Introduction to
Approximation Algorithms

Mong-Jen Kao (高孟駿)

Friday 13:20 – 15:10

Outline

- Linear Programming based Methods
  - Linear Programming (LP)
  - Formulating Optimization Problems as Linear Programs
- A 2-Approximation for Vertex Cover by *LP-rounding*
- Typical Flowchart for Rounding-based Methods
  - Formulation, Relaxation, Rounding

Linear Programming based Methods

"*A large fraction of the theory for approximation algorithms,*
as we know it today, is built around linear programming." by V. Vazirani, 2001.

"*This is still true today*." by M.-J. Kao, 2021.

Linear Programming (LP)

- ***Linear programming***, or, *linear optimization*, is ***to optimize a linear objective function under a set of linear constraints*** in the real multi-dimensional space.

$$
\begin{aligned}
\max \quad & 2x_1 + x_2 \\
\text{s.t.} \quad & x_1 - x_2 \geq -3, \\
& x_1 + x_2 \leq 13, \\
& 3 \leq x_1 \leq 7, \\
& x_2 \geq 2.
\end{aligned}
$$

Find $x_1, x_2 \in \mathbb{R}$

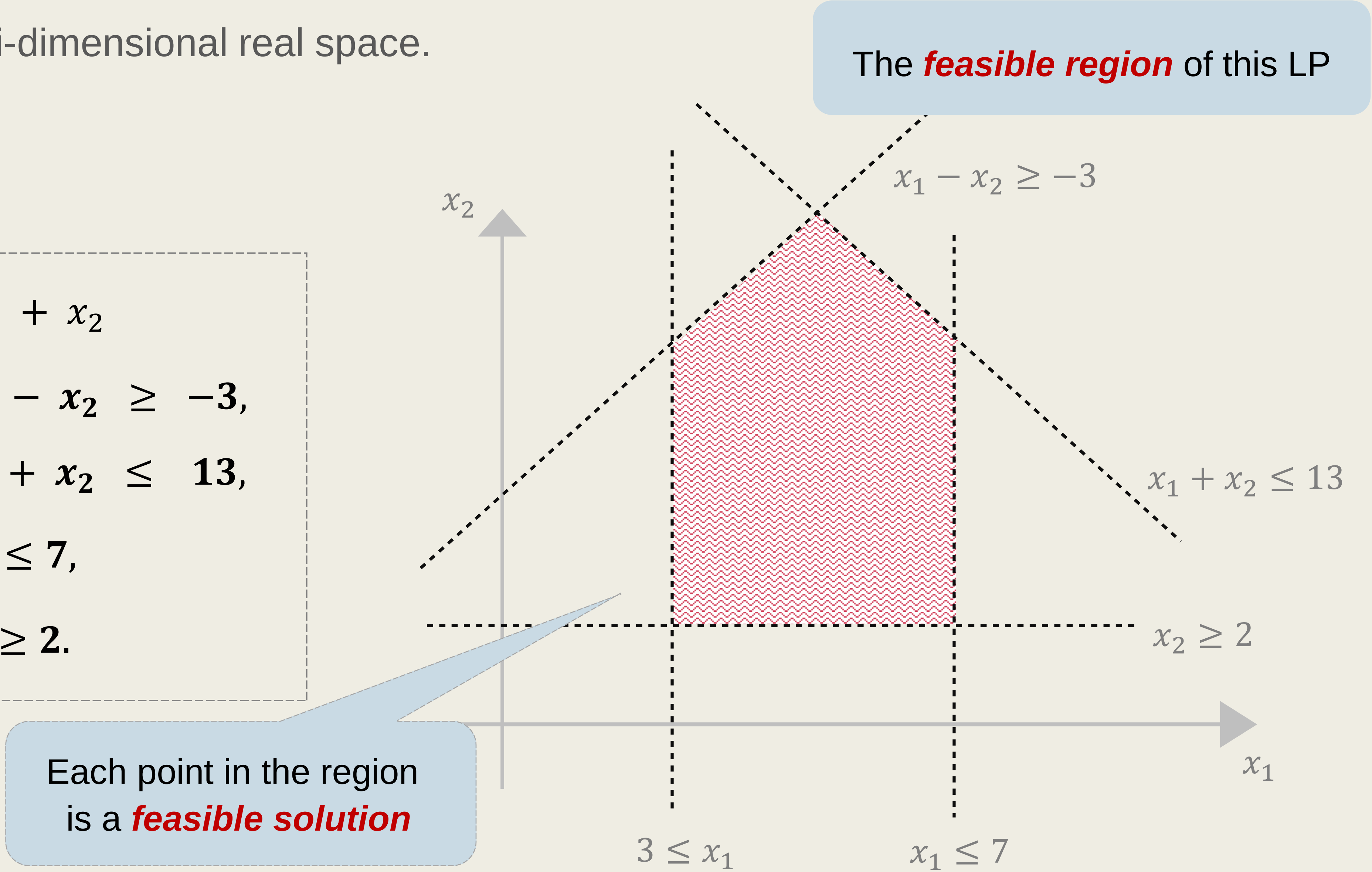
that satisfies

$$
\begin{aligned}
& x_1 - x_2 \geq -3, \\
& x_1 + x_2 \leq 13, \\
& 3 \leq x_1 \leq 7, \\
& x_2 \geq 2.
\end{aligned}
$$

such that $2x_1 + x_2$ is maximized

- ***Linear programming***, or, *linear optimization*, is ***to optimize <u>a linear objective function</u> under <u>a set of linear constraints</u>*** in the multi-dimensional real space.

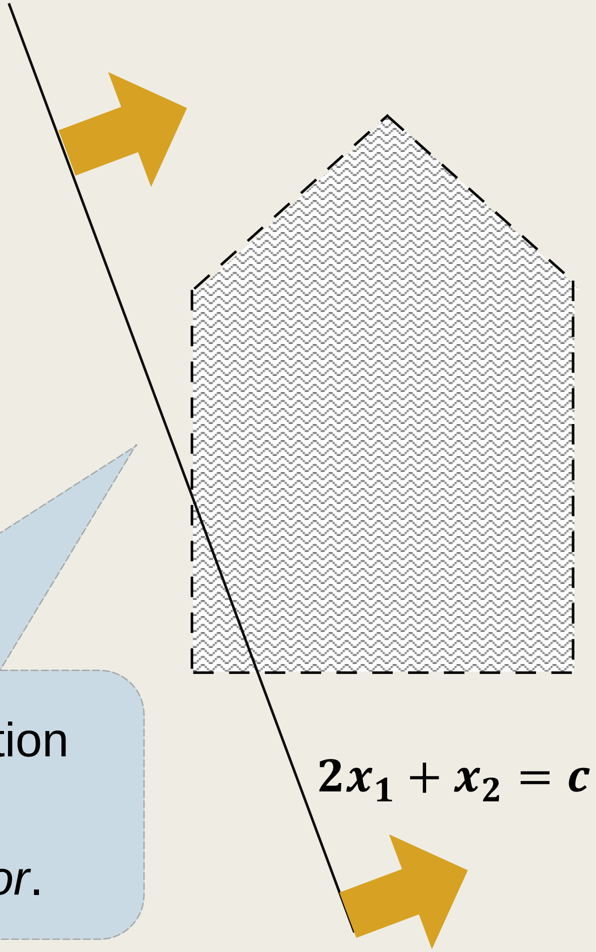
$$
\begin{aligned}
\max \quad & 2x_1 + x_2 \\
\text{s.t.} \quad & \boldsymbol{x_1 - x_2 \geq -3,} \\
& \boldsymbol{x_1 + x_2 \leq 13,} \\
& \boldsymbol{3 \leq x_1 \leq 7,} \\
& \boldsymbol{x_2 \geq 2.}
\end{aligned}
$$



- ***Linear programming***, or, *linear optimization*, is ***to optimize <u>a linear objective function</u> under <u>a set of linear constraints</u>*** in the multi-dimensional real space.

$$\begin{aligned}
\max \quad & 2x_1 + x_2 \\
\text{s.t.} \quad & x_1 - x_2 \geq -3, \\
& x_1 + x_2 \leq 13, \\
& 3 \leq x_1 \leq 7, \\
& x_2 \geq 2.
\end{aligned}$$

The process of optimizing $\boldsymbol{2x_1 + x_2}$



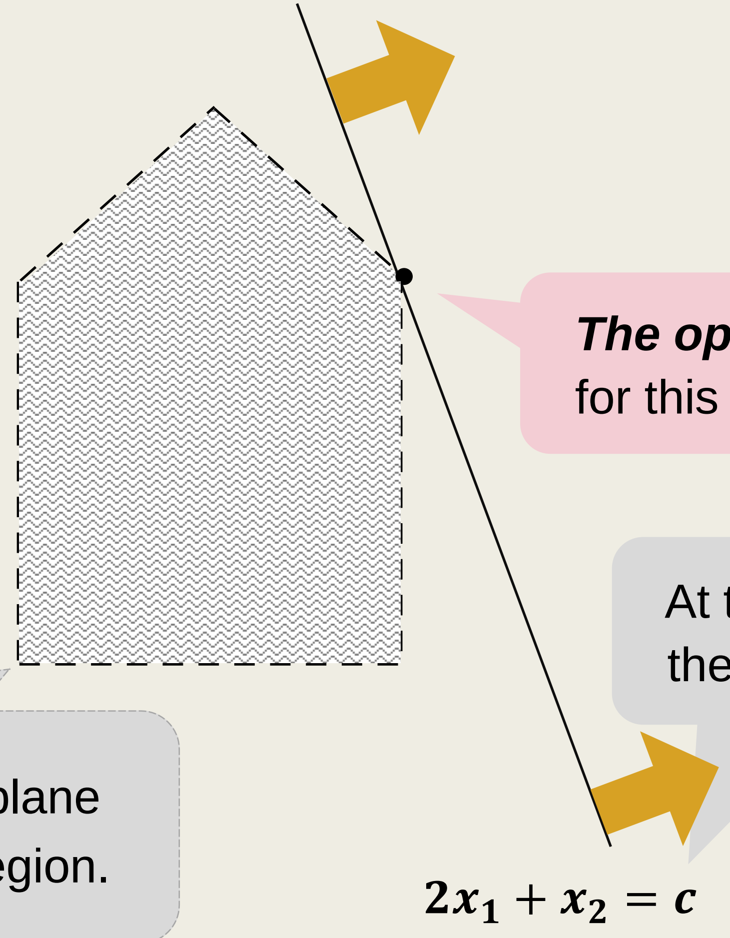
The value of the objective function ***increases*** as *the hyperplane moves towards its normal vector*.

Hyperplane with normal vector $\overrightarrow{(2,1)}$

- ***Linear programming***, or, *linear optimization*, is ***to optimize <u>a linear objective function</u> under <u>a set of linear constraints</u>*** in the multi-dimensional real space.

$$
\begin{aligned}
\max \quad & 2x_1 + x_2 \\
\text{s.t.} \quad & x_1 - x_2 \geq -3, \\
& x_1 + x_2 \leq 13, \\
& 3 \leq x_1 \leq 7, \\
& x_2 \geq 2.
\end{aligned}
$$

The process of optimizing $2x_1 + x_2$



The optimal solution for this LP.

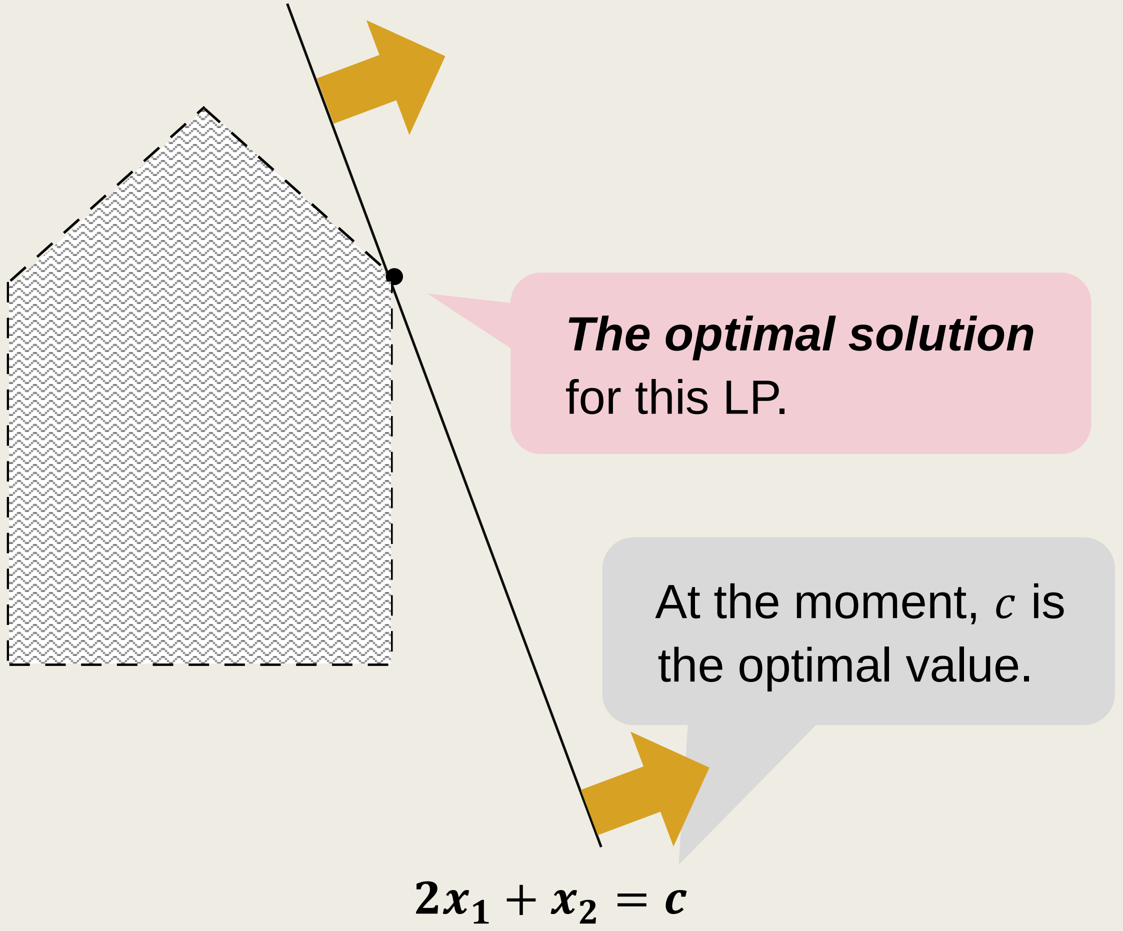
At the moment, c is the optimal value.

We reach the optimal solution when the hyperplane is about to become disjoint with the feasible region.

- ***Linear programming***, or, *linear optimization*, is ***to optimize a linear objective function under a set of linear constraints*** in the multi-dimensional real space.

The process of optimizing $2x_1 + x_2$

In fact, it can be shown that, if the considered LP has an optimal solution, then there must be a ***vertex* on the boundary of the polytope** that is optimal.



Linear Programming

as a Computation Problem

Linear Programming as a Computation Problem

- Linear programming **can be solved in *weakly polynomial time***.
  i.e., in time polynomial in the input length
  but not necessarily in the number of variables and constraints.

- Given $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$,
  we can compute an optimal $x \in \mathbb{R}^n$ for the program

$$\max \ \boldsymbol{c}^T \cdot \boldsymbol{x}$$
$$A \cdot \boldsymbol{x} \leq \boldsymbol{b},$$
$$\boldsymbol{x} \geq 0.$$

in time *polynomial in the input length*.

- Linear programming **can be solved in *weakly polynomial time***.
  i.e., in time polynomial in the input length
  but not necessarily in the number of variables and constraints.

  - There are a number of nice algorithms.
    - *Simplex method*, *Interior-point method*, *Ellipsoid methods*, etc.

Practically useful, not poly-time.

weakly poly-time.

- *Whether or not LP can be solved **in strongly polynomial time***
  is listed as *one of 18 greatest open problems in Mathematics in the 21st century*.

Formulating Combinatorial Optimization Problems

as Linear Programming Problems

or, more generally, Mathematical Programming Problems

Sometimes simple;
Most of the time an Art.

Formulating the Combinatorial Optimization Problems

- In the combinatorial optimization problems,
  we always have *some decisions* to make.

- The idea is to
  - Encode *each decision* as a ***decision variable***.
  - Convert problem requirements into *linear constraints*.

Ex. Vertex Cover

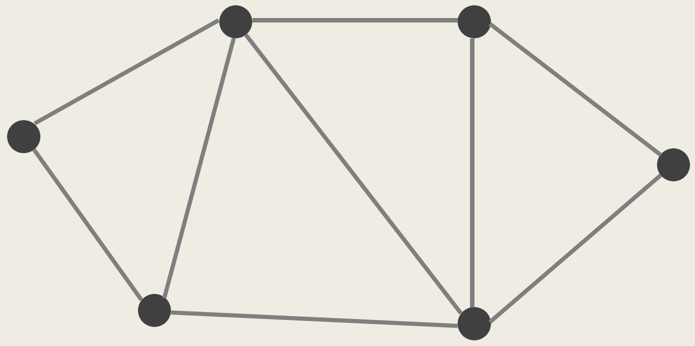
- Given a graph $G = (V, E)$ and a weight function $w : V \to Q^+$, compute a minimum-weight vertex subset $U \subseteq V$ such that, any $e \in E$ has at least one endpoint in U.

  – *Decisions to make*

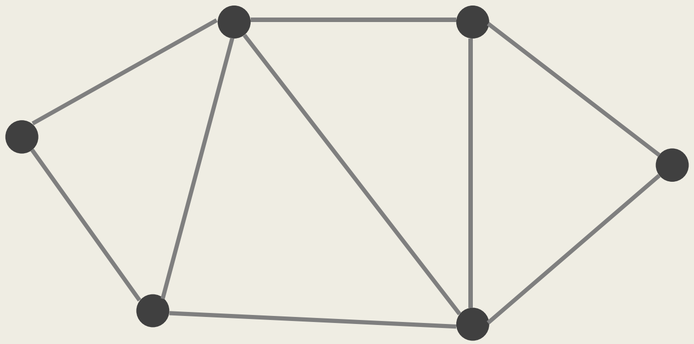
  For each vertex $v \in V$, should v be picked?

Variable x_v with $x_v \in \{0,1\}$

$x_v = 1 \leftrightarrow$ chosen



- Given a graph $G = (V, E)$, compute a minimum-weight subset $U \subseteq V$ such that, any $e \in E$ is adjacent to some $v \in U$.



  - *Decision encoding*

    For each $v \in V$, create $x_v \in \{0,1\}$

    $x_v = 1 \leftrightarrow$ chosen

  - *Linear constraints*

    Each $e = (u, v) \in E$ has to be covered.

    For each $(u, v) \in E$,
    $x_u + x_v \geq 1$ must hold.

    At least one of u, v needs to be selected.

  - *Objective*

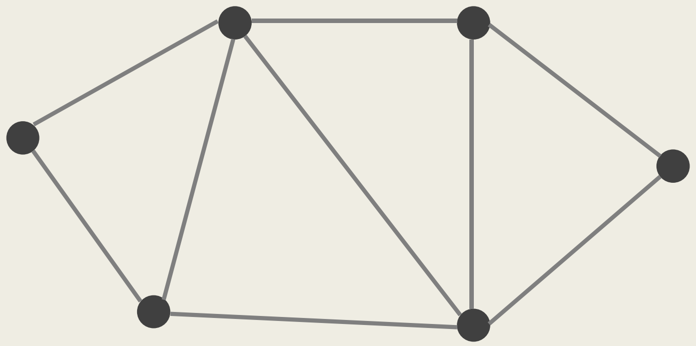
    To minimize $\sum_{v \in V} w_v \cdot x_v$

An Integer Linear Program for Vertex Cover

$$\min \sum_{v \in V} w_v \cdot x_v$$

$$\text{s.t.} \quad x_u + x_v \geq 1, \qquad \forall\, (u, v) \in E,$$

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

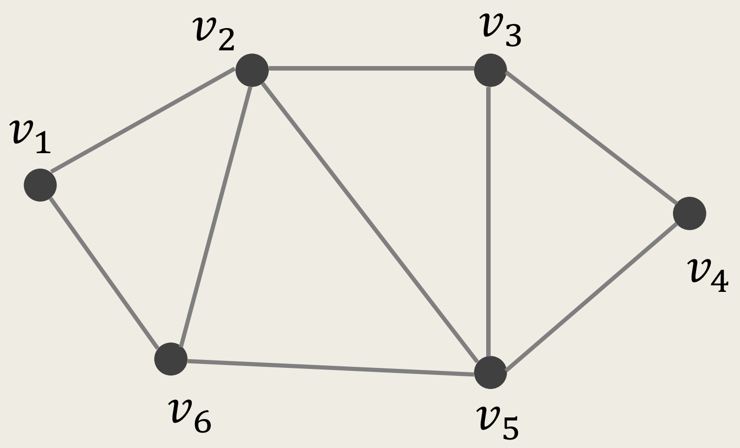


- We have formulated Vertex Cover as an ***Integer Linear Program (ILP)*** in a natural way.

An Integer Linear Program for Vertex Cover

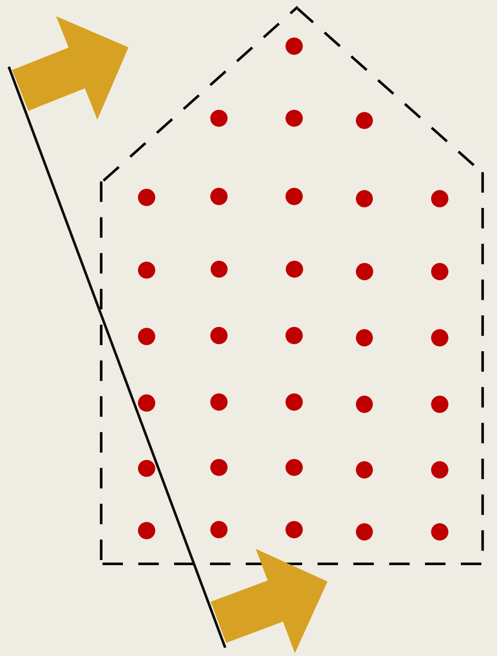
- We have formulated Vertex Cover as an ***Integer Linear Program (ILP)*** in a natural way.

$$
\begin{aligned}
\min \quad & x_1 + x_2 + \cdots + x_6 \\
\text{s.t.} \quad & \mathrm{x}_1 + \mathrm{x}_2 \geq 1, \quad x_2 + x_3 \geq 1, \\
& x_3 + x_4 \geq 1, \quad x_4 + x_5 \geq 1, \\
& x_5 + x_6 \geq 1, \quad x_6 + x_1 \geq 1, \\
& x_2 + x_5 \geq 1, \quad x_2 + x_6 \geq 1, \\
& x_3 + x_5 \geq 1, \\
& x_1, x_2, \ldots, x_6 \in \{0,1\}.
\end{aligned}
$$



ILP is NP-hard

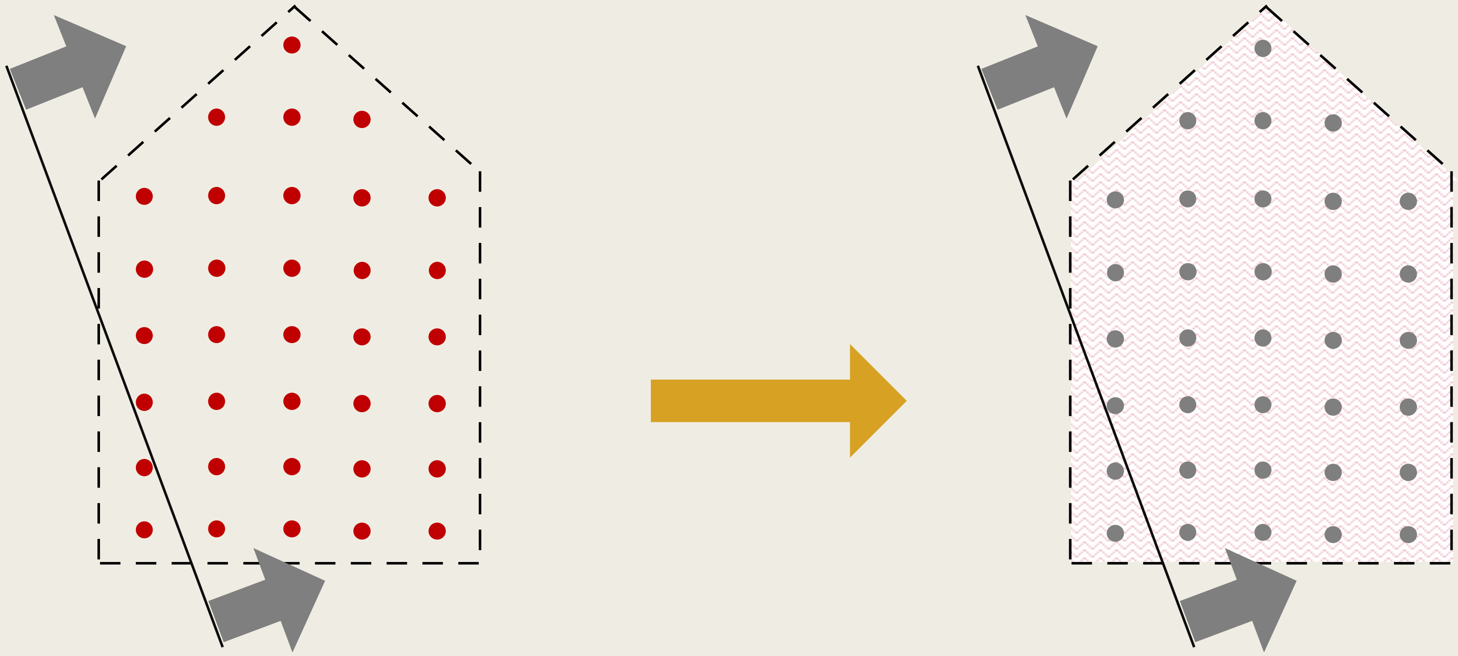
- Solving Integer Linear Program (ILP) is in general NP-hard.
  - It has to be, since many NP-hard problems can be formulated as ILPs in a natural way, including vertex cover.



Computing ***the optimal grid-point is hard in general***, as it should be.

LP Relaxations for ILPs

- By relaxing the range of the variables to real numbers,
  we get an ***LP relaxation***, which can be solved (in weakly poly-time).



An LP Relaxation for Vertex Cover

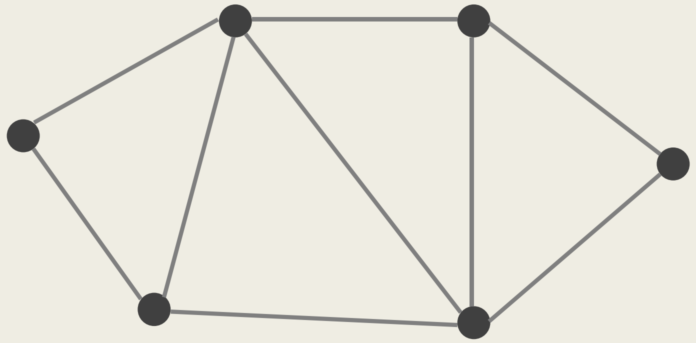
- By relaxing the range of the variables to real numbers, we get an ***LP relaxation***, which can be solved (in weakly poly-time).

$$\min \sum_{v\in V} x_v$$

$$\text{s.t.} \quad x_u + x_v \geq 1, \qquad \forall (u,v) \in E,$$

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

$$\boldsymbol{x_v \geq 0,}$$

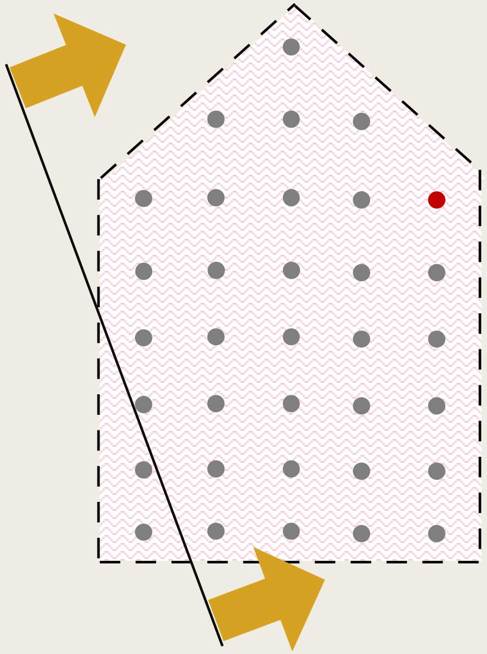


The constraint $x_v \leq 1$ is ***not needed here***. Why?

LP Relaxation for Bounds on ILP

- A very nice & useful property given by relaxations is that,

  optimal solutions for the relaxations directly give ***bounds*** to *the value of the original ILP*.



The **optimal solution for the ILP**,
which is *one of the grid points*,
is ***a feasible solution for the relaxed LP***.

Hence, the ***optimal value of the relaxed LP*** must be ***no worse*** than ***that of the ILP***.

LP Relaxation for Bounds on ILP

- *Optimal solutions for the relaxations* directly give ***bounds*** to *the value of the original ILP*.

$$\begin{aligned} \min \quad & \sum_{v \in V} w_v \cdot x_v & (**) \\ \text{s.t.} \quad & x_u + x_v \geq 1, & \forall\, (u, v) \in E, \\ & x_v \in \{0,1\}, & \forall\, v \in V. \end{aligned}$$

$$\begin{aligned} \min \quad & \sum_{v \in V} w_v \cdot x_v & (*) \\ \text{s.t.} \quad & x_u + x_v \geq 1, & \forall\, (u, v) \in E, \\ & x_v \geq 0, & \forall\, v \in V. \end{aligned}$$

- Then, $\text{Val}(*) \leq \text{Val}(**)$.

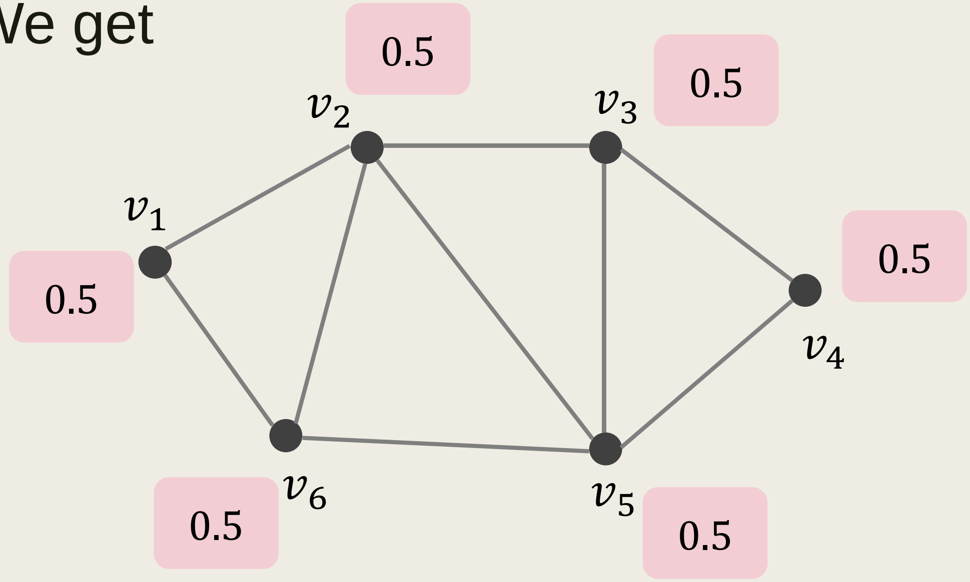
Optimal Fractional Solution of LP Relaxation

--- The *"Secret Message"* from the *Almighty Oracle*

- Solve the LP relaxation for an optimal (fractional) solution.

$$
\begin{aligned}
\min \quad & x_1 + x_2 + \cdots + x_6 \\
\text{s.t.} \quad & x_1 + x_2 \geq 1, \quad x_2 + x_3 \geq 1, \\
& x_3 + x_4 \geq 1, \quad x_4 + x_5 \geq 1, \\
& x_5 + x_6 \geq 1, \quad x_6 + x_1 \geq 1, \\
& x_2 + x_5 \geq 1, \quad x_2 + x_6 \geq 1, \\
& x_3 + x_5 \geq 1, \\
& x_1, x_2, \ldots, x_6 \geq 0.
\end{aligned}
$$

We get



"The best way for <u>your graph</u>, my friend,"

*"is **to select one-half of each vertex**,"* said the almighty oracle.

"Does this help in our problem?" We ask.

A Simple 2-approximation for Vertex Cover

1. Solve LP (*) for an optimal x^*.

2. (***rounding***)

   For each $v \in V$, define

$$\widehat{x_v} := \begin{cases} 1, & \text{if } x_v^* \geq \frac{1}{2}, \\ 0, & \text{otherwise.} \end{cases}$$

3. Output $\hat{x}$,

   i.e., the set described by $\hat{x}$.

$$\min \sum_{v \in V} w_v \cdot x_v \qquad (*)$$

$$\text{s.t.} \quad x_u + x_v \geq 1, \qquad \forall\, (u, v) \in E,$$

$$x_v \geq 0, \qquad \forall\, v \in V.$$

The Feasibility

- We need to show that, $\hat{x}$ is a feasible solution for the vertex cover problem, i.e., feasible for the ILP (**).
  - Consider any $(u, v) \in E$.
    - We have $x_u^* + x_v^* \geq 1$, since x^* is feasible for LP (*).

      Hence, at least one of x_u^*, x_v^* is at least $1/2$.

      This means that at least one of u, v will be rounded up, and $\widehat{x_u} + \widehat{x_v} \geq 1$ holds as well.

The Approximation Guarantee

- Since we only round up x_u^* when it is at least $1/2$,

  it follows that

$$\sum_{v \in V} \widehat{x_v} \leq 2 \cdot \sum_{v \in V} x_v^* = 2 \cdot OPT_f \leq 2 \cdot OPT,$$

where OPT_f is the value of the optimal solution for LP (*) and OPT is the optimal solution of the vertex cover instance.

Typical Flowchart

for Rounding-based Methods

Typical Flowchart for LP-based Methods (so far)

Problem Π of Interests

↕ **Design (properly)**

An ILP Formulation for Π

- Given a graph $G = (V, E)$, compute a minimum size subset $U \subseteq V$ such that, any $e \in E$ is adjacent to some $v \in U$.

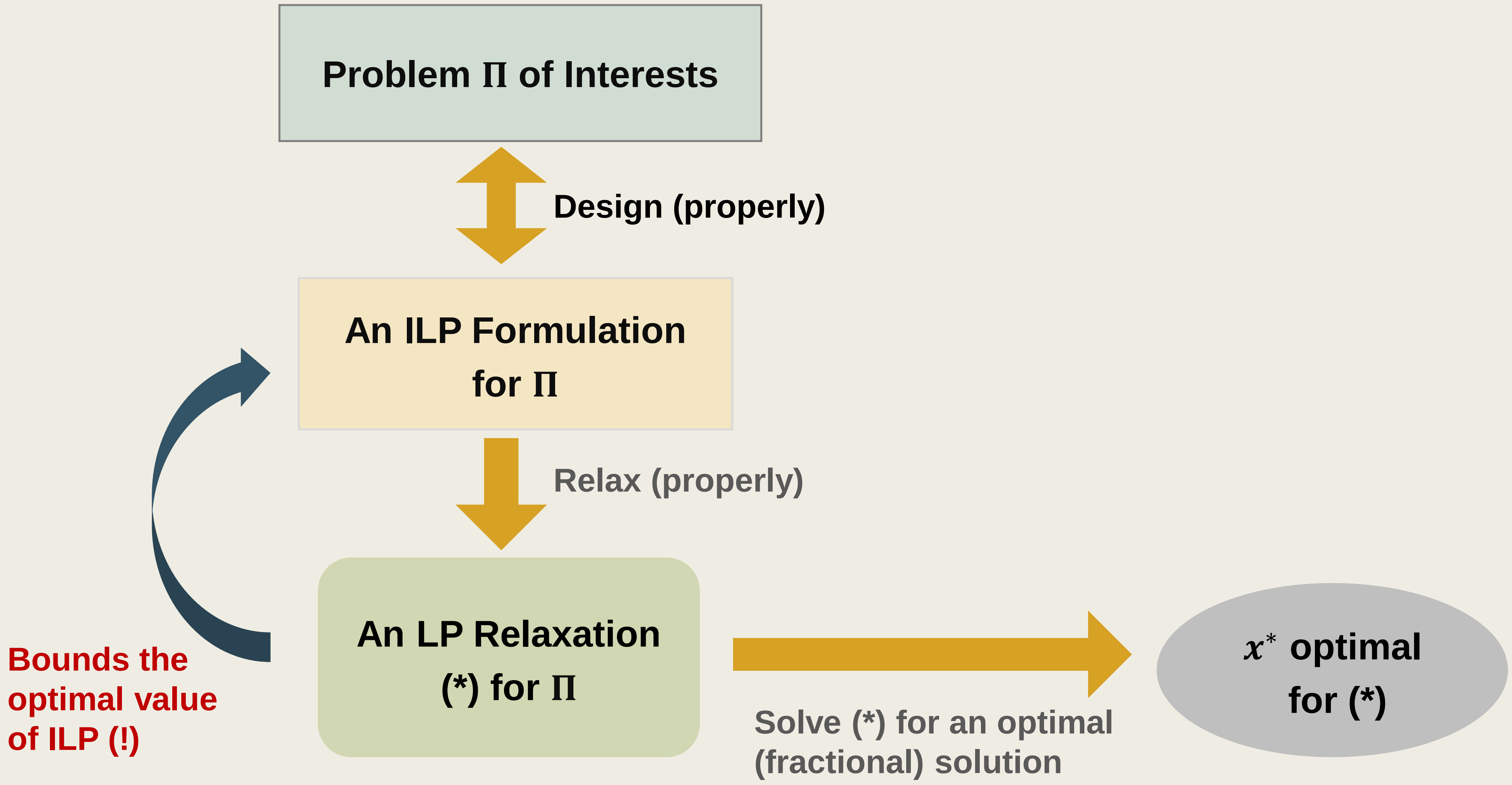
↕

$$\min \sum_{v \in V} x_v$$

$$\text{s.t.} \quad x_u + x_v \geq 1, \qquad \forall\, (u, v) \in E,$$

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

Typical Flowchart for LP-based Methods (so far)



Typical Flowchart for ***LP Rounding-based*** Methods