Set Cover Problem

We have n jobs to be completed: $s_1, s_2, \ldots, s_n$

We need to buy some machines to complete all jobs.

There are m types of machines.

The price of each machine j is w_j: $w_1, w_2, \ldots, w_m$

Each machine j can handle a subset S_j of the n jobs:

$$S_j \subset \{s_1, s_2, \ldots, s_n\}, j = 1, 2, \ldots, m$$

Problem: Choice of machines to minimize the cost.

Example 1 (3 jobs and 4 machines)

Machine 1: 200\$, $S_1 = \{s_1, s_2\}$

Machine 2: 250\$, $S_2 = \{s_2, s_3\}$

Machine 3: 170\$, $S_3 = \{s_3, s_1\}$

Machine 4: 150\$, $S_4 = \{s_1\}$

Your choice: Machine ? and Machine ? (Total Cost: ___ \$)

Solve the following problem

Example 2 (5 jobs and 6 machines)

Machine 1: 50$, $S_1 = \{s_1\}$

Machine 2: 100$, $S_2 = \{s_3\}$

Machine 3: 150$, $S_3 = \{s_5\}$

Machine 4: 200$, $S_4 = \{s_1, s_2, s_3\}$

Machine 5: 250$, $S_5 = \{s_1, s_2, s_4\}$

Machine 6: 300$, $S_6 = \{s_2, s_4, s_5\}$

Your choice: Machines ____________ (Total Cost: ___ $)

Example 3 (10 jobs and 10 machines)

Machine 01: 350$, $S_1 = \{s_1, s_2, s_3, s_4, s_5\}$

Machine 02: 220$, $S_2 = \{s_1, s_2, s_3\}$

Machine 03: 190$, $S_3 = \{s_7\}$

Machine 04: 400$, $S_4 = \{s_3, s_4, s_5, s_6, s_7\}$

Machine 05: 240$, $S_5 = \{s_8\}$

Machine 06: 280$, $S_6 = \{s_9\}$

Machine 07: 420$, $S_7 = \{s_4, s_5, s_6, s_7, s_8\}$

Machine 08: 500$, $S_8 = \{s_6, s_{10}\}$

Machine 09: 770$, $S_9 = \{s_7, s_8, s_{10}\}$

Machine 10: 880$, $S_{10} = \{s_8, s_9, s_{10}\}$

Your choice: Machines _______________ (Total Cost: ___ $)

Formulation: Set Cover Problem

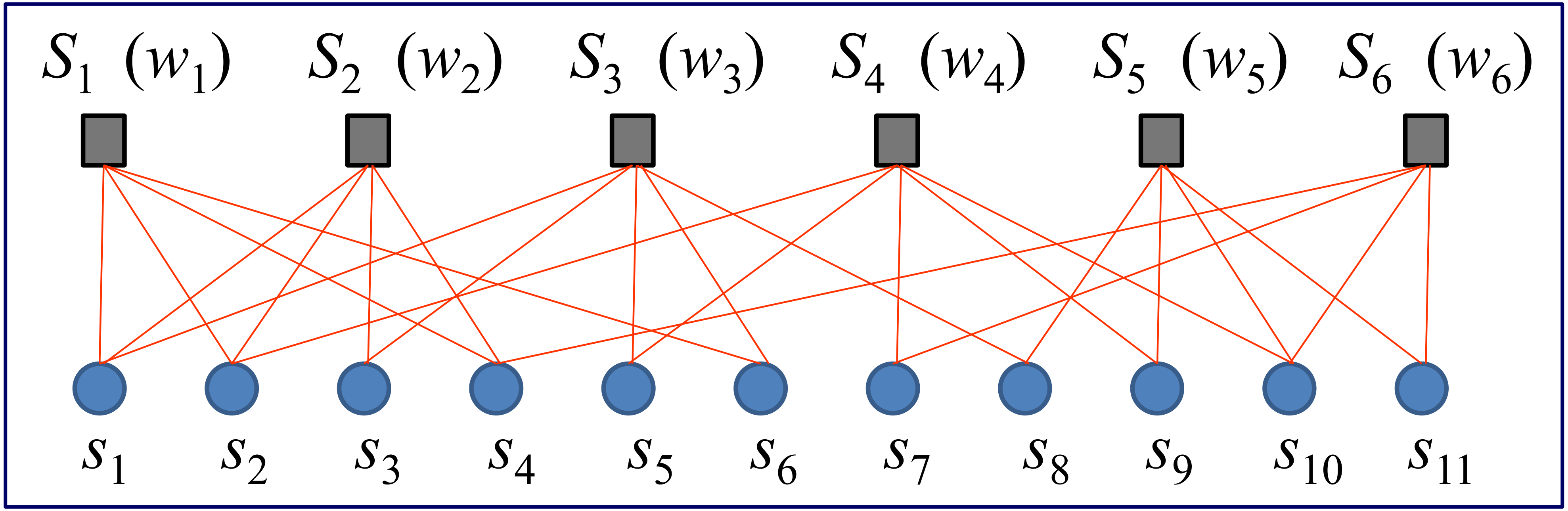
Input: n elements: $U = \{s_1, s_2, \ldots, s_n\}$

m subsets of U : $S_1, S_2, \ldots, S_m$ $(S_i \subset U)$

Weight (cost) of each subset: w_i $(i = 1, 2, \ldots, m)$

Output: Cover C (Selection from m subsets): $\bigcup_{S_i \in C} S_i = U$

Objective: Minimize the total weight: $\sum_{S_i \in C} w_i$



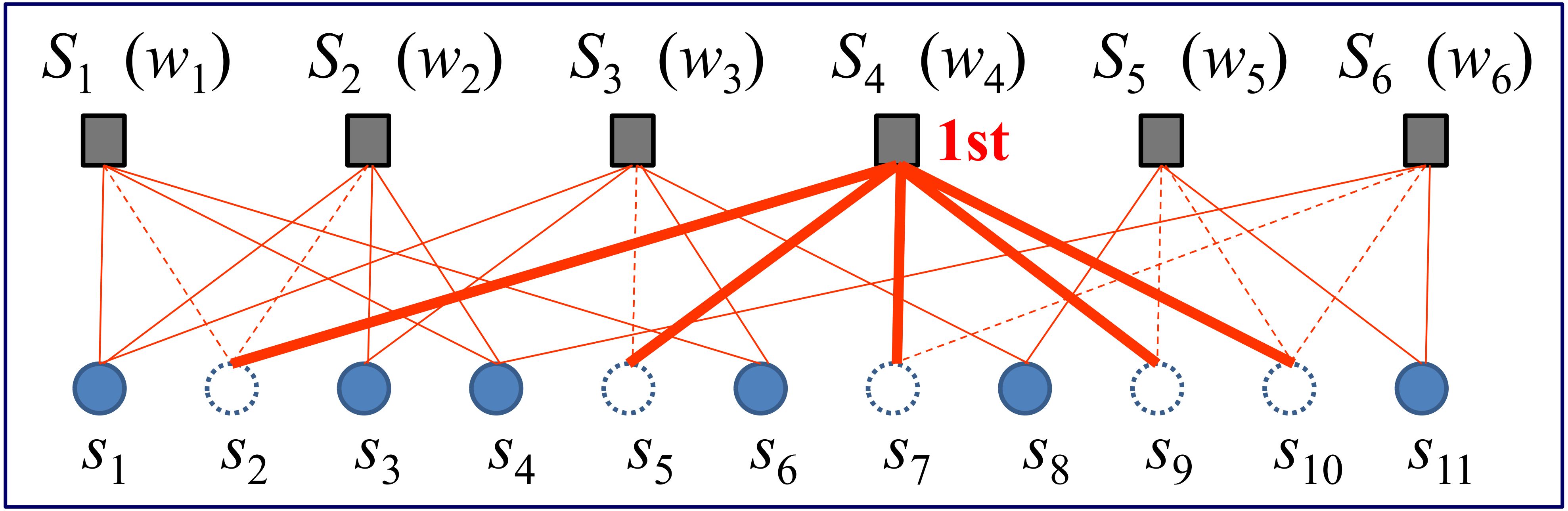
$$\text{Minimize } w(C) = \sum_{S_i \in C} w_i \quad \text{subject to} \quad \bigcup_{S_i \in C} S_i = U$$

Good subset: Small weight with many elements $\dfrac{w_i}{|S_i|}$

After some elements are covered
(*R*: remaining uncovered elements): $\dfrac{w_i}{|S_i \cap R|}$

Greedy Set Cover Algorithm:

Select the best subset with the best evaluation one by one.



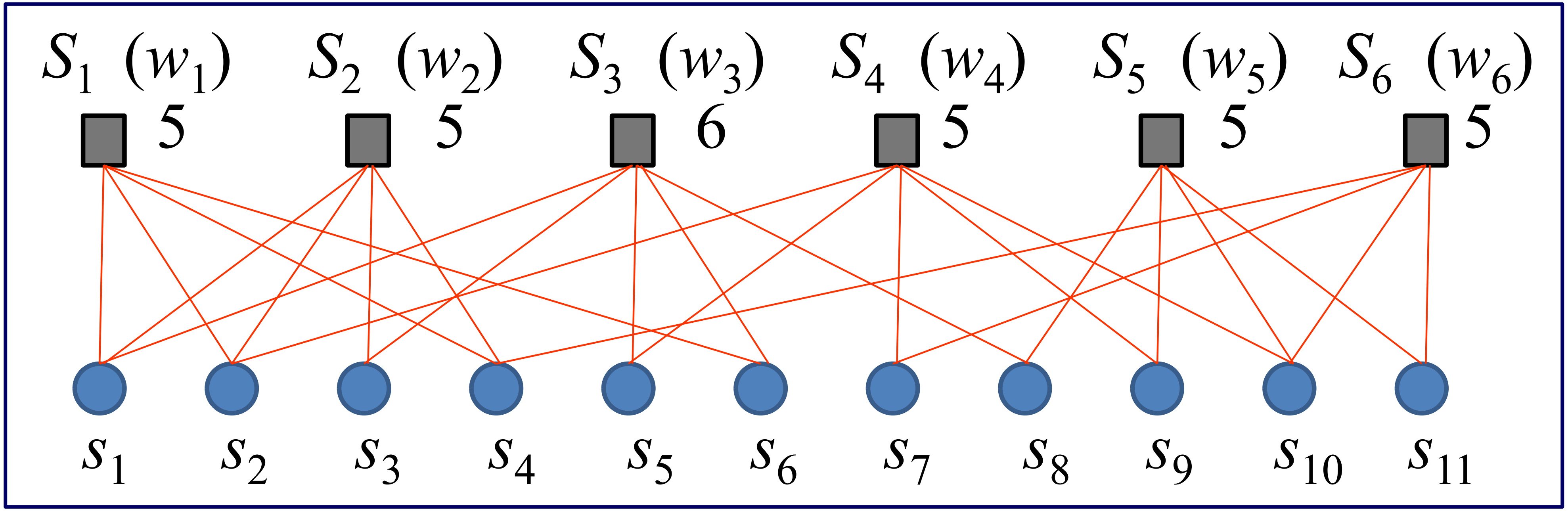
$$\text{Minimize } w(C) = \sum_{S_i \in C} w_i \text{ subject to } \bigcup_{S_i \in C} S_i = U$$

Good subset: Small weight with many elements $\dfrac{w_i}{|S_i|}$

After some elements are covered (*R*: remaining uncovered elements): $\dfrac{w_i}{|S_i \cap R|}$

Greedy Set Cover Algorithm:

Select the best subset with the best evaluation one by one.



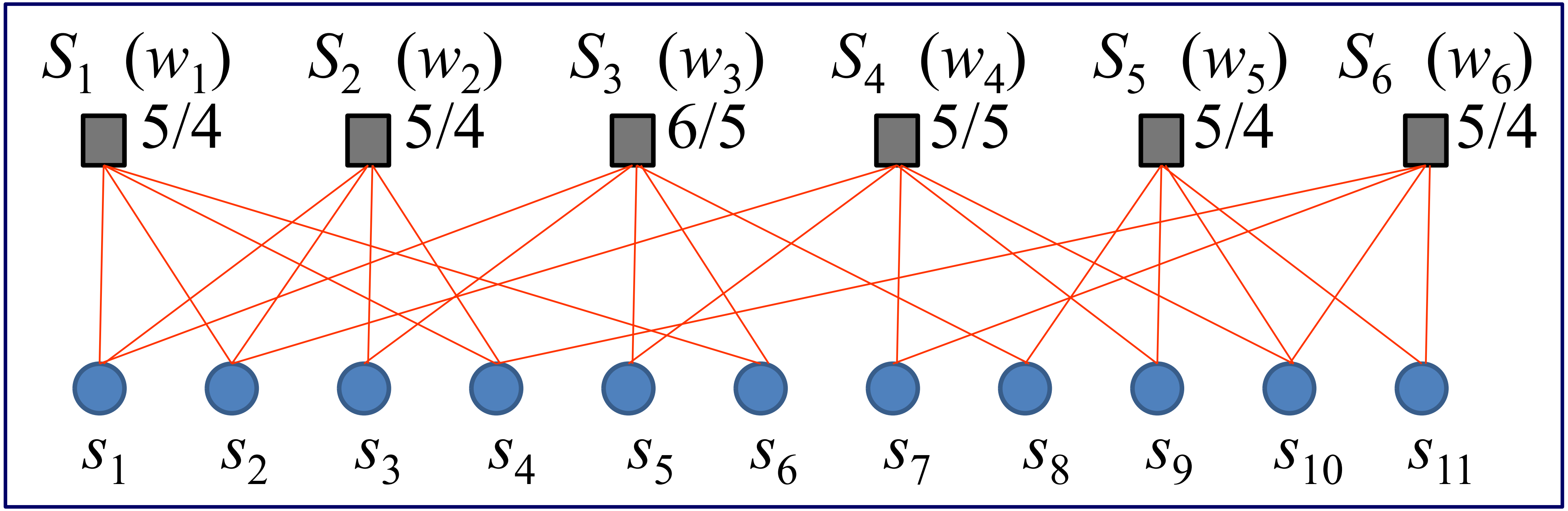
$$\text{Minimize } w(C) = \sum_{S_i \in C} w_i \text{ subject to } \bigcup_{S_i \in C} S_i = U$$

Good subset: Small weight with many elements $\dfrac{w_i}{|S_i|}$

After some elements are covered
(*R*: remaining uncovered elements): $\dfrac{w_i}{|S_i \cap R|}$

Greedy Set Cover Algorithm:

Select the best subset with the best evaluation one by one.



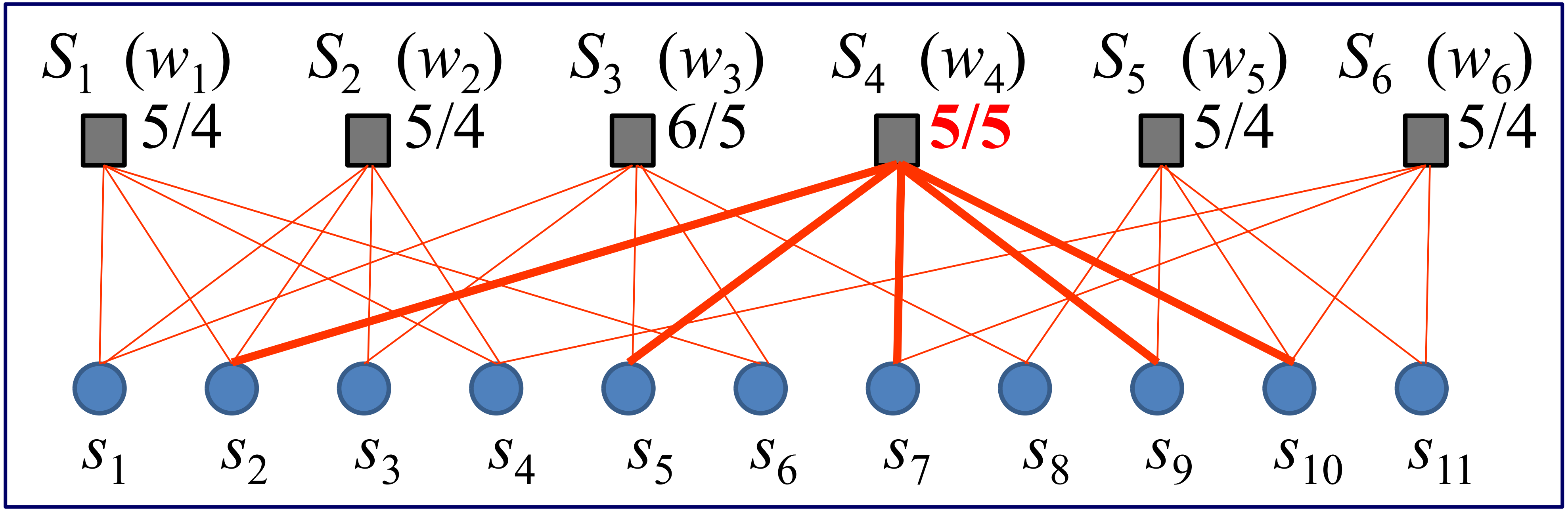
$$\text{Minimize } w(C) = \sum_{S_i \in C} w_i \text{ subject to } \bigcup_{S_i \in C} S_i = U$$

Good subset: Small weight with many elements $\dfrac{w_i}{|S_i|}$

After some elements are covered (*R*: remaining uncovered elements): $\dfrac{w_i}{|S_i \cap R|}$

Greedy Set Cover Algorithm:

Select the best subset with the best evaluation one by one.



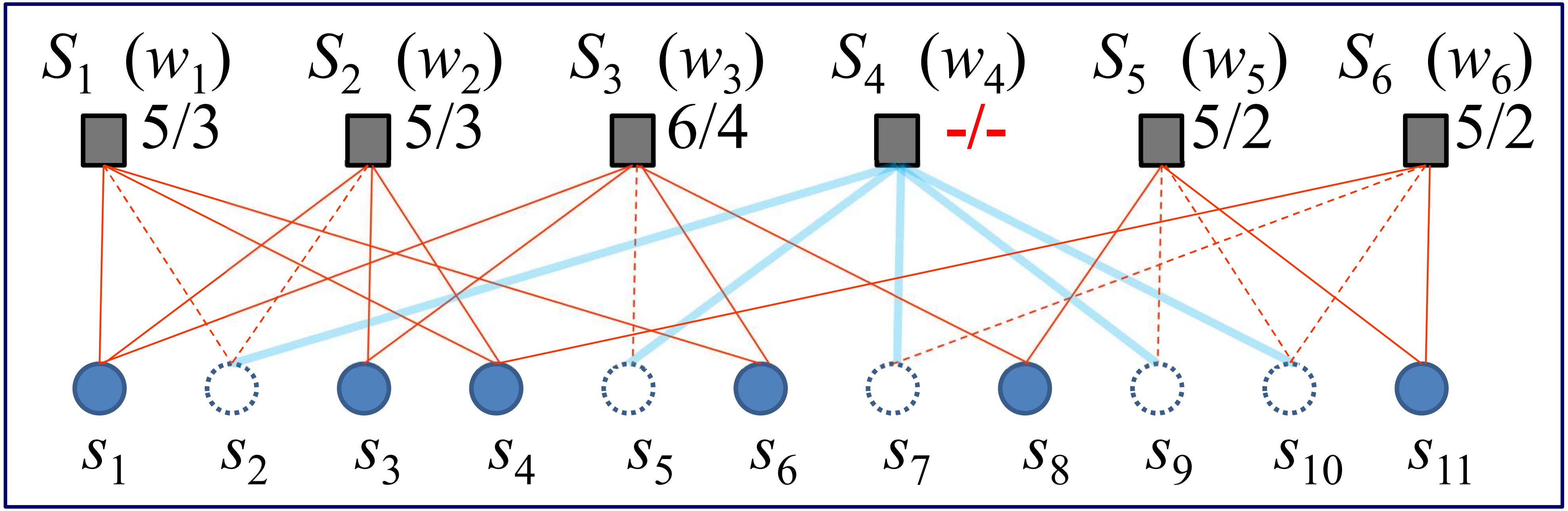
$$\text{Minimize } w(C) = \sum_{S_i \in C} w_i \text{ subject to } \bigcup_{S_i \in C} S_i = U$$

Good subset: Small weight with many elements $\dfrac{w_i}{|S_i|}$

After some elements are covered
(*R*: remaining uncovered elements): $\dfrac{w_i}{|S_i \cap R|}$

Greedy Set Cover Algorithm:

Select the best subset with the best evaluation one by one.



Greedy Set Cover Algorithm

Select the best subset with the best evaluation.

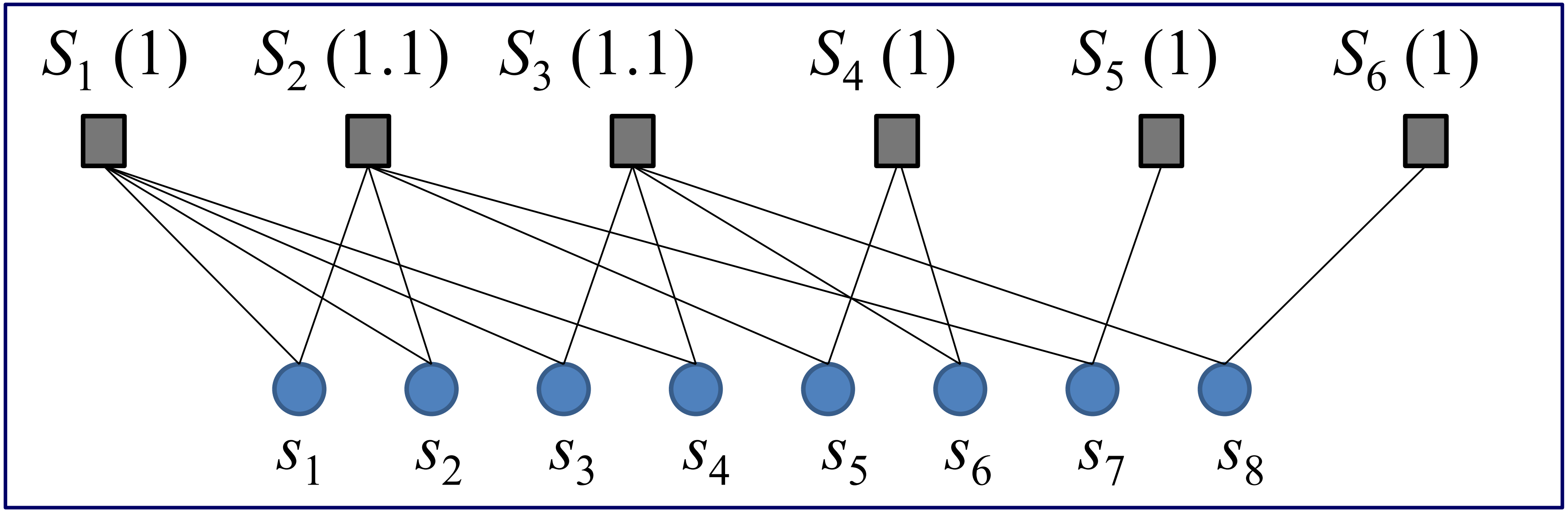
procedure GREEDY-SET-COVER
  Start with $R = U$ and no sets selected
  while $R \neq \emptyset$ **do**
    Select set S_i that minimizes $\frac{w_i}{|S_i \cap R|}$
    Delete set S_i from R
  end while
  Return the selected sets
end procedure

Exercise 8-1:

Create a simple example of the set cover problem where a good solution is not obtained by the greedy algorithm. Create another example which maximizes the value of $w(C)/w(C^*)$ where C is the obtained cover and C^* is the optimal solution.

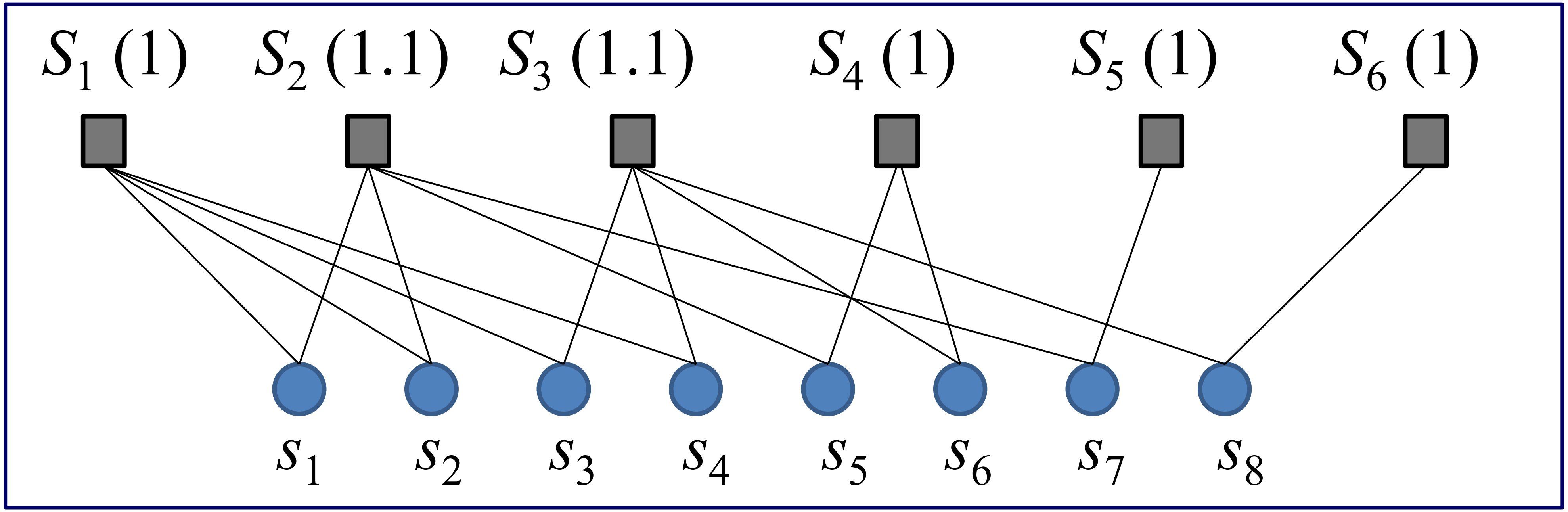
Simple Example: *n* = 8, *m* = 6

Optimal solution $C^* = \{$ ______ $\}$, $w(C^*) =$ _____ .



Simple Example: n = 8, m = 6

Greedy solution C = { _________ }, $w(C)$ = ____

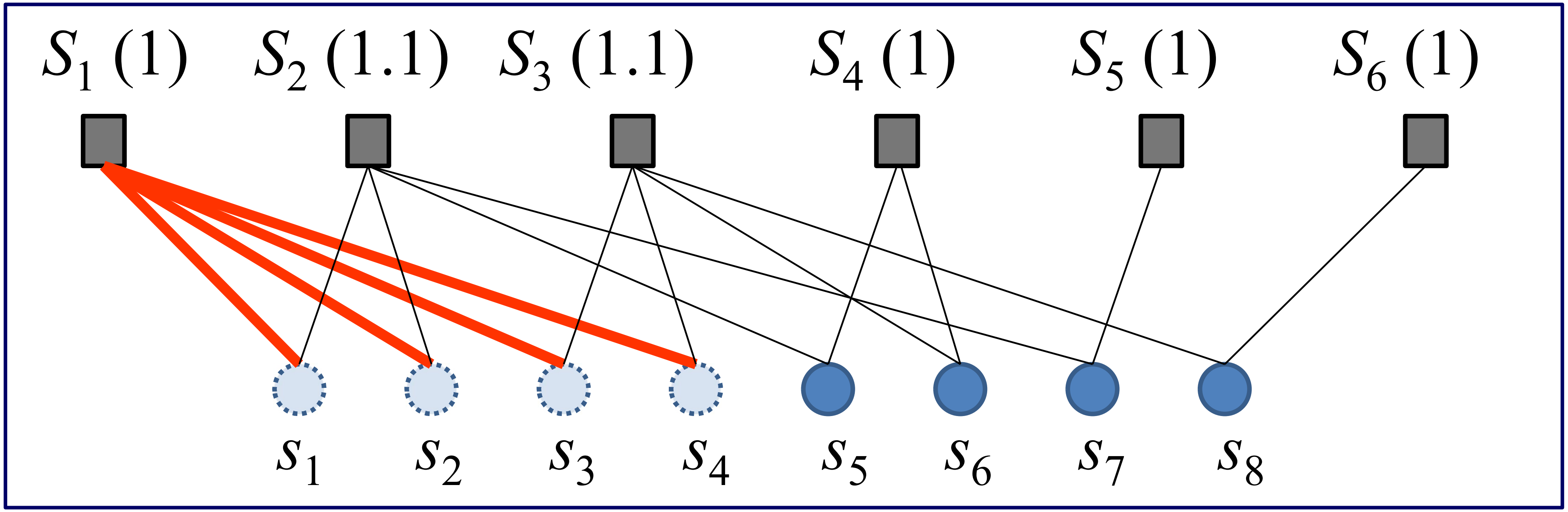


Approximation Quality of Algorithm: ?-approximation

When an element s is covered by S_i, the cost c_s paid by s is

$$c_s = \frac{w_i}{|S_i \cap R|} \quad \text{for all } s \in S_i \cap R$$

(since the total cost paid by all elements covered by S_i is w_i.)



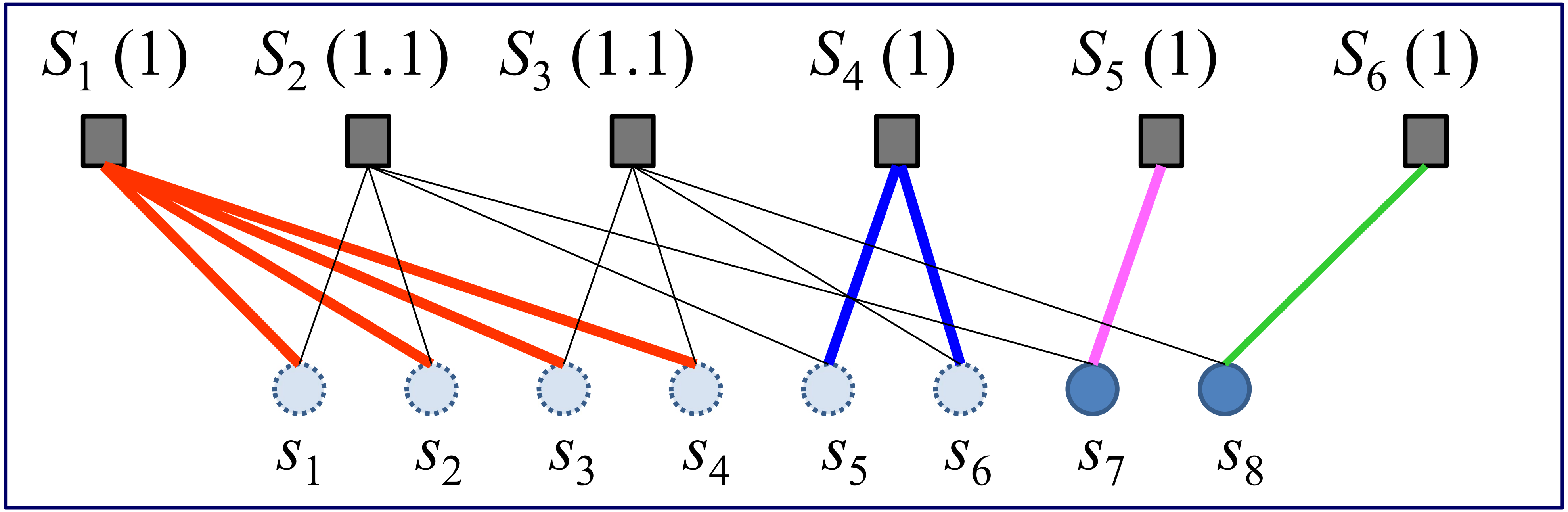
$$c_{s1} = c_{s2} = c_{s3} = c_{s4} = 1/4$$

Approximation Quality of Algorithm: ?-approximation

When an element s is covered by S_i, the cost c_s paid by s is

$$c_s = \frac{w_i}{|S_i \cap R|} \quad \text{for all } s \in S_i \cap R$$

(since the total cost paid by all elements covered by S_i is w_i.)



$$c_{s1} = c_{s2} = c_{s3} = c_{s4} = 1/4,\ \ c_{s5} = c_{s6} = 1/2,\ \ c_{s7} = 1,\ \ c_{s8} = 1$$

Approximation Quality of Algorithm: ?-approximation

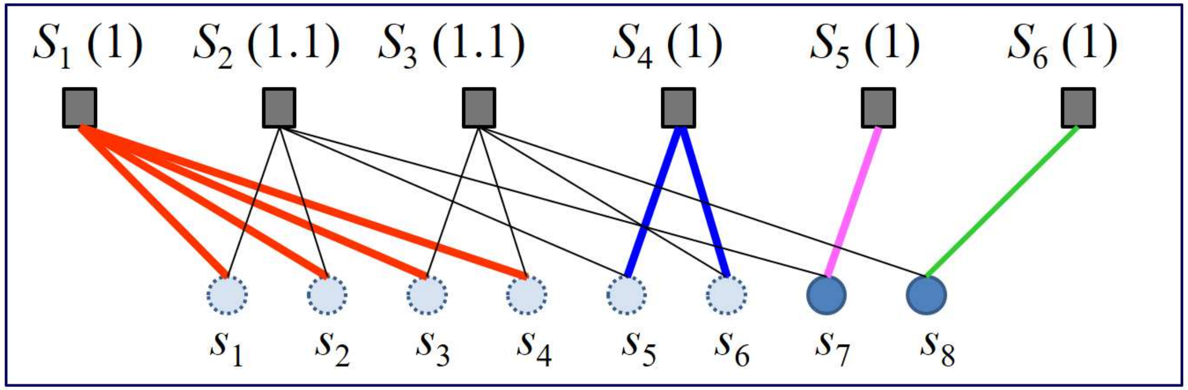
When an element s is covered by S_i, the cost c_s paid by s is

$$c_s = \frac{w_i}{|S_i \cap R|} \quad \text{for all } s \in S_i \cap R$$

(since the total cost paid by all elements covered by S_i is w_i.)

If C is the cover obtained by the greedy set cover algorithm and c_s is calculated during the execution of the algorithm,

$$\sum_{S_i \in C} w_i = \sum_{s \in U} c_s$$ (the right-hand side will be evaluated)



$$\underbrace{c_{s1} = c_{s2} = c_{s3} = c_{s4}}_{w_1} = 1/4, \quad \underbrace{c_{s5} = c_{s6}}_{w_4} = 1/2, \quad \underbrace{c_{s7}}_{w_5} = 1, \quad \underbrace{c_{s8}}_{w_6} = 1$$

Preparation

Harmonic Function: $H(n) = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$

(i) For every set S_k,

(c_s is calculated during the execution of the greedy algorithm)

Let us assume that d elements in $S_k = \{s_1, s_2, ..., s_d\}$ are covered in the order of $s_1, s_2, ..., s_d$ by the greedy algorithm. Consider the iteration when s_j is covered. Before this iteration, $\{s_j, s_{j+1}, ..., s_d\} \subset R$. Thus $\dfrac{w_k}{|S_k \cap R|} = \dfrac{w_k}{d - j + 1}$

At this iteration, the algorithm selects S_i with the minimum average cost. So, $c_{s_j} = \dfrac{w_i}{|S_i \cap R|} \leq \dfrac{w_k}{|S_k \cap R|} = \dfrac{w_k}{d - j + 1}$

Thus

$$\sum_{s \in S_k} c_s = \sum_{j=1}^{d} c_{s_j} \leq \sum_{j=1}^{d} \frac{w_k}{d - j + 1} = \frac{w_k}{d} + \frac{w_k}{d-1} + \cdots + \frac{w_k}{1} = w_k H(d)$$

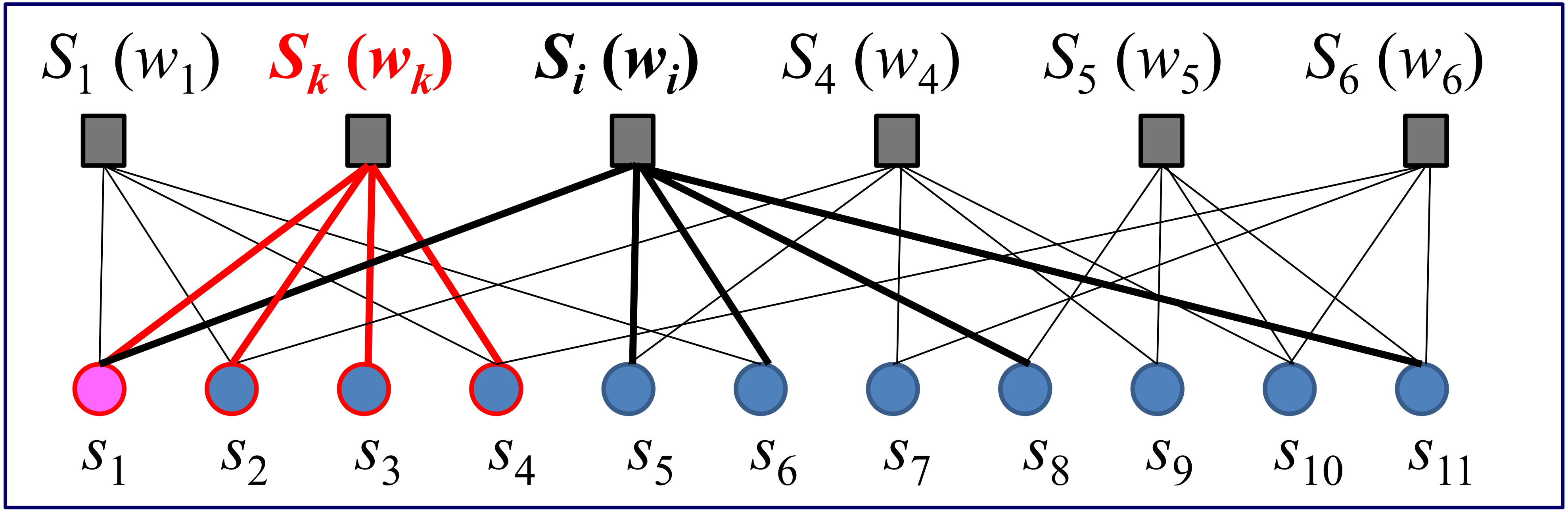
Example (Figure below)

If s_1 is covered by S_i (not S_k), the following relation holds:

$$c_1 = \frac{w_i}{|S_i \cap R|} \le \frac{w_k}{|S_k \cap R|} = \frac{w_k}{d-j+1} = \frac{w_k}{4} \quad (d=4,\ j=1)$$

If s_1 is covered by S_k, the following relation holds:

$$c_1 = \frac{w_k}{|S_k \cap R|} = \frac{w_k}{d-j+1} = \frac{w_k}{4}$$



$$S_k = \{s_1, s_2, s_3, s_4\}$$

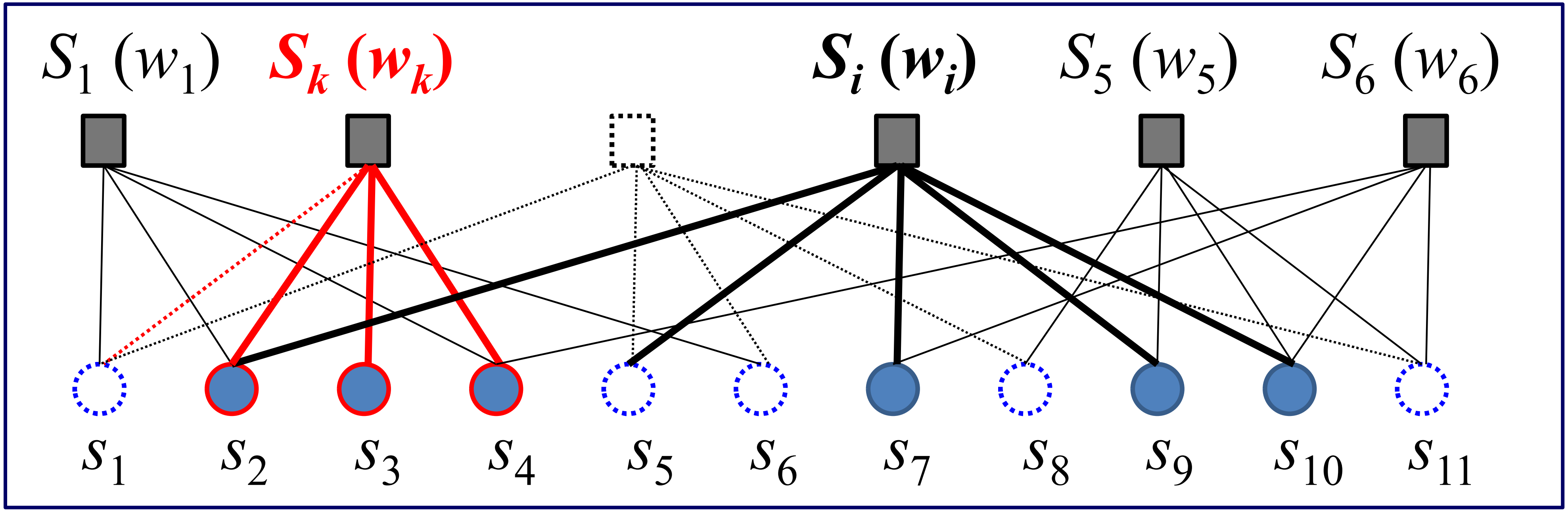
Example

If s_2 is covered by S_i (not S_k), the following relation holds:

$$c_2 = \frac{w_i}{|S_i \cap R|} \le \frac{w_k}{|S_k \cap R|} = \frac{w_k}{d-j+1} = \frac{w_k}{3} \quad (d=4,\ j=2)$$

If s_2 is covered by S_k, the following relation holds:

$$c_2 = \frac{w_k}{|S_k \cap R|} = \frac{w_k}{d-j+1} = \frac{w_k}{3}$$



$$\sum_{s \in S_k} c_s = \sum_{j=1}^{d} c_{s_j} \le \sum_{j=1}^{d} \frac{w_k}{d-j+1} = \frac{w_k}{d} + \frac{w_k}{d-1} + \cdots + \frac{w_k}{1} = w_k H(d)$$

(ii) $w \le H(\max_k |S_k|)\, w^*$

> The obtained weight w by the greedy algorithm is not worse than $H(d^*)$ times of the optimal weight w^* where $d^* = \max_k |S_k|$.

Let C^* be the optimal set cover: $w^* = \sum_{S_i \in C^*} w_i$

From (i), we have

$$\sum_{s \in S_i} c_s \le H(|S_i|) w_i \le H(d^*) w_i \Rightarrow w_i \ge \frac{1}{H(d^*)} \sum_{s \in S_i} c_s$$

Since C^* is a set cover, $\sum_{S_i \in C^*} \sum_{s \in S_i} c_s \ge \sum_{s \in U} c_s$

$$\bigcup_{S_i \in C} S_i = U$$

s can be included in multiple S_i.

(ii) $w \le H(\max_k |S_k|)\, w^*$

$\boxed{\text{The obtained weight } w \text{ by the greedy algorithm is not worse than } H(d^*) \text{ times of the optimal weight } w^* \text{ where } d^* = \max_k |S_k|.}$

Let C^* be the optimal set cover: $w^* = \sum_{S_i \in C^*} w_i$

From (i), we have

$$\sum_{s \in S_i} c_s \le H(|S_i|) w_i \le H(d^*) w_i \Rightarrow w_i \ge \frac{1}{H(d^*)} \sum_{s \in S_i} c_s$$

Since C^* is a set cover, $\sum_{S_i \in C^*} \sum_{s \in S_i} c_s \ge \sum_{s \in U} c_s$

$$w^* = \sum_{S_i \in C^*} w_i \ge \sum_{S_i \in C^*} \left[\frac{1}{H(d^*)} \sum_{s \in S_i} c_s \right] \ge \frac{1}{H(d^*)} \sum_{s \in U} c_s = \frac{1}{H(d^*)} \sum_{S_i \in C} w_i$$

$$w^* \ge \frac{w}{H(d^*)} \Rightarrow w \le H(d^*)\, w^*$$

$\boxed{H(d^*)\text{-approximation}}$

Exercise 8-2:

Create two examples (one simple example and one interesting example) of the set cover problem where all of the following three types of solutions can be obtained by the greedy set cover algorithm depending on the choice of a tie-breaking mechanism in each iteration in the algorithm: the best solution with $w(C) = w(C^*)$, the worst solution with $w(C) = H(d^*)w(C^*)$, and some other solutions with $w(C^*) < w(C) < H(d^*)w(C^*)$ where C is the obtained solution by the greedy set cover algorithm, C^* is the optimal solution, and $d^* = \max_k |S_k|$.