Approximation Algorithms
LP Relaxation and Rounding

Advanced Algorithms

Nanjing University, Fall 2018

Linear Programming (LP)

LP: Optimize (minimize/maximize) a *linear* objective function, subject to *linear* equality and *linear* inequality constraints.

Linear Programming (LP)

LP: Optimize (minimize/maximize) a *linear* objective function, subject to *linear* equality and *linear* inequality constraints.

- A factory produces two types of chocolates A and B.
- Revenue for A is 1 per kg and the revenue for B is 6 per kg.
- The demand for A is 200kg per day and the demand for B is 300kg per day.
- The factory can produce at most 400kg each day.
- How to arrange production to maximize revenue?

Linear Programming (LP)

LP: Optimize (minimize/maximize) a *linear* objective function, subject to *linear* equality and *linear* inequality constraints.

- A factory produces two types of chocolates A and B.
- Revenue for A is 1 per kg and the revenue for B is 6 per kg.
- The demand for A is 200kg per day and the demand for B is 300kg per day.
- The factory can produce at most 400kg each day.
- How to arrange production to maximize revenue?

Objective function:
$\max x_1 + 6x_2$

Linear Programming (LP)

LP: Optimize (minimize/maximize) a *linear* objective function, subject to *linear* equality and *linear* inequality constraints.

- A factory produces two types of chocolates A and B.
- Revenue for A is 1 per kg and the revenue for B is 6 per kg.
- The demand for A is 200kg per day and the demand for B is 300kg per day.
- The factory can produce at most 400kg each day.
- How to arrange production to maximize revenue?

Objective function:
$\max x_1 + 6x_2$

Constraints:
$x_1 \leq 200$
$x_2 \leq 300$
$x_1 + x_2 \leq 400$
$x_1 \geq 0, x_2 \geq 0$

General Form of LP

Objective function:
$\max x_1 + 6x_2$

Constraints:
$x_1 \leq 200$
$x_2 \leq 300$
$x_1 + x_2 \leq 400$
$x_1 \geq 0, x_2 \geq 0$

General Form of LP

Objective function:
$\max x_1 + 6x_2$

Constraints:
$x_1 \leq 200$
$x_2 \leq 300$
$x_1 + x_2 \leq 400$
$x_1 \geq 0, x_2 \geq 0$

General form of LP:

matrix $A = \{a_{ij}\}_{m \times n}$
sets $M \subseteq [m]$ and $N \subseteq [n]$

$$
\begin{array}{lll}
\text{minimize} & \vec{c}^T \vec{x} & \\
\text{subject to} & \vec{a_i}\, \vec{x} \geq b_i & i \in M \\
 & \vec{a_i}\, \vec{x} = b_i & i \in \overline{M} \\
 & x_j \geq 0 & j \in N \\
 & x_j \text{ free} & j \in \overline{N}
\end{array}
$$

General Form of LP

Objective function:

$\max x_1 + 6x_2$

Constraints:

$x_1 \leq 200$

$x_2 \leq 300$

$x_1 + x_2 \leq 400$

$x_1 \geq 0, x_2 \geq 0$

$\vec{x} = [x_1, x_2]^T$

General form of LP:

matrix $A = \{a_{ij}\}_{m \times n}$

sets $M \subseteq [m]$ and $N \subseteq [n]$

minimize	$\vec{c}^T \vec{x}$	
subject to	$\overrightarrow{a_i}\, \vec{x} \geq b_i$	$i \in M$
	$\overrightarrow{a_i}\, \vec{x} = b_i$	$i \in \overline{M}$
	$x_j \geq 0$	$j \in N$
	x_j free	$j \in \overline{N}$

General Form of LP

Objective function:

$\max x_1 + 6x_2$

Constraints:

$x_1 \leq 200$

$x_2 \leq 300$

$x_1 + x_2 \leq 400$

$x_1 \geq 0, x_2 \geq 0$

$\vec{x} = [x_1, x_2]^T \qquad \vec{c} = [-1, -6]^T$

General form of LP:

matrix $A = \{a_{ij}\}_{m \times n}$

sets $M \subseteq [m]$ and $N \subseteq [n]$

minimize	$\vec{c}^T \vec{x}$	
subject to	$\overrightarrow{a_i}\, \vec{x} \geq b_i$	$i \in M$
	$\overrightarrow{a_i}\, \vec{x} = b_i$	$i \in \overline{M}$
	$x_j \geq 0$	$j \in N$
	x_j free	$j \in \overline{N}$

General Form of LP

Objective function:

$\max x_1 + 6x_2$

Constraints:

$x_1 \leq 200$

$x_2 \leq 300$

$x_1 + x_2 \leq 400$

$x_1 \geq 0, x_2 \geq 0$

$\vec{x} = [x_1, x_2]^T \qquad \vec{c} = [-1, -6]^T$

$n = 2, m = 3$

General form of LP:

matrix $A = \{a_{ij}\}_{m \times n}$

sets $M \subseteq [m]$ and $N \subseteq [n]$

minimize	$\vec{c}^T \vec{x}$	
subject to	$\overrightarrow{a_i}\,\vec{x} \geq b_i$	$i \in M$
	$\overrightarrow{a_i}\,\vec{x} = b_i$	$i \in \overline{M}$
	$x_j \geq 0$	$j \in N$
	x_j free	$j \in \overline{N}$

General Form of LP

Objective function:

$\max x_1 + 6x_2$

Constraints:

$x_1 \leq 200$

$x_2 \leq 300$

$x_1 + x_2 \leq 400$

$x_1 \geq 0, x_2 \geq 0$

$\vec{x} = [x_1, x_2]^T \qquad \vec{c} = [-1, -6]^T$

$n = 2, m = 3$

$N = \{1,2\}, M = \{1,2,3\}$

General form of LP:

matrix $A = \{a_{ij}\}_{m \times n}$

sets $M \subseteq [m]$ and $N \subseteq [n]$

minimize	$\vec{c}^T \vec{x}$	
subject to	$\vec{a_i}\, \vec{x} \geq b_i$	$i \in M$
	$\vec{a_i}\, \vec{x} = b_i$	$i \in \overline{M}$
	$x_j \geq 0$	$j \in N$
	x_j free	$j \in \overline{N}$

General Form of LP

Objective function:
$\max x_1 + 6x_2$

Constraints:
$x_1 \le 200$
$x_2 \le 300$
$x_1 + x_2 \le 400$
$x_1 \ge 0, x_2 \ge 0$

$\vec{x} = [x_1, x_2]^T \qquad \vec{c} = [-1, -6]^T$

$n = 2, m = 3$

$N = \{1,2\}, M = \{1,2,3\}$

$$A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ -1 & -1 \end{bmatrix}, \vec{b} = \begin{bmatrix} -200 \\ -300 \\ -400 \end{bmatrix}$$

General form of LP:

matrix $A = \{a_{ij}\}_{m \times n}$

sets $M \subseteq [m]$ and $N \subseteq [n]$

minimize	$\vec{c}^T \vec{x}$	
subject to	$\vec{a_i}\, \vec{x} \ge b_i$	$i \in M$
	$\vec{a_i}\, \vec{x} = b_i$	$i \in \overline{M}$
	$x_j \ge 0$	$j \in N$
	x_j free	$j \in \overline{N}$

General form of LP:

matrix $A = \{a_{ij}\}_{m \times n}$

sets $M \subseteq [m]$ and $N \subseteq [n]$

minimize	$\vec{c}^T \vec{x}$	
subject to	$\overrightarrow{a_i}\, \vec{x} \geq b_i$	$i \in M$
	$\overrightarrow{a_i}\, \vec{x} = b_i$	$i \in \overline{M}$
	$x_j \geq 0$	$j \in N$
	x_j free	$j \in \overline{N}$

Canonical Form of LP

General form of LP:

matrix $A = \{a_{ij}\}_{m \times n}$

sets $M \subseteq [m]$ and $N \subseteq [n]$

minimize $\vec{c}^T \vec{x}$

subject to

$\vec{a_i} \vec{x} \geq b_i \quad i \in M$

$\vec{a_i} \vec{x} = b_i \quad i \in \overline{M}$

$x_j \geq 0 \quad j \in N$

x_j free $\quad j \in \overline{N}$

Canonical form of LP:

matrix $A = \{a_{ij}\}_{m \times n}$

vectors $\vec{b}$ and $\vec{c}$

minimize $\vec{c}^T \vec{x}$

subject to

$A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$

Canonical Form of LP

General form of LP:

matrix $A = \{a_{ij}\}_{m \times n}$

sets $M \subseteq [m]$ and $N \subseteq [n]$

minimize	$\vec{c}^T \vec{x}$	
subject to	$\overrightarrow{a_i}\,\vec{x} \geq b_i$	$i \in M$
	$\overrightarrow{a_i}\,\vec{x} = b_i$	$i \in \overline{M}$
	$x_j \geq 0$	$j \in N$
	x_j free	$j \in \overline{N}$

Canonical form of LP:

matrix $A = \{a_{ij}\}_{m \times n}$

vectors $\vec{b}$ and $\vec{c}$

minimize	$\vec{c}^T \vec{x}$
subject to	$A\vec{x} \geq \vec{b}$
	$\vec{x} \geq 0$

Canonical Form of LP

General form of LP:

matrix $A = \{a_{ij}\}_{m\times n}$

sets $M \subseteq [m]$ and $N \subseteq [n]$

minimize	$\vec{c}^T \vec{x}$	
subject to	$\overrightarrow{a_i}\,\vec{x} \geq b_i$	$i \in M$
	$\overrightarrow{a_i}\,\vec{x} = b_i$	$i \in \overline{M}$
	$x_j \geq 0$	$j \in N$
	x_j free	$j \in \overline{N}$

Canonical form of LP:

matrix $A = \{a_{ij}\}_{m\times n}$

vectors $\vec{b}$ and $\vec{c}$

minimize	$\vec{c}^T \vec{x}$
subject to	$A\vec{x} \geq \vec{b}$
	$\vec{x} \geq 0$

$$\overrightarrow{a_i}\,\vec{x} = b_i \iff \begin{cases} \overrightarrow{a_i}\,\vec{x} \geq b_i \\ -\overrightarrow{a_i}\,\vec{x} \geq -b_i \end{cases}$$

Canonical Form of LP

General form of LP:

matrix $A = \{a_{ij}\}_{m \times n}$

sets $M \subseteq [m]$ and $N \subseteq [n]$

minimize	$\vec{c}^T \vec{x}$	
subject to	$\overrightarrow{a_i}\, \vec{x} \geq b_i$	$i \in M$
	$\overrightarrow{a_i}\, \vec{x} = b_i$	$i \in \overline{M}$
	$x_j \geq 0$	$j \in N$
	x_j free	$j \in \overline{N}$

Canonical form of LP:

matrix $A = \{a_{ij}\}_{m \times n}$

vectors $\vec{b}$ and $\vec{c}$

minimize	$\vec{c}^T \vec{x}$
subject to	$A\vec{x} \geq \vec{b}$
	$\vec{x} \geq 0$

$$\overrightarrow{a_i}\, \vec{x} = b_i \iff \begin{cases} \overrightarrow{a_i}\, \vec{x} \geq b_i \\ -\overrightarrow{a_i}\, \vec{x} \geq -b_i \end{cases}$$

Canonical Form of LP

General form of LP:

matrix $A = \{a_{ij}\}_{m\times n}$

sets $M \subseteq [m]$ and $N \subseteq [n]$

minimize	$\vec{c}^T\vec{x}$	
subject to	$\overrightarrow{a_i}\,\vec{x} \geq b_i$	$i \in M$
	$\overrightarrow{a_i}\,\vec{x} = b_i$	$i \in \overline{M}$
	$x_j \geq 0$	$j \in N$
	x_j free	$j \in \overline{N}$

Canonical form of LP:

matrix $A = \{a_{ij}\}_{m\times n}$

vectors $\vec{b}$ and $\vec{c}$

minimize	$\vec{c}^T\vec{x}$
subject to	$A\vec{x} \geq \vec{b}$
	$\vec{x} \geq 0$

$$\overrightarrow{a_i}\,\vec{x} = b_i \iff \begin{cases} \overrightarrow{a_i}\,\vec{x} \geq b_i \\ -\overrightarrow{a_i}\,\vec{x} \geq -b_i \end{cases}$$

$$x_j \text{ unconstrained} \iff \begin{cases} x_j^+ \geq 0 \\ x_j^- \geq 0 \\ x_j = x_j^+ - x_j^- \end{cases}$$

Convex Polytope

Canonical form of LP:

minimize $\vec{c}^T\vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$

Hyperplane:

Subspace of dimension $n - 1$

$\sum_{j=1}^{n} a_{ij}x_j = b_i$

Halfspace:

$\sum_{j=1}^{n} a_{ij}x_j \geq b_i$

(Convex) Polytope:

Bounded and nonempty intersection of finite number of halfspaces

Convex Polytope

Canonical form of LP:

minimize $\vec{c}^T \vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$

Hyperplane:

Subspace of dimension $n-1$

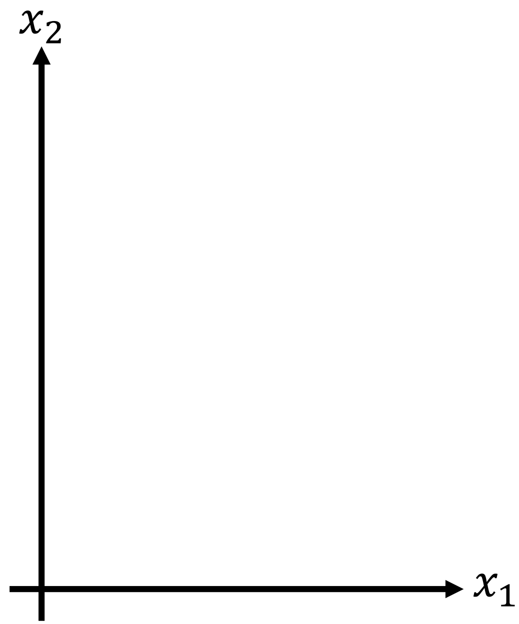
$\sum_{j=1}^{n} a_{ij} x_j = b_i$

Halfspace:

$\sum_{j=1}^{n} a_{ij} x_j \geq b_i$

(Convex) Polytope:

Bounded and nonempty intersection of finite number of halfspaces



Objective function:

$\max x_1 + 6x_2$

Constraints:

$x_1 \leq 200$

$x_2 \leq 300$

$x_1 + x_2 \leq 400$

$x_1 \geq 0, x_2 \geq 0$

Convex Polytope

Canonical form of LP:

minimize $\vec{c}^T \vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$

Hyperplane:

Subspace of dimension $n-1$

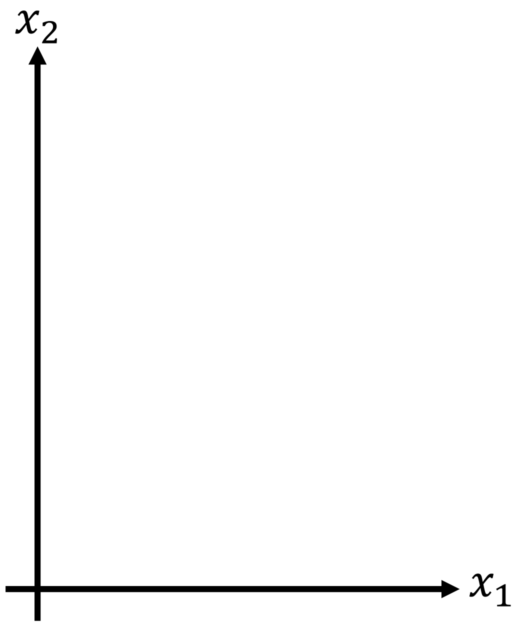
$\sum_{j=1}^{n} a_{ij} x_j = b_i$

Halfspace:

$\sum_{j=1}^{n} a_{ij} x_j \geq b_i$

(Convex) Polytope:

Bounded and nonempty intersection of finite number of halfspaces



Objective function:

$\max x_1 + 6x_2$

Constraints:

$x_1 \leq 200$

$x_2 \leq 300$

$x_1 + x_2 \leq 400$

$x_1 \geq 0, x_2 \geq 0$

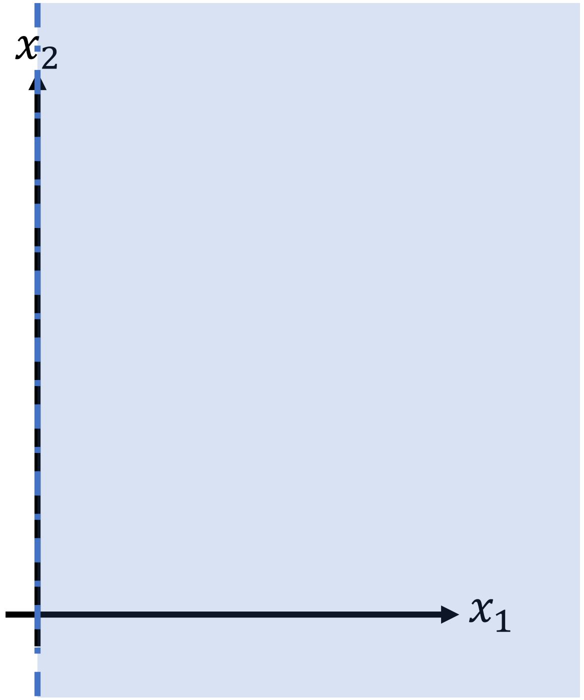
Convex Polytope

Canonical form of LP:

minimize $\vec{c}^T \vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$



Hyperplane:

Subspace of dimension $n-1$

$\sum_{j=1}^{n} a_{ij} x_j = b_i$

Halfspace:

$\sum_{j=1}^{n} a_{ij} x_j \geq b_i$

(Convex) Polytope:

Bounded and nonempty intersection of finite number of halfspaces

Objective function:

$\max x_1 + 6x_2$

Constraints:

$x_1 \leq 200$

$x_2 \leq 300$

$x_1 + x_2 \leq 400$

$x_1 \geq 0, x_2 \geq 0$

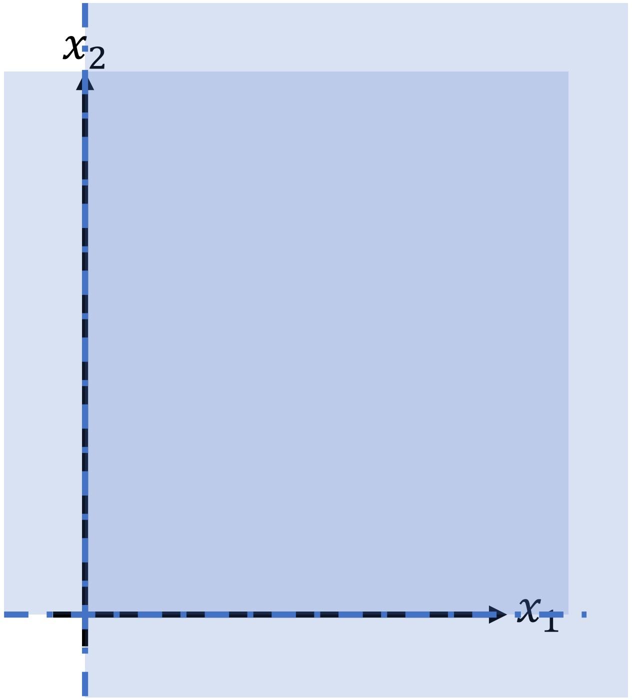
Convex Polytope

Canonical form of LP:

minimize $\vec{c}^T\vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$



Hyperplane:

Subspace of dimension $n-1$

$\sum_{j=1}^{n} a_{ij}x_j = b_i$

Halfspace:

$\sum_{j=1}^{n} a_{ij}x_j \geq b_i$

(Convex) Polytope:

Bounded and nonempty intersection of finite number of halfspaces

Objective function:

$\max x_1 + 6x_2$

Constraints:

$x_1 \leq 200$

$x_2 \leq 300$

$x_1 + x_2 \leq 400$

$x_1 \geq 0, x_2 \geq 0$

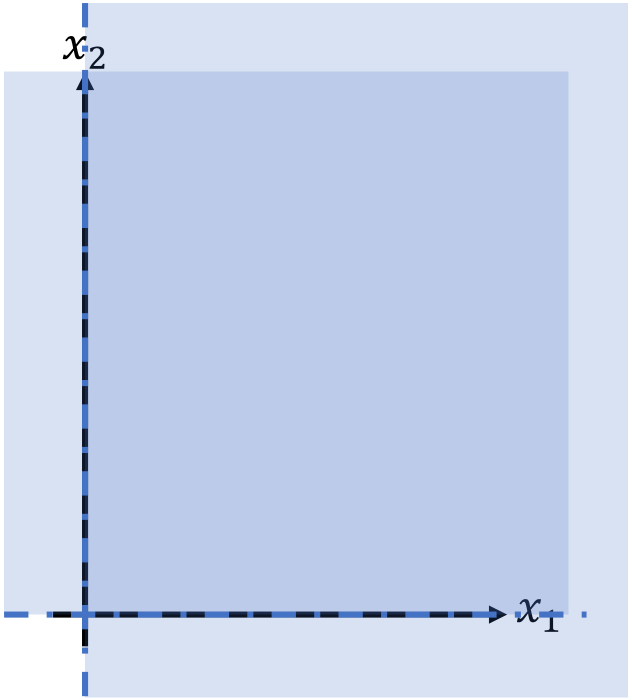
Convex Polytope

Canonical form of LP:

minimize $\vec{c}^T \vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$



Hyperplane:

Subspace of dimension $n - 1$

$\sum_{j=1}^{n} a_{ij} x_j = b_i$

Halfspace:

$\sum_{j=1}^{n} a_{ij} x_j \geq b_i$

(Convex) Polytope:

Bounded and nonempty intersection of finite number of halfspaces

Objective function:

$\max x_1 + 6x_2$

Constraints:

$x_1 \leq 200$

$x_2 \leq 300$

$x_1 + x_2 \leq 400$

$x_1 \geq 0, x_2 \geq 0$

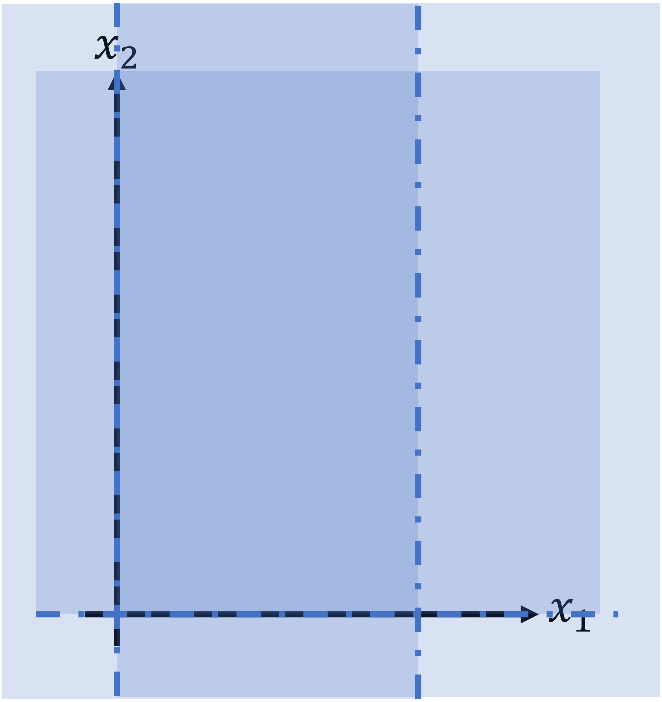
Convex Polytope

Canonical form of LP:

minimize $\vec{c}^T \vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$



Hyperplane:

Subspace of dimension $n-1$

$\sum_{j=1}^{n} a_{ij} x_j = b_i$

Halfspace:

$\sum_{j=1}^{n} a_{ij} x_j \geq b_i$

(Convex) Polytope:

Bounded and nonempty intersection of finite number of halfspaces

Objective function:

$\max x_1 + 6x_2$

Constraints:

$x_1 \leq 200$

$x_2 \leq 300$

$x_1 + x_2 \leq 400$

$x_1 \geq 0, x_2 \geq 0$

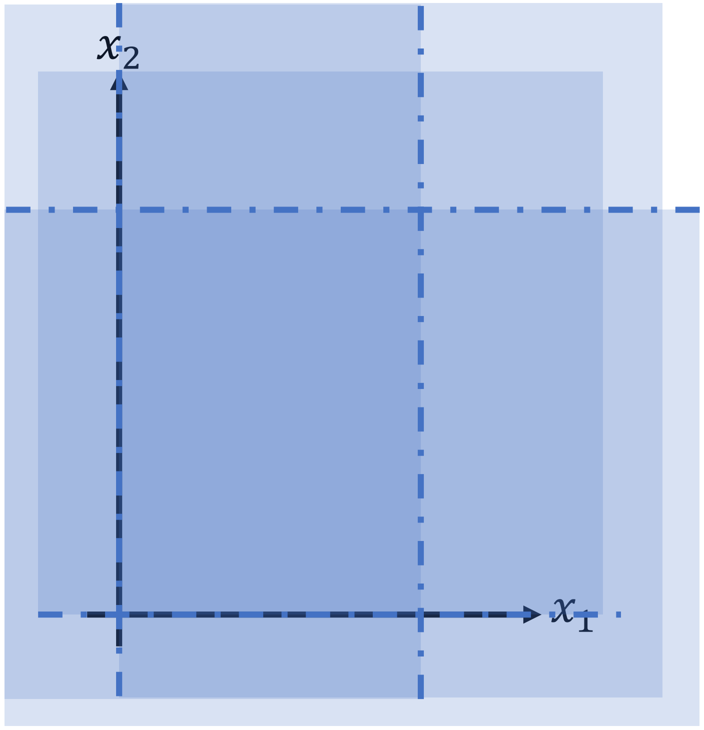
Convex Polytope

Canonical form of LP:

minimize $\vec{c}^T \vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$



Hyperplane:

Subspace of dimension $n - 1$

$\sum_{j=1}^{n} a_{ij} x_j = b_i$

Halfspace:

$\sum_{j=1}^{n} a_{ij} x_j \geq b_i$

(Convex) Polytope:

Bounded and nonempty intersection of finite number of halfspaces

Objective function:

$\max x_1 + 6x_2$

Constraints:

$x_1 \leq 200$

$x_2 \leq 300$

$x_1 + x_2 \leq 400$

$x_1 \geq 0, x_2 \geq 0$

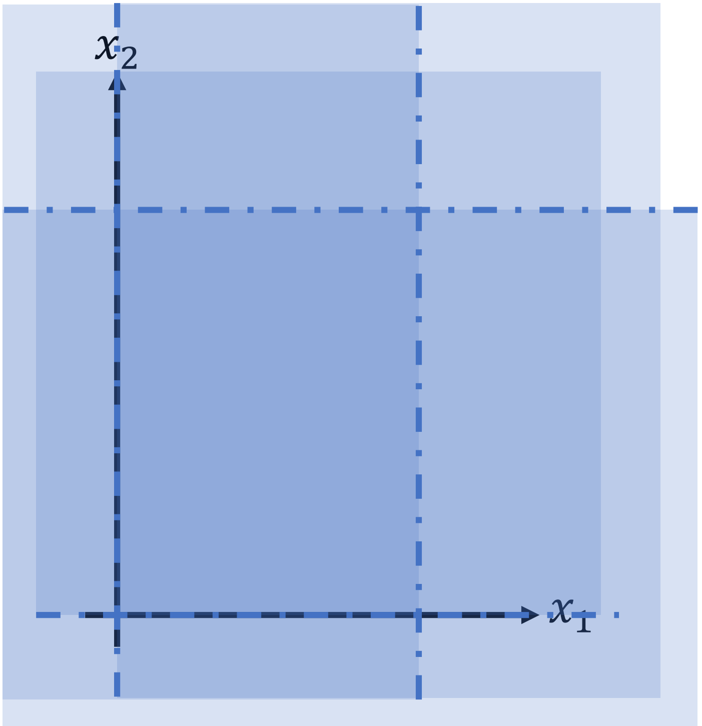
Convex Polytope

Canonical form of LP:

minimize $\vec{c}^T \vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$



Hyperplane:

Subspace of dimension $n - 1$

$\sum_{j=1}^{n} a_{ij} x_j = b_i$

Halfspace:

$\sum_{j=1}^{n} a_{ij} x_j \geq b_i$

(Convex) Polytope:

Bounded and nonempty intersection of finite number of halfspaces

Objective function:

$\max x_1 + 6x_2$

Constraints:

$x_1 \leq 200$

$x_2 \leq 300$

$x_1 + x_2 \leq 400$

$x_1 \geq 0, x_2 \geq 0$

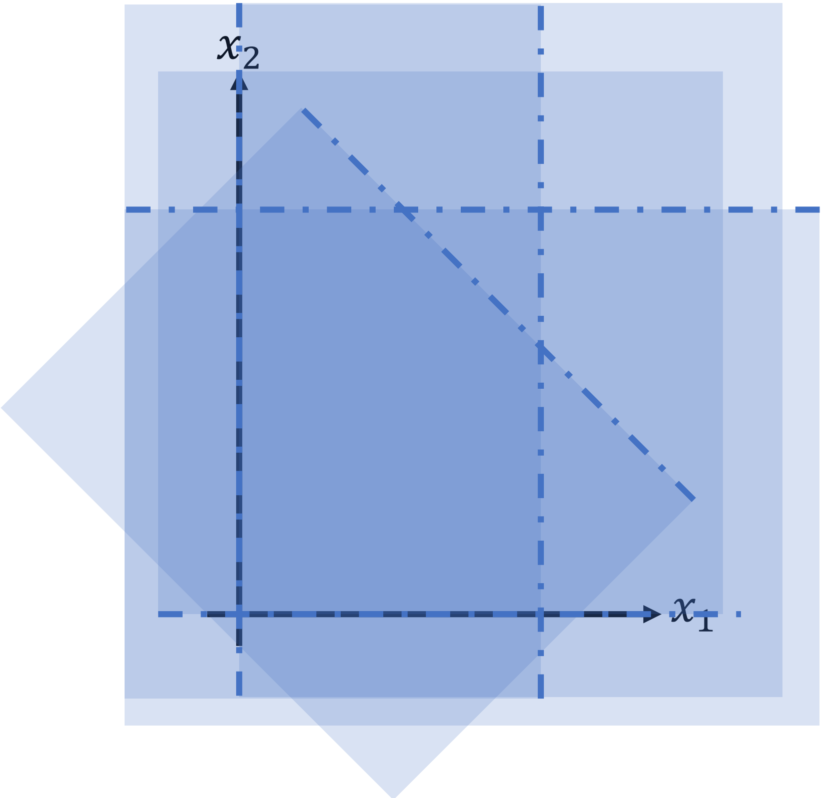
Convex Polytope

Canonical form of LP:

minimize $\vec{c}^T \vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$



Hyperplane:

Subspace of dimension $n-1$

$\sum_{j=1}^{n} a_{ij} x_j = b_i$

Halfspace:

$\sum_{j=1}^{n} a_{ij} x_j \geq b_i$

(Convex) Polytope:

Bounded and nonempty intersection of finite number of halfspaces

Objective function:

$\max x_1 + 6x_2$

Constraints:

$x_1 \leq 200$

$x_2 \leq 300$

$x_1 + x_2 \leq 400$

$x_1 \geq 0, x_2 \geq 0$

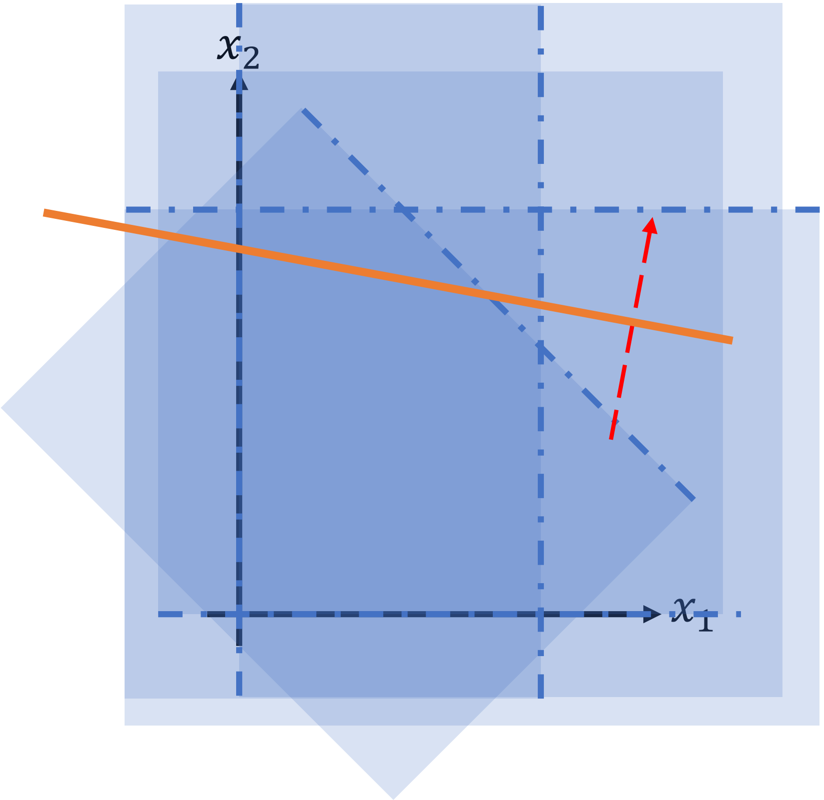
Convex Polytope

Canonical form of LP:

minimize $\vec{c}^T \vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$



Hyperplane:

Subspace of dimension $n-1$

$\sum_{j=1}^{n} a_{ij} x_j = b_i$

Halfspace:

$\sum_{j=1}^{n} a_{ij} x_j \geq b_i$

(Convex) Polytope:

Bounded and nonempty intersection of finite number of halfspaces

Objective function:

$\max x_1 + 6x_2$

Constraints:

$x_1 \leq 200$

$x_2 \leq 300$

$x_1 + x_2 \leq 400$

$x_1 \geq 0, x_2 \geq 0$

Integer Linear Programming (ILP)

Canonical form of LP:

matrix $A = \{a_{ij}\}_{m \times n}$

vectors $\vec{b}$ and $\vec{c}$

minimize $\vec{c}^T \vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$

Canonical form of ILP:

integer matrix $A = \{a_{ij}\}_{m \times n}$

integer vectors $\vec{b}$ and $\vec{c}$

minimize $\vec{c}^T \vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$

$\vec{x} \in \mathbb{Z}^n$

Integer Linear Programming (ILP)

Canonical form of LP:

matrix $A = \{a_{ij}\}_{m\times n}$

vectors $\vec{b}$ and $\vec{c}$

minimize $\vec{c}^T\vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$

Canonical form of ILP:

integer matrix $A = \{a_{ij}\}_{m\times n}$

integer vectors $\vec{b}$ and $\vec{c}$

minimize $\vec{c}^T\vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$

$\vec{x} \in \mathbb{Z}^n$

Integer Linear Programming (ILP)

Canonical form of LP:

matrix $A = \{a_{ij}\}_{m\times n}$

vectors $\vec{b}$ and $\vec{c}$

minimize $\vec{c}^T\vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$

Canonical form of ILP:

integer matrix $A = \{a_{ij}\}_{m\times n}$

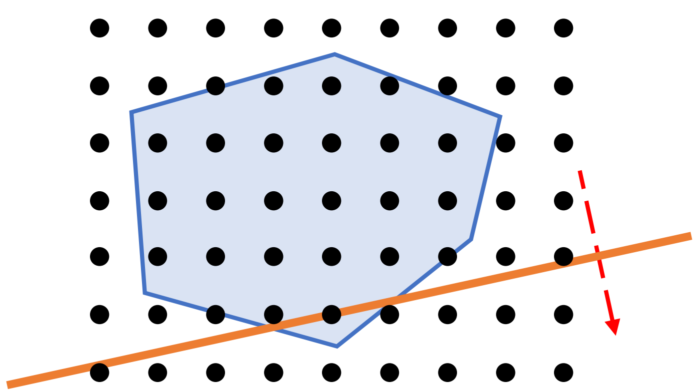
integer vectors $\vec{b}$ and $\vec{c}$

minimize $\vec{c}^T\vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$

$\vec{x} \in \mathbb{Z}^n$



Integer Linear Programming (ILP)

Canonical form of LP:

matrix $A = \{a_{ij}\}_{m\times n}$

vectors $\vec{b}$ and $\vec{c}$

minimize $\vec{c}^T\vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$

Canonical form of ILP:

integer matrix $A = \{a_{ij}\}_{m\times n}$

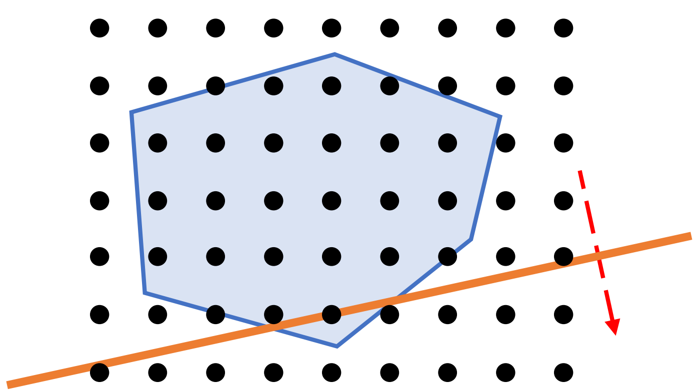
integer vectors $\vec{b}$ and $\vec{c}$

minimize $\vec{c}^T\vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$

$\vec{x} \in \mathbb{Z}^n$



LP is polynomial time solvable,
ILP is NP-hard.

Vertex Cover

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover: Find smallest $C \subseteq V$ s.t. $\forall e \in E: e \cap C \neq \emptyset$.

Instance of set cover
with frequency = 2
for all elements.

Vertex Cover

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover: Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

- This problem is NP-hard.
- $\ln n$-approx. alg. by greedy set cover.
- 2-approx. alg. by finding *maximal* matching.
- Assuming the *unique game conjecture*, there is no poly-time $(2 - \epsilon)$-approx. alg.

Vertex Cover as ILP

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover: Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

Vertex Cover as ILP

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover: Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

Define a 0-1 variable x_v for each vertex v denoting whether $v \in C$

Vertex Cover as ILP

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover: Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

Define a 0-1 variable x_v for each vertex v denoting whether $v \in C$

Objective function:

Vertex Cover as ILP

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover: Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

Define a 0-1 variable x_v for each vertex v denoting whether $v \in C$

Objective function: $\min \sum_{v \in V} x_v$

Vertex Cover as ILP

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover: Find smallest $C \subseteq V$ s.t. $\forall e \in E: e \cap C \neq \emptyset$.

Define a 0-1 variable x_v for each vertex v denoting whether $v \in C$

Objective function: $\min \sum_{v \in V} x_v$

Constraints:

Vertex Cover as ILP

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover: Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

Define a 0-1 variable x_v for each vertex v denoting whether $v \in C$

Objective function: $\min \sum_{v \in V} x_v$

Constraints: $\sum_{v \in e} x_v \geq 1, \quad e \in E$

Vertex Cover as ILP

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover: Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

Define a 0-1 variable x_v for each vertex v denoting whether $v \in C$

Objective function: $\min \sum_{v \in V} x_v$

Constraints: $\sum_{v \in e} x_v \geq 1, \quad e \in E$

$$x_v \in \{0,1\}, \qquad v \in V$$

Vertex Cover as ILP

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover: Find smallest $C \subseteq V$ s.t. $\forall e \in E: e \cap C \neq \emptyset$.

Define a 0-1 variable x_v for each vertex v denoting whether $v \in C$

Objective function: $\min \sum_{v \in V} x_v$

Constraints: $\sum_{v \in e} x_v \geq 1, \quad e \in E$

$x_v \in \{0,1\}, \quad v \in V$

Canonical form of ILP:

integer matrix $A = \{a_{ij}\}_{m \times n}$

integer vectors $\vec{b}$ and $\vec{c}$

minimize $\vec{c}^T \vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$

$\vec{x} \in \mathbb{Z}^n$

Vertex Cover as ILP

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover: Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

Define a 0-1 variable x_v for each vertex v denoting whether $v \in C$

Objective function: $\min \sum_{v \in V} x_v$

Constraints: $\sum_{v \in e} x_v \geq 1, \quad e \in E$

$x_v \in \{0,1\}, \quad v \in V$

ILP is NP-hard!

Canonical form of ILP:

integer matrix $A = \{a_{ij}\}_{m \times n}$

integer vectors $\vec{b}$ and $\vec{c}$

minimize $\vec{c}^T \vec{x}$

subject to $A\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$

$\vec{x} \in \mathbb{Z}^n$

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover: Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

Define a 0-1 variable x_v for each vertex v denoting whether $v \in C$

Objective function: $\min \sum_{v \in V} x_v$

Constraints: $\sum_{v \in e} x_v \geq 1, \quad e \in E$

$x_v \in \{0,1\}, \quad v \in V$

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover: Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

~~Define a 0-1 variable x_v for each vertex v denoting whether $v \in C$~~

Objective function: $\min \sum_{v \in V} x_v$

Constraints: $\sum_{v \in e} x_v \geq 1, \quad e \in E$

~~$x_v \in \{0,1\}, \quad v \in V$~~

$x_v \in [0,1], \quad v \in V$

LP Relaxation for Vertex Cover

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover: Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

~~Define a 0-1 variable x_v for each vertex v denoting whether $v \in C$~~

Objective function: $\min \sum_{v \in V} x_v$

Constraints: $\sum_{v \in e} x_v \geq 1, \quad e \in E$

~~$x_v \in \{0,1\}, \quad v \in V$~~

$x_v \in [0,1], \quad v \in V$

LP is poly-time solvable,
so we can solve LP relaxation of vertex cover in poly-time.

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover (VC): Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

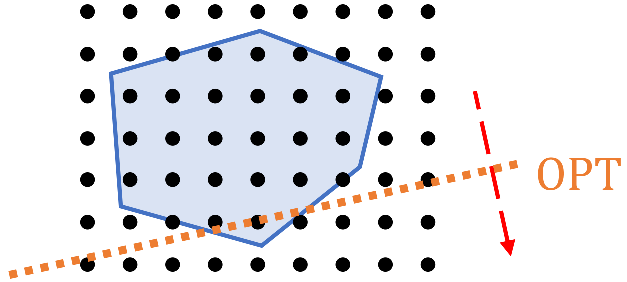
<u>VC as ILP:</u>

$$\min \sum_{v \in V} x_v$$
$$\text{s.t.} \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in \{0,1\}, \quad v \in V$$

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover (VC): Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

<u>VC as ILP:</u>

$$\min \sum_{v \in V} x_v$$
$$\text{s.t.} \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in \{0,1\}, \quad v \in V$$



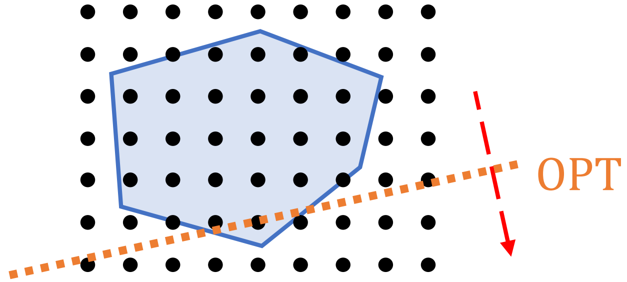
Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover (VC): Find smallest $C \subseteq V$ s.t. $\forall e \in E: e \cap C \neq \emptyset$.

VC as ILP:

$$\min \sum_{v \in V} x_v$$
$$\text{s.t. } \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in \{0,1\}, \quad v \in V$$

LP relaxation of VC:

$$\min \sum_{v \in V} x_v$$
$$\text{s.t. } \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in [0,1], \quad v \in V$$



Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover (VC): Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

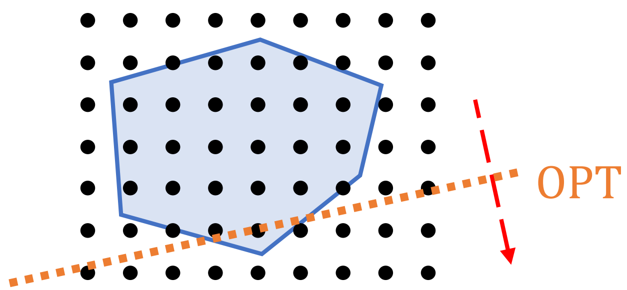
<u>VC as ILP:</u>

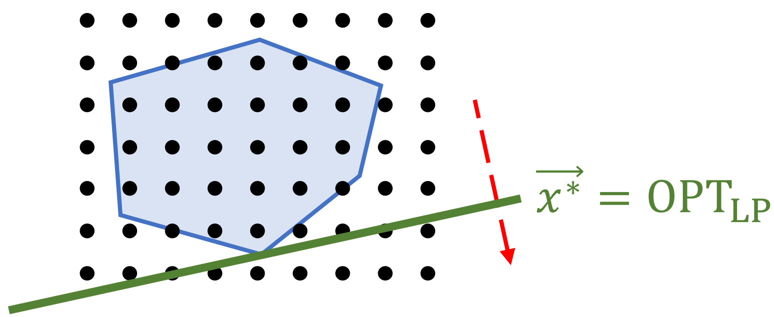
$$\begin{aligned} \min \textstyle\sum_{v\in V} x_v & \\ \text{s.t. } \textstyle\sum_{v\in e} x_v \geq 1, & \quad e \in E \\ x_v \in \{0,1\}, & \quad v \in V \end{aligned}$$

<u>LP relaxation of VC:</u>

$$\begin{aligned} \min \textstyle\sum_{v\in V} x_v & \\ \text{s.t. } \textstyle\sum_{v\in e} x_v \geq 1, & \quad e \in E \\ x_v \in [0,1], & \quad v \in V \end{aligned}$$

fractional and optimal solution
$\overrightarrow{x^*} \in [0,1]^{|V|}$
(found in poly-time)





Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover (VC): Find smallest $C \subseteq V$ s.t. $\forall e \in E: e \cap C \neq \emptyset$.

<u>VC as ILP:</u>

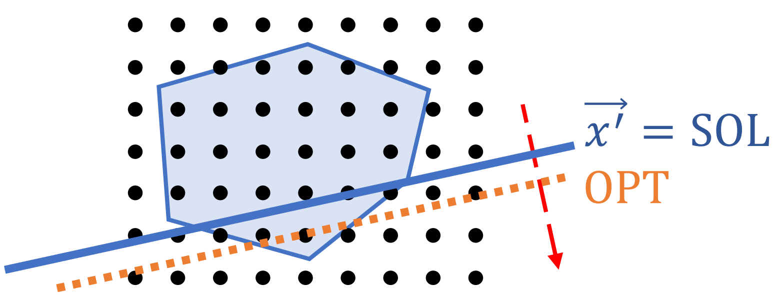
$$\begin{aligned} &\min \textstyle\sum_{v \in V} x_v \\ &\text{s.t. } \textstyle\sum_{v \in e} x_v \geq 1, \quad e \in E \\ &\qquad x_v \in \{0,1\}, \quad v \in V \end{aligned}$$

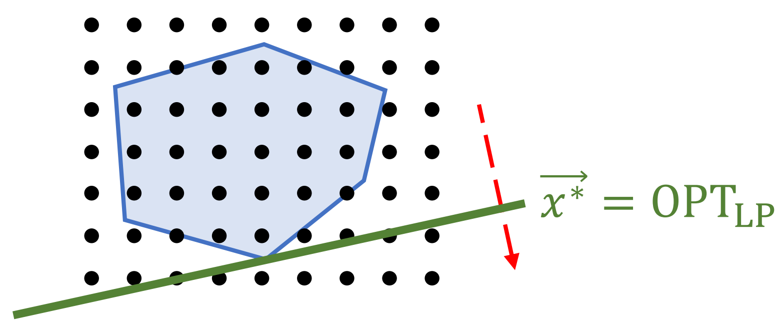
<u>LP relaxation of VC:</u>

$$\begin{aligned} &\min \textstyle\sum_{v \in V} x_v \\ &\text{s.t. } \textstyle\sum_{v \in e} x_v \geq 1, \quad e \in E \\ &\qquad x_v \in [0,1], \quad v \in V \end{aligned}$$

integral and feasible solution
$\overrightarrow{x'} \in [0,1]^{|V|}$
(not too bad, found in poly-time)

fractional and optimal solution
$\overrightarrow{x^*} \in [0,1]^{|V|}$
(found in poly-time)





Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover (VC): Find smallest $C \subseteq V$ s.t. $\forall e \in E: e \cap C \neq \emptyset$.

VC as ILP:

$$\min \sum_{v \in V} x_v$$
$$\text{s.t. } \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in \{0,1\}, \quad v \in V$$

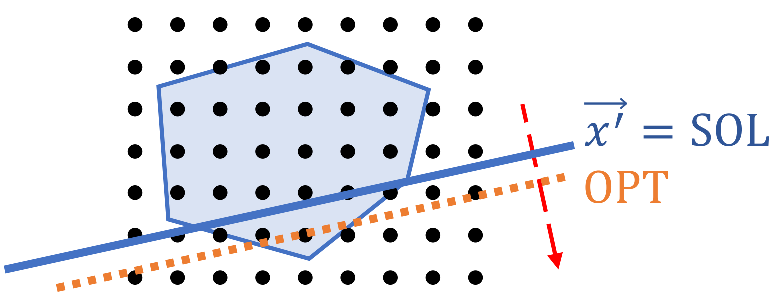
LP relaxation of VC:

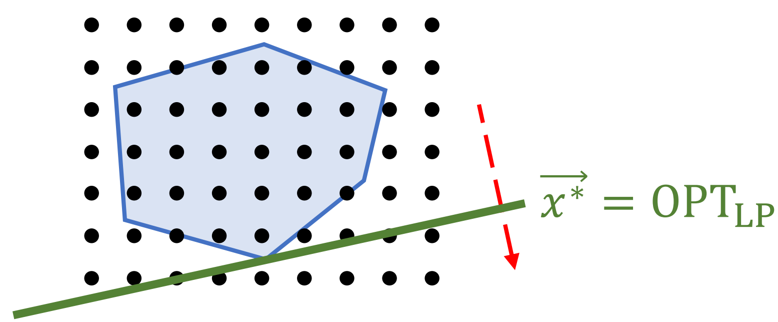
$$\min \sum_{v \in V} x_v$$
$$\text{s.t. } \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in [0,1], \quad v \in V$$

integral and feasible solution $\overrightarrow{x'} \in [0,1]^{|V|}$ (not too bad, found in poly-time)

rounding

fractional and optimal solution $\overrightarrow{x^*} \in [0,1]^{|V|}$ (found in poly-time)





Relaxation & Rounding

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover (VC): Find smallest $C \subseteq V$ s.t. $\forall e \in E: e \cap C \neq \emptyset$.

VC as ILP:

$$\min \sum_{v \in V} x_v$$
$$\text{s.t. } \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in \{0,1\}, \quad v \in V$$

→ relaxation →

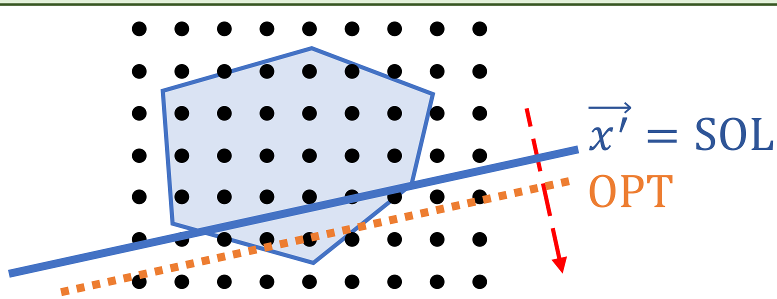
LP relaxation of VC:

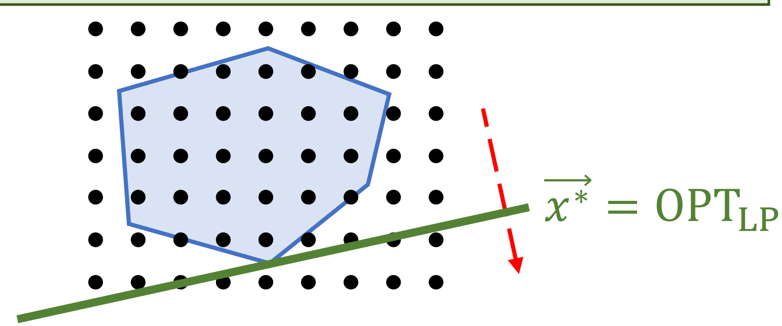
$$\min \sum_{v \in V} x_v$$
$$\text{s.t. } \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in [0,1], \quad v \in V$$

fractional and optimal solution
$\overrightarrow{x^*} \in [0,1]^{|V|}$
(found in poly-time)

← rounding ←

integral and feasible solution
$\overrightarrow{x'} \in [0,1]^{|V|}$
(not too bad, found in poly-time)





Relaxation & Rounding

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover (VC): Find smallest $C \subseteq V$ s.t. $\forall e \in E: e \cap C \neq \emptyset$.

VC as ILP:

$$\min \sum_{v \in V} x_v$$
$$\text{s.t. } \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in \{0,1\}, \quad v \in V$$

relaxation →

LP relaxation of VC:

$$\min \sum_{v \in V} x_v$$
$$\text{s.t. } \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in [0,1], \quad v \in V$$

fractional and optimal solution
$\overrightarrow{x^*} \in [0,1]^{|V|}$
(found in poly-time)

← rounding

integral and feasible solution
$\overrightarrow{x'} \in [0,1]^{|V|}$
(not too bad, found in poly-time)

Relaxation & Rounding

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover (VC): Find smallest $C \subseteq V$ s.t. $\forall e \in E: e \cap C \neq \emptyset$.

VC as ILP:

$$\begin{aligned} &\min \textstyle\sum_{v \in V} x_v \\ &\text{s.t. } \textstyle\sum_{v \in e} x_v \geq 1, \quad e \in E \\ &\qquad x_v \in \{0,1\}, \quad v \in V \end{aligned}$$

relaxation →

LP relaxation of VC:

$$\begin{aligned} &\min \textstyle\sum_{v \in V} x_v \\ &\text{s.t. } \textstyle\sum_{v \in e} x_v \geq 1, \quad e \in E \\ &\qquad x_v \in [0,1], \quad v \in V \end{aligned}$$

fractional and optimal solution
$\overrightarrow{x^*} \in [0,1]^{|V|}$
(found in poly-time)

← rounding

integral and feasible solution
$\overrightarrow{x'} \in [0,1]^{|V|}$
(not too bad, found in poly-time)

$$x'_v = \begin{cases} 1 & \text{if } x^*_v \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Relaxation & Rounding

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover (VC): Find smallest $C \subseteq V$ s.t. $\forall e \in E: e \cap C \neq \emptyset$.

VC as ILP:

$$\min \textstyle\sum_{v \in V} x_v$$
$$\text{s.t. } \textstyle\sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in \{0,1\}, \quad v \in V$$

→ relaxation →

LP relaxation of VC:

$$\min \textstyle\sum_{v \in V} x_v$$
$$\text{s.t. } \textstyle\sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in [0,1], \quad v \in V$$

integral and feasible solution
$\overrightarrow{x'} \in [0,1]^{|V|}$
(not too bad, found in poly-time)

← rounding ←

fractional and optimal solution
$\overrightarrow{x^*} \in [0,1]^{|V|}$
(found in poly-time)

$$x'_v = \begin{cases} 1 & \text{if } x^*_v \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

For each $e \in E$,
$\sum_{v \in e} x^*_v \geq 1$ implies $\sum_{v \in e} x'_v \geq 1$

Relaxation & Rounding

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover (VC): Find smallest $C \subseteq V$ s.t. $\forall e \in E: e \cap C \neq \emptyset$.

VC as ILP:

$$\begin{aligned} \min & \textstyle\sum_{v \in V} x_v \\ \text{s.t.} & \textstyle\sum_{v \in e} x_v \geq 1, \quad e \in E \\ & x_v \in \{0,1\}, \quad v \in V \end{aligned}$$

relaxation →

LP relaxation of VC:

$$\begin{aligned} \min & \textstyle\sum_{v \in V} x_v \\ \text{s.t.} & \textstyle\sum_{v \in e} x_v \geq 1, \quad e \in E \\ & x_v \in [0,1], \quad v \in V \end{aligned}$$

fractional and optimal solution
$\overrightarrow{x^*} \in [0,1]^{|V|}$
(found in poly-time)

← rounding

integral and feasible solution
$\overrightarrow{x'} \in [0,1]^{|V|}$
(not too bad, found in poly-time)

$$x'_v = \begin{cases} 1 & \text{if } x^*_v \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

For each $e \in E$,
$\sum_{v \in e} x^*_v \geq 1$ implies $\sum_{v \in e} x'_v \geq 1$

Thus $\overrightarrow{x'}$ is an integral feasible solution!

Relaxation & Rounding

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover (VC): Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

VC as ILP:

$$\min \sum_{v \in V} x_v$$
$$\text{s.t. } \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in \{0,1\}, \quad v \in V$$

relaxation →

LP relaxation of VC:

$$\min \sum_{v \in V} x_v$$
$$\text{s.t. } \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in [0,1], \quad v \in V$$

fractional and optimal solution
$\overrightarrow{x^*} \in [0,1]^{|V|}$
(found in poly-time)

← rounding

integral and feasible solution
$\overrightarrow{x'} \in [0,1]^{|V|}$
(not too bad, found in poly-time)

$$x'_v = \begin{cases} 1 & \text{if } x^*_v \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Relaxation & Rounding

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover (VC): Find smallest $C \subseteq V$ s.t. $\forall e \in E: e \cap C \neq \emptyset$.

VC as ILP:

$$\min \sum_{v \in V} x_v$$
$$\text{s.t.} \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in \{0,1\}, \quad v \in V$$

relaxation →

LP relaxation of VC:

$$\min \sum_{v \in V} x_v$$
$$\text{s.t.} \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in [0,1], \quad v \in V$$

integral and feasible solution
$\overrightarrow{x'} \in [0,1]^{|V|}$
(not too bad, found in poly-time)

← rounding

fractional and optimal solution
$\overrightarrow{x^*} \in [0,1]^{|V|}$
(found in poly-time)

$$x'_v = \begin{cases} 1 & \text{if } x^*_v \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{OPT} \geq \text{OPT}_{\text{LP}} = \sum_{v \in V} x^*_v$$

Relaxation & Rounding

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover (VC): Find smallest $C \subseteq V$ s.t. $\forall e \in E$: $e \cap C \neq \emptyset$.

VC as ILP:

$$\min \sum_{v \in V} x_v$$
$$\text{s.t.} \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in \{0,1\}, \quad v \in V$$

relaxation

LP relaxation of VC:

$$\min \sum_{v \in V} x_v$$
$$\text{s.t.} \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in [0,1], \quad v \in V$$

fractional and optimal solution
$\overrightarrow{x^*} \in [0,1]^{|V|}$
(found in poly-time)

rounding

integral and feasible solution
$\overrightarrow{x'} \in [0,1]^{|V|}$
(not too bad, found in poly-time)

$$x'_v = \begin{cases} 1 & \text{if } x^*_v \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{OPT} \geq \text{OPT}_{\text{LP}} = \sum_{v \in V} x^*_v$$
$$\text{SOL} = \sum_{v \in V} x'_v \leq \sum_{v \in V} 2 \cdot x^*_v \leq 2 \cdot \text{OPT}$$

Relaxation & Rounding

Instance: An undirected simple graph $G = (V, E)$.
Vertex Cover (VC): Find smallest $C \subseteq V$ s.t. $\forall e \in E: e \cap C \neq \emptyset$.

VC as ILP:

$$\min \sum_{v \in V} x_v$$
$$\text{s.t. } \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in \{0,1\}, \quad v \in V$$

relaxation

LP relaxation of VC:

$$\min \sum_{v \in V} x_v$$
$$\text{s.t. } \sum_{v \in e} x_v \geq 1, \quad e \in E$$
$$x_v \in [0,1], \quad v \in V$$

fractional and optimal solution
$\overrightarrow{x^*} \in [0,1]^{|V|}$
(found in poly-time)

rounding

integral and feasible solution
$\overrightarrow{x'} \in [0,1]^{|V|}$
(not too bad, found in poly-time)

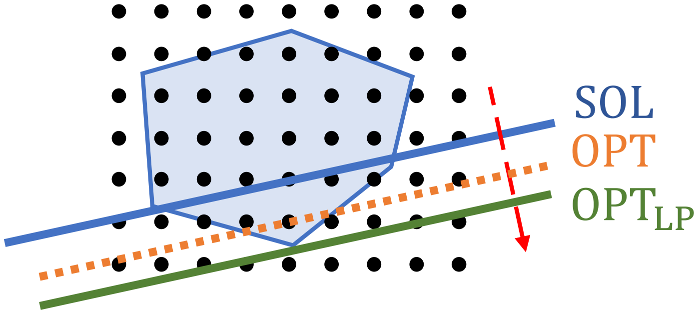
$$x'_v = \begin{cases} 1 & \text{if } x^*_v \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{OPT} \geq \mathrm{OPT}_{\mathrm{LP}} = \sum_{v \in V} x^*_v$$
$$\mathrm{SOL} = \sum_{v \in V} x'_v \leq \sum_{v \in V} 2 \cdot x^*_v \leq 2 \cdot \mathrm{OPT}$$

This is a poly-time 2-approx. alg. for VC.

LP Relaxation & Rounding

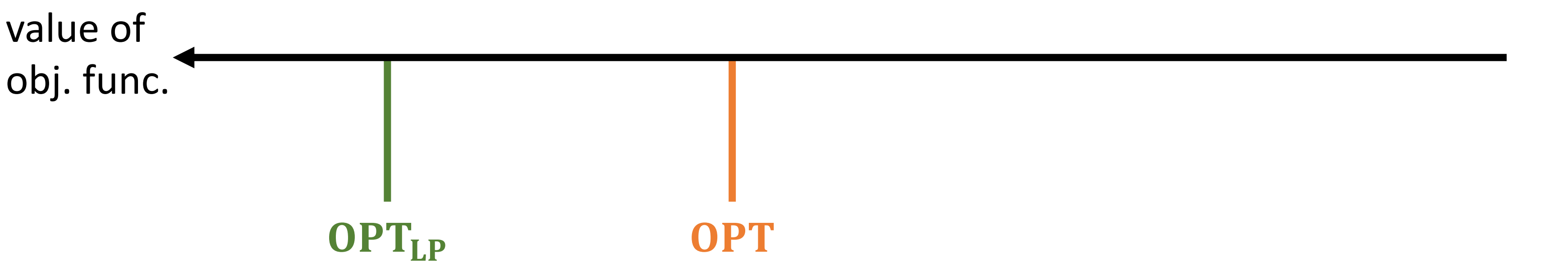
- Model the problem as an ILP.
  (Finding **OPT** of this ILP will be NP-hard.)
- Relaxation: relax the ILP to an LP.
- Find *optimal fraction* solution of the LP, call it $\mathbf{OPT_{LP}}$.
  (Can be done in poly-time via ellipsoid, interior-point, etc.)
- Rounding: round $\mathbf{OPT_{LP}}$ to a *feasible integral* solution **SOL**.
  (This is a tricky step: *how* to do rounding?)
- Show **SOL** is not far from **OPT**.
  (Notice $\mathbf{OPT_{LP}}$ provides a natural lower bound for **OPT**.)
  (Thus usually compare **SOL** with $\mathbf{OPT_{LP}}$.)



Integrality Gap

Consider a problem that can be modeled as a (minimization) ILP

Integrality Gap

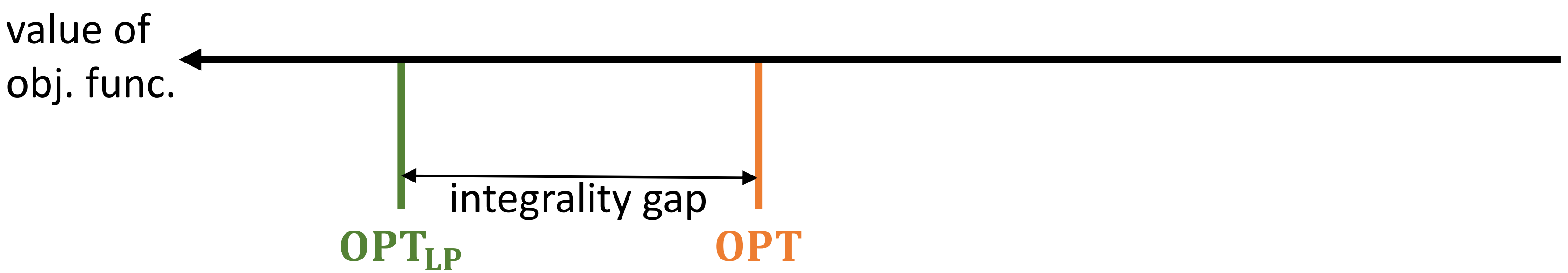


Consider a problem that can be modeled as a (minimization) ILP

$\mathbf{OPT}(I)$: optimal value (of ILP) on instance I

$\mathbf{OPT}_{\mathbf{LP}}(I)$: optimal value of LP relaxation on instance I

Integrality Gap



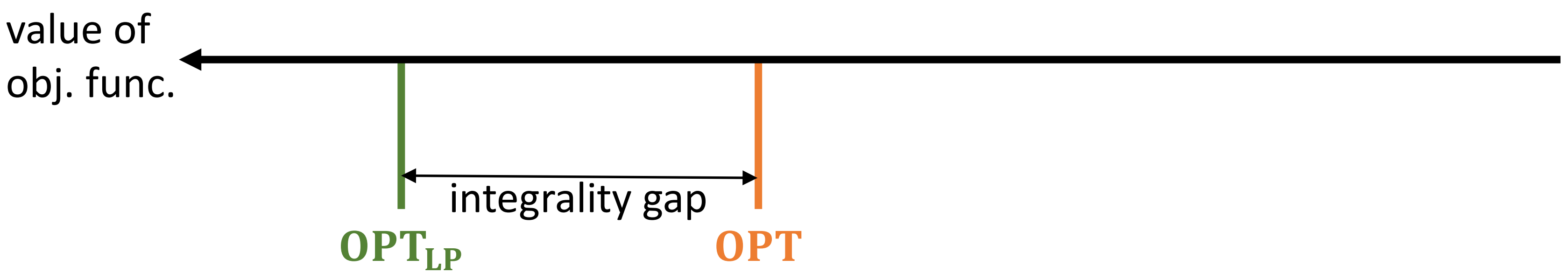
Consider a problem that can be modeled as a (minimization) ILP

$\mathbf{OPT}(I)$: optimal value (of ILP) on instance I

$\mathbf{OPT}_{\mathbf{LP}}(I)$: optimal value of LP relaxation on instance I

$$\text{Integrality Gap} = \sup_I \frac{\mathbf{OPT}(I)}{\mathbf{OPT}_{\mathbf{LP}}(I)}$$

Integrality Gap



Consider a problem that can be modeled as a (minimization) ILP

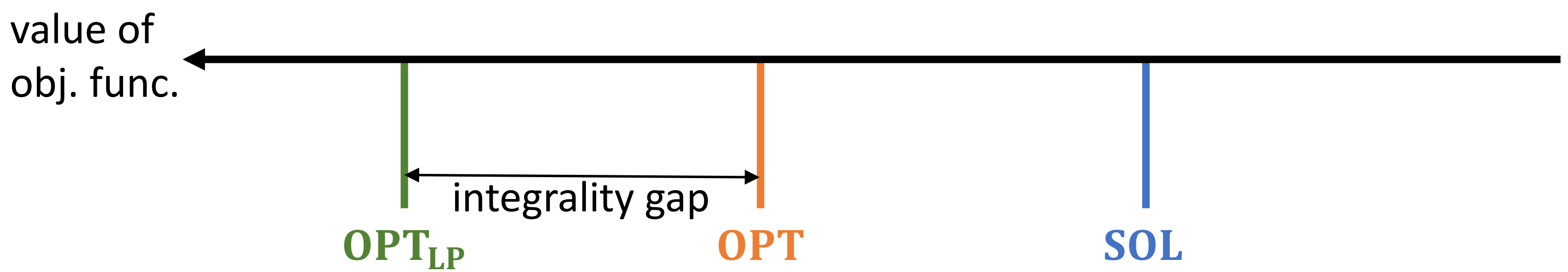
$\mathbf{OPT}(I)$: optimal value (of ILP) on instance I

$\mathbf{OPT}_{\mathbf{LP}}(I)$: optimal value of LP relaxation on instance I

$$\text{Integrality Gap} = \sup_I \frac{\mathbf{OPT}(I)}{\mathbf{OPT}_{\mathbf{LP}}(I)}$$

For the LP relaxation of vertex cover: integrality gap = 2.

Integrality Gap



Consider a problem that can be modeled as a (minimization) ILP

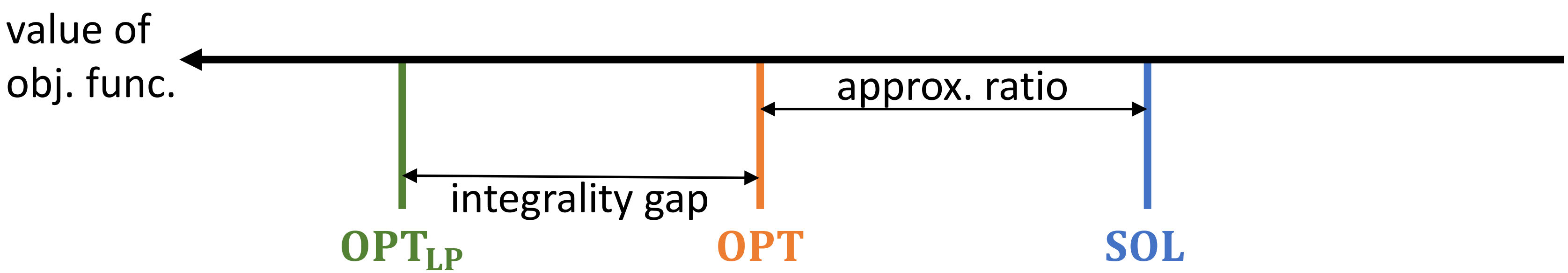
$\mathbf{OPT}(I)$: optimal value (of ILP) on instance I

$\mathbf{OPT}_{\mathbf{LP}}(I)$: optimal value of LP relaxation on instance I

$$\text{Integrality Gap} = \sup_I \frac{\mathbf{OPT}(I)}{\mathbf{OPT}_{\mathbf{LP}}(I)}$$

For the LP relaxation of vertex cover: integrality gap = 2.

Integrality Gap



Consider a problem that can be modeled as a (minimization) ILP

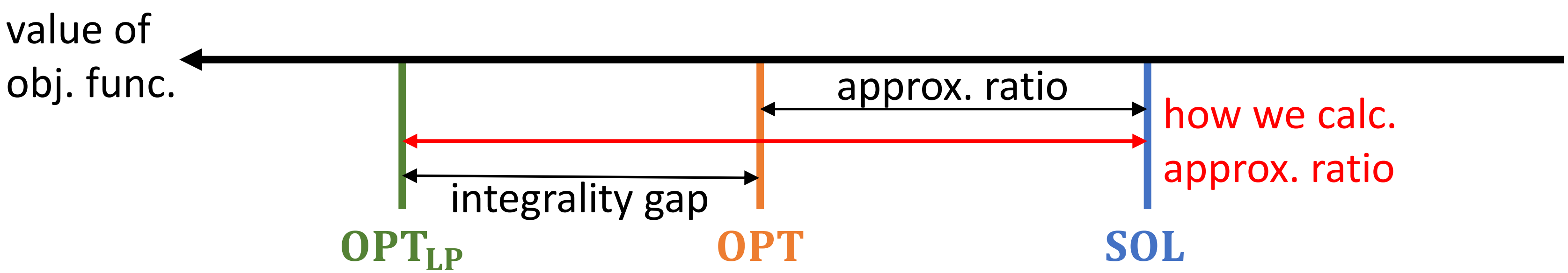
$\mathbf{OPT}(I)$: optimal value (of ILP) on instance I

$\mathbf{OPT_{LP}}(I)$: optimal value of LP relaxation on instance I

$$\text{Integrality Gap} = \sup_I \frac{\mathbf{OPT}(I)}{\mathbf{OPT_{LP}}(I)}$$

For the LP relaxation of vertex cover: integrality gap = 2.

Integrality Gap



Consider a problem that can be modeled as a (minimization) ILP

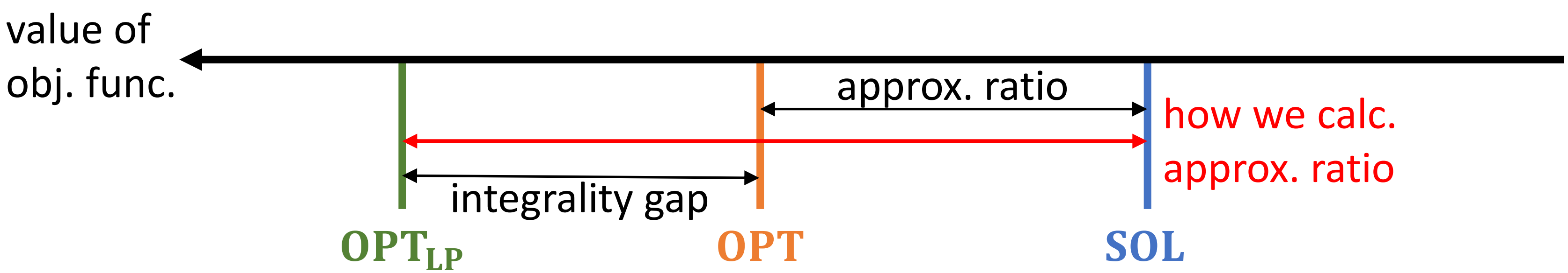
$\mathbf{OPT}(I)$: optimal value (of ILP) on instance I

$\mathbf{OPT}_{\mathbf{LP}}(I)$: optimal value of LP relaxation on instance I

$$\text{Integrality Gap} = \sup_I \frac{\mathbf{OPT}(I)}{\mathbf{OPT}_{\mathbf{LP}}(I)}$$

For the LP relaxation of vertex cover: integrality gap = 2.

Integrality Gap



Consider a problem that can be modeled as a (minimization) ILP

$\mathbf{OPT}(I)$: optimal value (of ILP) on instance I

$\mathbf{OPT_{LP}}(I)$: optimal value of LP relaxation on instance I

$$\text{Integrality Gap} = \sup_I \frac{\mathbf{OPT}(I)}{\mathbf{OPT_{LP}}(I)}$$

For the LP relaxation of vertex cover: integrality gap = 2.

Using LP relaxation & rounding, can approx. ratio beat integrality gap?

MAX-SAT

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find assignment $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize number of *satisfied* clauses.

- Boolean variables: $x_1, x_2, \cdots, x_n$
- literal: x_i or $\overline{x_i}$
- clause: $\vee$ of literals

$$C_1 = (x_1 \vee \overline{x_2} \vee \overline{x_3})$$

$$C_2 = (x_1 \vee x_4)$$

$$C_3 = (x_2 \vee \overline{x_3} \vee x_4)$$

$$C_4 = (x_3)$$

MAX-SAT

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find assignment $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize number of *satisfied* clauses.

- Boolean variables: $x_1, x_2, \cdots, x_n$
- literal: x_i or $\overline{x_i}$
- clause: $\vee$ of literals

$$C_1 = (x_1 \vee \overline{x_2} \vee \overline{x_3})$$
$$C_2 = (x_1 \vee x_4)$$
$$C_3 = (x_2 \vee \overline{x_3} \vee x_4)$$
$$C_4 = (x_3)$$

MAX-SAT is NP-hard, even MAX-E2SAT is NP-hard!
(Recall 2SAT is in P, and 3SAT is NP-hard.)

Random Solution for MAX-SAT

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find assignment $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize number of satisfied clauses.

Random Solution for MAX-SAT

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find assignment $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize number of satisfied clauses.

Assign each variable with true or false *uniformly* and *independently* at random.

Random Solution for MAX-SAT

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find assignment $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize number of satisfied clauses.

Assign each variable with true or false *uniformly* and *independently* at random.

A clause $C = (l_1 \vee l_2 \vee \cdots \vee l_k)$ where each $l_j \in \{x_i, \overline{x_i}\}$

$$\Pr[\text{clause } C \text{ is satisfied}] = 1 - 2^{-k} \geq \frac{1}{2}$$

Random Solution for MAX-SAT

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find assignment $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize number of satisfied clauses.

Assign each variable with true or false *uniformly* and *independently* at random.

A clause $C = (l_1 \vee l_2 \vee \cdots \vee l_k)$ where each $l_j \in \{x_i, \overline{x_i}\}$

$$\Pr[\text{clause } C \text{ is satisfied}] = 1 - 2^{-k} \geq \frac{1}{2}$$

$$\mathbb{E}[\#\text{ of satisfied clauses}] \geq \frac{m}{2} \geq \frac{1}{2} \cdot \mathrm{OPT}$$

Random Solution for MAX-SAT

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find assignment $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize number of satisfied clauses.

Assign each variable with true or false *uniformly* and *independently* at random.

A clause $C = (l_1 \vee l_2 \vee \cdots \vee l_k)$ where each $l_j \in \{x_i, \overline{x_i}\}$

$$\Pr[\text{clause } C \text{ is satisfied}] = 1 - 2^{-k} \geq \frac{1}{2}$$

$$\mathbb{E}[\#\text{ of satisfied clauses}] \geq \frac{m}{2} \geq \frac{1}{2} \cdot \text{OPT}$$

Does this imply a (1/2)-approx. alg. for MAX-SAT?

Derandomization via

The Method of Conditional Expectation

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Let W denote # of satisfied clauses. $\mathbb{E}[W] \geq \text{OPT}/2$.

For $i = 1$ to n:

If $\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \geq \mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}]$

$a_i = \text{true}.$

Else

$a_i = \text{false}.$

Return $\vec{a}$.

Derandomization via

The Method of Conditional Expectation

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

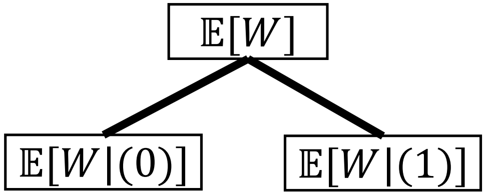
Let W denote # of satisfied clauses. $\mathbb{E}[W] \geq \text{OPT}/2$.
For $i = 1$ to n:
    If $\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \geq$
    $\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}]$
        $a_i = \text{true}$.
    Else
        $a_i = \text{false}$.
Return $\vec{a}$.

$\mathbb{E}[W]$

Derandomization via
The Method of Conditional Expectation

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Let W denote # of satisfied clauses. $\mathbb{E}[W] \geq \mathrm{OPT}/2$.
For $i = 1$ to n:
  If $\mathbb{E}[W \mid x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \geq \mathbb{E}[W \mid x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}]$
    $a_i = \text{true}$.
  Else
    $a_i = \text{false}$.
Return $\vec{a}$.



Derandomization via

The Method of Conditional Expectation

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Let W denote # of satisfied clauses. $\mathbb{E}[W] \geq \text{OPT}/2$.

For $i = 1$ to n:

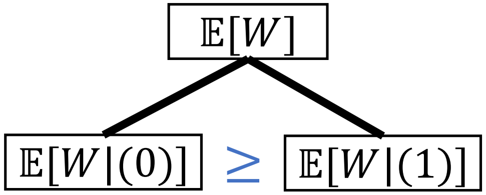
If $\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \geq \mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}]$

$a_i = \text{true}.$

Else

$a_i = \text{false}.$

Return $\vec{a}$.

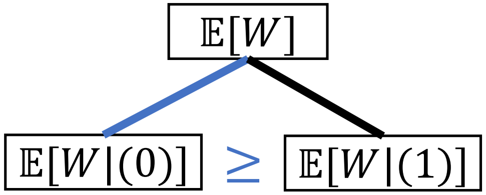


Derandomization via

The Method of Conditional Expectation

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

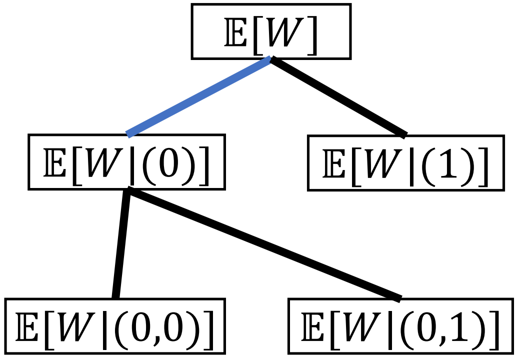
Let W denote # of satisfied clauses. $\mathbb{E}[W] \geq \mathrm{OPT}/2$.
For $i = 1$ to n:
    If $\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \geq$
    $\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}]$
        $a_i = \text{true}$.
    Else
        $a_i = \text{false}$.
Return $\vec{a}$.



Derandomization via
The Method of Conditional Expectation

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Let W denote # of satisfied clauses. $\mathbb{E}[W] \geq \text{OPT}/2$.
For $i = 1$ to n:
  If $\mathbb{E}[W|x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \geq \mathbb{E}[W|x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}]$
    $a_i = \text{true}$.
  Else
    $a_i = \text{false}$.
Return $\vec{a}$.



Derandomization via

The Method of Conditional Expectation

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Let W denote # of satisfied clauses. $\mathbb{E}[W] \geq \text{OPT}/2$.

For $i = 1$ to n:

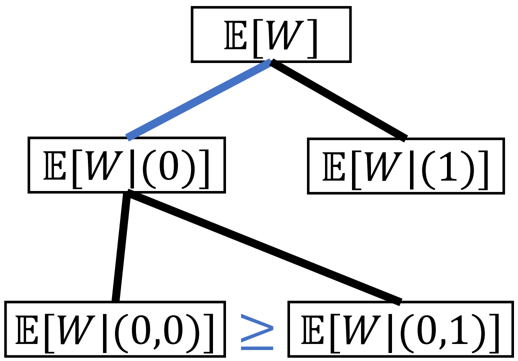
If $\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \geq \mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}]$

$a_i = \text{true}.$

Else

$a_i = \text{false}.$

Return $\vec{a}$.

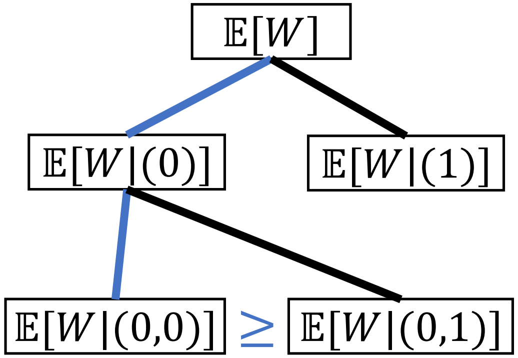


Derandomization via

The Method of Conditional Expectation

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Let W denote # of satisfied clauses. $\mathbb{E}[W] \geq \text{OPT}/2$.
For $i = 1$ to n:
    If $\mathbb{E}[W|x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \geq \mathbb{E}[W|x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}]$
        $a_i = \text{true}.$
    Else
        $a_i = \text{false}.$
Return $\vec{a}$.

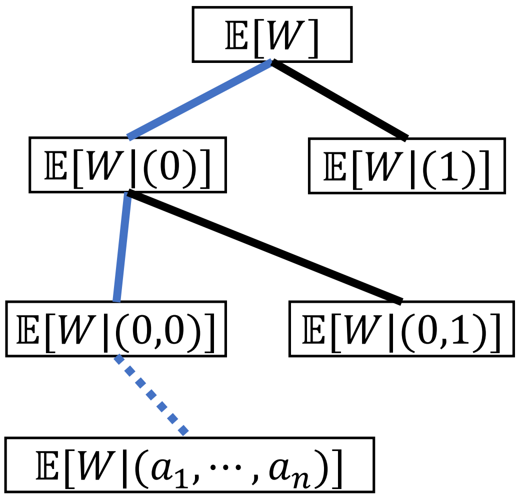


Derandomization via

The Method of Conditional Expectation

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Let W denote # of satisfied clauses. $\mathbb{E}[W] \geq \text{OPT}/2$.
For $i = 1$ to n:
  If $\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \geq$
  $\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}]$
    $a_i = \text{true}$.
  Else
    $a_i = \text{false}$.
Return $\vec{a}$.

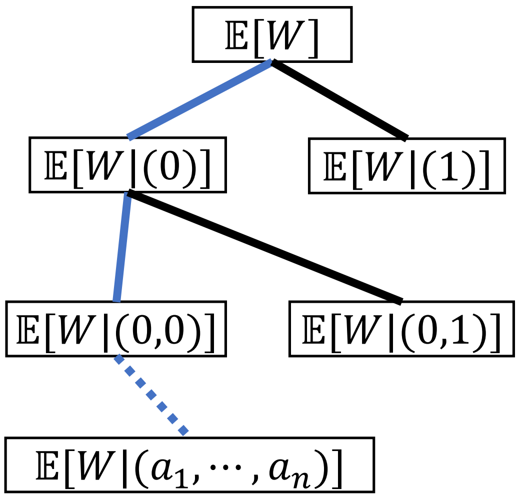


Derandomization via

The Method of Conditional Expectation

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Let W denote # of satisfied clauses. $\mathbb{E}[W] \geq \mathrm{OPT}/2$.
For $i = 1$ to n:
  If $\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \geq \mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}]$
    $a_i = \text{true}$.
  Else
    $a_i = \text{false}$.
Return $\vec{a}$.

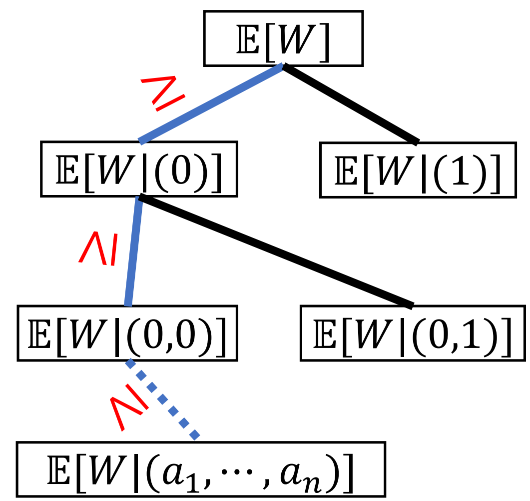


$$\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}] = \mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \cdot (1/2) + \mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}] \cdot (1/2)$$

Derandomization via
The Method of Conditional Expectation

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Let W denote # of satisfied clauses. $\mathbb{E}[W] \geq \text{OPT}/2$.
For $i = 1$ to n:
  If $\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \geq \mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}]$
    $a_i = \text{true}.$
  Else
    $a_i = \text{false}.$
Return $\vec{a}$.



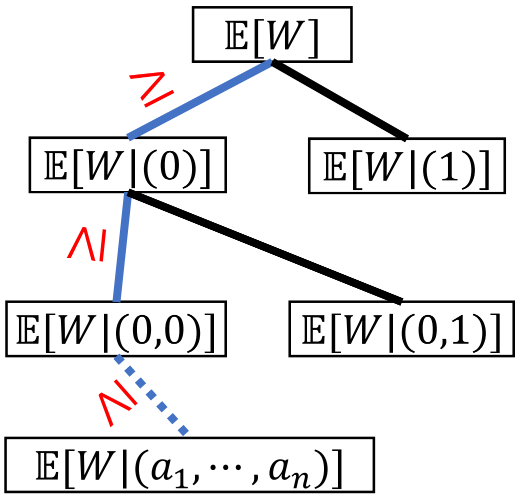
$$\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}] = \mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \cdot (1/2) + \mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}] \cdot (1/2)$$

$$\max\{\mathbb{E}[W | (a_1, \cdots, a_{i-1}, \text{true})], \mathbb{E}[W | (a_1, \cdots, a_{i-1}, \text{false})]\} \geq \mathbb{E}[W | (a_1, \cdots, a_{i-1})]$$

Derandomization via
The Method of Conditional Expectation

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Let W denote # of satisfied clauses. $\mathbb{E}[W] \geq \text{OPT}/2$.
For $i = 1$ to n:
    If $\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \geq \mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}]$
        $a_i = \text{true}$.
    Else
        $a_i = \text{false}$.
Return $\vec{a}$.



$$\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}] = \mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \cdot (1/2) + \mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}] \cdot (1/2)$$

$$\max\{\mathbb{E}[W | (a_1, \cdots, a_{i-1}, \text{true})], \mathbb{E}[W | (a_1, \cdots, a_{i-1}, \text{false})]\} \geq \mathbb{E}[W | (a_1, \cdots, a_{i-1})]$$

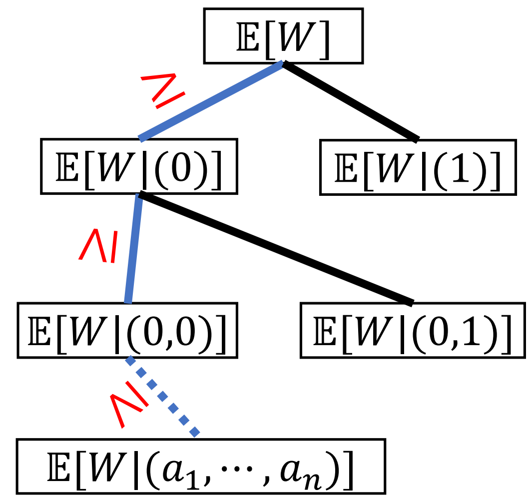
For any partial assignment, $\mathbb{E}[W | (a_1, \cdots, a_i)]$ can be computed in poly-time.

Derandomization via
The Method of Conditional Expectation

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Let W denote # of satisfied clauses. $\mathbb{E}[W] \geq \text{OPT}/2$.
For $i = 1$ to n:
  If $\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \geq \mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}]$
    $a_i = \text{true}$.
  Else
    $a_i = \text{false}$.
Return $\vec{a}$.

Poly-time deterministic (1/2)-approx. algorithm!



$$\mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}] = \mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{true}] \cdot (1/2) + \mathbb{E}[W | x_1 = a_1, \cdots, x_{i-1} = a_{i-1}, x_i = \text{false}] \cdot (1/2)$$

$$\max\{\mathbb{E}[W | (a_1, \cdots, a_{i-1}, \text{true})], \mathbb{E}[W | (a_1, \cdots, a_{i-1}, \text{false})]\} \geq \mathbb{E}[W | (a_1, \cdots, a_{i-1})]$$

For any partial assignment, $\mathbb{E}[W | (a_1, \cdots, a_i)]$ can be computed in poly-time.

MAX-SAT as ILP

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find assignment $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize number of satisfied clauses.

MAX-SAT as ILP

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find assignment $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize number of satisfied clauses.

Boolean variables: $x_1, x_2, \cdots, x_n \in \{0,1\}$

$x_i = 0$ means $x_i = \text{false}$

$x_i = 1$ means $x_i = \text{true}$

MAX-SAT as ILP

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find assignment $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize number of satisfied clauses.

Boolean variables: $x_1, x_2, \cdots, x_n \in \{0,1\}$

$x_i = 0$ means $x_i = \text{false}$

$x_i = 1$ means $x_i = \text{true}$

clause: $C_j = (l_1 \vee l_2 \vee \cdots \vee l_k)$ where $l_r \in \{x_i, \overline{x_i} \mid 1 \leq i \leq n\}$

MAX-SAT as ILP

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find assignment $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize number of satisfied clauses.

Boolean variables: $x_1, x_2, \cdots, x_n \in \{0,1\}$
$x_i = 0$ means $x_i = \text{false}$
$x_i = 1$ means $x_i = \text{true}$

clause: $C_j = (l_1 \vee l_2 \vee \cdots \vee l_k)$ where $l_r \in \{x_i, \overline{x_i} \mid 1 \leq i \leq n\}$

C_j is satisfied

MAX-SAT as ILP

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find assignment $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize number of satisfied clauses.

Boolean variables: $x_1, x_2, \cdots, x_n \in \{0,1\}$
$x_i = 0$ means $x_i = \text{false}$
$x_i = 1$ means $x_i = \text{true}$

clause: $C_j = (l_1 \vee l_2 \vee \cdots \vee l_k)$ where $l_r \in \{x_i, \overline{x_i} \mid 1 \leq i \leq n\}$
S_j^+: set of i such that x_i is in C_j
S_j^-: set of i such that $\overline{x_i}$ is in C_j

C_j is satisfied

MAX-SAT as ILP

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find assignment $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize number of satisfied clauses.

Boolean variables: $x_1, x_2, \cdots, x_n \in \{0,1\}$

$x_i = 0$ means $x_i = \text{false}$

$x_i = 1$ means $x_i = \text{true}$

clause: $C_j = (l_1 \vee l_2 \vee \cdots \vee l_k)$ where $l_r \in \{x_i, \overline{x_i} \mid 1 \leq i \leq n\}$

S_j^+: set of i such that x_i is in C_j

S_j^-: set of i such that $\overline{x_i}$ is in C_j

C_j is satisfied $\Longleftrightarrow$ $$\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq 1$$

MAX-SAT as ILP

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find assignment $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize number of satisfied clauses.

Boolean variables: $x_1, x_2, \cdots, x_n \in \{0,1\}$

clause: $C_1, C_2, \cdots, C_m$

S_j^+: set of i such that x_i is in C_j

S_j^-: set of i such that $\overline{x_i}$ is in C_j

MAX-SAT as ILP:

maximize $\sum_{j=1}^{m} y_j$

subject to

$$\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j, \quad 1 \leq j \leq m$$

$$x_i \in \{0,1\}, \quad 1 \leq i \leq n$$

$$y_j \in \{0,1\}, \quad 1 \leq j \leq m$$

LP Relaxation

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find assignment $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize number of satisfied clauses.

Boolean variables: $x_1, x_2, \cdots, x_n \in \{0,1\}$

clause: $C_1, C_2, \cdots, C_m$

S_j^+: set of i such that x_i is in C_j

S_j^-: set of i such that $\overline{x_i}$ is in C_j

<u>LP relaxation of MAX-SAT-ILP:</u>

maximize $\sum_{j=1}^{m} y_j$

subject to $\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j, \quad 1 \leq j \leq m$

~~$x_i \in \{0,1\}$~~, $x_i \in [0,1]$, $\quad 1 \leq i \leq n$

~~$y_j \in \{0,1\}$~~, $y_j \in [0,1]$, $\quad 1 \leq j \leq m$

Randomized Rounding

MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

s.t. $\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \quad 1 \leq j \leq m$

$x_i \in \{0,1\} \quad 1 \leq i \leq n$

$y_j \in \{0,1\} \quad 1 \leq j \leq m$

LP relaxation of MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

s.t. $\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \quad 1 \leq j \leq m$

$x_i \in [0,1] \quad 1 \leq i \leq n$

$y_j \in [0,1] \quad 1 \leq j \leq m$

Randomized Rounding

MAX-SAT-ILP:

$$\max \sum_{j=1}^{m} y_j$$

$$\text{s.t.} \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$

$$x_i \in \{0,1\} \qquad 1 \leq i \leq n$$

$$y_j \in \{0,1\} \qquad 1 \leq j \leq m$$

LP relaxation of MAX-SAT-ILP:

$$\max \sum_{j=1}^{m} y_j$$

$$\text{s.t.} \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$

$$x_i \in [0,1] \qquad 1 \leq i \leq n$$

$$y_j \in [0,1] \qquad 1 \leq j \leq m$$

Optimal *integral* solution **OPT**

Randomized Rounding

MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

s.t. $\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \quad 1 \leq j \leq m$

$x_i \in \{0,1\} \quad 1 \leq i \leq n$

$y_j \in \{0,1\} \quad 1 \leq j \leq m$

Optimal *integral* solution $\mathbf{OPT}$

LP relaxation of MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

s.t. $\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \quad 1 \leq j \leq m$

$x_i \in [0,1] \quad 1 \leq i \leq n$

$y_j \in [0,1] \quad 1 \leq j \leq m$

Optimal *fractional* solution $\mathbf{OPT_{LP}}$:

$\overrightarrow{x^*} \in [0,1]^n$, $\overrightarrow{y^*} \in [0,1]^m$

Randomized Rounding

MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$

$$x_i \in \{0,1\} \qquad 1 \leq i \leq n$$

$$y_j \in \{0,1\} \qquad 1 \leq j \leq m$$

LP relaxation of MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$

$$x_i \in [0,1] \qquad 1 \leq i \leq n$$

$$y_j \in [0,1] \qquad 1 \leq j \leq m$$

Optimal *integral* solution $\mathbf{OPT}$

Clearly $\mathbf{OPT} \leq \mathbf{OPT_{LP}}$

Optimal *fractional* solution $\mathbf{OPT_{LP}}$:

$\overrightarrow{x^*} \in [0,1]^n$, $\overrightarrow{y^*} \in [0,1]^m$

Randomized Rounding

MAX-SAT-ILP:

$$\max \sum_{j=1}^{m} y_j$$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$

$$x_i \in \{0,1\} \qquad 1 \leq i \leq n$$

$$y_j \in \{0,1\} \qquad 1 \leq j \leq m$$

LP relaxation of MAX-SAT-ILP:

$$\max \sum_{j=1}^{m} y_j$$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$

$$x_i \in [0,1] \qquad 1 \leq i \leq n$$

$$y_j \in [0,1] \qquad 1 \leq j \leq m$$

Optimal *integral* solution $\mathbf{OPT}$

Clearly $\mathbf{OPT} \leq \mathbf{OPT_{LP}}$

Optimal *fractional* solution $\mathbf{OPT_{LP}}$:

$\overrightarrow{x^*} \in [0,1]^n$, $\overrightarrow{y^*} \in [0,1]^m$

Randomly generate *integral* solution $\mathbf{SOL}$ from $\mathbf{OPT_{LP}}$:

$\overrightarrow{x'} \in \{0,1\}^n$, specifically, $x_i' = \begin{cases} 1 \text{ with probability } x_i^* \\ 0 \text{ with probability } 1 - x_i^* \end{cases}$

Randomized Rounding

MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$
$$x_i \in \{0,1\} \qquad 1 \leq i \leq n$$
$$y_j \in \{0,1\} \qquad 1 \leq j \leq m$$

LP relaxation of MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$
$$x_i \in [0,1] \qquad 1 \leq i \leq n$$
$$y_j \in [0,1] \qquad 1 \leq j \leq m$$

Optimal *integral* solution $\mathbf{OPT}$

Clearly $\mathbf{OPT} \leq \mathbf{OPT_{LP}}$

Optimal *fractional* solution $\mathbf{OPT_{LP}}$:

$\overrightarrow{x^*} \in [0,1]^n$, $\overrightarrow{y^*} \in [0,1]^m$

Randomly generate *integral* solution $\mathbf{SOL}$ from $\mathbf{OPT_{LP}}$:

Is $\mathbf{SOL}$ feasible?

$\overrightarrow{x'} \in \{0,1\}^n$, specifically, $x_i' = \begin{cases} 1 \text{ with probability } x_i^* \\ 0 \text{ with probability } 1 - x_i^* \end{cases}$

Randomized Rounding

MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$
$$x_i \in \{0,1\} \qquad 1 \leq i \leq n$$
$$y_j \in \{0,1\} \qquad 1 \leq j \leq m$$

LP relaxation of MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$
$$x_i \in [0,1] \qquad 1 \leq i \leq n$$
$$y_j \in [0,1] \qquad 1 \leq j \leq m$$

Optimal *integral* solution $\mathbf{OPT}$

Clearly $\mathbf{OPT} \leq \mathbf{OPT_{LP}}$

Optimal *fractional* solution $\mathbf{OPT_{LP}}$:

$\overrightarrow{x^*} \in [0,1]^n$, $\overrightarrow{y^*} \in [0,1]^m$

Randomly generate *integral* solution $\mathbf{SOL}$ from $\mathbf{OPT_{LP}}$:

$\overrightarrow{x'} \in \{0,1\}^n$, specifically, $x_i' = \begin{cases} 1 \text{ with probability } x_i^* \\ 0 \text{ with probability } 1 - x_i^* \end{cases}$

How good is $\mathbf{SOL}$?

Randomized Rounding

MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

s.t. $\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \quad 1 \leq j \leq m$

$x_i \in \{0,1\} \quad 1 \leq i \leq n$

$y_j \in \{0,1\} \quad 1 \leq j \leq m$

LP relaxation of MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

s.t. $\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \quad 1 \leq j \leq m$

$x_i \in [0,1] \quad 1 \leq i \leq n$

$y_j \in [0,1] \quad 1 \leq j \leq m$

Optimal *integral* solution $\mathbf{OPT}$

Clearly $\mathbf{OPT} \leq \mathbf{OPT_{LP}}$

Optimal *fractional* solution $\mathbf{OPT_{LP}}$:

$\overrightarrow{x^*} \in [0,1]^n$, $\overrightarrow{y^*} \in [0,1]^m$

Randomly generate *integral* solution $\mathbf{SOL}$ from $\mathbf{OPT_{LP}}$:

$\overrightarrow{x'} \in \{0,1\}^n$, specifically, $x'_i = \begin{cases} 1 \text{ with probability } x_i^* \\ 0 \text{ with probability } 1 - x_i^* \end{cases}$

How good is $\mathbf{SOL}$?

$$\Pr[C_j \text{ satisfied in } \mathbf{SOL}] = 1 - \left(\prod_{i \in S_j^+} (1 - x_i^*)\right)\left(\prod_{i \in S_j^-} x_i^*\right)$$

Randomized Rounding

MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$
$$x_i \in \{0,1\} \qquad 1 \leq i \leq n$$
$$y_j \in \{0,1\} \qquad 1 \leq j \leq m$$

LP relaxation of MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$
$$x_i \in [0,1] \qquad 1 \leq i \leq n$$
$$y_j \in [0,1] \qquad 1 \leq j \leq m$$

Optimal *integral* solution $\mathbf{OPT}$

Clearly $\mathbf{OPT} \leq \mathbf{OPT_{LP}}$

Optimal *fractional* solution $\mathbf{OPT_{LP}}$:

$\overrightarrow{x^*} \in [0,1]^n$, $\overrightarrow{y^*} \in [0,1]^m$

Randomly generate *integral* solution $\mathbf{SOL}$ from $\mathbf{OPT_{LP}}$:

$\overrightarrow{x'} \in \{0,1\}^n$, specifically, $x_i' = \begin{cases} 1 \text{ with probability } x_i^* \\ 0 \text{ with probability } 1 - x_i^* \end{cases}$

How good is $\mathbf{SOL}$?

$$\Pr[C_j \text{ satisfied in } \mathbf{SOL}] = 1 - \left(\prod_{i \in S_j^+} (1 - x_i^*)\right)\left(\prod_{i \in S_j^-} x_i^*\right) \geq 1 - \left(1 - y_j^*/k\right)^k$$

Randomized Rounding

MAX-SAT-ILP:

$$\max \sum_{j=1}^{m} y_j$$

$$\text{s.t.} \quad \sum_{i\in S_j^+} x_i + \sum_{i\in S_j^-} (1-x_i) \geq y_j \qquad 1 \leq j \leq m$$

$$x_i \in \{0,1\} \qquad 1 \leq i \leq n$$

$$y_j \in \{0,1\} \qquad 1 \leq j \leq m$$

LP relaxation of MAX-SAT-ILP:

$$\max \sum_{j=1}^{m} y_j$$

$$\text{s.t.} \quad \sum_{i\in S_j^+} x_i + \sum_{i\in S_j^-} (1-x_i) \geq y_j \qquad 1 \leq j \leq m$$

$$x_i \in [0,1] \qquad 1 \leq i \leq n$$

$$y_j \in [0,1] \qquad 1 \leq j \leq m$$

Optimal *integral* solution $\mathbf{OPT}$

Optimal *fractional* solution $\mathbf{OPT_{LP}}$:

$^* \in [0,1]^m$

Clearly $\mathbf{OPT} \leq \mathbf{OPT}$

Randomly generate

arithmetic-geometric mean inequality:

for $a_1, a_2, \cdots, a_k \in \mathbb{R}^{\geq 0}$,

$(a_1 + \cdots + a_k)/k \geq (a_1 \times \cdots \times a_k)^{1/k}$

$\overrightarrow{x'} \in \{0,1\}^n$, specifically, $x_i' = \begin{cases} 1 \text{ with probability } x_i^* \\ 0 \text{ with probability } 1 - x_i^* \end{cases}$

How good is $\mathbf{SOL}$?

$$\Pr[C_j \text{ satisfied in } \mathbf{SOL}] = 1 - \left(\prod_{i\in S_j^+} (1-x_i^*)\right)\left(\prod_{i\in S_j^-} x_i^*\right) \geq 1 - \left(1 - y_j^*/k\right)^k$$

Randomized Rounding

MAX-SAT-ILP:

$$\max \sum_{j=1}^{m} y_j$$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$

$$x_i \in \{0,1\} \qquad 1 \leq i \leq n$$

$$y_j \in \{0,1\} \qquad 1 \leq j \leq m$$

LP relaxation of MAX-SAT-ILP:

$$\max \sum_{j=1}^{m} y_j$$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$

$$x_i \in [0,1] \qquad 1 \leq i \leq n$$

$$y_j \in [0,1] \qquad 1 \leq j \leq m$$

Optimal *integral* solution $\mathbf{OPT}$

Clearly $\mathbf{OPT} \leq \mathbf{OPT_{LP}}$

Optimal *fractional* solution $\mathbf{OPT_{LP}}$:

$\overrightarrow{x^*} \in [0,1]^n$, $\overrightarrow{y^*} \in [0,1]^m$

Randomly generate *integral* solution $\mathbf{SOL}$ from $\mathbf{OPT_{LP}}$:

$\overrightarrow{x'} \in \{0,1\}^n$, specifically, $x_i' = \begin{cases} 1 \text{ with probability } x_i^* \\ 0 \text{ with probability } 1 - x_i^* \end{cases}$

How good is $\mathbf{SOL}$?

$$\Pr[C_j \text{ satisfied in } \mathbf{SOL}] = 1 - \left(\prod_{i \in S_j^+} (1 - x_i^*)\right)\left(\prod_{i \in S_j^-} x_i^*\right) \geq 1 - \left(1 - y_j^*/k\right)^k$$

$$\geq [1 - (1 - 1/k)^k] \cdot y_j^*$$

Randomized Rounding

MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

s.t. $\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \quad 1 \leq j \leq m$

$x_i \in \{0,1\} \quad 1 \leq i \leq n$

$y_j \in \{0,1\} \quad 1 \leq j \leq m$

LP relaxation of MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

s.t. $\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \quad 1 \leq j \leq m$

$x_i \in [0,1] \quad 1 \leq i \leq n$

$y_j \in [0,1] \quad 1 \leq j \leq m$

Optimal *integral* solution $\mathbf{OPT}$

Clearly $\mathbf{OPT} \leq \mathbf{OPT_{LP}}$

Optimal *fractional* solution $\mathbf{OPT_{LP}}$:

$\overrightarrow{x^*} \in [0,1]^n$, $\overrightarrow{y^*} \in [0,1]^m$

Randomly generate *integral* solution $\mathbf{SOL}$ from $\mathbf{OPT_{LP}}$:

$\overrightarrow{x'} \in \{0,1\}^n$, specifically, $x_i' = \begin{cases} 1 \text{ with probability } x_i^* \\ 0 \text{ with probability } 1 - x_i^* \end{cases}$

How good is $\mathbf{SOL}$?

$$\Pr[C_j \text{ satisfied in } \mathbf{SOL}] = 1 - \left(\prod_{i \in S_j^+} (1 - x_i^*)\right)\left(\prod_{i \in S_j^-} x_i^*\right) \geq 1 - \left(1 - y_j^*/k\right)^k$$

$$\geq [1 - (1 - 1/k)^k] \cdot y_j^*$$

$f(y_j^*) = 1 - \left(1 - y_j^*/k\right)^k$ is concave when $y_j^* \in [0,1]$

Randomized Rounding

MAX-SAT-ILP:

$$\max \sum_{j=1}^{m} y_j$$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$

$$x_i \in \{0,1\} \qquad 1 \leq i \leq n$$

$$y_j \in \{0,1\} \qquad 1 \leq j \leq m$$

LP relaxation of MAX-SAT-ILP:

$$\max \sum_{j=1}^{m} y_j$$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$

$$x_i \in [0,1] \qquad 1 \leq i \leq n$$

$$y_j \in [0,1] \qquad 1 \leq j \leq m$$

Optimal *integral* solution $\mathbf{OPT}$

Clearly $\mathbf{OPT} \leq \mathbf{OPT_{LP}}$

Optimal *fractional* solution $\mathbf{OPT_{LP}}$:

$\overrightarrow{x^*} \in [0,1]^n$, $\overrightarrow{y^*} \in [0,1]^m$

Randomly generate *integral* solution $\mathbf{SOL}$ from $\mathbf{OPT_{LP}}$:

$\overrightarrow{x'} \in \{0,1\}^n$, specifically, $x'_i = \begin{cases} 1 \text{ with probability } x_i^* \\ 0 \text{ with probability } 1 - x_i^* \end{cases}$

How good is $\mathbf{SOL}$?

$$\Pr[C_j \text{ satisfied in } \mathbf{SOL}] = 1 - \left(\prod_{i \in S_j^+} (1 - x_i^*)\right)\left(\prod_{i \in S_j^-} x_i^*\right) \geq 1 - \left(1 - y_j^*/k\right)^k$$

$$\geq [1 - (1 - 1/k)^k] \cdot y_j^*$$

$$\geq (1 - 1/e) \cdot y_j^*$$

Randomized Rounding

MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$
$$x_i \in \{0,1\} \qquad 1 \leq i \leq n$$
$$y_j \in \{0,1\} \qquad 1 \leq j \leq m$$

LP relaxation of MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

$$\text{s.t.} \quad \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$
$$x_i \in [0,1] \qquad 1 \leq i \leq n$$
$$y_j \in [0,1] \qquad 1 \leq j \leq m$$

Optimal *integral* solution $\mathbf{OPT}$

Clearly $\mathbf{OPT} \leq \mathbf{OPT_{LP}}$

Optimal *fractional* solution $\mathbf{OPT_{LP}}$:

$\overrightarrow{x^*} \in [0,1]^n$, $\overrightarrow{y^*} \in [0,1]^m$

Randomly generate *integral* solution $\mathbf{SOL}$ from $\mathbf{OPT_{LP}}$:

$\overrightarrow{x'} \in \{0,1\}^n$, specifically, $x_i' = \begin{cases} 1 \text{ with probability } x_i^* \\ 0 \text{ with probability } 1 - x_i^* \end{cases}$

How good is $\mathbf{SOL}$?

$$\Pr[C_j \text{ satisfied in } \mathbf{SOL}] \geq (1 - 1/e) \cdot y_j^*$$

Randomized Rounding

MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

s.t. $\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \quad 1 \leq j \leq m$

$x_i \in \{0,1\} \quad 1 \leq i \leq n$

$y_j \in \{0,1\} \quad 1 \leq j \leq m$

LP relaxation of MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

s.t. $\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \quad 1 \leq j \leq m$

$x_i \in [0,1] \quad 1 \leq i \leq n$

$y_j \in [0,1] \quad 1 \leq j \leq m$

Optimal *integral* solution $\mathbf{OPT}$

Clearly $\mathbf{OPT} \leq \mathbf{OPT_{LP}}$

Optimal *fractional* solution $\mathbf{OPT_{LP}}$:

$\overrightarrow{x^*} \in [0,1]^n$, $\overrightarrow{y^*} \in [0,1]^m$

Randomly generate *integral* solution $\mathbf{SOL}$ from $\mathbf{OPT_{LP}}$:

$\overrightarrow{x'} \in \{0,1\}^n$, specifically, $x_i' = \begin{cases} 1 \text{ with probability } x_i^* \\ 0 \text{ with probability } 1 - x_i^* \end{cases}$

How good is $\mathbf{SOL}$?

$$\Pr[C_j \text{ satisfied in } \mathbf{SOL}] \geq (1 - 1/e) \cdot y_j^*$$

$$\mathbb{E}[\#\text{ clauses satisfied in } \mathbf{SOL}] \geq \sum_{j=1}^{m} \left(1 - \frac{1}{e}\right) \cdot y_j^*$$

$$= \left(1 - \frac{1}{e}\right) \cdot \mathbf{OPT_{LP}} \geq \left(1 - \frac{1}{e}\right) \cdot \mathbf{OPT}$$

Randomized Rounding

MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

s.t. $\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \quad 1 \leq j \leq m$

$x_i \in \{0,1\} \quad 1 \leq i \leq n$

$y_j \in \{0,1\} \quad 1 \leq j \leq m$

LP relaxation of MAX-SAT-ILP:

max $\sum_{j=1}^{m} y_j$

s.t. $\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \quad 1 \leq j \leq m$

$x_i \in [0,1] \quad 1 \leq i \leq n$

$y_j \in [0,1] \quad 1 \leq j \leq m$

Optimal *integral* solution $\mathbf{OPT}$

Clearly $\mathbf{OPT} \leq \mathbf{OPT_{LP}}$

Optimal *fractional* solution $\mathbf{OPT_{LP}}$:

$\overrightarrow{x^*} \in [0,1]^n$, $\overrightarrow{y^*} \in [0,1]^m$

Randomly generate *integral* solution $\mathbf{SOL}$ from $\mathbf{OPT_{LP}}$:

$\overrightarrow{x'} \in \{0,1\}^n$, specifically, $x_i' = \begin{cases} 1 \text{ with probability } x_i^* \\ 0 \text{ with probability } 1 - x_i^* \end{cases}$

$\Pr[C_j \text{ satisfied in } \mathbf{SOL}] \geq (1 - 1/e) \cdot y_j^*$

$\mathbb{E}[\# \text{ clauses satisfied in } \mathbf{SOL}] \geq \sum_{j=1}^{m} \left(1 - \frac{1}{e}\right) \cdot y_j^*$

$= \left(1 - \frac{1}{e}\right) \cdot \mathbf{OPT_{LP}} \geq \left(1 - \frac{1}{e}\right) \cdot \mathbf{OPT}$

How good is $\mathbf{SOL}$?

Derandomize to get deterministic (1-1/e)-approx. alg.

Putting two algorithms together

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Assign each variable with true or false *uniformly* and *independently* at random.

Model problem as an ILP.
Obtain LP relaxation of the ILP.
Get optimal solution $\overrightarrow{x^*}$ of the LP.
Randomized rounding $\overrightarrow{x^*}$ to $\overrightarrow{x'}$ as SOL.

Putting two algorithms together

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Assign each variable with true or false *uniformly* and *independently* at random.

Model problem as an ILP.
Obtain LP relaxation of the ILP.
Get optimal solution $\overrightarrow{x^*}$ of the LP.
Randomized rounding $\overrightarrow{x^*}$ to $\overrightarrow{x'}$ as SOL.

Suppose $C_j = (l_1 \vee l_2 \vee \cdots \vee l_k)$

- Random assignment:
  - 1/2-approximation
  - $\Pr[C_j \text{ satisfied}] = 1 - 2^{-k}$

Putting two algorithms together

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Assign each variable with true or false *uniformly* and *independently* at random.

Model problem as an ILP.
Obtain LP relaxation of the ILP.
Get optimal solution $\overrightarrow{x^*}$ of the LP.
Randomized rounding $\overrightarrow{x^*}$ to $\overrightarrow{x'}$ as SOL.

Suppose $C_j = (l_1 \vee l_2 \vee \cdots \vee l_k)$

- Random assignment:
  - 1/2-approximation
  - $\Pr\left[C_j \text{ satisfied}\right] = 1 - 2^{-k}$
- LP relaxation and randomized rounding:
  - (1-1/e)-approximation
  - $\Pr\left[C_j \text{ satisfied}\right] = [1 - (1 - 1/k)^k] \cdot y_j^*$

Putting two algorithms together

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Assign each variable with true or false *uniformly* and *independently* at random.

Model problem as an ILP.
Obtain LP relaxation of the ILP.
Get optimal solution $\overrightarrow{x^*}$ of the LP.
Randomized rounding $\overrightarrow{x^*}$ to $\overrightarrow{x'}$ as SOL.

Suppose $C_j = (l_1 \vee l_2 \vee \cdots \vee l_k)$

- Random assignment:
  - 1/2-approximation
  - $\Pr[C_j \text{ satisfied}] = 1 - 2^{-k}$ Good when k is large.
- LP relaxation and randomized rounding:
  - (1-1/e)-approximation
  - $\Pr[C_j \text{ satisfied}] = [1 - (1 - 1/k)^k] \cdot y_j^*$ Good when k is small.

Putting two algorithms together

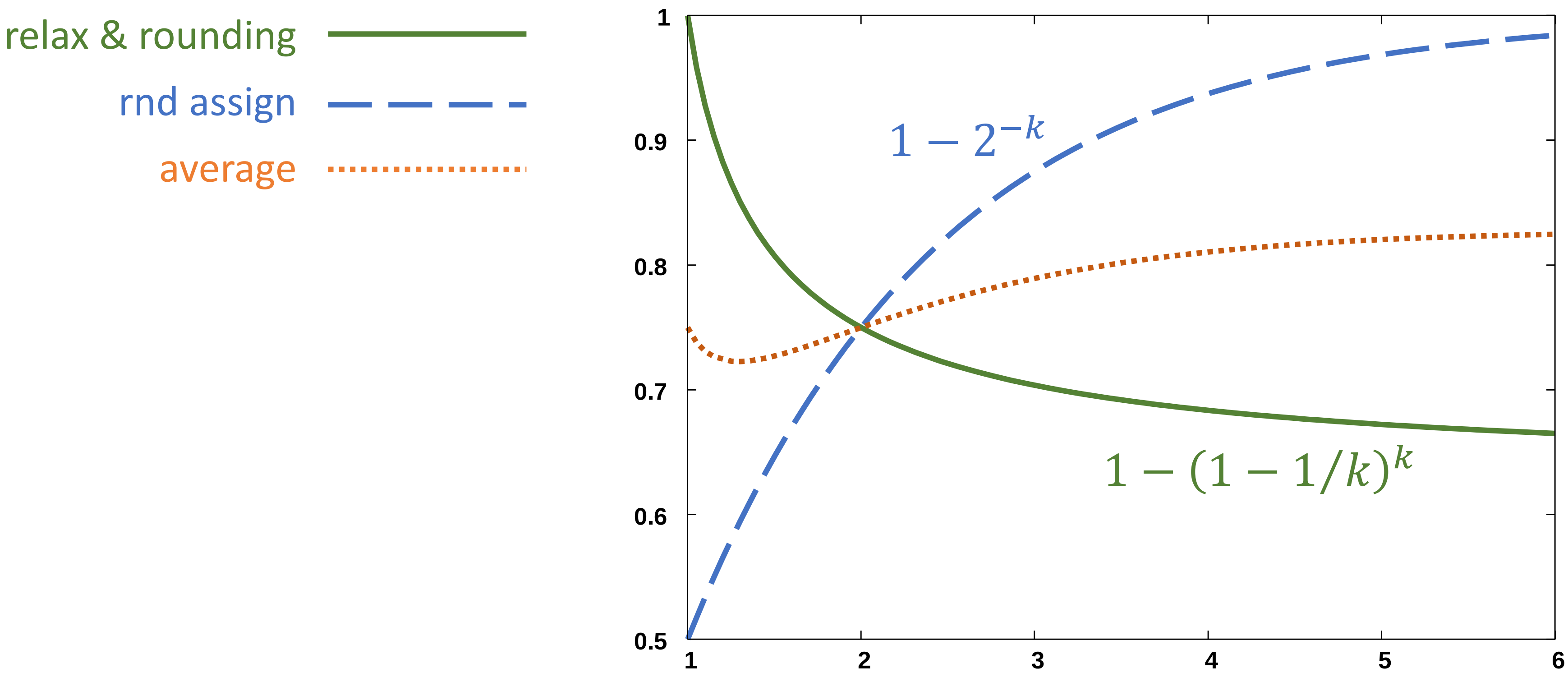
Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Random assignment:

- 1/2-approximation
- $\Pr[C_j \text{ satisfied}] = 1 - 2^{-k}$

LP relaxation and randomized rounding:

- (1-1/e)-approximation
- $\Pr[C_j \text{ satisfied}] = [1 - (1 - 1/k)^k] \cdot y_j^*$



Putting two algorithms together

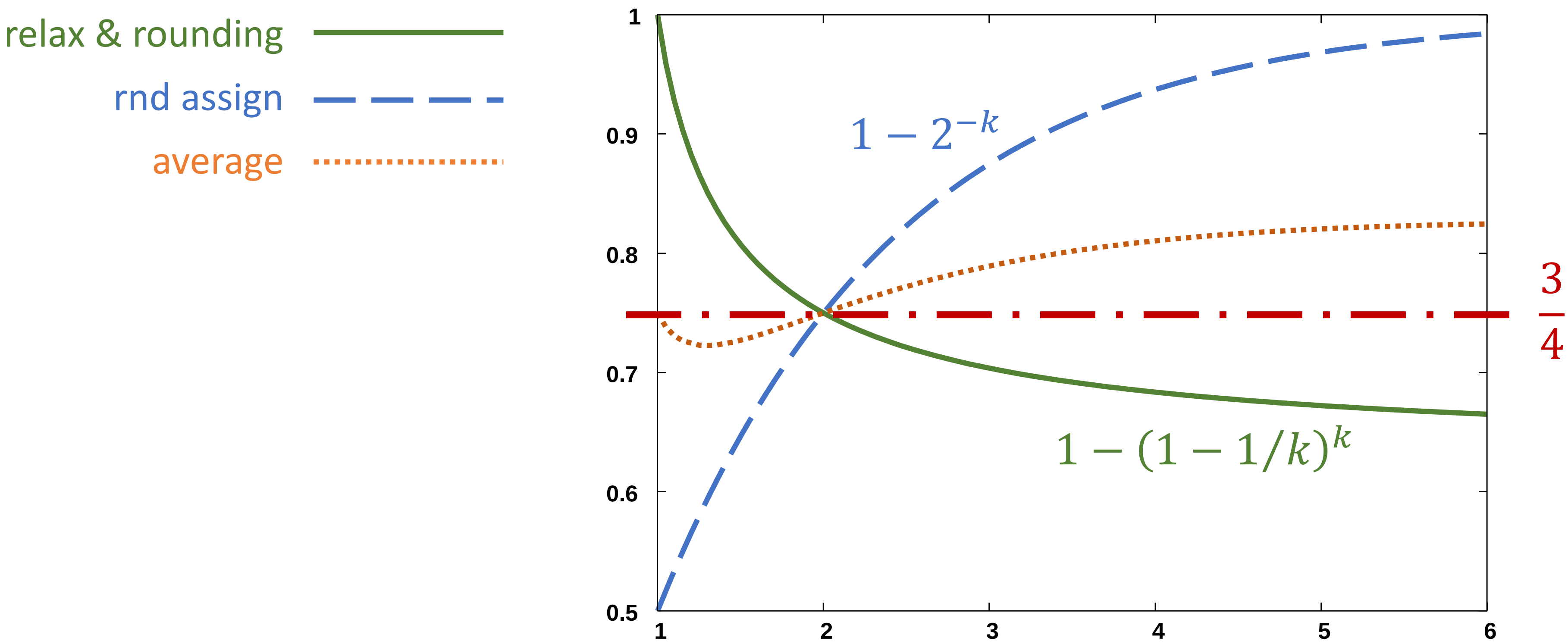
Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Random assignment:

- 1/2-approximation
- $\Pr[C_j \text{ satisfied}] = 1 - 2^{-k}$

LP relaxation and randomized rounding:

- (1-1/e)-approximation
- $\Pr[C_j \text{ satisfied}] = [1 - (1 - 1/k)^k] \cdot y_j^*$



Putting two algorithms together

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Assign each variable with true or false *uniformly* and *independently* at random.

Model problem as an ILP.
Obtain LP relaxation of the ILP.
Get optimal solution $\overrightarrow{x^*}$ of the LP.
Randomized rounding $\overrightarrow{x^*}$ to $\overrightarrow{x'}$ as SOL.

Run each of the two algorithms once, return better solution.

Putting two algorithms together

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Assign each variable with true or false *uniformly* and *independently* at random.

Model problem as an ILP.
Obtain LP relaxation of the ILP.
Get optimal solution $\overrightarrow{x^*}$ of the LP.
Randomized rounding $\overrightarrow{x^*}$ to $\overrightarrow{x'}$ as SOL.

Run each of the two algorithms once, return better solution.

Assume the solution of the rnd. assign. alg. satisfies m_1 clauses.
Assume the solution of the LP relaxation & rounding alg. satisfies m_2 clauses.

$$\mathbb{E}[\#\text{ of satisfied clauses}] = \mathbb{E}[\max\{m_1, m_2\}] \geq \mathbb{E}\left[\frac{m_1 + m_2}{2}\right]$$

Putting two algorithms together

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Run each of the two algorithms once, return better solution.

Assume the solution of the rnd. assign. alg. satisfies m_1 clauses.
Assume the solution of the LP relaxation & rounding alg. satisfies m_2 clauses.

$$\mathbb{E}[\text{\# of satisfied clauses}] = \mathbb{E}[\max\{m_1, m_2\}] \geq \mathbb{E}\left[\frac{m_1 + m_2}{2}\right]$$

Putting two algorithms together

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Run each of the two algorithms once, return better solution.

Assume the solution of the rnd. assign. alg. satisfies m_1 clauses.
Assume the solution of the LP relaxation & rounding alg. satisfies m_2 clauses.

$$\mathbb{E}[\#\text{ of satisfied clauses}] = \mathbb{E}[\max\{m_1, m_2\}] \geq \mathbb{E}\left[\frac{m_1 + m_2}{2}\right]$$

Assume clause C_j has k_j literals, and $\left(\overrightarrow{x^*}, \overrightarrow{y^*}\right)$ is the opt. sol. of relaxed-ILP

Putting two algorithms together

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Run each of the two algorithms once, return better solution.

Assume the solution of the rnd. assign. alg. satisfies m_1 clauses.
Assume the solution of the LP relaxation & rounding alg. satisfies m_2 clauses.

$$\mathbb{E}[\#\text{ of satisfied clauses}] = \mathbb{E}[\max\{m_1, m_2\}] \geq \mathbb{E}\left[\frac{m_1 + m_2}{2}\right]$$

Assume clause C_j has k_j literals, and $\left(\overrightarrow{x^*}, \overrightarrow{y^*}\right)$ is the opt. sol. of relaxed-ILP

$$\mathbb{E}[m_2] \geq \sum_{j=1}^{m} \left(1 - \left(1 - \frac{1}{k_j}\right)^{k_j}\right) \cdot y_j^*$$

Putting two algorithms together

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Run each of the two algorithms once, return better solution.

Assume the solution of the rnd. assign. alg. satisfies m_1 clauses.
Assume the solution of the LP relaxation & rounding alg. satisfies m_2 clauses.

$$\mathbb{E}[\text{\# of satisfied clauses}] = \mathbb{E}[\max\{m_1, m_2\}] \geq \mathbb{E}\left[\frac{m_1 + m_2}{2}\right]$$

Assume clause C_j has k_j literals, and $\left(\overrightarrow{x^*}, \overrightarrow{y^*}\right)$ is the opt. sol. of relaxed-ILP

$$\mathbb{E}[m_2] \geq \sum_{j=1}^{m} \left(1 - \left(1 - \frac{1}{k_j}\right)^{k_j}\right) \cdot y_j^*$$

$$\mathbb{E}[m_1] = \sum_{j=1}^{m} \left(1 - 2^{-k_j}\right) \geq \sum_{j=1}^{m} \left(1 - 2^{-k_j}\right) \cdot y_j^*$$

Putting two algorithms together

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Run each of the two algorithms once, return better solution.

Assume the solution of the rnd. assign. alg. satisfies m_1 clauses.
Assume the solution of the LP relaxation & rounding alg. satisfies m_2 clauses.

$$\mathbb{E}[\text{\# of satisfied clauses}] = \mathbb{E}[\max\{m_1, m_2\}] \geq \mathbb{E}\left[\frac{m_1 + m_2}{2}\right]$$

$$\geq \sum_{j=1}^{m} \left[\left(1 - 2^{-k_j}\right) + \left(1 - \left(1 - 1/k_j\right)^{k_j}\right)\right] \cdot \frac{y_j^*}{2}$$

Assume clause C_j has k_j literals, and $\left(\overrightarrow{x^*}, \overrightarrow{y^*}\right)$ is the opt. sol. of relaxed-ILP

$$\mathbb{E}[m_2] \geq \sum_{j=1}^{m} \left(1 - \left(1 - \frac{1}{k_j}\right)^{k_j}\right) \cdot y_j^*$$

$$\mathbb{E}[m_1] = \sum_{j=1}^{m} \left(1 - 2^{-k_j}\right) \geq \sum_{j=1}^{m} \left(1 - 2^{-k_j}\right) \cdot y_j^*$$

Putting two algorithms together

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Run each of the two algorithms once, return better solution.

Assume the solution of the rnd. assign. alg. satisfies m_1 clauses.
Assume the solution of the LP relaxation & rounding alg. satisfies m_2 clauses.
Assume clause C_j has k_j literals, and $\left(\overrightarrow{x^*}, \overrightarrow{y^*}\right)$ is the opt. sol. of relaxed-ILP

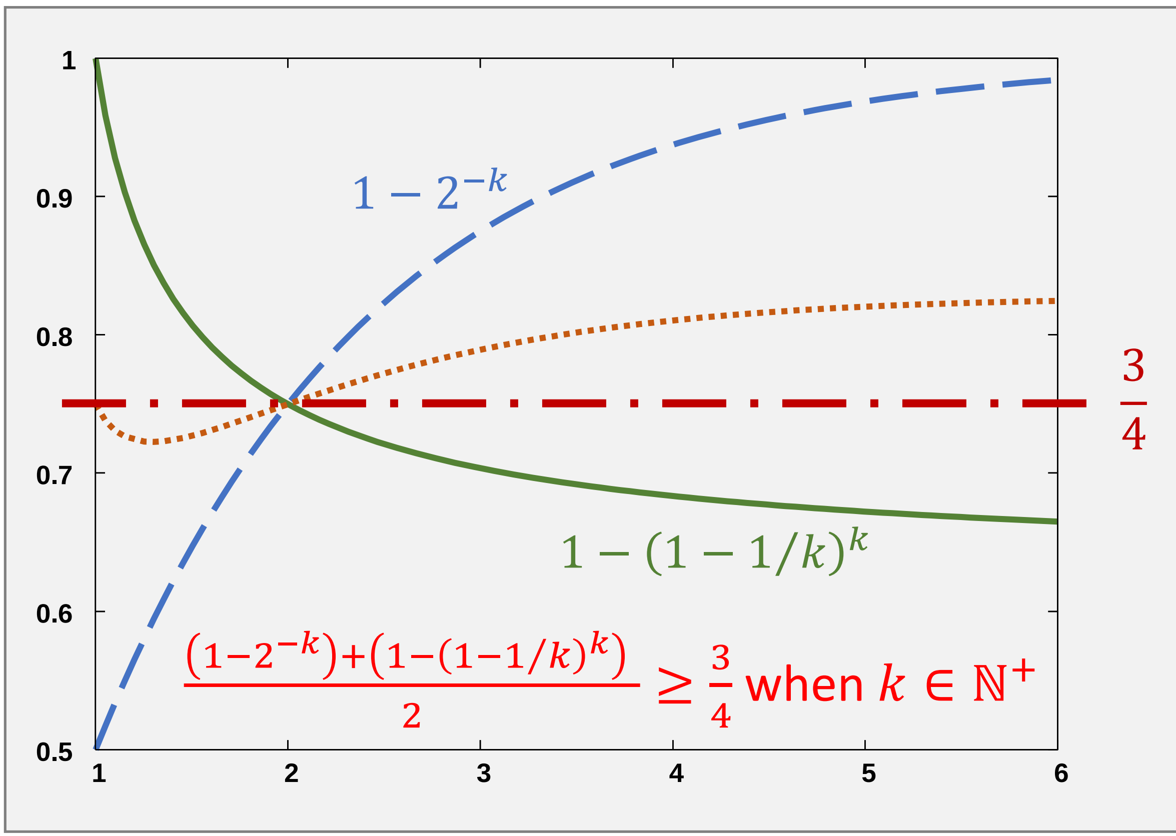
$$\mathbb{E}[\#\text{ of satisfied clauses}] = \mathbb{E}[\max\{m_1, m_2\}] \geq \mathbb{E}\left[\frac{m_1 + m_2}{2}\right]$$

$$\geq \sum_{j=1}^{m} \left[\left(1 - 2^{-k_j}\right) + \left(1 - \left(1 - 1/k_j\right)^{k_j}\right)\right] \cdot \frac{y_j^*}{2}$$

Putting two algorithms together

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Run each of the two algorithms once, return better solution.



$_1$ clauses.

lg. satisfies m_2 clauses.

pt. sol. of relaxed-ILP

$$\frac{\quad + m_2}{2}\Big]$$

$$\left(1 - 1/k_j\right)^{k_j}\Big)\Big] \cdot \frac{y_j^*}{2}$$

Putting two algorithms together

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Run each of the two algorithms once, return better solution.

Assume the solution of the rnd. assign. alg. satisfies m_1 clauses.
Assume the solution of the LP relaxation & rounding alg. satisfies m_2 clauses.
Assume clause C_j has k_j literals, and $(\overrightarrow{x^*}, \overrightarrow{y^*})$ is the opt. sol. of relaxed-ILP

$$\mathbb{E}[\#\text{ of satisfied clauses}] = \mathbb{E}[\max\{m_1, m_2\}] \geq \mathbb{E}\left[\frac{m_1 + m_2}{2}\right]$$

$$\geq \sum_{j=1}^{m} \left[\left(1 - 2^{-k_j}\right) + \left(1 - \left(1 - 1/k_j\right)^{k_j}\right)\right] \cdot \frac{y_j^*}{2}$$

$$\geq \sum_{j=1}^{m} \frac{3}{4} \cdot y_j^* = \frac{3}{4} \cdot \text{OPT}_{\text{LP}} \geq \frac{3}{4} \cdot \text{OPT}$$

MAX-SAT

Instance: A set of clauses $C_1, C_2, \cdots, C_m$.
MAX-SAT: Find $\vec{x} \in \{\text{true}, \text{false}\}^n$ that maximize # of satisfied clauses.

Assign each variable with true or false *uniformly* and *independently* at random.

Model problem as an ILP.
Obtain LP relaxation of the ILP.
Get optimal solution $\overrightarrow{x^*}$ of the LP.
Randomized rounding $\overrightarrow{x^*}$ to $\overrightarrow{x'}$ as SOL.

Run each of the two algorithms once, return better solution.

- A rnd. alg. that satisfies at least $(3/4) \cdot \mathrm{OPT}$ clauses *in expectation*.
- Can derandomize above alg. via the method of conditional expectation.
- The integrality gap of the LP relaxation for MAX-SAT is $3/4$.
- MAX-3SAT has a $(7/8)$-approx. alg. by semidefinite programming, and cannot have better approx. alg. in poly-time unless P=NP.
- How about MAX-E3SAT?