof.

- $G = (L \uplus R, E)$: the bipartite graph
- A': obtained from A by negating columns correspondent to R

Example

Lemma The edge-vertex incidence matrix A of a bipartite graph is totally-unimodular.

Proof.

- $G = (L \uplus R, E)$: the bipartite graph
- A': obtained from A by negating columns correspondent to R
- each row of A' has exactly one $+1$, and exactly one -1

Example

Lemma The edge-vertex incidence matrix A of a bipartite graph is totally-unimodular.

Proof.

- $G = (L \uplus R, E)$: the bipartite graph
- A': obtained from A by negating columns correspondent to R
- each row of A' has exactly one $+1$, and exactly one -1
- $\implies A'$ is TUM $\iff A$ is TUM □

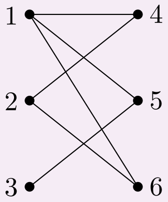
Example

Lemma The edge-vertex incidence matrix A of a bipartite graph is totally-unimodular.

Proof.

- $G = (L \uplus R, E)$: the bipartite graph
- A': obtained from A by negating columns correspondent to R
- each row of A' has exactly one $+1$, and exactly one -1
- $\implies A'$ is TUM $\iff A$ is TUM $\square$

Example

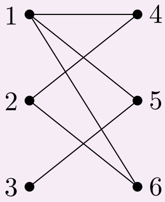


Lemma The edge-vertex incidence matrix A of a bipartite graph is totally-unimodular.

Proof.

- $G = (L \uplus R, E)$: the bipartite graph
- A': obtained from A by negating columns correspondent to R
- each row of A' has exactly one $+1$, and exactly one -1
- $\implies A'$ is TUM $\iff A$ is TUM $\square$

Example



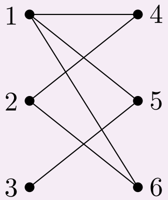
$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Lemma The edge-vertex incidence matrix A of a bipartite graph is totally-unimodular.

Proof.

- $G = (L \uplus R, E)$: the bipartite graph
- A': obtained from A by negating columns correspondent to R
- each row of A' has exactly one $+1$, and exactly one -1
- $\implies A'$ is TUM $\iff A$ is TUM □

Example



$$\begin{pmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & -1 & 0 \end{pmatrix}$$

A different proof for the theorem we proved:

Theorem $\mathcal{P}_{\mathrm{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \leq 1, \forall v \in L \cup R; \qquad x_e \geq 0, \forall e \in E.$$

A different proof for the theorem we proved:

Theorem $\mathcal{P}_{\mathrm{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \leq 1, \forall v \in L \cup R; \qquad x_e \geq 0, \forall e \in E.$$

Proof.

The coefficient matrix for the constraints $\sum_{e \in \delta(v)} x_e \leq 1, \forall v \in L \cup R$ is the vertex-edge incidence matrix of the graph G. Therefore, the polytope is integral. □

A different proof for the theorem we proved:

Theorem $\mathcal{P}_{\mathrm{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \le 1, \forall v \in L \cup R; \qquad x_e \ge 0, \forall e \in E.$$

Proof.

The coefficient matrix for the constraints $\sum_{e \in \delta(v)} x_e \le 1, \forall v \in L \cup R$ is the vertex-edge incidence matrix of the graph G. Therefore, the polytope is integral. □

- remark: bipartiteness is needed. The edge-vertex incidence matrix $\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$ of a triangle has determinent 2.

Outline

1 Linear Programming
- Introduction
- Methods for Solving Linear Programs

2 Polytope with Polynomial Number of Facets
- Bipartite Matching Polytope
- Polytopes with Totally Unimodular Coefficient Matrices

3 Polytopes with Efficient Separation Oracles
- s-t Cut Polytope
- Spanning Tree Polytope
- General Graph (Perfect) Matching Polytope

4 Matroid, Matroid Basis and Matroid Intersection Polytopes *
- Preliminaries on Matroid Theory
- Matroid Polytope
- Matroid Basis and Matroid Intersection Polytope

Def. A separation oracle for a polytope $\mathcal{P} \subseteq \mathbb{R}^n$ is an algorithm that, given some $x^* \in \mathbb{R}^n$,

- either correctly claims that $x \in \mathcal{P}$,
- or outputs a linear constraint $a^{\mathrm{T}}x \leq b$ that separating x^* from $\mathcal{P}$: every $x \in \mathcal{P}$ satisfies $a^{\mathrm{T}}x \leq b$, but $a^{\mathrm{T}}x^* > b$. We say $a^{\mathrm{T}}x \leq b$ is a separation plane for x^*.

The separation oracle is efficient if its running time is polynomial in the size of the instance plus the size of x

Def. A separation oracle for a polytope $\mathcal{P} \subseteq \mathbb{R}^n$ is an algorithm that, given some $x^* \in \mathbb{R}^n$,

- either correctly claims that $x \in \mathcal{P}$,
- or outputs a linear constraint $a^{\mathrm{T}}x \leq b$ that separating x^* from $\mathcal{P}$: every $x \in \mathcal{P}$ satisfies $a^{\mathrm{T}}x \leq b$, but $a^{\mathrm{T}}x^* > b$. We say $a^{\mathrm{T}}x \leq b$ is a separation plane for x^*.

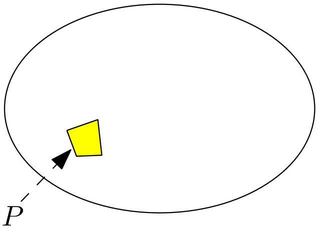
The separation oracle is efficient if its running time is polynomial in the size of the instance plus the size of x

- Clearly, if $\mathcal{P} \subseteq \mathbb{R}^n$ can be described using a polynomial-size LP, then it has an efficient separation oracle.
- However, there are cases where $\mathcal{P} \subseteq \mathbb{R}^n$ has exponential number of facets, but still admits an efficient separation oracle.

- We can use ellipsoid method to solve the LP $\min / \max w^{\mathrm{T}}x, x \in \mathcal{P}$, when $\mathcal{P}$ has an efficient separation oracle, using the ellipsoid method.

Ellipsoid Method

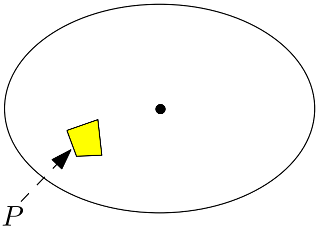
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



- We can use ellipsoid method to solve the LP $\min / \max w^{\mathrm{T}}x, x \in \mathcal{P}$, when $\mathcal{P}$ has an efficient separation oracle, using the ellipsoid method.

Ellipsoid Method

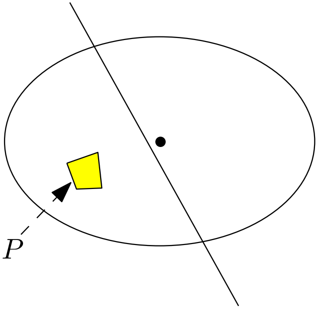
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



- We can use ellipsoid method to solve the LP $\min / \max w^{\mathrm{T}} x, x \in \mathcal{P}$, when $\mathcal{P}$ has an efficient separation oracle, using the ellipsoid method.

Ellipsoid Method

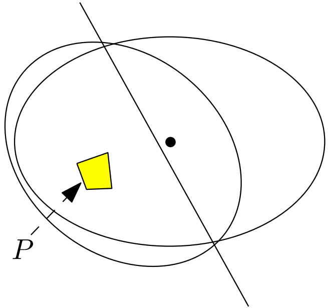
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



- We can use ellipsoid method to solve the LP $\min / \max w^{\mathrm{T}} x, x \in \mathcal{P}$, when $\mathcal{P}$ has an efficient separation oracle, using the ellipsoid method.

Ellipsoid Method

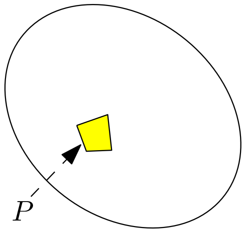
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



- We can use ellipsoid method to solve the LP $\min / \max w^{\mathrm{T}} x, x \in \mathcal{P}$, when $\mathcal{P}$ has an efficient separation oracle, using the ellipsoid method.

Ellipsoid Method

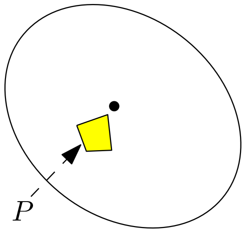
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



- We can use ellipsoid method to solve the LP $\min / \max w^{\mathrm{T}} x, x \in \mathcal{P}$, when $\mathcal{P}$ has an efficient separation oracle, using the ellipsoid method.

Ellipsoid Method

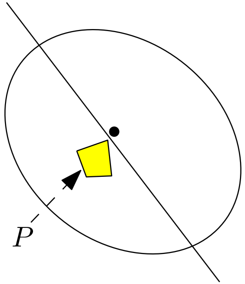
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



- We can use ellipsoid method to solve the LP $\min / \max w^{\mathrm{T}} x, x \in \mathcal{P}$, when $\mathcal{P}$ has an efficient separation oracle, using the ellipsoid method.

Ellipsoid Method

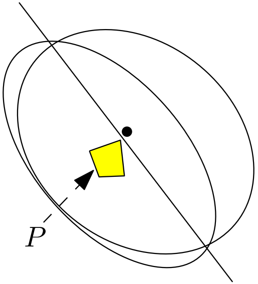
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



- We can use ellipsoid method to solve the LP $\min / \max w^{\mathrm{T}} x, x \in \mathcal{P}$, when $\mathcal{P}$ has an efficient separation oracle, using the ellipsoid method.

Ellipsoid Method

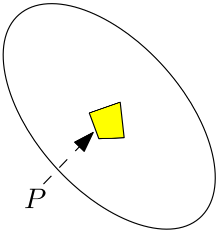
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



- We can use ellipsoid method to solve the LP $\min / \max w^{\mathrm{T}} x, x \in \mathcal{P}$, when $\mathcal{P}$ has an efficient separation oracle, using the ellipsoid method.

Ellipsoid Method

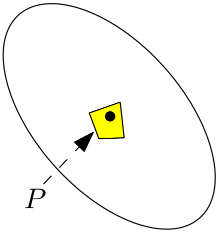
- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



- We can use ellipsoid method to solve the LP $\min / \max w^{\mathrm{T}} x, x \in \mathcal{P}$, when $\mathcal{P}$ has an efficient separation oracle, using the ellipsoid method.

Ellipsoid Method

- maintain an ellipsoid that contains the feasible region
- query a separation oracle if the center of ellipsid is in the feasible region:
  - yes: then the feasible region is not empty
  - no: cut the elliposid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



Outline

1. Linear Programming
   - Introduction
   - Methods for Solving Linear Programs
2. Polytope with Polynomial Number of Facets
   - Bipartite Matching Polytope
   - Polytopes with Totally Unimodular Coefficient Matrices
3. Polytopes with Efficient Separation Oracles
   - s-t Cut Polytope
   - Spanning Tree Polytope
   - General Graph (Perfect) Matching Polytope
4. Matroid, Matroid Basis and Matroid Intersection Polytopes *
   - Preliminaries on Matroid Theory
   - Matroid Polytope
   - Matroid Basis and Matroid Intersection Polytope

s-t Cut Polytope

Def. Given a digraph $G = (V, E)$, C is a s-t cut in G, if s and t are disconnected in $(V, E \setminus C)$.

- $\mathcal{P}_{\mathrm{min-cut}} :=$ conv$\big(\{\chi^C : C$ is a s-t cut in $G\}\big)$

Theorem $\mathcal{P}_{\mathrm{min-cut}}$ is the set of vectors $x \in \mathbb{R}^E$ satisfying the following inequalities:

$$\sum_{e \in P} x_e \geq 1 \qquad \forall \text{ simple } s\text{-}t \text{ path } P \qquad (*)$$

$$x_e \in [0, 1] \qquad \forall e \in E$$

s-t Cut Polytope

Def. Given a digraph $G = (V, E)$, C is a s-t cut in G, if s and t are disconnected in $(V, E \setminus C)$.

- $\mathcal{P}_{\mathrm{min-cut}} := \mathrm{conv}\big(\{\chi^C : C \text{ is a } s\text{-}t \text{ cut in } G\}\big)$

Theorem $\mathcal{P}_{\mathrm{min-cut}}$ is the set of vectors $x \in \mathbb{R}^E$ satisfying the following inequalities:

$$\sum_{e \in P} x_e \geq 1 \qquad \forall \text{ simple } s\text{-}t \text{ path } P \qquad (*)$$

$$x_e \in [0, 1] \qquad \forall e \in E$$

Q: Given $x \in [0, 1]^E$, how can we check if x satisfies all constraints in (*)?

s-t Cut Polytope

Def. Given a digraph $G = (V, E)$, C is a s-t cut in G, if s and t are disconnected in $(V, E \setminus C)$.

- $\mathcal{P}_{\mathrm{min-cut}} := \mathrm{conv}\big(\{\chi^C : C \text{ is a } s\text{-}t \text{ cut in } G\}\big)$

Theorem $\mathcal{P}_{\mathrm{min-cut}}$ is the set of vectors $x \in \mathbb{R}^E$ satisfying the following inequalities:

$$\sum_{e \in P} x_e \geq 1 \qquad \forall \text{ simple } s\text{-}t \text{ path } P \qquad (*)$$

$$x_e \in [0, 1] \qquad \forall e \in E$$

Q: Given $x \in [0, 1]^E$, how can we check if x satisfies all constraints in (*)?

A: Use shortest path algorithm with weights $(x_e)_{e \in E}$.

Theorem $\mathcal{P}_{\text{min-cut}}$ is the set of vectors $x \in \mathbb{R}^E$ satisfying the following inequalities:

$$\sum_{e \in P} x_e \geq 1 \qquad \forall \text{ simple } s\text{-}t \text{ path } P \qquad (*)$$

$$x_e \in [0, 1] \qquad \forall e \in E$$

Proof of Lemma.

- Given $x \in [0, 1]^E$ satisfying (*)
- $d_x(v), v \in V$: length of shortest path from s to v, with x being the weights; so $d_x(s) = 0$ and $d_x(t) \geq 1$

Theorem $\mathcal{P}_{\min-\text{cut}}$ is the set of vectors $x \in \mathbb{R}^E$ satisfying the following inequalities:

$$\sum_{e \in P} x_e \geq 1 \qquad \forall \text{ simple } s\text{-}t \text{ path } P \qquad (*)$$

$$x_e \in [0, 1] \qquad \forall e \in E$$

Proof of Lemma.

- Given $x \in [0, 1]^E$ satisfying (*)
- $d_x(v), v \in V$: length of shortest path from s to v, with x being the weights; so $d_x(s) = 0$ and $d_x(t) \geq 1$
- randomly choose a real $\theta \in (0, 1)$
- $S := \{v \in V : d_x(v) \leq \theta\}, T := V \setminus S = \{v \in V : d_x(v) > \theta\}$
- $C := E(S, T)$

Claim For an edge $(u, v) \in E$, we have

$$\Pr[(u, v) \in C] \leq \max\{d_x(v) - d_x(u), 0\}.$$

Proof.

- $(u, v) \in C$ happens only if $d_x(u) < \theta \leq d_x(v)$.
- This happens with probability at most $\max\{d_x(v) - d_x(u), 0\} \leq x_{(u,v)}$. □

Proof of Lemma, Continued

- $\mathbb{E}_\theta[\chi^C] \leq x$
- We can define a random set C' so that $C' \supseteq C$ happens with probability 1, and $\mathbb{E}_\theta[\chi^{C'}] = x$.
- So $x \in \text{conv}(\{\chi^{C'} : C' \text{ is a } s\text{-}t \text{ cut in } G\})$ □

Outline

1. Linear Programming
   - Introduction
   - Methods for Solving Linear Programs
2. Polytope with Polynomial Number of Facets
   - Bipartite Matching Polytope
   - Polytopes with Totally Unimodular Coefficient Matrices
3. Polytopes with Efficient Separation Oracles
   - s-t Cut Polytope
   - Spanning Tree Polytope
   - General Graph (Perfect) Matching Polytope
4. Matroid, Matroid Basis and Matroid Intersection Polytopes *
   - Preliminaries on Matroid Theory
   - Matroid Polytope
   - Matroid Basis and Matroid Intersection Polytope

Spanning Tree Polytope

- Given a connected graph $G = (V, E)$
- $\mathcal{P}_{\text{ST}} :=$ conv $\left(\left\{\chi^T : T \subseteq E \text{ is a spanning tree of } G\right\}\right)$

Theorem (Spanning Tree Polytope Theorem) $\mathcal{P}_{\text{ST}}$ is the set of vectors $x \in \mathbb{R}^E$ satisfying the following inequalities:

$$\begin{aligned} \sum_{e \in E} x_e &= n - 1 & & \\ \sum_{e \in E[S]} x_e &\leq |S| - 1 & \forall S \subseteq V, 2 \leq |S| \leq n - 1 \quad & (*) \\ x_e &\geq 0 & \forall e \in E & \end{aligned}$$

- Spanning trees correspond to bases of graphic matroid for G
- Later we prove a more general theorem on matroid polytopes

Theorem (Spanning Tree Polytope Theorem) $\mathcal{P}_{\mathrm{ST}}$ is the set of vectors $x \in \mathbb{R}^E$ satisfying the following inequalities:

$$\begin{aligned} \sum_{e \in E} x_e &= n - 1 & & \\ \sum_{e \in E[S]} x_e &\leq |S| - 1 & \forall S \subseteq V, 2 \leq |S| \leq n - 1 \qquad (*) \\ x_e &\geq 0 & \forall e \in E \end{aligned}$$

Q: How can we check if all constraints in (*) are satisfied?

Theorem (Spanning Tree Polytope Theorem) $\mathcal{P}_{\mathrm{ST}}$ is the set of vectors $x \in \mathbb{R}^E$ satisfying the following inequalities:

$$\sum_{e \in E} x_e = n - 1$$

$$\sum_{e \in E[S]} x_e \le |S| - 1 \qquad \forall S \subseteq V, 2 \le |S| \le n-1 \qquad (*)$$

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

Q: How can we check if all constraints in (*) are satisfied?

A: $\xrightarrow{\text{reduce}}$ densest sub-graph $\xrightarrow{\text{reduce}}$ maximum flow

Checking if $\sum_{e \in E[S]} x_e \le |S| - 1, \forall S \subseteq V$

- We need to check if $\exists S \subseteq V, \frac{\sum_{e \in E[S]} x_e}{|S|-1} > 1$:

- Guess a vertex $v \in S$; set $w_v = 0$ and $w_u = 1$ for every $u \in V \setminus \{v\}$
- The problem becomes to check if $\exists S \subseteq V, \frac{\sum_{e \in E[S]} x_e}{\sum_{u \in S} w_u} > 1$

Checking if $\sum_{e\in E[S]} x_e \leq |S|-1, \forall S \subseteq V$

- We need to check if $\exists S \subseteq V, \frac{\sum_{e\in E[S]} x_e}{|S|-1} > 1$:

- Guess a vertex $v \in S$; set $w_v = 0$ and $w_u = 1$ for every $u \in V \setminus \{v\}$
- The problem becomes to check if $\exists S \subseteq V, \frac{\sum_{e\in E[S]} x_e}{\sum_{u\in S} w_u} > 1$
- This is a (weighted) densest subgraph problem

Checking if $\sum_{e\in E[S]} x_e \leq |S| - 1, \forall S \subseteq V$

- We need to check if $\exists S \subseteq V, \frac{\sum_{e\in E[S]} x_e}{|S|-1} > 1$:

- Guess a vertex $v \in S$; set $w_v = 0$ and $w_u = 1$ for every $u \in V \setminus \{v\}$
- The problem becomes to check if $\exists S \subseteq V, \frac{\sum_{e\in E[S]} x_e}{\sum_{u\in S} w_u} > 1$
- This is a (weighted) densest subgraph problem
- Exercise: It can be solved using maximum flow

Outline

1. Linear Programming
   - Introduction
   - Methods for Solving Linear Programs
2. Polytope with Polynomial Number of Facets
   - Bipartite Matching Polytope
   - Polytopes with Totally Unimodular Coefficient Matrices
3. Polytopes with Efficient Separation Oracles
   - s-t Cut Polytope
   - Spanning Tree Polytope
   - General Graph (Perfect) Matching Polytope
4. Matroid, Matroid Basis and Matroid Intersection Polytopes *
   - Preliminaries on Matroid Theory
   - Matroid Polytope
   - Matroid Basis and Matroid Intersection Polytope

General Graph Perfect Matching Polytope

General Perfect Matching Polytope

- Given a graph $G = (V, E)$, where $|V|$ is even
- $\mathcal{P}_{\mathrm{GPM}} :=$ conv $\left(\left\{\chi^M : M \subseteq E \text{ is a perfect matching in } G\right\}\right)$

General Graph Perfect Matching Polytope

General Perfect Matching Polytope

- Given a graph $G = (V, E)$, where $|V|$ is even
- $\mathcal{P}_{\mathrm{GPM}} := \text{conv}\left(\{\chi^M : M \subseteq E \text{ is a perfect matching in } G\}\right)$

Theorem (General Perfect Matching Polytope Theorem)

$\mathcal{P}_{\mathrm{GPM}}$ is the set of vectors $x \in \mathbb{R}^E$ satisfying the following inequalities:

$$\sum_{e \in \delta(v)} x_e = 1 \qquad \forall v \in V$$

$$\sum_{e \in E(S, V \setminus S)} x_e \geq 1 \qquad \forall S \subseteq V, |S| \text{ is odd} \qquad (*)$$

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

Theorem (General Perfect Matching Polytope Theorem)
$\mathcal{P}_{\mathrm{GPM}}$ is the set of vectors $x \in \mathbb{R}^E$ satisfying the following inequalities:

$$\sum_{e\in\delta(v)} x_e = 1 \qquad \forall v \in V$$

$$\sum_{e\in E(S,V\setminus S)} x_e \geq 1 \qquad \forall S \subseteq V, |S| \text{ is odd} \qquad (*)$$

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

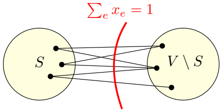
Proof of General Perfect Matching Polytope Theorem

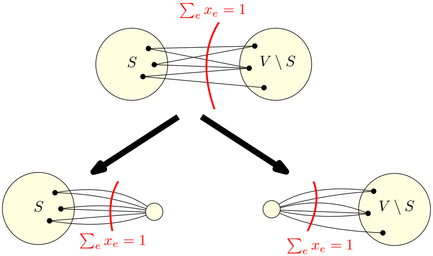
- Clearly, every $x \in \mathcal{P}_{\text{GPM}}$ satisfies all the LP constraints
- We prove the LP polytope is integral; this implies lemma
- We choose the counter-example G with the smallest $|V| + |E|$, and focus on a non-integral vertex x of the LP polytope

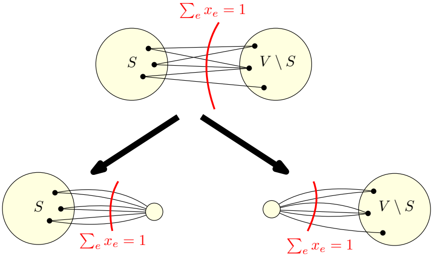
Proof of General Perfect Matching Polytope Theorem

- $x_e = 0$ for some $e \in E$: e could be removed.
- $x_e = 1$ for some $e \in E$: e and its 2 end vertices could be removed.
- So $x_e \in (0, 1)$ for every $e \in E$.
- Every $v \in V$ has degree at least 2.
- Every $v \in V$ has degree exactly 2: G is union of disjoint cycles, x would not be a vertex of LP polytope.
- Assume some $v \in V$ has degree at least 3; $|E| \geq |V| + 1$.
- x is the unique solution to a system ot n linear equations from the LP.
- So, some linear equation is

$$\sum_{e \in E(S, V \setminus S)} x_e = 1 \text{ for some } S \subseteq V \text{ with } |S| \geq 3, |V \setminus S| \geq 3$$

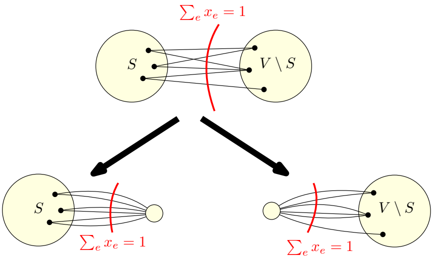

$\sum_e x_e = 1$
S
$V \setminus S$


$\sum_e x_e = 1$
S
$V \setminus S$
S
$\sum_e x_e = 1$
$V \setminus S$
$\sum_e x_e = 1$



Proof of General Perfect Matching Polytope Theorem

- Consider two instances: $(G/V, x'), (G/(V \setminus S), x'')$
- Both x' and x'' satisfy the LP constraints for their respective graphs.

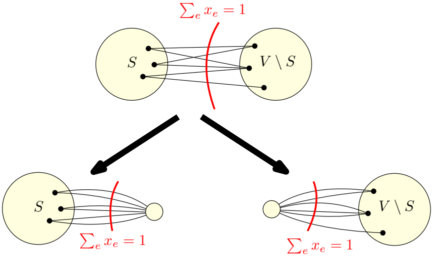


Proof of General Perfect Matching Polytope Theorem

- Consider two instances: $(G/V, x'), (G/(V \setminus S), x'')$
- Both x' and x'' satisfy the LP constraints for their respective graphs.

- By the minimality assumption:

$$x' \in \text{conv}(\{\chi^M : M \text{ is a perfect matching in } G/S\})$$
$$x'' \in \text{conv}(\{\chi^M : M \text{ is a perfect matching in } G/(V \setminus S)\})$$



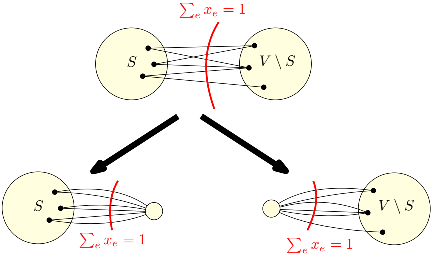
Proof of General Perfect Matching Polytope Theorem

- Consider two instances: $(G/V, x'), (G/(V \setminus S), x'')$
- Both x' and x'' satisfy the LP constraints for their respective graphs.

- By the minimality assumption:

$$x' \in \text{conv}(\{\chi^M : M \text{ is a perfect matching in } G/S\})$$
$$x'' \in \text{conv}(\{\chi^M : M \text{ is a perfect matching in } G/(V \setminus S)\})$$

- Decompose x' and x'' into a convex combinations of matchings



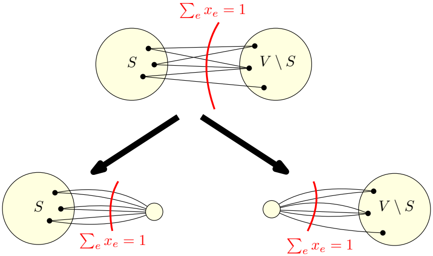
Proof of General Perfect Matching Polytope Theorem

- Consider two instances: $(G/V, x'), (G/(V \setminus S), x'')$
- Both x' and x'' satisfy the LP constraints for their respective graphs.

- By the minimality assumption:

$$x' \in \text{conv}(\{\chi^M : M \text{ is a perfect matching in } G/S\})$$
$$x'' \in \text{conv}(\{\chi^M : M \text{ is a perfect matching in } G/(V \setminus S)\})$$

- Decompose x' and x'' into a convex combinations of matchings
- Each $e \in E(S, V \setminus S)$ has the same fraction in combinations



Proof of General Perfect Matching Polytope Theorem

- Consider two instances: $(G/V, x'), (G/(V \setminus S), x'')$
- Both x' and x'' satisfy the LP constraints for their respective graphs.

- By the minimality assumption:

$$x' \in \text{conv}(\{\chi^M : M \text{ is a perfect matching in } G/S\})$$
$$x'' \in \text{conv}(\{\chi^M : M \text{ is a perfect matching in } G/(V \setminus S)\})$$

- Decompose x' and x'' into a convex combinations of matchings
- Each $e \in E(S, V \setminus S)$ has the same fraction in combinations
- "Concatenate" two convex combinations into one convex combinations of matching in G. So x can not be a vertex. $\square$

Theorem (General Perfect Matching Polytope Theorem)
$\mathcal{P}_{\text{GPM}}$ is the set of vectors $x \in \mathbb{R}^E$ satisfying the following inequalities:

$$\sum_{e \in \delta(v)} x_e = 1 \qquad \forall v \in V$$

$$\sum_{e \in E(S, V \setminus S)} x_e \geq 1 \qquad \forall S \subseteq V, |S| \text{ is odd} \qquad (*)$$

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

Q: How can we check if all constraints in (*) are satisfied?

Theorem (General Perfect Matching Polytope Theorem)
$\mathcal{P}_{\text{GPM}}$ is the set of vectors $x \in \mathbb{R}^E$ satisfying the following inequalities:

$$\sum_{e \in \delta(v)} x_e = 1 \qquad \forall v \in V$$

$$\sum_{e \in E(S, V \setminus S)} x_e \geq 1 \qquad \forall S \subseteq V, |S| \text{ is odd} \qquad (*)$$

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

Q: How can we check if all constraints in (*) are satisfied?

A: Use the Gomory-Hu Tree structure.

Theorem (General Perfect Matching Polytope Theorem)
$\mathcal{P}_{\mathrm{GPM}}$ is the set of vectors $x \in \mathbb{R}^E$ satisfying the following inequalities:

$$\sum_{e \in \delta(v)} x_e = 1 \qquad \forall v \in V$$

$$\sum_{e \in E(S, V \setminus S)} x_e \geq 1 \qquad \forall S \subseteq V, |S| \text{ is odd} \qquad (*)$$

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

Q: How can we check if all constraints in (*) are satisfied?

A: Use the Gomory-Hu Tree structure.

- inequality in (*) can be replaced by $\sum_{e \in E[S]} x_e \leq \frac{|S|-1}{2}$
- more convenient to obtain general matching polytope

General Matching Polytope

- Given a graph $G = (V, E)$
- $\mathcal{P}_{\mathrm{GM}} := \text{conv}\left(\left\{\chi^M : M \subseteq E \text{ is a matching in } G\right\}\right)$

General Matching Polytope

- Given a graph $G = (V, E)$
- $\mathcal{P}_{\mathrm{GM}} := \text{conv}\left(\left\{\chi^M : M \subseteq E \text{ is a matching in } G\right\}\right)$

Theorem (General Matching Polytope Theorem) $\mathcal{P}_{\mathrm{GM}}$ is the set of vectors $x \in \mathbb{R}^E$ satisfying the following inequalities:

$$\begin{aligned} \sum_{e \in \delta(v)} x_e &\leq 1 && \forall v \in V \\ \sum_{e \in E(S)} x_e &\leq \frac{|S| - 1}{2} && \forall S \subseteq V, |S| \text{ is odd} \\ x_e &\geq 0 && \forall e \in E \end{aligned} \tag{1}$$

Remark

- For all the polytopes, we identified a set of linear inequalities that are sufficient to define the polytope.
- However, not all the constraints are facet-defining.
- Only facet-defining constraints are necessarily; other constraints could be removed. (We keep all the constraints.for convenience of description.)

Remark

- For all the polytopes, we identified a set of linear inequalities that are sufficient to define the polytope.
- However, not all the constraints are facet-defining.
- Only facet-defining constraints are necessarily; other constraints could be removed. (We keep all the constraints.for convenience of description.)
- Example: in spanning tree polytope, $\sum_{e \in E[S]} x_e \leq |S| - 1$ is not needed if $(S, E[S])$ is disconnected, or contains a bridge. In this case, the constraint does not define a facet.

Outline

1. Linear Programming
   - Introduction
   - Methods for Solving Linear Programs
2. Polytope with Polynomial Number of Facets
   - Bipartite Matching Polytope
   - Polytopes with Totally Unimodular Coefficient Matrices
3. Polytopes with Efficient Separation Oracles
   - s-t Cut Polytope
   - Spanning Tree Polytope
   - General Graph (Perfect) Matching Polytope
4. Matroid, Matroid Basis and Matroid Intersection Polytopes *
   - Preliminaries on Matroid Theory
   - Matroid Polytope
   - Matroid Basis and Matroid Intersection Polytope

Outline

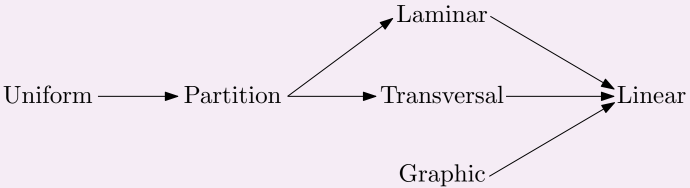
1. Linear Programming
   - Introduction
   - Methods for Solving Linear Programs
2. Polytope with Polynomial Number of Facets
   - Bipartite Matching Polytope
   - Polytopes with Totally Unimodular Coefficient Matrices
3. Polytopes with Efficient Separation Oracles
   - s-t Cut Polytope
   - Spanning Tree Polytope
   - General Graph (Perfect) Matching Polytope
4. Matroid, Matroid Basis and Matroid Intersection Polytopes *
   - Preliminaries on Matroid Theory
   - Matroid Polytope
   - Matroid Basis and Matroid Intersection Polytope

Recall Definition and Examples of Matroid

Def. A (finite) matroid $\mathcal{M}$ is a pair $(E, \mathcal{I})$, where E is a finite set (called the ground set) and $\mathcal{I}$ is a family of subsets of E (called independent sets) with the following properties:

1. $\emptyset \in \mathcal{I}$.
2. (downward-closed property) If $B \subsetneq A \in \mathcal{I}$, then $B \in \mathcal{I}$.
3. (augmentation/exchange property) If $A, B \in \mathcal{I}$ and $|B| < |A|$, then there exists $e \in A \setminus B$ such that $B \cup \{e\} \in \mathcal{I}$.

Relationship between matroids



Other Terminologies Related To a Matroid $\mathcal{M} = (E, \mathcal{I})$

- A subset of E that is not independent is dependent.
- A maximal independent set is called a basis (plural: bases)
- A minimal dependent set is called a circuit

Other Terminologies Related To a Matroid $\mathcal{M} = (E, \mathcal{I})$

- A subset of E that is not independent is dependent.
- A maximal independent set is called a basis (plural: bases)
- A minimal dependent set is called a circuit

- Graphic matroid for a connected graph $G = (V, E)$:
  basis $\Longleftrightarrow$ spanning tree   circuit $\Longleftrightarrow$ cycle

Lemma All bases of a matroid have the same size.

Proof.

- Assume two A and A' are both bases of $\mathcal{M}$ and $|A| > |A'|$
- By exchange property: $\exists i \in A \setminus A', A' \cup \{i\} \in \mathcal{I}$
- contradiction with that A' is a basis $\square$

- Recall: Matroid Rank Function:

Def. Given a matroid $\mathcal{M} = (E, \mathcal{I})$, the rank of any $A \subseteq E$ is defined as

$$r_{\mathcal{M}}(A) = \max\left\{|A'| : A' \subseteq A, A' \in \mathcal{I}\right\}.$$

The function $r_{\mathcal{M}} : 2^E \to \mathbb{Z}_{\geq 0}$ is called the rank function of $\mathcal{M}$.

- $r_{\mathcal{M}}(A)$ is size of maximum independent subset of A

Trivial properties of $r_{\mathcal{M}}$

- $r_{\mathcal{M}}(\emptyset) = 0$
- $r_{\mathcal{M}}(A \cup \{i\}) - r_{\mathcal{M}}(A) \in \{0, 1\}$ for every $A \subseteq E, i \in E \setminus A$

Theorem The rank function $r_{\mathcal{M}}$ of a matroid $\mathcal{M} = (E, \mathcal{I})$ is submodular.

Greedy algorithm finds max ind. subset of any given $X \subseteq E$:

1: $S \leftarrow \emptyset$
2: **while** $\exists e \in X \setminus S$ s.t. $S \cup \{e\} \in \mathcal{I}$ **do**
3: let e be an arbitrary element satisfying the condition
4: $S \leftarrow S \cup \{e\}$
5: **return** S

Proof of Submodularity of $r_{\mathcal{M}}$.

Greedy algorithm finds max ind. subset of any given $X \subseteq E$:

```
S ← ∅
while ∃e ∈ X \ S s.t. S ∪ {e} ∈ 𝓘 do
    let e be an arbitrary element satisfying the condition
    S ← S ∪ {e}
return S
```

Proof of Submodularity of $r_{\mathcal{M}}$.

- Take $A \subsetneq E, i, j \in E \setminus A, i \neq j$, need to prove: $r_{\mathcal{M}}(A \cup \{i, j\}) - r_{\mathcal{M}}(A \cup \{i\}) \leq r_{\mathcal{M}}(A \cup \{j\}) - r_{\mathcal{M}}(A)$

Greedy algorithm finds max ind. subset of any given $X \subseteq E$:

$S \leftarrow \emptyset$
while $\exists e \in X \setminus S$ s.t. $S \cup \{e\} \in \mathcal{I}$ **do**
    let e be an arbitrary element satisfying the condition
    $S \leftarrow S \cup \{e\}$
return S

Proof of Submodularity of $r_\mathcal{M}$.

- Take $A \subsetneq E, i, j \in E \setminus A, i \neq j$, need to prove: $r_\mathcal{M}(A \cup \{i, j\}) - r_\mathcal{M}(A \cup \{i\}) \leq r_\mathcal{M}(A \cup \{j\}) - r_\mathcal{M}(A)$
- if not, then LHS $= 1$, RHS $= 0$

Greedy algorithm finds max ind. subset of any given $X \subseteq E$:

1: $S \leftarrow \emptyset$
2: **while** $\exists e \in X \setminus S$ s.t. $S \cup \{e\} \in \mathcal{I}$ **do**
3: let e be an arbitrary element satisfying the condition
4: $S \leftarrow S \cup \{e\}$
5: **return** S

Proof of Submodularity of $r_{\mathcal{M}}$.

- Take $A \subsetneq E, i, j \in E \setminus A, i \neq j$, need to prove: $r_{\mathcal{M}}(A \cup \{i,j\}) - r_{\mathcal{M}}(A \cup \{i\}) \leq r_{\mathcal{M}}(A \cup \{j\}) - r_{\mathcal{M}}(A)$
- if not, then LHS $= 1$, RHS $= 0$
- S: max ind. subset of A, S': max ind. subset of $A \cup \{i\}$
- $|S| = r_{\mathcal{M}}(A), |S'| = r_{\mathcal{M}}(A \cup \{i\})$, $S' = S$ or $S' = S \cup \{i\}$

Greedy algorithm finds max ind. subset of any given $X \subseteq E$:

1: $S \leftarrow \emptyset$
2: **while** $\exists e \in X \setminus S$ s.t. $S \cup \{e\} \in \mathcal{I}$ **do**
3: let e be an arbitrary element satisfying the condition
4: $S \leftarrow S \cup \{e\}$
5: **return** S

Proof of Submodularity of $r_{\mathcal{M}}$.

- Take $A \subsetneq E, i, j \in E \setminus A, i \neq j$, need to prove:
  $r_{\mathcal{M}}(A \cup \{i, j\}) - r_{\mathcal{M}}(A \cup \{i\}) \leq r_{\mathcal{M}}(A \cup \{j\}) - r_{\mathcal{M}}(A)$
- if not, then LHS $= 1$, RHS $= 0$
- S: max ind. subset of A, S': max ind. subset of $A \cup \{i\}$
- $|S| = r_{\mathcal{M}}(A), |S'| = r_{\mathcal{M}}(A \cup \{i\})$, $S' = S$ or $S' = S \cup \{i\}$
- RHS $= 0 \implies S \cup \{j\} \notin \mathcal{I}$, LHS $= 1 \implies S' \cup \{j\} \in \mathcal{I}$
- contradiction □

Lemma A function $r : 2^E \to \mathbb{R}$ is the rank function of a matroid if and only if

1. $r(\emptyset) = 0$
2. $r(A \cup \{i\}) - r(A) \in \{0, 1\}$ for all $A \subseteq E, i \notin E \setminus A$
3. r is submodular.

Proof.

- Define $\mathcal{I} = \{A \subseteq E : r(A) = |A|\}$.
- Claim: $(E, \mathcal{I})$ is a matroid and r is its rank function.

Lemma A function $r : 2^E \to \mathbb{R}$ is the rank function of a matroid if and only if

1. $r(\emptyset) = 0$
2. $r(A \cup \{i\}) - r(A) \in \{0, 1\}$ for all $A \subseteq E, i \notin E \setminus A$
3. r is submodular.

Proof.

- Define $\mathcal{I} = \big\{A \subseteq E : r(A) = |A|\big\}$.
- Claim: $(E, \mathcal{I})$ is a matroid and r is its rank function.
- ①, ② $\implies$ $\mathcal{I}$ is closed under taking subsets

Lemma A function $r : 2^E \to \mathbb{R}$ is the rank function of a matroid if and only if

1. $r(\emptyset) = 0$
2. $r(A \cup \{i\}) - r(A) \in \{0, 1\}$ for all $A \subseteq E, i \notin E \setminus A$
3. r is submodular.

Proof.

- Define $\mathcal{I} = \big\{A \subseteq E : r(A) = |A|\big\}$.
- Claim: $(E, \mathcal{I})$ is a matroid and r is its rank function.
- ①, ② $\implies$ $\mathcal{I}$ is closed under taking subsets
- $A, A' : r(A) = |A|, r(A') = |A'|, |A| < |A'|$
- $U := A \cup A' : r(U) \geq r(A') > r(A), \qquad A \subsetneq U$

Lemma A function $r : 2^E \to \mathbb{R}$ is the rank function of a matroid if and only if

1. $r(\emptyset) = 0$
2. $r(A \cup \{i\}) - r(A) \in \{0, 1\}$ for all $A \subseteq E, i \notin E \setminus A$
3. r is submodular.

Proof.

- Define $\mathcal{I} = \big\{A \subseteq E : r(A) = |A|\big\}$.
- Claim: $(E, \mathcal{I})$ is a matroid and r is its rank function.
- ①, ② $\implies$ $\mathcal{I}$ is closed under taking subsets
- $A, A' : r(A) = |A|, r(A') = |A'|, |A| < |A'|$
- $U := A \cup A' : r(U) \geq r(A') > r(A), \qquad A \subsetneq U$
- ③ $\implies$ $\exists i \in U \setminus A = A' \setminus A : r(A \cup \{i\}) > r(A)$
- $i \in A' \setminus A$ and $r(A \cup \{i\}) = r(A) + 1 = |A \cup \{i\}|$
- so, $A \cup \{i\} \in \mathcal{I} \implies$ exchange property □

Derivatives of Matroids

Def. Given a matroid $\mathcal{M} = (E, \mathcal{I})$ and an element $e \in E$, the matroid obtained from $\mathcal{M}$ by removing e, denoted as $\mathcal{M} \setminus e$, is defined as follows:

$$\mathcal{M} \setminus e = (E \setminus e, \{A \subseteq E \setminus e : A \in \mathcal{I}\}).$$

Derivatives of Matroids

Def. Given a matroid $\mathcal{M} = (E, \mathcal{I})$ and an element $e \in E$, the matroid obtained from $\mathcal{M}$ by removing e, denoted as $\mathcal{M} \setminus e$, is defined as follows:

$$\mathcal{M} \setminus e = (E \setminus e, \{A \subseteq E \setminus e : A \in \mathcal{I}\}).$$

Def. Given a matroid $\mathcal{M} = (E, \mathcal{I})$ and an element $e \in E$, the matroid obtained from $\mathcal{M}$ by contracting e, denoted as $\mathcal{M}/e$, is defined as follows:

$$\mathcal{M}/e = (E \setminus e, \{A \subseteq E \setminus e : A \cup \{e\} \in \mathcal{I}\}).$$

Derivatives of Matroids

Def. Given a matroid $\mathcal{M} = (E, \mathcal{I})$ and a subset $E' \subseteq E$, the matroid of $\mathcal{M}$ restricted to E', denoted as $\mathcal{M}[E']$, is defined as follows:

$$\mathcal{M}[E'] = (E', \{A \subseteq E' : A \in \mathcal{I}\}).$$

Derivatives of Matroids

Def. Given a matroid $\mathcal{M} = (E, \mathcal{I})$ and a subset $E' \subseteq E$, the matroid of $\mathcal{M}$ restricted to E', denoted as $\mathcal{M}[E']$, is defined as follows:

$$\mathcal{M}[E'] = (E', \{A \subseteq E' : A \in \mathcal{I}\}).$$

Def. For a matroid $\mathcal{M} = (E, \mathcal{I})$, the dual matroid $\mathcal{M}^* = (E, \mathcal{I}^*)$ is defined so that the bases in $\mathcal{M}^*$ are exactly the complements of the bases in $\mathcal{I}$.

Theorem $\mathcal{M}^*$ is a matroid.

Outline

1. Linear Programming
   - Introduction
   - Methods for Solving Linear Programs
2. Polytope with Polynomial Number of Facets
   - Bipartite Matching Polytope
   - Polytopes with Totally Unimodular Coefficient Matrices
3. Polytopes with Efficient Separation Oracles
   - s-t Cut Polytope
   - Spanning Tree Polytope
   - General Graph (Perfect) Matching Polytope
4. Matroid, Matroid Basis and Matroid Intersection Polytopes *
   - Preliminaries on Matroid Theory
   - Matroid Polytope
   - Matroid Basis and Matroid Intersection Polytope

Matroid Polytope

- Given a matroid $\mathcal{M} = (E, \mathcal{I})$
- The matroid polytope for $\mathcal{M}$ is defined as

$$\mathcal{P}_{\mathcal{M}} := \text{conv}(\{\chi^A : A \in \mathcal{I}\}).$$

- Recall: $\chi^A \in \{0,1\}^E, \quad \chi_i^A = \begin{cases} 1 & i \in A \\ 0 & i \notin A \end{cases}$

Matroid Polytope

- Given a matroid $\mathcal{M} = (E, \mathcal{I})$
- The matroid polytope for $\mathcal{M}$ is defined as
$$\mathcal{P}_{\mathcal{M}} := \text{conv}(\{\chi^A : A \in \mathcal{I}\}).$$

- Recall: $\chi^A \in \{0,1\}^E, \quad \chi_i^A = \begin{cases} 1 & i \in A \\ 0 & i \notin A \end{cases}$

Theorem (Matroid Polytope Theorem) For a matroid $\mathcal{M} = (E, \mathcal{I})$, we have

$$\mathcal{P}_{\mathcal{M}} = \left\{ x \in [0,1]^E : x(S) \leq r_{\mathcal{M}}(S), \forall S \subseteq E \right\},$$

where $x(S) := \sum_{i \in S} x_i$ for every $S \subseteq E$.

Proof of Matroid Polytope Theorem

- $\mathcal{Q} := \left\{ x \in [0,1]^E : \sum_{i \in A} x_i \leq r_{\mathcal{M}}(A), \forall A \subseteq E \right\}$
- $\mathcal{Q} \cap \{0,1\}^E = \{\chi^A : A \in \mathcal{I}\}$; it suffices to prove $\mathcal{Q}$ is integral

Proof of Matroid Polytope Theorem

- $\mathcal{Q} := \left\{x \in [0,1]^E : \sum_{i \in A} x_i \leq r_{\mathcal{M}}(A), \forall A \subseteq E\right\}$
- $\mathcal{Q} \cap \{0,1\}^E = \{\chi^A : A \in \mathcal{I}\}$; it suffices to prove $\mathcal{Q}$ is integral
- Focus on the counter example with the smallest $|E|$
- assume some vertex x of $\mathcal{Q}$ is non-integral

Proof of Matroid Polytope Theorem

- $\mathcal{Q} := \left\{ x \in [0,1]^E : \sum_{i \in A} x_i \leq r_{\mathcal{M}}(A), \forall A \subseteq E \right\}$
- $\mathcal{Q} \cap \{0,1\}^E = \{\chi^A : A \in \mathcal{I}\}$; it suffices to prove $\mathcal{Q}$ is integral
- Focus on the counter example with the smallest $|E|$
- assume some vertex x of $\mathcal{Q}$ is non-integral
- If $x_e = 0$ for some $e \in E$, removing e gives a smaller counterexample
- If $x_e = 1$ for some $e \in E$, contracting e gives a smaller counterexample
- So, $x_e \in (0,1)$ for every $e \in E$.

Proof of Matroid Polytope Theorem

Def. We say a set $A \subseteq E$ is tight if $x(A) = r_{\mathcal{M}}(A)$. Let $\mathcal{T}$ be the family of all tight subsets of E.

Lemma If $A, B \in \mathcal{T}$, then both $A \cup B$ and $A \cap B$ are in $\mathcal{T}$.

Proof of Matroid Polytope Theorem

Def. We say a set $A \subseteq E$ is tight if $x(A) = r_{\mathcal{M}}(A)$. Let $\mathcal{T}$ be the family of all tight subsets of E.

Lemma If $A, B \in \mathcal{T}$, then both $A \cup B$ and $A \cap B$ are in $\mathcal{T}$.

Proof.

$$\begin{aligned} x(A) + x(B) &= r_{\mathcal{M}}(A) + r_{\mathcal{M}}(B) \\ &\geq r_{\mathcal{M}}(A \cup B) + r_{\mathcal{M}}(A \cap B) \geq x(A \cup B) + x(A \cap B). \end{aligned}$$

- equality: A and B are tight
- first inequality: $r_{\mathcal{M}}$ is submodular
- second inequality: $x(S) \leq r_{\mathcal{M}}(S)$ for every $S \subseteq E$

But $x(A) + x(B) = x(A \cup B) + x(A \cap B)$. So, both inequalities hold with equality. □

Proof of Matroid Polytope Theorem

Def. A chain is a sequence of subsets $S_1 \subsetneq S_2 \subsetneq \cdots \subsetneq S_t$ of E.

- We use span$(\mathcal{S})$ for span$(\{\chi^S : S \in \mathcal{S}\})$, for any $\mathcal{S} \subseteq \mathcal{T}$.

Proof of Matroid Polytope Theorem

Def. A chain is a sequence of subsets $S_1 \subsetneq S_2 \subsetneq \cdots \subsetneq S_t$ of E.

- We use span$(\mathcal{S})$ for span$(\{\chi^S : S \in \mathcal{S}\})$, for any $\mathcal{S} \subseteq \mathcal{T}$.

Lemma (Key Lemma) Let $\mathcal{C}$ be a longest chain of tight subsets of E (i.e., subsets in $\mathcal{T}$). Then, we have span$(\mathcal{C})$ = span$(\mathcal{T})$.

Proof of Matroid Polytope Theorem

Def. A chain is a sequence of subsets $S_1 \subsetneq S_2 \subsetneq \cdots \subsetneq S_t$ of E.

- We use span$(\mathcal{S})$ for span$(\{\chi^S : S \in \mathcal{S}\})$, for any $\mathcal{S} \subseteq \mathcal{T}$.

Lemma (Key Lemma) Let $\mathcal{C}$ be a longest chain of tight subsets of E (i.e., subsets in $\mathcal{T}$). Then, we have span$(\mathcal{C})$ = span$(\mathcal{T})$.

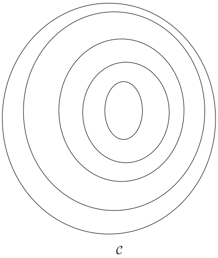
Proof of Key Lemma

- We say two sets B and T conflict with each other, if $B \not\subseteq T$ and $T \not\subseteq B$.
- Define $\tau(B) := \{T \in \mathcal{C} : B \text{ conflicts with } T\}, \forall B$
- Assume span$(\mathcal{C}) \subsetneq$ span$(\mathcal{T})$
- Let $B = \arg\min_{B \in \mathcal{T}, \chi^B \notin \text{span}(\mathcal{C})} |\tau(B)|$

Proof of Matroid Polytope Theorem

Proof of Key Lemma

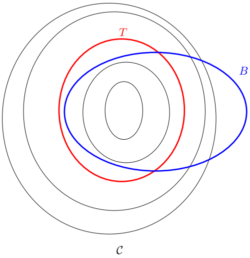
- Let $T \in \mathcal{C}$ be a set contradicting with B;
- We prove $\tau(B \cup T), \tau(B \cap T) \subsetneq \tau(B)$.



Proof of Matroid Polytope Theorem

Proof of Key Lemma

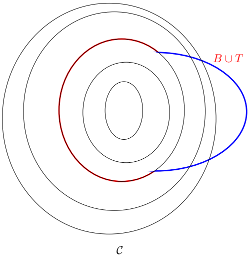
- Let $T \in \mathcal{C}$ be a set contradicting with B;
- We prove $\tau(B \cup T), \tau(B \cap T) \subsetneq \tau(B)$.



Proof of Matroid Polytope Theorem

Proof of Key Lemma

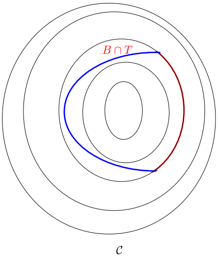
- Let $T \in \mathcal{C}$ be a set contradicting with B;
- We prove $\tau(B \cup T), \tau(B \cap T) \subsetneq \tau(B)$.



Proof of Matroid Polytope Theorem

Proof of Key Lemma

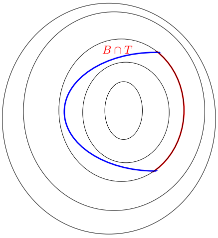
- Let $T \in \mathcal{C}$ be a set contradicting with B;
- We prove $\tau(B \cup T), \tau(B \cap T) \subsetneq \tau(B)$.



Proof of Matroid Polytope Theorem

Proof of Key Lemma

- Let $T \in \mathcal{C}$ be a set contradicting with B;
- We prove $\tau(B \cup T), \tau(B \cap T) \subsetneq \tau(B)$.



- For $\tau(B \cup T) \subseteq \tau(B)$:
  - $S \subsetneq T$: S does not conflict with $B \cup T$, and may conflict with B.
  - $S \supsetneq T$: S not conflict with $B \implies S$ not conflict with $B \cup T$.
- For $\tau(B \cap T) \subseteq \tau(B)$:
  - $S \subsetneq T$: S not conflict with $B \implies S$ not conflict with $B \cap T$.
  - $S \supsetneq T$: S does not conflict with $B \cap T$, and may conflict with B.
- "$\neq$" : B conflicts with T, but $B \cup T$ and $B \cap T$ do not.

Proof of Matroid Polytope Theorem

Proof of Key Lemma

- By our choice of B, we have $\chi^{B\cup T}, \chi^{B\cap T} \in \text{span}(\mathcal{C})$.
- However, as $\chi^B = \chi^{B\cup T} + \chi^{B\cap T} - \chi^T$ and all the three vectors are in $\text{span}(\mathcal{T})$, contradiction with $\chi^B \notin \text{span}(\mathcal{C})$. □

Recall the key lemma:

Lemma (Key Lemma) Let $\mathcal{C}$ be a longest chain of tight subsets of E (i.e., subsets in $\mathcal{T}$). Then, we have $\text{span}(\mathcal{C}) = \text{span}(\mathcal{T})$.

- Therefore, $x \in [0,1]^E$ is defined by the system of linear equations correspondent to $\mathcal{C}$.
- $|\mathcal{C}| = |E|$, the chain $\mathcal{C}$ is of full length.
- The system gives an integer solution x. Contradiction. □

What we proved:

Matroid Polytope

- Given a matroid $\mathcal{M} = (E, \mathcal{I})$
- The matroid polytope for $\mathcal{M}$ is defined as

$$\mathcal{P}_{\mathcal{M}} := \text{conv}(\{\chi^A : A \in \mathcal{I}\}).$$

Theorem (Matroid Polytope Theorem) For a matroid $\mathcal{M} = (E, \mathcal{I})$, we have

$$\mathcal{P}_{\mathcal{M}} = \left\{ x \in [0,1]^E : x(S) \le r_{\mathcal{M}}(S), \forall S \subseteq E \right\},$$

where $x(S) := \sum_{i \in S} x_i$ for every $S \subseteq E$.

Outline

1. Linear Programming
   - Introduction
   - Methods for Solving Linear Programs
2. Polytope with Polynomial Number of Facets
   - Bipartite Matching Polytope
   - Polytopes with Totally Unimodular Coefficient Matrices
3. Polytopes with Efficient Separation Oracles
   - s-t Cut Polytope
   - Spanning Tree Polytope
   - General Graph (Perfect) Matching Polytope
4. Matroid, Matroid Basis and Matroid Intersection Polytopes *
   - Preliminaries on Matroid Theory
   - Matroid Polytope
   - Matroid Basis and Matroid Intersection Polytope

Matroid Basis Polytope

- Given a matroid $\mathcal{M} = (E, \mathcal{I})$
- The matroid basis polytope for $\mathcal{M}$ is defined as

$$\mathcal{P}_{\mathcal{M}}^{\text{basis}} := \text{conv}(\{\chi^A : A \in \mathcal{I}, \text{rank}_{\mathcal{M}}(A) = \text{rank}_{\mathcal{M}}(E)\}).$$

Theorem (Matroid Basis Polytope Theorem) For a matroid $\mathcal{M} = (E, \mathcal{I})$, we have

$$\mathcal{P}_{\mathcal{M}}^{\text{basis}} = \left\{x \in [0,1]^E : x(S) \leq r_{\mathcal{M}}(S), \forall S \subseteq E; x(E) = r_{\mathcal{M}}(E)\right\},$$

where $x(S) := \sum_{i \in S} x_i$ for every $S \subseteq E$.

Matroid Basis Polytope

- Given a matroid $\mathcal{M} = (E, \mathcal{I})$
- The matroid basis polytope for $\mathcal{M}$ is defined as

$$\mathcal{P}^{\text{basis}}_{\mathcal{M}} := \text{conv}(\{\chi^A : A \in \mathcal{I}, \text{rank}_{\mathcal{M}}(A) = \text{rank}_{\mathcal{M}}(E)\}).$$

Theorem (Matroid Basis Polytope Theorem) For a matroid $\mathcal{M} = (E, \mathcal{I})$, we have

$$\mathcal{P}^{\text{basis}}_{\mathcal{M}} = \left\{ x \in [0,1]^E : x(S) \leq r_{\mathcal{M}}(S), \forall S \subseteq E; x(E) = r_{\mathcal{M}}(E) \right\},$$

where $x(S) := \sum_{i \in S} x_i$ for every $S \subseteq E$.

Proof.

- $\mathcal{P}^{\text{basis}}_{\mathcal{M}}$ is a face (not necessarily a facet) of $\mathcal{P}_{\mathcal{M}}$.
- $\mathcal{P}_{\mathcal{M}}$ is integral $\implies$ $\mathcal{P}^{\text{basis}}_{\mathcal{M}}$ is integral □

Recall: Spanning Tree Polytope

Spanning Tree Polytope

- Given a connected graph $G = (V, E)$
- $\mathcal{P}_{\mathrm{ST}} :=$ conv $\left(\left\{\chi^T : T \subseteq E \text{ is a spanning tree of } G\right\}\right)$

Theorem (Spanning Tree Polytope Theorem) $\mathcal{P}_{\mathrm{ST}}$ is the set of vectors $x \in \mathbb{R}^E$ satisfying the following inequalities:

$$\sum_{e \in E} x_e = n - 1$$

$$\sum_{e \in E[S]} x_e \leq |S| - 1 \qquad \forall S \subseteq V, 2 \leq |S| \leq n-1 \qquad (*)$$

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

- Graphic matroid:
  - independent sets $\leftrightarrow$ spanning forests
  - bases $\leftrightarrow$ spanning trees.

- Graphic matroid:
  - independent sets $\leftrightarrow$ spanning forests
  - bases $\leftrightarrow$ spanning trees.
- So, $\mathcal{P}_{\mathrm{ST}}$ is the set of $x \in [0,1]^E$ satisfying

$$x(E') \leq n - \mathsf{CC}(E'), \forall E' \subseteq E; \quad x(E) = n - 1,$$

where $\mathsf{CC}(E')$ is the number of connected components in (V, E').

- Graphic matroid:
  - independent sets $\leftrightarrow$ spanning forests
  - bases $\leftrightarrow$ spanning trees.
- So, $\mathcal{P}_{\mathrm{ST}}$ is the set of $x \in [0,1]^E$ satisfying

$$x(E') \leq n - \mathsf{CC}(E'), \forall E' \subseteq E; \quad x(E) = n - 1,$$

  where $\mathsf{CC}(E')$ is the number of connected components in (V, E').
- It suffices to consider the case where $E' = E[S]$ for some connected set $S \subseteq V$, in which case $n - \mathrm{CC}(E') = |S| - 1$.
- $\implies$ Spanning Tree Polytope Theorem.

Theorem (Matroid Intersection Polytope Theorem) Let $\mathcal{M}_1 = (E, \mathcal{I}_1)$ and $\mathcal{M}_2 = (E, \mathcal{I}_2)$ be two matroids with the common ground set E. Then

$$\text{conv}\big(\{\chi^A : A \in \mathcal{I}_1 \cap \mathcal{I}_2\}\big) = \mathcal{P}_{\mathcal{M}_1} \cap \mathcal{P}_{\mathcal{M}_2}$$

$$= \Big\{x \in [0,1]^E : x(S) \le r_{\mathcal{M}_1}(S), x(S) \le r_{\mathcal{M}_2}(S), \forall S \subseteq E\Big\}.$$

Theorem (Matroid Intersection Polytope Theorem) Let $\mathcal{M}_1 = (E, \mathcal{I}_1)$ and $\mathcal{M}_2 = (E, \mathcal{I}_2)$ be two matroids with the common ground set E. Then

$$\mathrm{conv}\big(\{\chi^A : A \in \mathcal{I}_1 \cap \mathcal{I}_2\}\big) = \mathcal{P}_{\mathcal{M}_1} \cap \mathcal{P}_{\mathcal{M}_2}$$
$$= \Big\{x \in [0,1]^E : x(S) \le r_{\mathcal{M}_1}(S), x(S) \le r_{\mathcal{M}_2}(S), \forall S \subseteq E\Big\}.$$

- We will not prove the theorem.
- A similar theorem works if we require A to be a basis for the matroid $\mathcal{M}_1$ or $\mathcal{M}_2$:

$$\mathrm{conv}\big(\{\chi^A : A \in \mathcal{I}_1 \cap \mathcal{I}_2, \mathrm{rank}_{\mathcal{M}_1}(A) = \mathrm{rank}_{\mathcal{M}_1}(E)\}\big)$$
$$= \mathcal{P}^{\mathrm{basis}}_{\mathcal{M}_1} \cap \mathcal{P}_{\mathcal{M}_2}$$

Applications

Bipartite Matching Polytope

- Given bipartite graph $G = (L \cup R, E)$
- $\mathcal{P}_{\text{BM}} := \text{conv}\big(\{\chi^M : M \text{ is a matching in } G\}\big)$

Theorem $\mathcal{P}_{\text{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \leq 1, \forall v \in L \cup R; \qquad x_e \geq 0, \forall e \in E.$$

Applications

Bipartite Matching Polytope

- Given bipartite graph $G = (L \cup R, E)$
- $\mathcal{P}_{\text{BM}} := \text{conv}\big(\{\chi^M : M \text{ is a matching in } G\}\big)$

Theorem $\mathcal{P}_{\text{BM}}$ is the set of $x \in \mathbb{R}^E$ satisfying the following constraints:

$$\sum_{e \in \delta(v)} x_e \leq 1, \forall v \in L \cup R; \qquad x_e \geq 0, \forall e \in E.$$

- A matching is an independent set of two partition matroids, one for each side of the bipartite graph.
- Matching polytope is intersection of two partition matroid polytopes.

Applications

Arborescence Polytope

- Given a directed graph $G = (V, E)$, a root $r \in V$
- $\mathcal{P}_{\mathrm{Arbo}} :=$ conv$(\{\chi^{E'} : E'$ is an arborescence of G rooted at $r\})$

Applications

Arborescence Polytope

- Given a directed graph $G = (V, E)$, a root $r \in V$
- $\mathcal{P}_{\mathrm{Arbo}} := \mathrm{conv}(\{\chi^{E'} : E' \text{ is an arborescence of } G \text{ rooted at } r\})$

- We define two matroids:
  - Graphic Matroid: we ignore the directions of G, and require E' to be a spanning forest
  - Partition Matroid: we require every vertex other than r has in-degree at most 1

Applications

Arborescence Polytope

- Given a directed graph $G = (V, E)$, a root $r \in V$
- $\mathcal{P}_{\mathrm{Arbo}} := \mathrm{conv}(\{\chi^{E'} : E' \text{ is an arborescence of } G \text{ rooted at } r\})$

- We define two matroids:
  - Graphic Matroid: we ignore the directions of G, and require E' to be a spanning forest
  - Partition Matroid: we require every vertex other than r has in-degree at most 1
- E' is an arborescence if it is a basis of both polytopes.

Summary

- linear programming, simplex method, interior point method, ellipsoid method
- Polytopes with totally-unimodular coefficient matrix:
  - integral LP polytopes: bipartite matching polytope, s-t flow polytope, weighted interval scheduling polytope

Summary

- linear programming, simplex method, interior point method, ellipsoid method
- Polytopes with totally-unimodular coefficient matrix:
  - integral LP polytopes: bipartite matching polytope, s-t flow polytope, weighted interval scheduling polytope
- Matroid Polytope