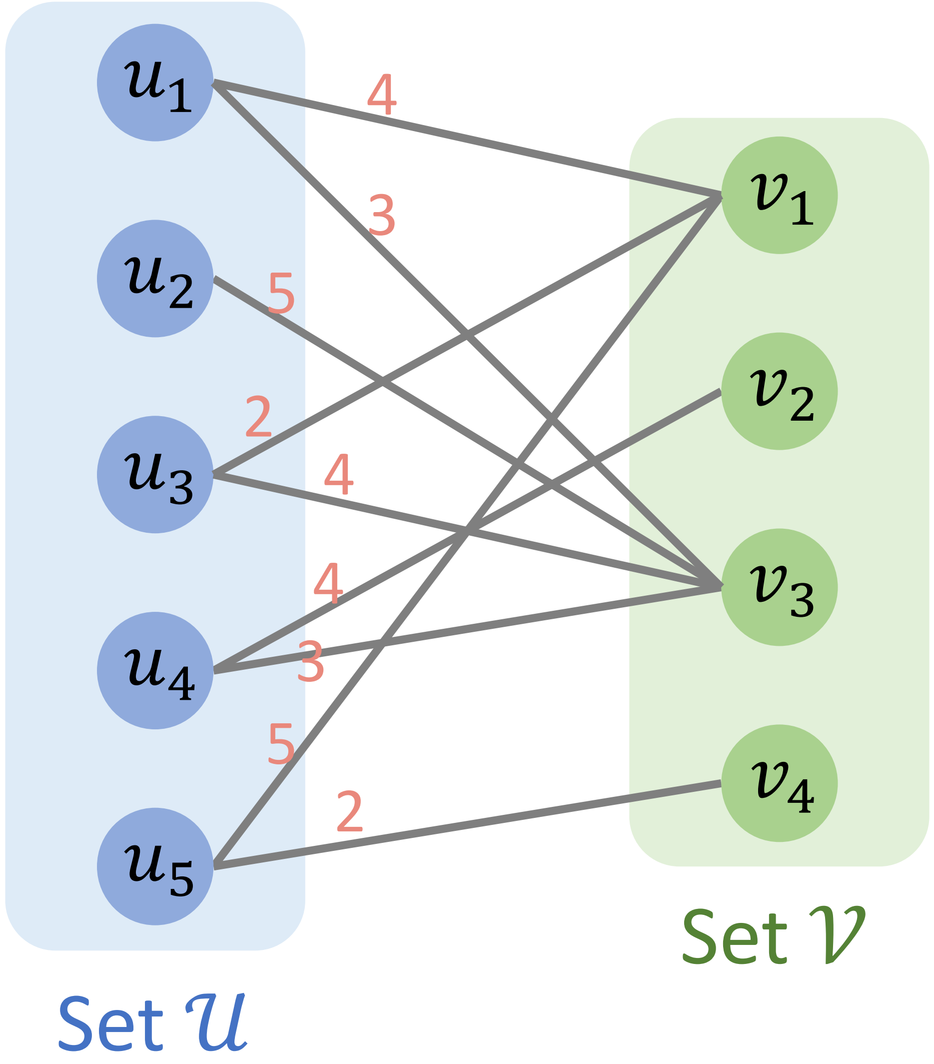
Maximum-Weight Bipartite Matching

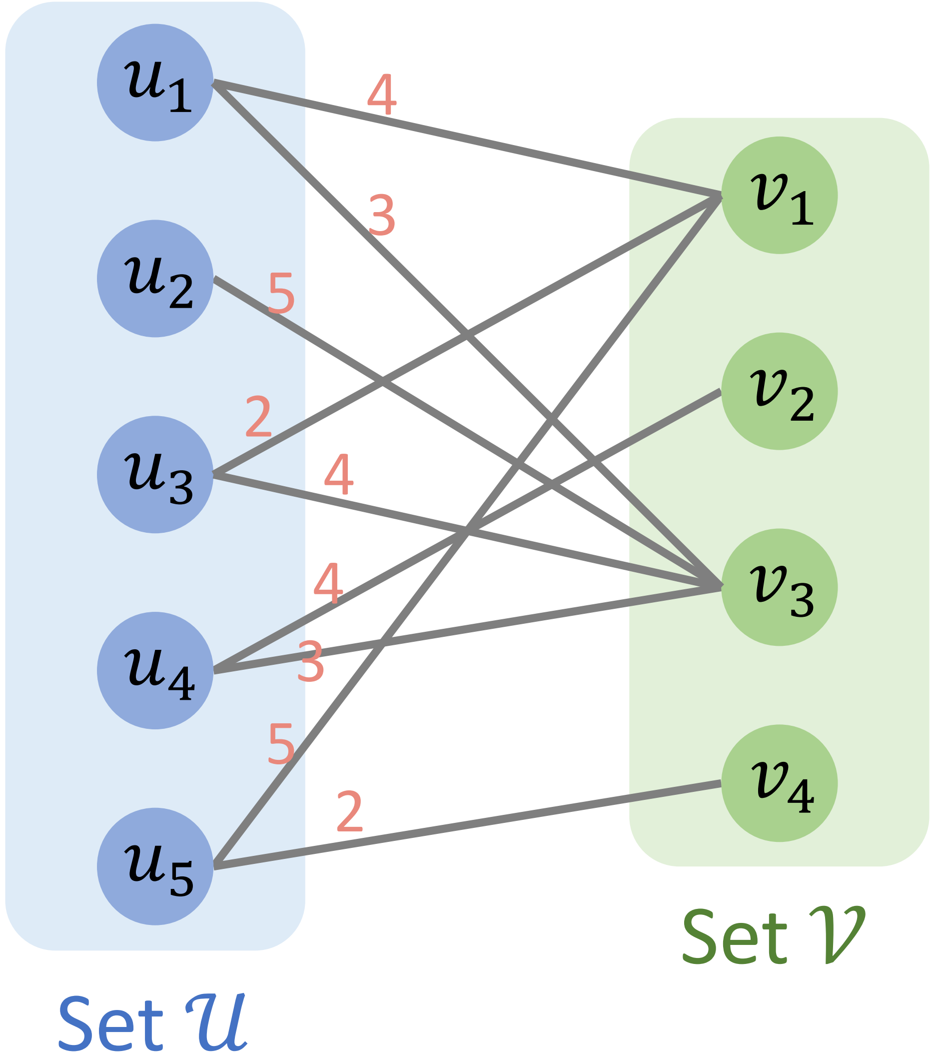
Shusen Wang

Weighted Bipartite Graph



- Bipartite graph: $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$.
- Edges have weights: $\boxed{w_{ij}.}$

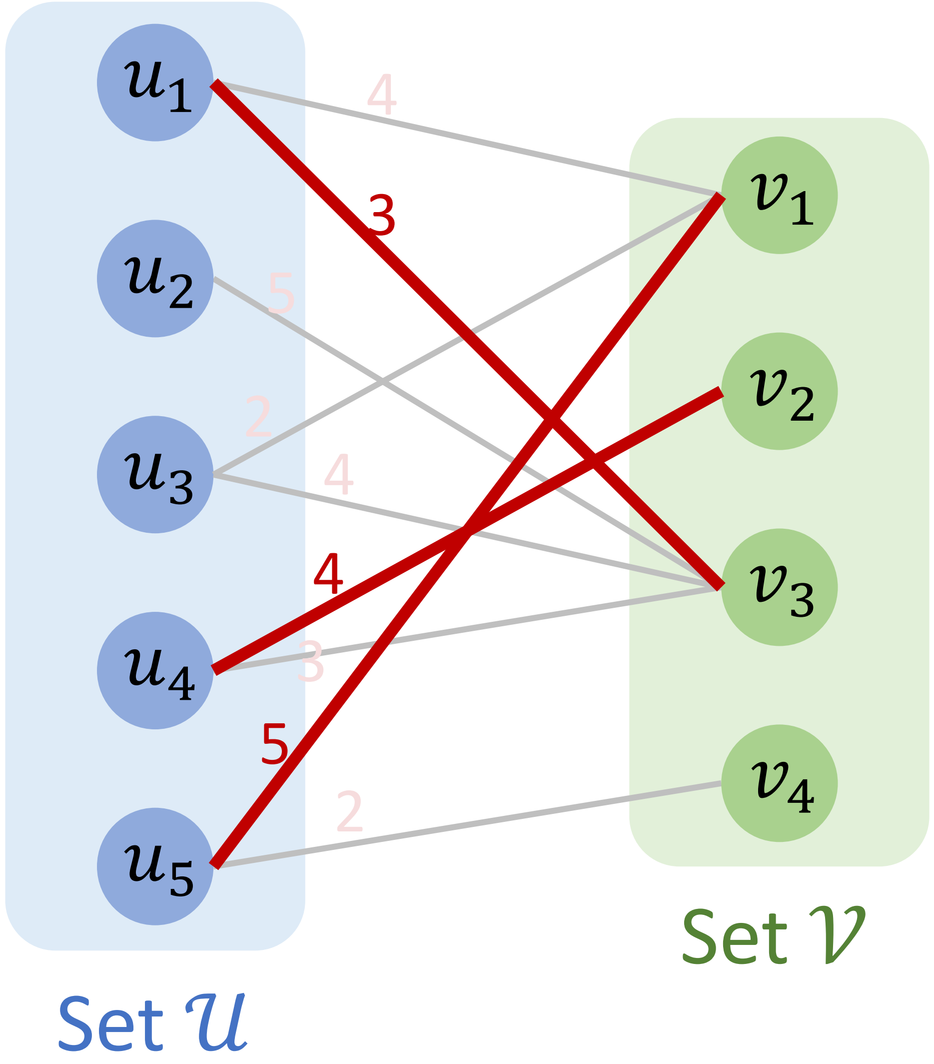
Weighted Bipartite Graph



- Bipartite graph: $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$.
- Edges have weights: w_{ij}.
- Adjacency matrix:

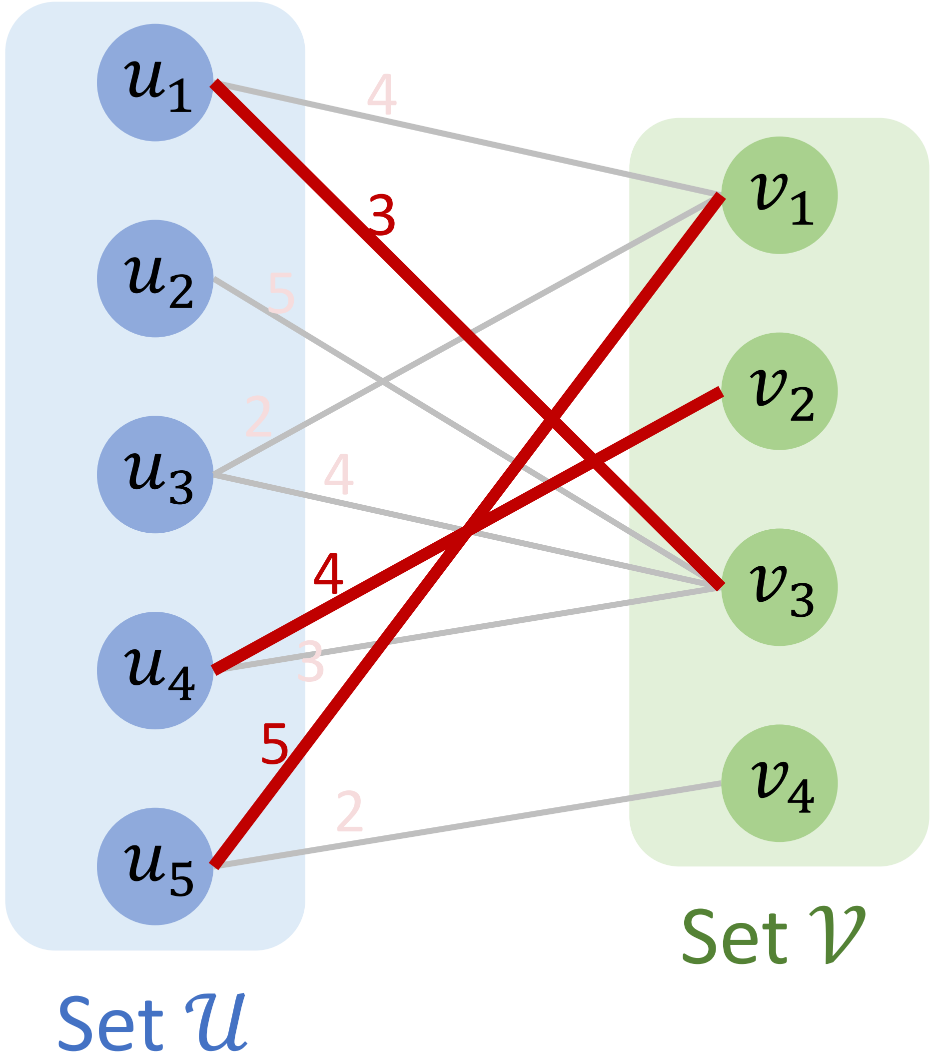
	v_1	v_2	v_3	v_4
u_1	**4**	0	**3**	0
u_2	0	0	**5**	0
u_3	**2**	0	**4**	0
u_4	0	**4**	**3**	0
u_5	**5**	0	0	**2**

Bipartite Matching in Weighted Graph



- Bipartite graph: $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$.
- Matching is a subset of edges without common vertices.
- Denote the matching by set $\mathcal{S} \subseteq \mathcal{E}$.

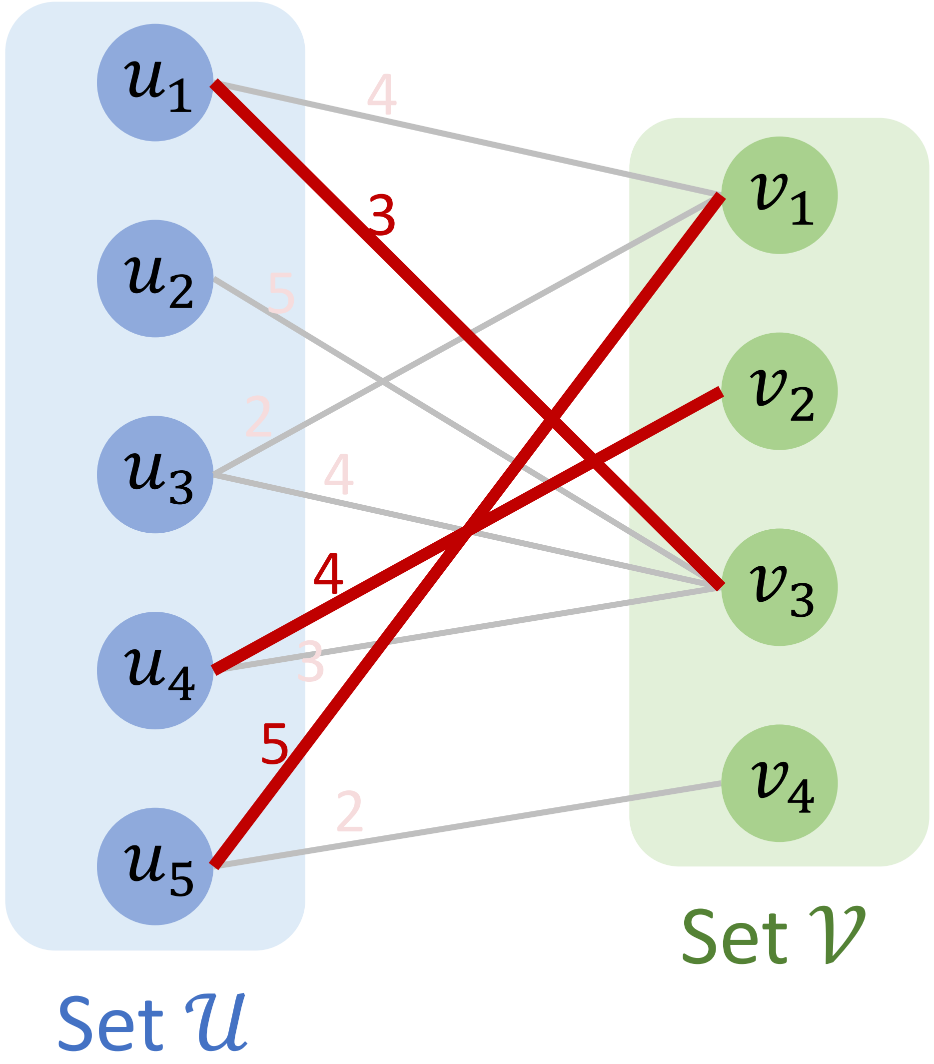
Bipartite Matching in Weighted Graph



- Sum of weights in matching $\mathcal{S}$:

$$f(\mathcal{S}) = \sum_{(u,v)\in\mathcal{S}} w_{uv}.$$

Bipartite Matching in Weighted Graph



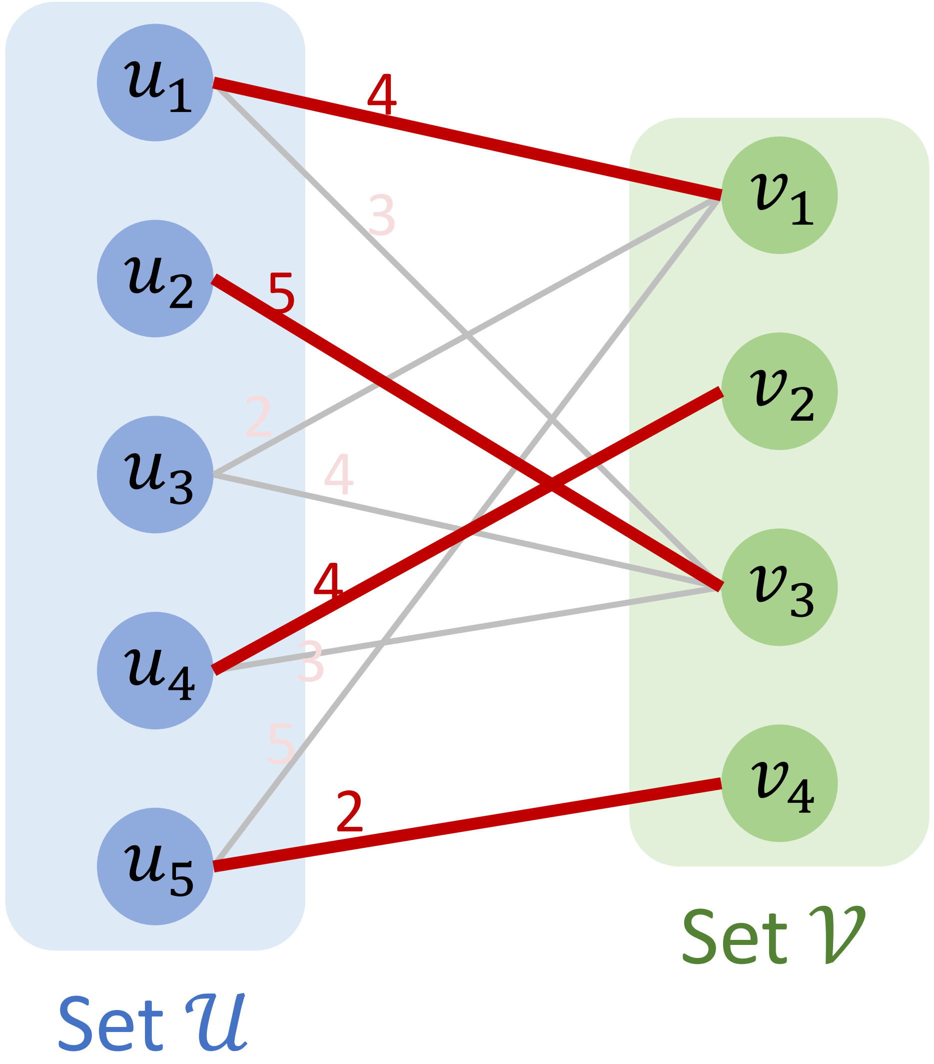
- Sum of weights in matching $\mathcal{S}$:

$$f(\mathcal{S}) = \sum_{(u,v)\in\mathcal{S}} w_{uv}.$$

- In this example,

$$f(\mathcal{S}) = 3 + 4 + 5 = 12.$$

Bipartite Matching in Weighted Graph



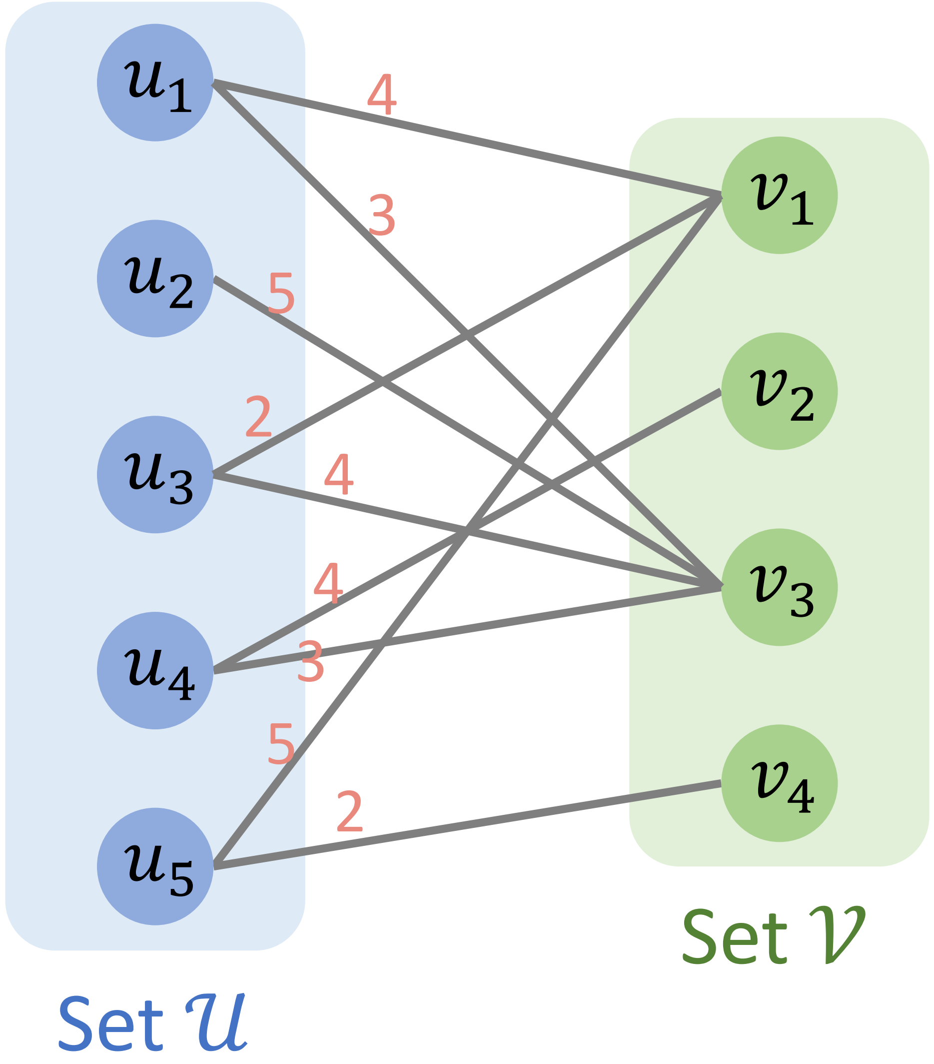
- Sum of weights in matching $\mathcal{S}$:

$$f(\mathcal{S}) = \sum_{(u,v)\in\mathcal{S}} w_{uv}.$$

- In this example,

$$f(\mathcal{S}) = 4 + 5 + 4 + 2 = 15.$$

Maximum-Weight Bipartite Matching



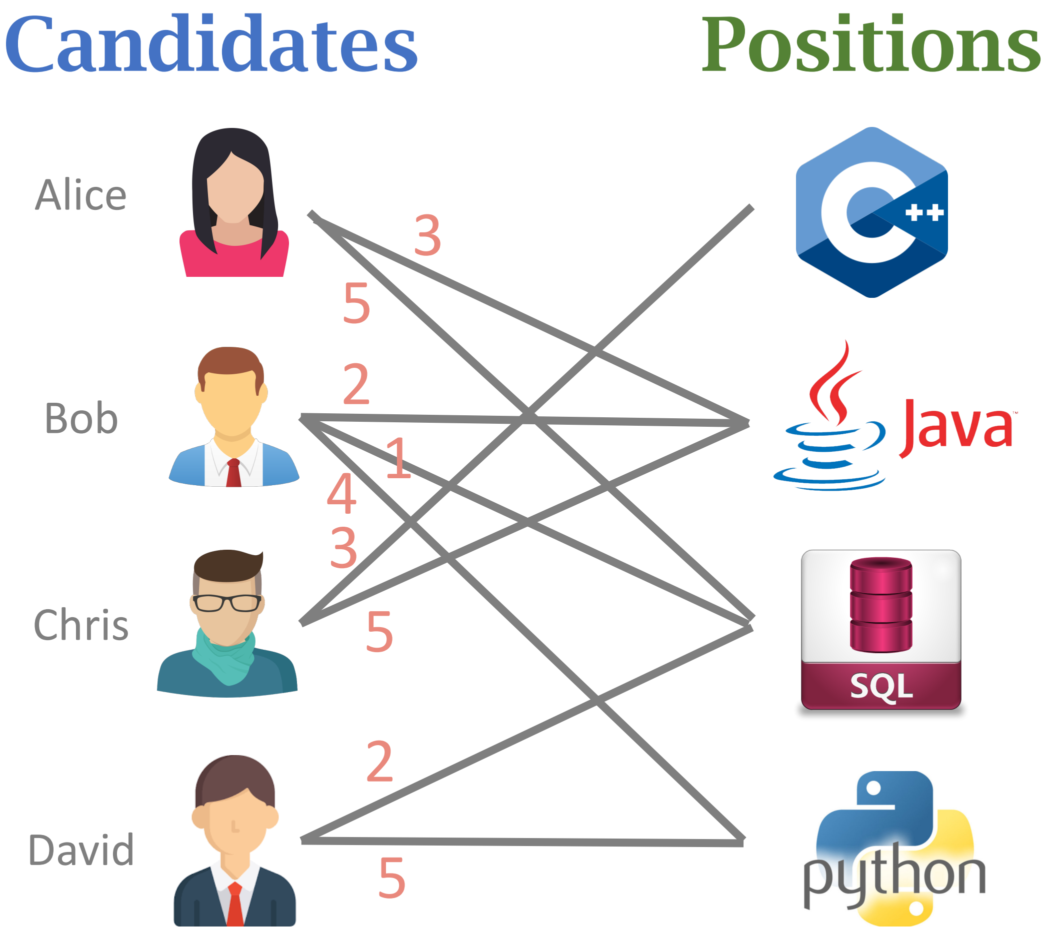
- Sum of weights in matching $\mathcal{S}$:

$$f(\mathcal{S}) = \sum_{(u,v)\in\mathcal{S}} w_{uv}.$$

- **Objective:** Finding matching $\mathcal{S}$ that has the maximum sum of weights:

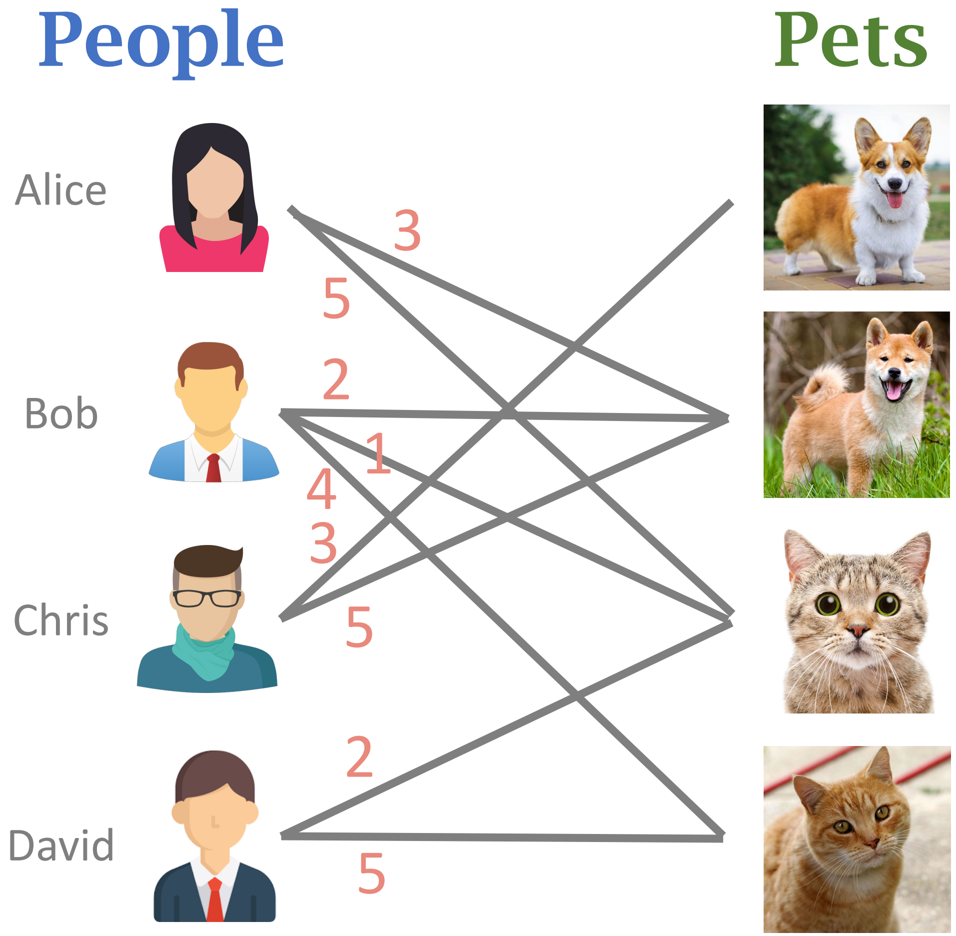
$$\boxed{\max_{\mathcal{S}} f(\mathcal{S}).}$$

Application 1: Match candidates and positions



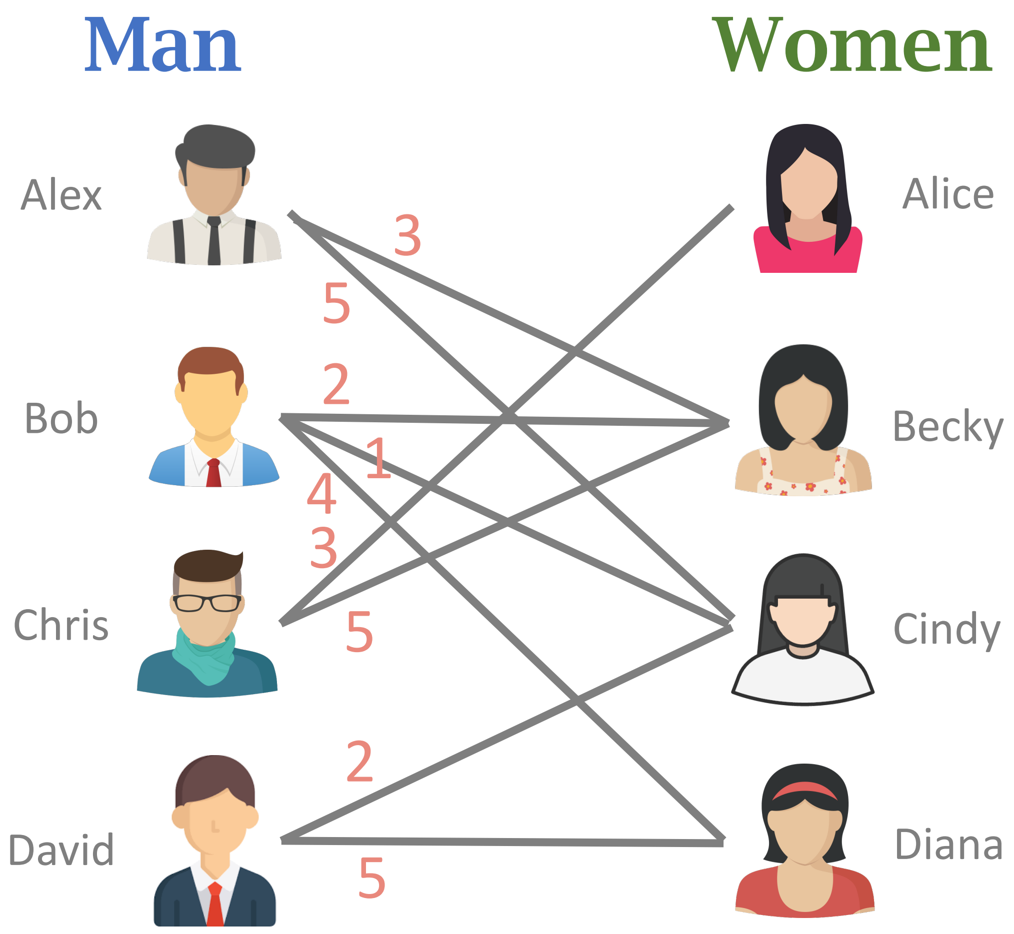
- Edge weights quantify candidates' skills.
- Maximize the weights of matching. (Match the right person with the right job position to maximize the company's interest.)

Application 2: Pet adoptions



- An edge weight quantifies how much a person loves a pet.
- Maximize the weights of matching. (Maximize people's happiness.)

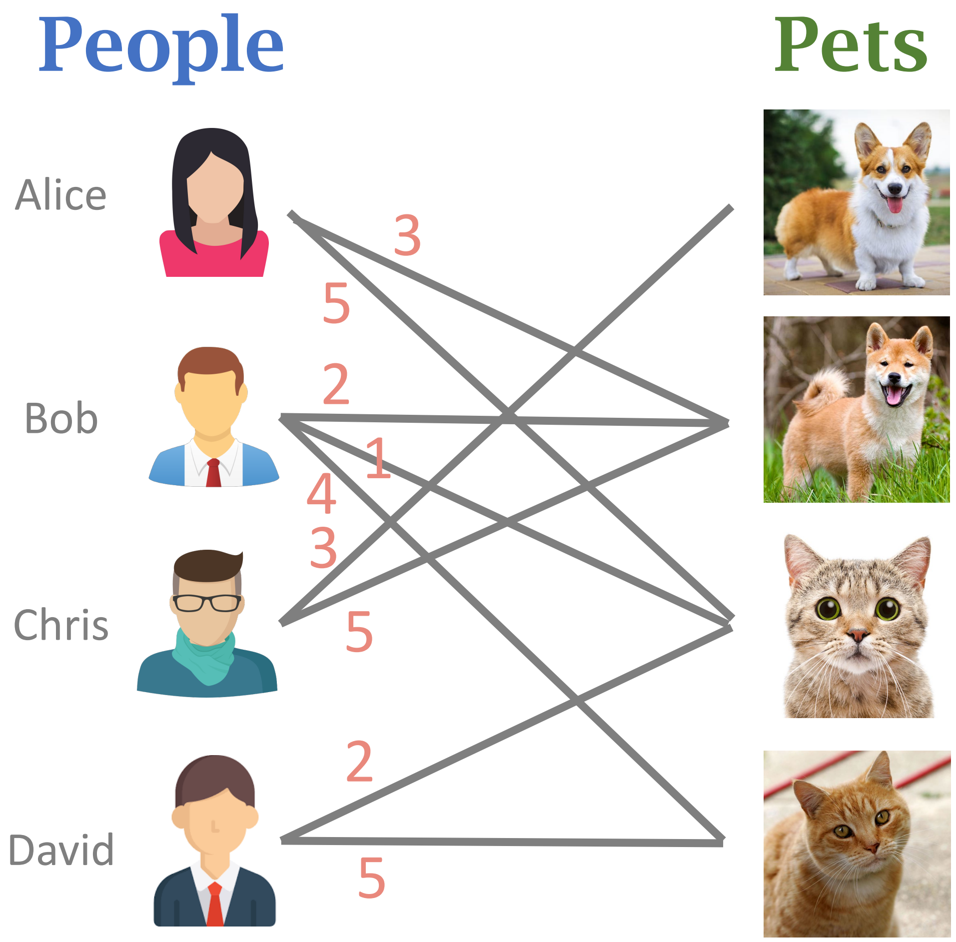
Application 3: Dating



- An edge weight quantifies how well two persons match (e.g., similar hoppy).
- Maximize the weights of matching. (Maximize the change of success.)

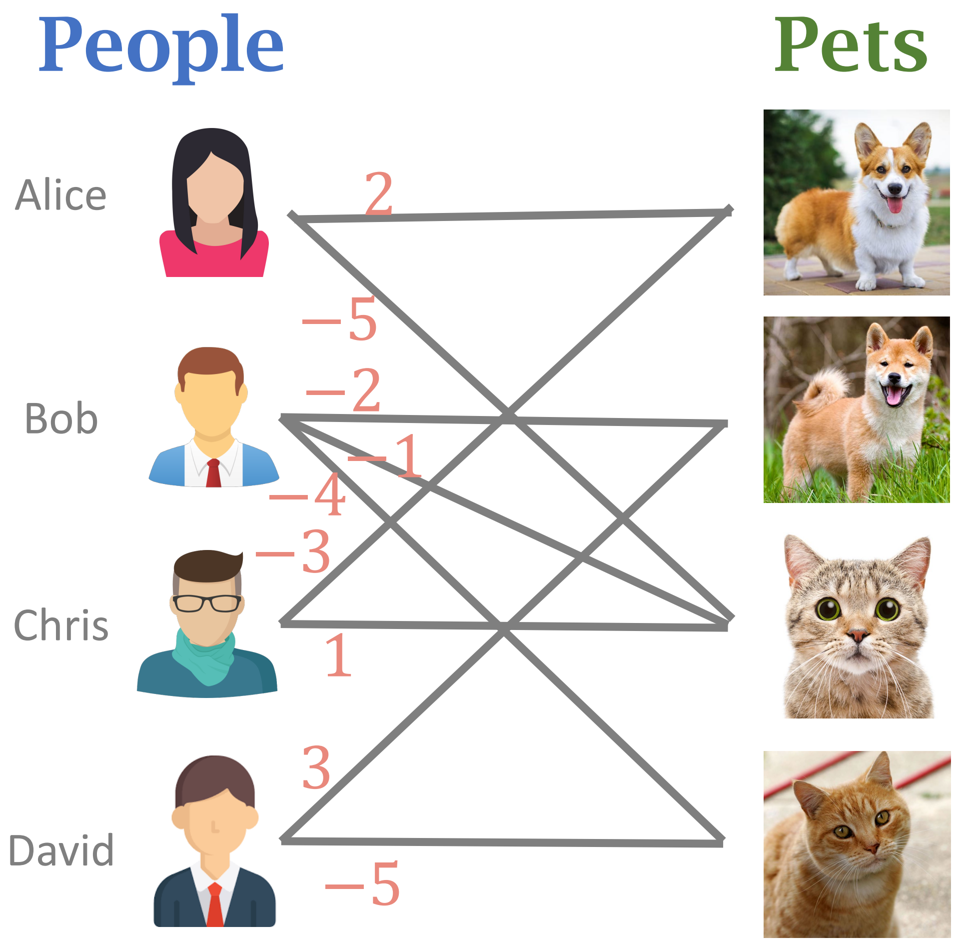
Maximum Matching ⟺ Minimum Matching

Maximum Matching



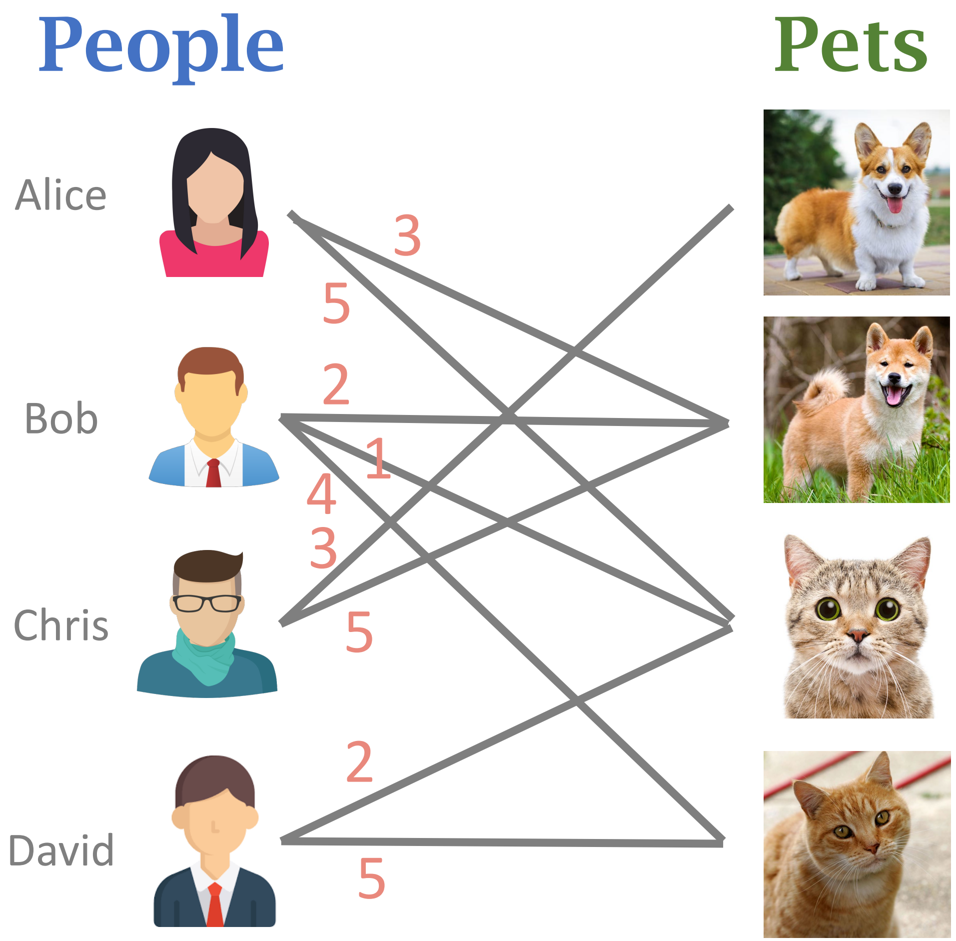
- Adopting a pet can bring happiness to people.
- A weight quantifies how much a person loves a pet.
- Maximize the weights of matching. (Maximize people's happiness.)

Minimum Matching



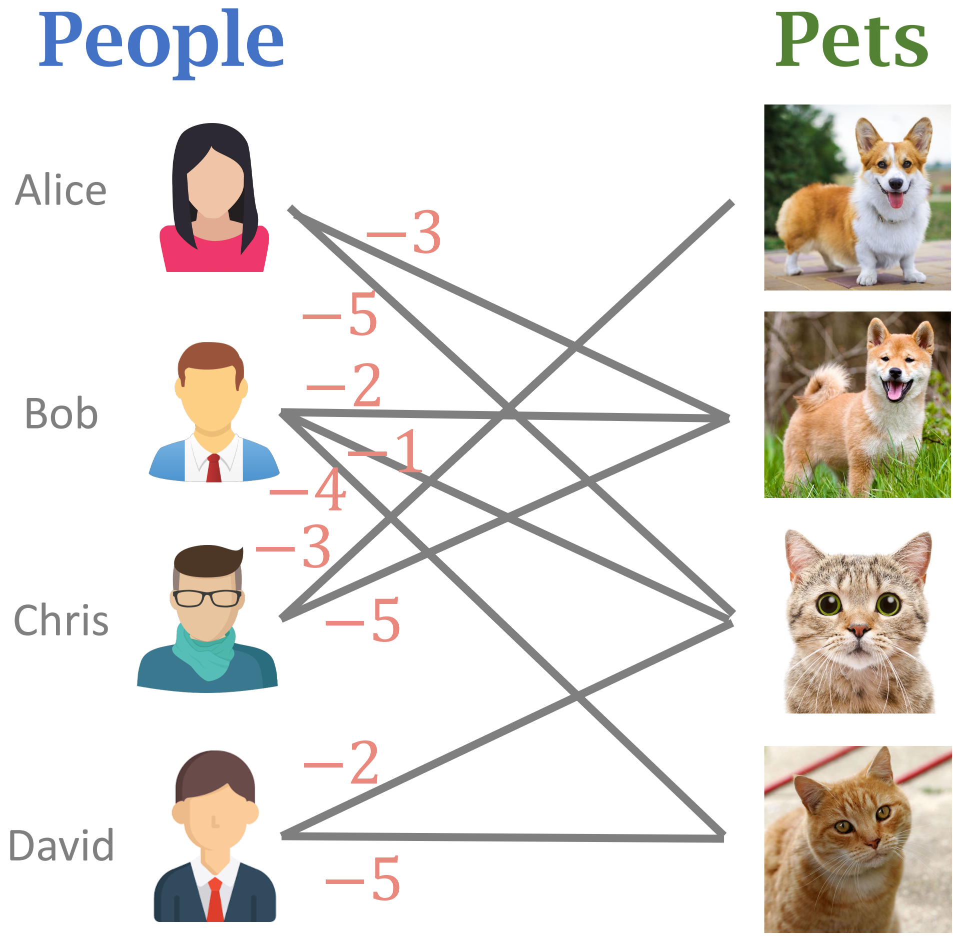
- Adopting a pet can cost time and money.
- A weight quantifies how a person dislike a pet.
- Minimize the weights of matching. (Maximize people's happiness.)

Maximum Matching ⟺ Minimum Matching



- If we have an algorithm for finding minimum matching.
- Then we can use it for finding maximum matching.

Maximum Matching ⟺ Minimum Matching



- If we have an algorithm for finding minimum matching.
- Then we can use it for finding maximum matching.
    1. Flip the signs of all the weights.
    2. Run the minimum matching algorithm.

Hungarian Algorithm

- Hungarian algorithm is for finding the minimum-weight bipartite matching.
- On the graph, the cardinality of $\mathcal{U}$ and $\mathcal{V}$ must be the same:

$$\boxed{|\mathcal{U}| = |\mathcal{V}| = n.}$$

Reference:

- Harold W. Kuhn. The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*, 2: 83–97, 1955.

Hungarian Algorithm

- Hungarian algorithm is for finding the minimum-weight bipartite matching.
- On the graph, the cardinality of $\mathcal{U}$ and $\mathcal{V}$ must be the same:

$$|\mathcal{U}| = |\mathcal{V}| = n.$$

- Time complexity: $O(n^3)$.

Reference:

- Harold W. Kuhn. The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*, 2: 83–97, 1955.

Hungarian Algorithm

- Hungarian algorithm is for finding the minimum-weight bipartite matching.
- How to solve maximum-weight bipartite matching?

- Flip the signs of all the weights.
- On the new graph, run Hungarian algorithm to find the minimum-matching.
- The outcome is the maximum-matching on the original graph.

Thank You!