Introduction to Modern Combinatorial Optimization **POSTECH, Winter 2025**
Lecture #5: Hungarian algorithm and matching markets January 15, 2025
Lecturer: Dabeen Lee

Outline

In this lecture, we explain the Hungarian algorithm for computing a maximum weight matching in a bipartite graph. Next, as an application of bipartite matching, we consider the problem of matching markets based on the Vickrey–Clarke–Groves pricing mechanism.

1 Hungarian algorithm for maximum weight bipartite matching

In Lecture 3, we learned an LP-based algorithm for computing a maximum weight matching in a bipartite graph. In this section, we introduce a combinatorial algorithm, that is known as the **Hungarian algorithm**.

Preprocessing step Let $G = (V, E)$ be a bipartite graphm and let $w \in \mathbb{R}^{|E|}$ be the edge weight vector.

1. First, as we are interested in a maximum weight matching, we may discard edges with a negative weight.
2. Up to adding dummy vertices and dummy edges with weight zero, we obtain a complete bipartite graph $K_{n,n}$ for some $n \geq 1$.

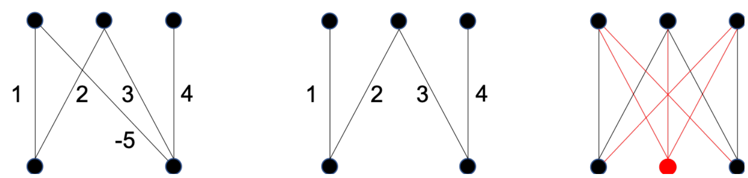

Figure 5.1: illustrating the preprocessing step

We may delete the dummy vertices and dummy edges later.

After the preprocessing step, we may assume that $G = K_{n,n}$ for some $n \geq 1$ and $w \in \mathbb{R}^{|E|}_+$, in which case the problem boils down to finding a **maximum weight perfect matching** in G. As before, let the vertex set V be partitioned into V_1 amd V_2 with $|V_1| = |V_2| = n$. Then a maximum weight matching in G can be computed by

$$\begin{aligned} \text{maximize} \quad & \sum_{e \in E} w_e x_e \\ \text{subject to} \quad & \sum_{v \in V_2} x_{uv} \leq 1 \quad \text{for all } u \in V_1, \\ & \sum_{u \in V_1} x_{uv} \leq 1 \quad \text{for all } v \in V_2, \\ & x_e \geq 0 \quad \text{for all } e \in E. \end{aligned} \tag{5.1}$$

Again, as $w_e \geq 0$ for all $e \in E$ and G is a complete bipartite graph, (5.1) has an optimal solution that corresponds to a perfect matching. Then it follows that (5.1) is equivalent to

$$\begin{aligned}
\text{maximize} \quad & \sum_{e\in E} w_e x_e \\
\text{subject to} \quad & \sum_{v\in V_2} x_{uv} = 1 \quad \text{for all } u \in V_1, \\
& \sum_{u\in V_1} x_{uv} = 1 \quad \text{for all } v \in V_2, \\
& x_e \geq 0 \quad \text{for all } e \in E.
\end{aligned} \tag{5.2}$$

The dual of (5.2) is given by

$$\begin{aligned}
\text{minimize} \quad & \sum_{u\in V_1} y_u + \sum_{v\in V_2} z_v \\
\text{subject to} \quad & y_u + z_v \geq w_{uv} \quad \text{for all } uv \in E.
\end{aligned} \tag{5.3}$$

The following result is a direct consequence of the **complementary slackness condition** for linear programming, while we state its direct proof.

Lemma 5.1. *Let M be a perfect matching in G. Suppose that there exists a feasible solution (y, z) to (5.3) that satisfies $y_u + z_v = w_{uv}$ for every $uv \in M$. Then M is a maximum weight matching.*

Proof. Let M' be a perfect matching in G, and let (y', z') be a solution satisfying the constraints of (5.3). Then Note that for any solution (y, z) satisfying the constraints of (5.3), we have

$$\sum_{uv\in M'} w_{uv} \leq \sum_{uv\in M'} (y'_u + z'_v) = \sum_{u\in V_1} y'_u + \sum_{v\in V_2} z'_v$$

where the equality holds because M' is a perfect matching. This implies that

$$\begin{aligned}
&\max\left\{\sum_{uv\in M'} w_{uv} : \ M' \text{ is a perfect matching}\right\} \\
&\leq \min\left\{\sum_{u\in V_1} y'_u + \sum_{v\in V_2} z'_v : \ y'_u + z'_v \geq w_{uv} \quad \text{for all } uv \in E\right\}
\end{aligned}$$

If some (y, z) satisfies $y_u + z_v = w_{uv}$ for every $uv \in M$, then it follows that

$$\sum_{uv\in M} w_{uv} = \sum_{uv\in M} (\bar{y}_u + \bar{z}_v) = \sum_{u\in V_1} \bar{y}_u + \sum_{v\in V_2} \bar{z}_v.$$

This indicates that the weight of M achieves the maximum possible, and therefore, M is a maximum weight matching. □

Based on Lemma 5.1, the main idea behind the Hungarian algorithm is as follows.

- (y, z) always remains feasible to (5.3), satisfying the constraints of (5.3).
- Only an edge $uv \in E$ satisfying $y_u + z_v = w_{uv}$ can be added to our matching M.

Once our matching M becomes a perfect matching, then it will satisfy the conditions of Lemma 5.1, which guarantees that M is a maximum weight matching. To implement this idea, we introduce the notion of **equality subgraphs**. Given a feasible solution (y, z) to (5.3), we define the subgraph of G taking the edges $uv \in E$ satisfying $y_u + z_v = w_{uv}$. We use notation $G_{y,z}$ to denote the equality subgraph of G associated with (y, z).

- Given a feasible solution (y, z) to (5.3), we take a maximum matching M in $G_{y,z}$.

Based on this, we deduce the Hungarian algorithm.

Algorithm 1 Hungarian algorithm for maximum weight bipartite matching

> **Input:** complete bipartite graph $G = (V, E)$ with $V = V_1 \cup V_2$ and $w \in \mathbb{R}_+^{|E|}$
> Initialize $y_u = \max_{v \in V_2} w_{uv}$ for $u \in V_1$, $z_v = 0$ for $v \in V_2$
> Initialize $M = \emptyset$ and $B = \emptyset$
> **while** M is not a perfect matching **do**
>   Construct the equality subgraph $G_{y,z}$ associated with (y, z)
>   Set M and B as a maximum matching and a minimum vertex cover in $G_{y,z}$, respectively
>   Set $R = V_1 \cap B$ and $T = V_2 \cap B$
>   Compute $\epsilon = \min \{y_u + z_v - w_{uv} : u \in V_1 - R, v \in V_2 - T\}$
>   Update $y_u = y_u - \epsilon$ for $u \in V_1 - R$ and $z_v = z_v + \epsilon$ for $v \in T$
> **end while**
> Return M

Example 5.2 (Example 3.2.10., West)**.** Let us consider an example with $G = K_{5,5}$ given by Figure 5.2. In each matrix, the rows correspond to the vertices in V_1, and the columns are for the

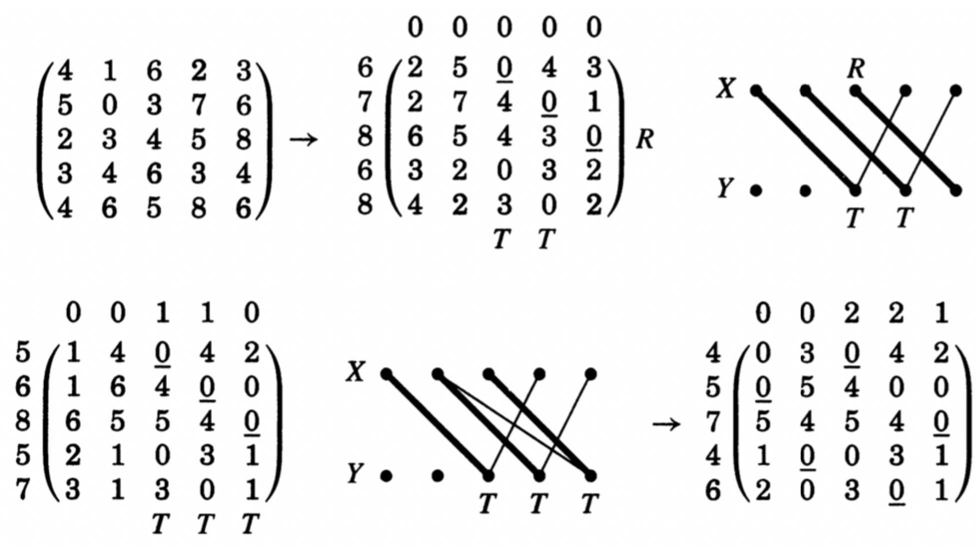

Figure 5.2: an example of running the Hungarian algorithm

vertices in V_2.

Theorem 5.3. *Let $G = (V, E)$ be a complete bipartite graph, and let $w \in \mathbb{R}_+^{|E|}$. Then Algorithm 1 finds a maximum weight pefect matching in G.*

Proof. First, we argue that (y, z) always remains feasible to (5.3). The initial solution with $y_u = \max_{v \in V_2} w_{uv}$ for $u \in V_1$, $z_v = 0$ for $v \in V_2$ is feasible because $\max_{v \in V_2} w_{uv} \geq w_{uv}$ for any $uv \in E$. Suppose that (y, z) is feasible to (5.3) at some point of running Algorithm 1. Let M and B be a maximum matching and a minimum vertex cover in $G_{y,z}$, respectively. Moreover, we take $R = V_1 \cap B$ and $T = V_2 \cap B$ and

$$\epsilon = \min \left\{ y_u + z_v - w_{uv} : \ u \in V_1 - R, \ v \in V_2 - T \right\}.$$

Assume that M is not a perfect matching. Then. let (y', z') denote what is obtained from (y, z) after the update. It is sufficient to check that $y'_u + z'_v \geq w_{uv}$ for $u \in V_1 - R$ and $v \in V_2$. Note that for $u \in V_1 - R$,

$$y'_u + z'_v = \begin{cases} y_u - \epsilon + z_v & \text{if } v \in V_2 - T, \\ y_u - \epsilon + z_v + \epsilon & \text{if } v \in T. \end{cases}$$

Moreover, for $u \in V_1 - R$ and $v \in V_2 - T$, we have

$$y_u + z_v - \epsilon \geq y_u + z_v - (y_u + z_v - w_{uv}) = w_{uv}.$$

Therefore, what remains is to argue that Algorithm 1 terminates with a perfect matching. Suppose that the current matching M is not a pefect matching, in which case $B = R \cup T$ is not a vertex cover of G. That meanas that there exists an edge not covered by B, which implies that the has not yet appeared in the equality subgraph $G_{y,z}$. Therefore, we must have $\epsilon > 0$. Let $u \in V_1 - R$ and $v \in V_2 - T$ be such that $y_u + z_v - w_{uv} = \epsilon$. Then after the update, we have

$$y'_u + z'_v - w_{uv} = y_u - \epsilon + z_v - w_{uv} = 0.$$

Hence, the edge uv newly enters the equality subgraph. That said, one instance of the while loop increases the number of edges in the equality subgraph by at least 1. Note that G has $O(|V|^2)$ edges in total, so the algorithm will terminate eventually. $\square$

2 Matching markets

Suppose that we have a nework of sellers and buyers for certain items in a market place. To simplify our discussion, let us assume that there are three sellers labeled u, v, and w and that we have a set of three buyers labeled x, y, and z. Each seller offers an item, and each buyer has certain valuations of the items as shown in Figure 5.3. The sellers, or the market, are supposed to set the prices of

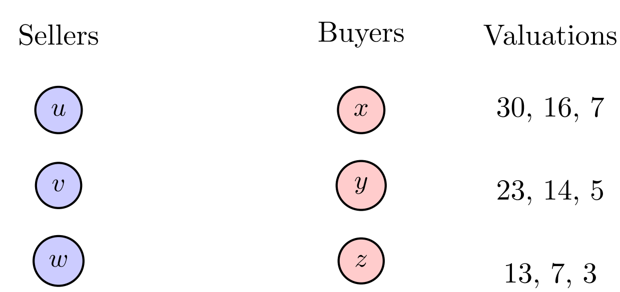

Figure 5.3: matching market example

items. For the item offered by seller $i \in \{u, v, w\}$, we use notation p_i for its price. We use notation v_{ij} to denote the valuation of buyer $j \in \{x, y, z\}$ for the item offered by seller $i \in \{u, v, w\}$. Suppose that buyer j is assigned to seller i and gets to buy its item. Then the **utility** of buyer j buying the item of seller i is given by

$$u_{ij} := v_{ij} - p_i.$$

We assume that the **rational behavior** of buyer j, which means that the buyer would decide to buy the item from seller i only if u_{ij} is nonnegative. It is natural that the assignment of buyers to sellers can be represented as a bipartite matching. Let $M \subseteq \{u, v, w\} \times \{x, y, z\}$ denote a matching or an assignment of buyers and sellers. Then the **social welfare** is defined as

the social welfare = the total profit of sellers + the total profit of buyers.

Then it follows that

$$\begin{aligned}\textbf{the social welfare} &= \sum_{ij \in M} (\textbf{the profit of buyer } i + \textbf{the profit of seller } j) \\ &= \sum_{ij \in M} (p_i + v_{ij} - p_i) \\ &= \sum_{ij \in M} v_{ij}.\end{aligned}$$

Therefore, the social welfare equals the valuation sum of items that are matched with buyers. Then the social welfare can be viewed as the weight of a matching M where each assignment between seller i and buyer j is given by the item valuation v_{ij}. In turn, this implies that the social welfare is maximized if the corresponding matching is a maximum weight matching.

We have just argued that finding a maximum weight matching leads to the maximum social welfare. However, individual buyers would behave rationally, so they will always target an item with the highest utility. It is quite likely to have conflicts between buyers. To respond to such scenarios, a market moderator would set a high price for a popular item. We call the set of prices are **market clearing** when a perfect matching is available under the prices. In this section, we will explain the **Vickrey–Clarke–Groves (VCG) mechanism** that is proven to be market clearing.

Let us explain how the VCG mechanism works. The basic idea is that whenever there is a conflict which forbids a perfect matching, we increase the price of some item. Here, a conflict can be captured by the notion of **preferred-seller graph**. For each buyer j, we draw an edge between buyer j and seller u for every $u \in \operatorname{argmax}\{u_{ij} = v_{ij} - p_i : i \in \{u, v, w\}\}$. To elaborate, let us set the prices of items to 0 initially, and the corresponding preferred-seller graph is given in Figure 5.4. By Hall's marriage theorem, the current preferred-seller graph does not have a perfect matching

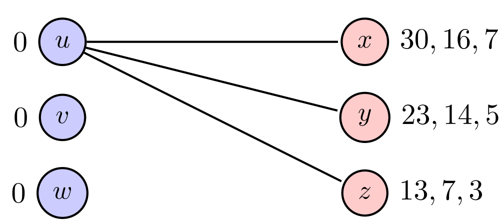

Figure 5.4: initial preferred-seller graph

because $N(\{x, y, z\}) = \{u\}$. We have just identified a conflict $S_1 = \{x, y, z\}$, which means that $|N(S_1)| < |S_1|$. Then we increase the price of the item of seller $u \in N(S_1)$ until we get a change in the preferred-seller graph. Let us set the price p_u to 6. Then we get the following new preferred-seller graph in Figure 5.5. The preferred-seller graph in Figure 5.5 also has a conflict $S_2 = \{x, y, z\}$

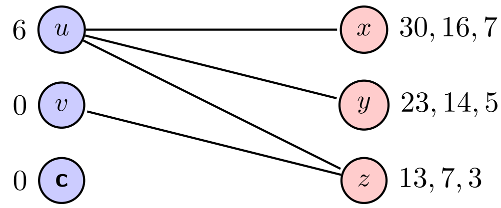

Figure 5.5: after increasing the price of the item in $N(S_1)$

as $N(S_2) = \{u, v\}$ and $|N(S_2)| < |S_2|$. Again, we increase the prices of items in $N(S_2) = \{u, v\}$ until we deduce a change in the preferred-seller graph. Let us increase p_u and p_v by 4. As a result, we deduce our new preferred-seller graph given in Figure 5.6. Note that Figure 5.6 still has a

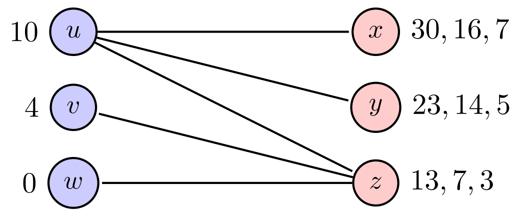

Figure 5.6: after increasing the prices of the items in $N(S_2)$

conflict, $S_3 = \{x, y\}$. As before, we increase the price p_u by 3, which gives us the new graph given in Figure 5.7. Finally, the preferred-seller graph admits a perfect matching without a conflict. The

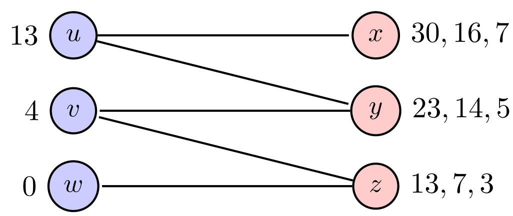

Figure 5.7: after increasingthe prices of the items in $N(S_3)$

perfect matching generates the assigned pair of (u, x), (v, y), and (w, z).

Theorem 5.4. *The Vickrey–Clarke–Groves (VCG) mechanism always finds a market clearing price that maximizes the social welfare in finite time.*

Proof. We prove finite termination first. Let I denote the set of sellers, and let J denote the set of buyers. We take the potential defined by

$$\Phi = \sum_{i \in I} p_i + \sum_{j \in J} \max\{v_{ij} - p_i : i \in I\}.$$

Initially, the price of each item is set to 0. Hence, the initial potential is given

$$\Phi_{\text{initial}} = \sum_{j \in J} \max\{v_{ij} : i \in I\}.$$

We next argue that the potential strictly decreses until we find a perfect matching. If there is no perfect matching in the current preferred-seller graph, there is a conflict set $S \subseteq J$ such that

$|N(S)| < |S|$. In this case, we increase the price of items in $N(S)$ by some δ. After the update, the total profit of sellers increases by $|N(S)|\delta$ while the total profit of buyers decreases by $|S|\delta$. As $|N(S)|$ is strictly less than $|S|$, it follows that the potential strictly decreases. Therefore, the algorithm terminates in finite time.

When the preferred-seller graph contains a perfect matching, the VCG mechanism selects a perfect matching M that maximizes

$$\sum_{ij \in M} (v_{ij} - p_i) = \sum_{ij \in M} v_{ij} - \sum_{i \in I} p_i,$$

which is equivalent to maximizing the social welfare. $\square$