Topic 4: Vertex Cover Problem

Input: Graph G: $G = (V, E)$

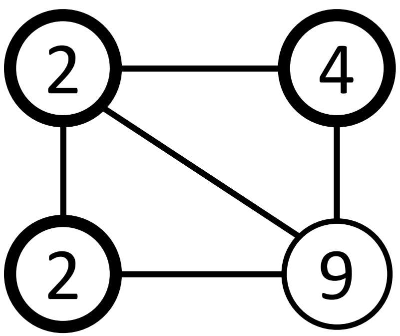
Weight of each vertex (node): w_i ($i \in V$)

Output: Vertex cover S with the minimum total weight

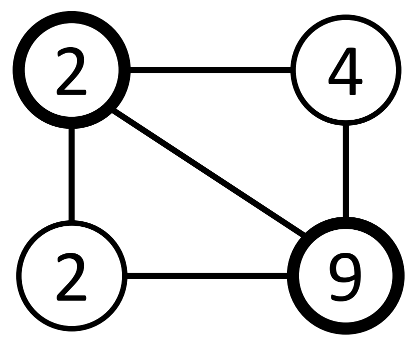
Minimize $w(S) = \sum_{i \in S} w_i$

where S ($S \subset V$) is a vertex cover (i.e., each edge in E has at least one end in S).

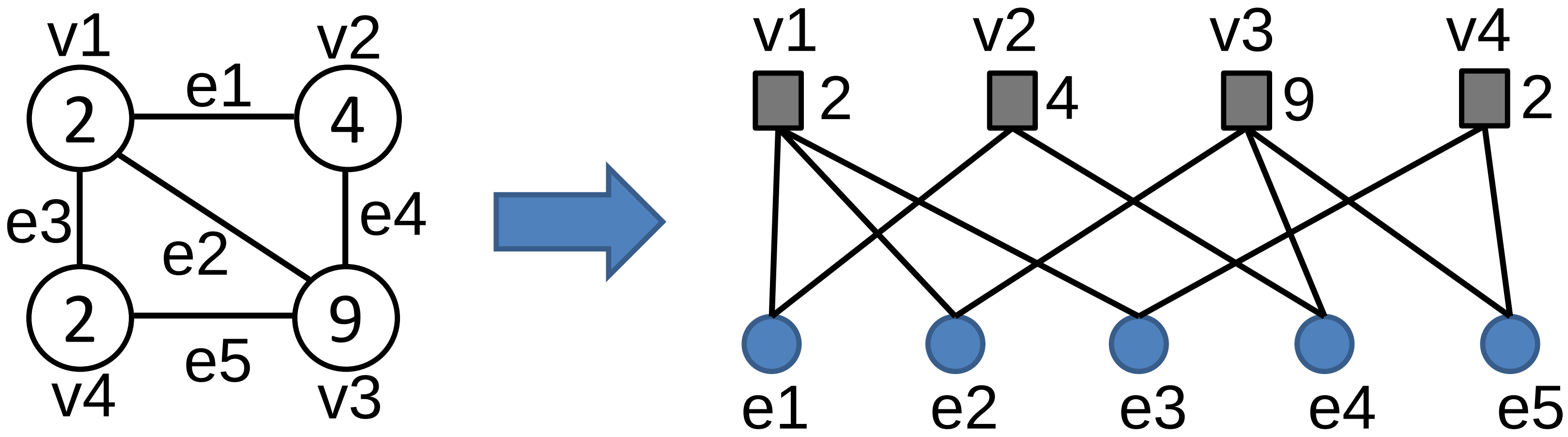
***w*(*S*) = 2 + 2 + 4**



***w*(*S*) = 2 + 9**



Vertex Cover Problem ➔ Set Cover Problem



We can use the greedy set cover algorithm for the vertex cover problem, which is an $H(d)$-approximation algorithm where d is the maximum degree of the graph (i.e., the maximum number of edges from each vertex (node)).

<u>Discussions</u>:

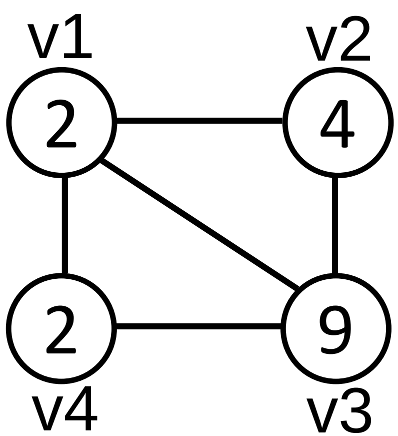
Let us assume that a set cover problem $U = \{s_1, s_2, ..., s_n\}$ with $S_1, S_2, ..., S_m$ is created from a vertex cover problem. Explain the characteristics of the created problem.

Pricing Method (Idea)

Edge e=(i, j) must be covered by vertex (node) v_i or v_j. Let p_e be the price that the edge e is willing to pay for being covered. The sum of prices over all edges incident to vertex (node) v_i should be equal to or less than w_i (since they do not have to pay more than the total cost w_i and they can use other nodes).

For each vertex v_i: $\sum_{e=(i,j)} p_e \le w_i$ $\qquad p_{(i,j)} = p_{(j,i)}$

If $\sum_{e=(i,j)} p_e = w_i$, v_i **is tight.**



$$v_1 : p_{(1,2)} + p_{(1,3)} + p_{(1,4)} \le w_1 = 2$$

$$v_2 : p_{(2,1)} + p_{(2,3)} \le w_2 = 4$$

$$v_3 : p_{(3,1)} + p_{(3,2)} + p_{(3,4)} \le w_3 = 9$$

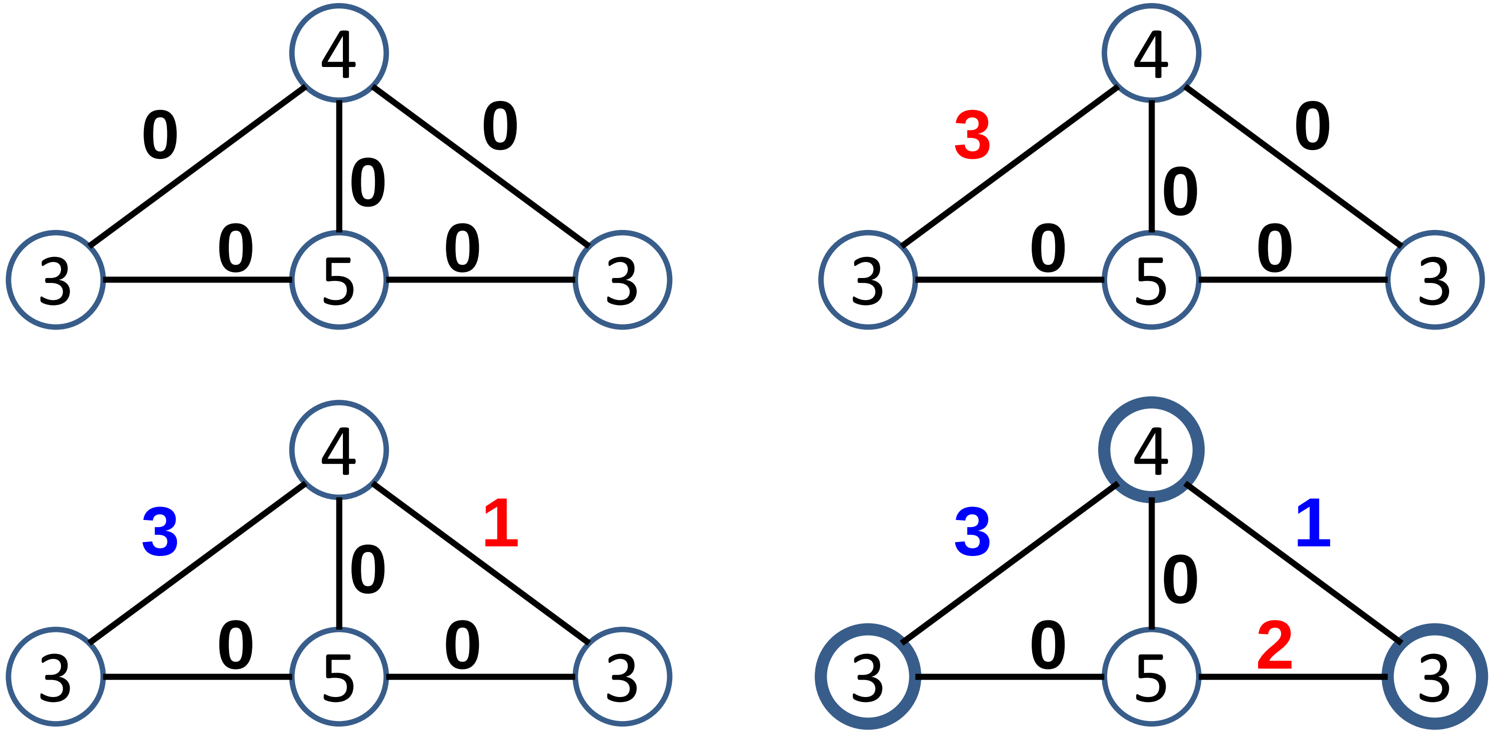
$$v_4 : p_{(4,1)} + p_{(4,3)} \le w_4 = 2$$

Pricing Method (Algorithm)

Initialization of p_e : $p_e = 0$ for each edge $e = (i, j)$.

Increase p_e : If neither vertex v_i nor v_j is tight, increase $p_{(i,j)}$ as much as possible under the condition: $\sum_{e=(i,j)} p_e \le w_i$

Selection of a vertex cover S: Select all tight vertexes.



Pricing Method (Algorithm)

We say a node i is *tight* (or "paid for") if $\sum_{e=(i,j)} p_e = w_i$.

procedure $\textsc{Vertex-Cover-Approx}(G, w)$
  Set $p_e = 0$ for all $e \in E$
  while $\exists$ edge $e = (i, j)$ such that neither i nor j is tight **do**
    Select e
    Increase p_e without violating fairness
  end while
  Let S = set of all tight nodes
  Return S.
end procedure

Fairness condition: $\sum_{e=(i,j)} p_e \leq w_i$

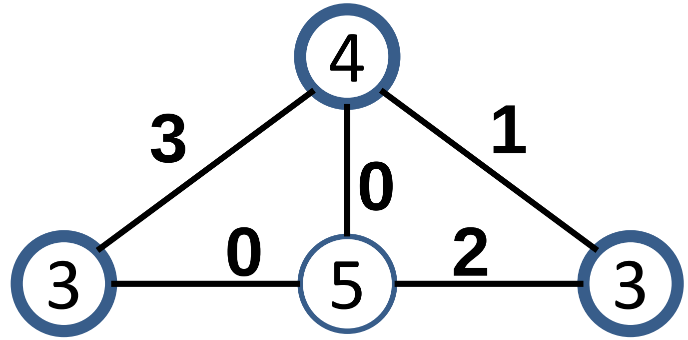
```
Weighted-Vertex-Cover-Approx(G, w) {
   foreach e in E
      pₑ = 0

   while (∃ edge i-j such that neither i nor j are tight)
      select such an edge e
      increase pₑ as much as possible until i or j tight
   }

   S ← set of all tight nodes
   return S
}
```

$$\sum_{e=(i,j)} p_e = w_i$$

Exercise 7-1:

Examine the dependency of the result S on the order of edges in which edges are selected to increase p_e. That is, create an example of the vertex cover problem where different results are obtained depending on the order of edges.

Pricing Method (Analysis)



Price Assignment: $\sum_{e=(i,j)} p_e \leq w_i$

For any vertex cover *S*: $\sum_{e \in E} p_e \leq \sum_{i \in S} \sum_{e=(i,j)} p_e \leq \sum_{i \in S} w_i = w(S)$

For the vertex cover *S* by the algorithm: $w(S) \leq 2w(S^*)$

Since all vertexes v_i in S are tight,

$$\sum_{e=(i,j)} p_e = w_i \Rightarrow w(S) = \sum_{i \in S} w_i = \sum_{i \in S} \sum_{e=(i,j)} p_e \leq 2 \sum_{e \in E} p_e$$

(An edge $e = (i, j)$ can be included at most twice.)

Since $\sum_{e \in E} p_e \leq w(S)$ holds for any vertex cover S including S^*,

$$w(S) \leq 2 \sum_{e \in E} p_e \leq 2w(S^*)$$

<u>Exercise 7-2</u>:

Create an example of the vertex cover problem where a good solution is not obtained by the pricing method (i.e., the obtained solution $w(S)$ is close to $2w(S^*)$.

<u>Exercise 7-3</u>:

Create an example of the vertex cover problem where better results are always obtained (independent of the order) by the greedy set cover algorithm than the pricing method.

<u>Exercise 7-4</u>:

Create an example of the vertex cover problem where better results are always obtained (independent of the order) by the pricing method than the greedy set cover algorithm.