

# Hungarian Method

## Centralized Hungarian Method

### Background

- **Bipartite Graph:** A Graph  $G = (V, E)$ , where the vertex set  $V$  is decomposed into two disjoint sets of vertices  $A$  and  $B$  respectively, such that no two vertices in the same set are adjacent. In general, it is said that  $G$  has a bipartition  $(A, B)$ .
- **Matching:** A set of edges without common vertices without common edges
- **Maximum Cardinality Matching:** A matching that contains the large possible number of edges.
- **Vertex Cover:** A set of vertices such that each edge is incident on at least one vertex of the set.
- **Minimum Vertex Cover:** A vertex cover that contains the smallest possible number of vertices.

### Remark 1

In a bipartite graph, the number of edges in a maximum cardinality matching equals the number of vertices in a minimum vertex cover. In fact, due to this inter-relation between a matching and a vertex cover, algorithms used for finding a maximum cardinality M, can be extended to finding a corresponding minimum vertex cover  $V_c \subset V$ .

Using the definitions above (as well as the remark) the **Minimum Weight Bipartite Matching Problem** can be stated as:

Given a graph  $G = (V, E)$  with bipartition  $(A, B)$  and weight function  $w : E \rightarrow \mathbb{R}$ , the objective is to find a maximum cardinality  $M$  of minimum cost, where the cost of matching  $M$  is given by  $c(M) = \sum_{e \in M} w(e)$ .

Next we can state the dual of the Minimum Weight Bipartite Matching Problem:

Given a graph  $G = (V, E)$  with bipartition  $(A, B)$  and weight function  $w : E \rightarrow \mathbb{R}$  and a vertex labeling function  $y : V \rightarrow \mathbb{R}$ , the objective is to find a feasible labeling of maximum cost, where a feasible labeling is a choice of labels  $y$  such that  $w(a, b) \geq y(a) + y(b) \forall (a, b) \in E$ , and the cost of labeling is given by  $c(y) = \sum_{a \in M} w(a) + \sum_{b \in M} w(b)$ .

Moreover, given a feasible labeling  $y$ , and **equality graph**  $G_y = (V, E_y)$  is a sub-graph of  $G$  such that  $E_y = \{(a, b) \mid y(a) + y(b) = w(a, b)\}$

### Theorem (Kuhn-Munkres)

Given a bipartite graph  $G = (V, E)$  with bipartition  $(A, B)$ , a weight function  $w : E \rightarrow \mathbb{R}$ , and a vertex labeling function  $y : V \rightarrow \mathbb{R}$ , let  $M$  and  $y$  be feasible ( $M$  is a perfect matching and  $y$  is a feasible labeling). Then  $M$  and  $y$  are optimal if and only if  $M \subseteq E_y$ , i.e. each edge in  $M$  is also in the set of equality subgraph edges  $E_y$ , given by (68).

**Initialization:** Given a graph  $G = (V, E)$  with bipartition  $(A, B)$ , and a weight function  $w$  (Figure 29a), the Hungarian Method begins with an arbitrary feasible labeling  $y$  (Figure 29b), generates the corresponding equality subgraph edges  $E_y$  using (68) (Figure 15c), finds a maximum cardinality matching  $M_y \subseteq E_y$ , and a corresponding minimum vertex cover  $V_{c_y} \subset V$ , with bipartition  $(A_{c_y}, B_{c_y})$  as per Remark 1 (Figure 15d). The algorithm then performs the following two-step iterations repeatedly, until  $M_y$  is a perfect matching:

1. The algorithm uses  $V_{c_y}$  to isolate a set of candidate edges  $E_{cand} \subseteq E \setminus E_y$  as per Remark 2, and calculates the following (Figure 16a):

$$\delta = \min_{(a, b) \in E_{cand}} \text{slack}(w, y, a, b)$$

where  $\text{slack}(w, y, a, b) = w(a, b) - (y(a) + y(b))$

- Using  $\delta$  and  $V_{c_y}$ , the algorithm updates  $y$  as follows (Figure 16b):

$$y(a) = y(a) - \delta, \forall a \in A_{c_y} \quad y(b) = y(b) + \delta, \forall b \in B \setminus B_{c_y}$$

2. For the updated  $y$ , the algorithm finds the corresponding equality subgraph edges  $E_y$  using (68) (Figure 16c), to find a maximum cardinality matching  $M_y \subseteq E_y$ , and a corresponding minimum vertex cover  $C_{c_y} \subset V$  as per Remark 1.

### Lemma

Given  $G = (V, E)$  with bipartition  $(A, B)$ , a weight function  $w$ , a feasible vertex labeling function  $y$ , and a corresponding matching  $M_y$ , every two-step iteration (steps (a) and (b)) of the Hungarian Method results in the following: (i) An updated  $y$  that remains feasible. (ii) An increase in the matching size  $|M_y|$ , or no change in the matching  $M_y$ , but an increase in  $|A_{c_y}|$  (and corresponding decrease in  $|B_{c_y}|$ , such that  $|A_{c_y}| + |B_{c_y}| = |M_y|$ ).

## Distributed Hungarian Method

Similar to the centralized Hungarian Method, the idea behind the distributed version is to execute in a finite amount of steps. However, execution of these steps are distributed across the robots. In particular, each robot shares and operates on limited information.

All robots begin by running their individual copies of the algorithm at some initial time  $t_0 \in \mathbb{R}_{\geq 0}$ , and continue to run it synchronously at intervals of  $T_s$  seconds ( $T_s \in \mathbb{R}_{>0}$ ). The robot maintains and updates a global iteration counter  $\alpha \in \mathbb{N}$ .

Each robot starts with initial information  $(R, P, c^i)$  and creates a bipartite graph  $G_{orig}^i = (V, E_{orig}^i)$  with bipartition  $(R, P)$ , and weight function  $w_{orig}^i : E \rightarrow \mathbb{R}$ . Moreover, throughout the execution of the algorithm, robot  $i$  shares, updates, and operates on a bipartite graph  $G_{lean}^i = (V, E_{lean}^i)$  with bipartition  $(R, P)$ , a corresponding weight function  $w_{lean}^i : E_{lean}^i \rightarrow \mathbb{R}$ , and a corresponding vertex labeling function  $y^i : V \rightarrow \mathbb{R}$ .

On every iteration of the algorithm, the following two actions are performed:

1. **Send:** Robot  $i$  sends a message  $msg^i$ , to each of its outgoing neighbors, where  $msg^i = (G_{lean}^i, w_{lean}^i, y^i, \gamma^i)$  where  $\gamma^i$  denotes the cumulative number of completed stages (two-step iterations).
2. **Receive and Compute:** Robot  $i$  receives the message from its incoming neighbors, and through computations involving the received information, its own message  $msg^i$  and its original information  $(G_{orig}^i, w_{orig}^i)$  robot  $i$  prepares its new message for sending.

Every robot extracts specific information from the messages it receives from its neighbor with the largest  $\gamma$  value (including its own). Robot  $i$  then combines the information into an updated  $(G_{lean}^i, w_{lean}^i, y^i, \gamma^i)$  and continues with the Hungarian Method. The caveat is that robot  $i$  must have sufficient candidate edges in  $E_{cand}^i$ , for it to be able to execute Step 2. If this is not the case, robot  $i$  gathers whatever pertinent candidate edges it has (drawing an edge if required, from its original information  $(G_{orig}^i, w_{orig}^i)$ ) and updates  $E_{cand}^i$  accordingly.