Matrix Form of VC-IP

Minimize $w_{\mathrm{IP}}(\boldsymbol{x}) = \boldsymbol{w}^t\, \boldsymbol{x}$

subject to $\mathbf{1} \geq \boldsymbol{x} \geq \mathbf{0}$, $A\boldsymbol{x} \geq \mathbf{1}$, and $\boldsymbol{x}$ is an integer vector.

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_{|V|})^t \qquad \mathbf{1} = (1, 1, \ldots, 1)^t$$

$$\boldsymbol{w} = (w_1, w_2, \ldots, w_{|V|})^t \qquad \mathbf{0} = (0, 0, \ldots, 0)^t$$

Matrix *A*: Rows of A correspond to edges in E
Columns of A correspond to vertexes in V

$$A[e, i] = \begin{cases} 1 & \text{if vertex } v_i \text{ is an end of edge } e \\ 0 & \text{otherwise} \end{cases}$$

If $\boldsymbol{x}^*$ is the optimal solution of VC-IP, $S = \{v_i \in V\colon x_i^* = 1\}$ is the optimal vertex cover S^* with the minimum total weight $w(S^*)$.

VC-LP: Linear Programming Relaxation of VC-IP

Minimize $w_{\mathrm{LP}}(\boldsymbol{x}) = \boldsymbol{w}^t\, \boldsymbol{x}$

subject to $\mathbf{1} \geq \boldsymbol{x} \geq \mathbf{0}$, $A\boldsymbol{x} \geq \mathbf{1}$~~, and $\boldsymbol{x}$ is an integer vector~~.

Optimal value of VC-LP $\leq$ Optimal value of VC-IP

$$w_{\mathrm{LP}}(\boldsymbol{x}^*_{\mathrm{LP}}) \leq w_{\mathrm{IP}}(\boldsymbol{x}^*_{\mathrm{IP}})$$

LP: Linear Programming

Most frequently used optimization method
(a number of software packages are available)

The following relation holds among the LP solution $\boldsymbol{x}^*_{\mathrm{LP}}$, the optimal solution S^*, and any greedy solution S:

$$w_{\mathrm{LP}}(\boldsymbol{x}^*_{\mathrm{LP}}) \leq w_{\mathrm{IP}}(\boldsymbol{x}^*_{\mathrm{IP}}) = w(S^*) \leq w(S)$$

VC-LP: Linear Programming Relaxation of VC-IP

Minimize $w_{\mathrm{LP}}(\boldsymbol{x}) = \boldsymbol{w}^t \boldsymbol{x}$

subject to $\mathbf{1} \geq \boldsymbol{x} \geq \mathbf{0}$, $A\boldsymbol{x} \geq \mathbf{1}$.

Different optimal solutions

Optimal value of VC-LP $\leq$ Optimal value of VC-IP

$$w_{\mathrm{LP}}(\boldsymbol{x}^*_{\mathrm{LP}}) \leq w_{\mathrm{IP}}(\boldsymbol{x}^*_{\mathrm{IP}})$$

This can be an invalid solution as a vertex cover (e.g., $x_1 = 0.5$, $x_2 = 0.5$, $x_3 = 1$).

LP: Linear Programming

Most frequently used optimization method
(a number of software packages are available)

The following relation holds among the LP solution $\boldsymbol{x}^*_{\mathrm{LP}}$, the optimal solution S^*, and any greedy solution S:

$$w_{\mathrm{LP}}(\boldsymbol{x}^*_{\mathrm{LP}}) \leq w_{\mathrm{IP}}(\boldsymbol{x}^*_{\mathrm{IP}}) = w(S^*) \leq w(S)$$

Creation of an IP solution from the LP optimal solution

Minimize $w_{\mathrm{LP}}(\boldsymbol{x}) = \boldsymbol{w}^t\,\boldsymbol{x}$

subject to $\mathbf{1} \geq \boldsymbol{x} \geq \mathbf{0}$, $\quad A\boldsymbol{x} \geq \mathbf{1}$

Let $\boldsymbol{x}^*$ be the optimal solution of VC-LP. A vertex cover S is obtained by $S = \{i \in V \mid x_i^* \geq 1/2\}$. **LP-based Method.**

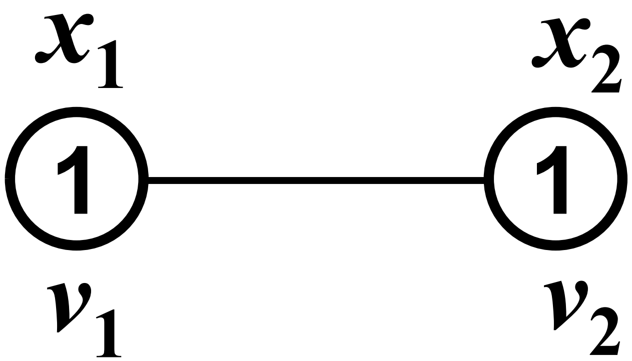
[Cover] Consider an edge $(i, j) \in E$. Since $x_i^* + x_j^* \geq 1$, either $x_i^* \geq 1/2$ or $x_j^* \geq 1/2$. Thus the edge (i, j) is covered by S.

[2-approximation] Let S^* be the optimal vertex cover. Then

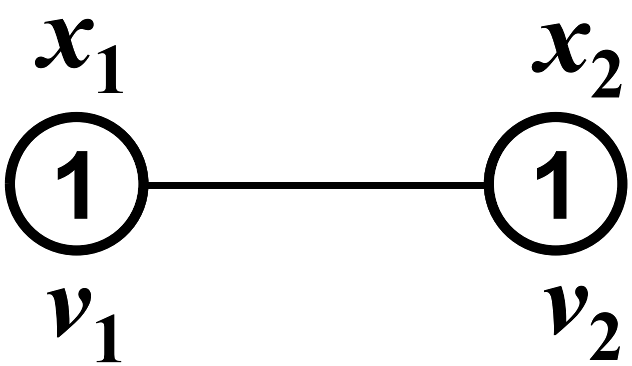
$$\sum_{i \in S^*} w_i \geq \sum_{i \in V} w_i\, x_i^* \geq \sum_{i \in S} w_i\, x_i^* \geq \frac{1}{2} \sum_{i \in S} w_i \quad \Rightarrow \quad 2w(S^*) \geq w(S)$$

$\boldsymbol{x}^*$ is the solution of the relaxation problem. $\quad x_i^* \geq 1/2$

A Simple Example



A Simple Example



Solve this simple example using the LP method (i.e., show the obtained cover(s) by the LP method).

Step 1: To formulate an LP problem.
Step 2: To solve the formulated LP problem.
Step 3: To generate a vertex cover *S*.

Q. How many solutions (i.e., vertex covers) can be obtained ?

General Discussions: Relaxation Problem

0-1 Integer Programming Problem

Minimize $z = f(\boldsymbol{x})$

subject to $g(\boldsymbol{x}) \geq 0, \quad x_i = 0 \text{ or } 1 \text{ for } i = 1, 2, \ldots, n$

Relaxation Problem

Minimize $y = f(\boldsymbol{x})$

subject to $g(\boldsymbol{x}) \geq 0, \quad 0 \leq x_i \leq 1 \text{ for } i = 1, 2, \ldots, n$

(1) The optimal value y^* of the relaxation problem is the same as or better than that of the original problem z^*: $y^* \leq z^*$

(2) If the optimal solution $\boldsymbol{x}^*$ of the relaxation problem is an integer vector, $\boldsymbol{x}^*$ is also the optimal solution of the original problem.

General Discussions: Relaxation Problem

Integer Programming Problem (L_i and U_i are integers)

Minimize $z = f(\boldsymbol{x})$

subject to $g(\boldsymbol{x}) \geq 0, \quad x_i \in \{L_i, L_i + 1, \ldots, U_i\}$ for $i = 1, 2, \ldots, n$

Relaxation Problem

Minimize $y = f(\boldsymbol{x})$

subject to $g(\boldsymbol{x}) \geq 0, \quad L_i \leq x_i \leq U_i$ for $i = 1, 2, \ldots, n$

(1) The optimal value y^* of the relaxation problem is the same as or better than that of the original problem z^*: $y^* \leq z^*$

(2) If the optimal solution $\boldsymbol{x}^*$ of the relaxation problem is an integer vector, $\boldsymbol{x}^*$ is also the optimal solution of the original problem.

General Discussions: Relaxation Problem

Integer Programming Problem (L_i and U_i are integers)

Minimize $z = f(\boldsymbol{x})$

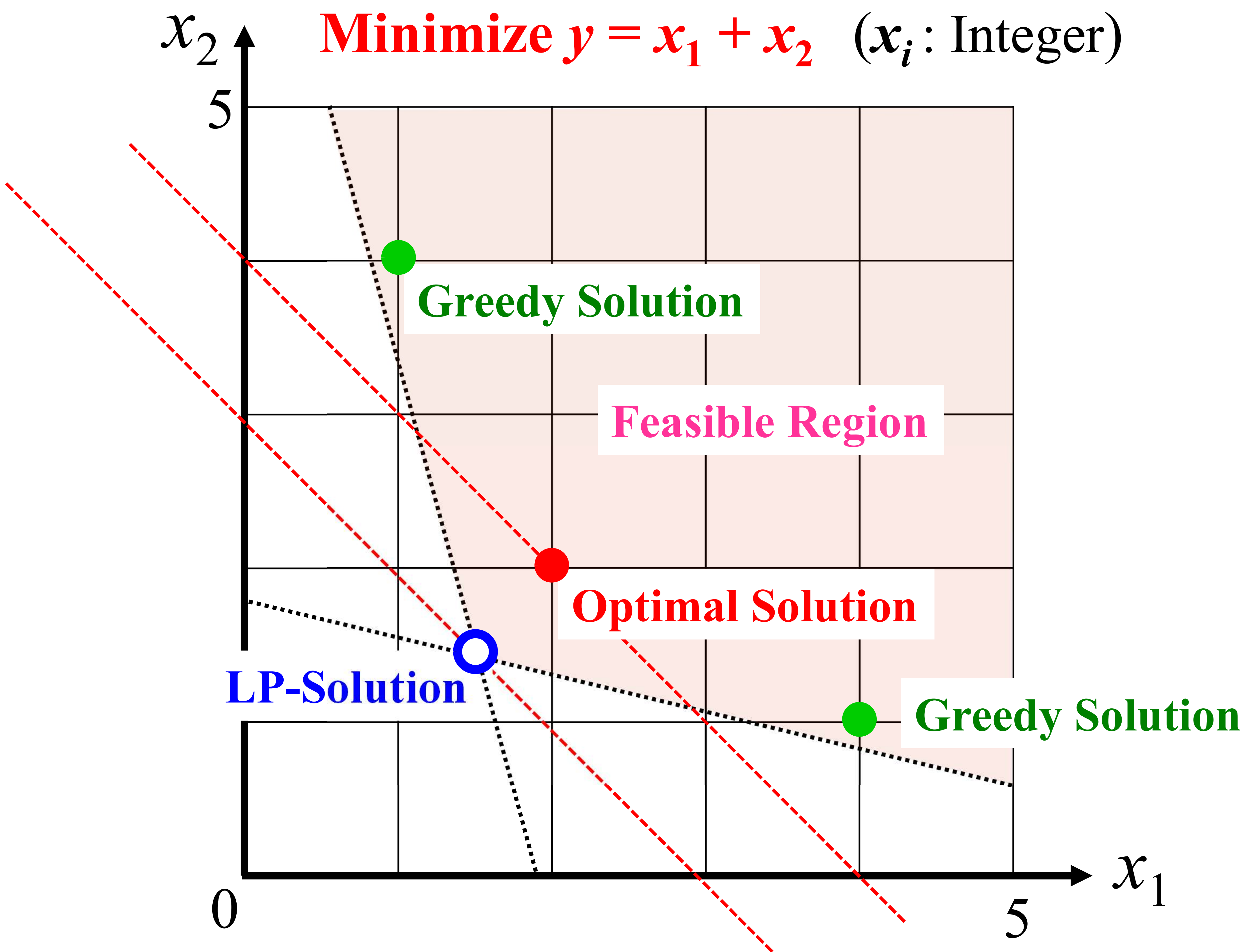
subject to $g(\boldsymbol{x}) \geq 0, \quad x_i \in \{L_i, L_i + 1, ..., U_i\}$ for $i = 1, 2, ..., n$

Relaxation Problem

Minimize $y = f(\boldsymbol{x})$

subject to $g(\boldsymbol{x}) \geq 0, \quad L_i \leq x_i \leq U_i$ for $i = 1, 2, ..., n$

The relaxation problem is used to evaluated the lower bound of the optimal value z^* (i.e., an optimistic estimation: z^* cannot be better than y^*). A greedy algorithm is used to evaluate the upper bound of the optimal value z^* (i.e., a pessimistic estimation: z^* cannot be worse than the greedy algorithm result z).


Minimize $y = x_1 + x_2$ (x_i : Integer)
x_2
5
Greedy Solution
Feasible Region
Optimal Solution
LP-Solution
Greedy Solution
0
5
x_1

Discussions: How to address the following question (i.e., how to compare the three algorithms):

Which is the best algorithm among the following three algorithms?

* **Greedy Set Cover:** $H(d)$-approximation algorithm
* **Pricing Method:** 2-approximation algorithm
* **LP-based Method:** 2-approximation algorithm

Which is the best algorithm among the following three algorithms?

* **Greedy Set Cover:** $H(d)$-approximation algorithm
* **Pricing Method:** 2-approximation algorithm
* **LP-based Method:** 2-approximation algorithm

Q1. Can we say that "Greedy set cover is inferior to the other two algorithms because its upper bound is worse than the others' upper bounds when $d \geq 4$" ?

Q2. Can we say that "Greedy set cover is superior to the other two algorithms because its upper bound is better than the others' upper bounds when $d \leq 3$" ?

Q3. How can we decide which is the best algorithm? In other words, how can we evaluate the performance of each algorithm?

Exercise 11-1:

Create an example for which the best solution is always obtained from the greedy set cover algorithm among the three methods (the greedy set cover algorithm, the pricing method, and the LP-based method) where "always" means "independent of the choice of a tie-breaking mechanism in the greedy set cover algorithm, the order of edges in the pricing method, and the choice of a single solution from multiple optimal solutions in the LP-based method".

Exercise 11-2:

Create an example for which the best solution is always obtained from the pricing method among the three method.

Exercise 11-3:

Create an example for which the best solution is always obtained from the LP-based method among the three method.

Important: In the LP-based method, you need to examine all optimal solutions if the LP problem has multiple optimal solutions in order to handle "**always**" in these three exercises.

Exercise 11-2:

Create an example for which the best solution is always obtained from the pricing method among the three method.

Please create a single connected graph instead of a combination of disconnected graphs

[Bad Example]