Lecture 4: König's theorem and the Hungarian algorithm

Dabeen Lee

Industrial and Systems Engineering, KAIST

2025 Winter Lecture Series on Combinatorial Optimization

January 14, 2025

Outline

- Vertex cover problem
- LP duality-based proof for König's theorem
- Hungarian algorithm for maximum weight bipartite matching

Vertex cover

- Given a graph $G = (V, E)$, a subset B of the vertex set V is called a **vertex cover** if for every edge $e \in E$, e has an endpoint in B.

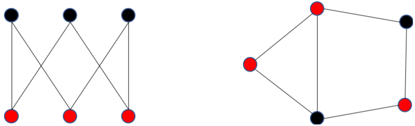


Figure: vertex cover examples

- The **vertex cover problem** is to find a vertex cover with the minimum number of vertices.

Connection to bipartite matching

Proposition

Let $G = (V, E)$ be a graph. Then the minimum size of a vertex cover for G is greater than or equal to the maximum size of a matching in G.

König's theorem

Theorem (König's theorem)

Let $G = (V, E)$ be a bipartite graph. Then the minimum size of a vertex cover for G equals the maximum size of a matching in G.

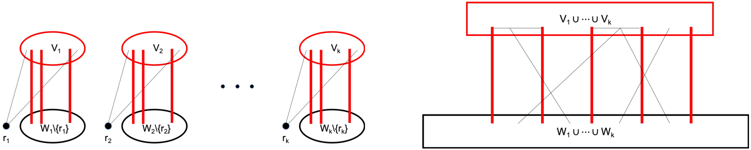


Figure: vertex set decomposition by the alternating tree procedure

Remarks

- The proof suggests that the augmenting path algorithm not only gives us a maximum matching but also a minimum vertex cover.
- This means that the vertex cover problem can be solved in polynomial time.
- However, the vertex cover problem for general graphs is known to be NP-hard.

LP formulation for vertex cover

- As for the matching problem, vertex cover also admits an integer linear programming formulation.
- For each vertex $v \in V$, we use a variable y_v to indicate whether v is picked for our vertex cover B or not, i.e.,

$$y_v = \begin{cases} 1 & \text{if } v \text{ is included in vertex cover } B, \\ 0 & \text{otherwise.} \end{cases}$$

- Then we may impose the condition that y corresponds to a vertex cover by setting

$$y_u + y_v \geq 1$$

  for all $uv \in E$.

LP formulation for vertex cover

- Therefore, the vertex cover problem can be equivalently formulated as the following integer linear program:

$$\begin{aligned} \text{minimize} \quad & \sum_{v \in V} y_v \\ \text{subject to} \quad & y_u + y_v \geq 1 \quad \text{for all } uv \in E, \\ & y_v \in \{0, 1\} \quad \text{for all } v \in V. \end{aligned} \tag{IP}$$

Proposition

Let $G = (V, E)$ be a graph, not necessarily bipartite. Then solving the optimization problem (IP) computes a minimum vertex cover for G.

LP formulation for vertex cover

- The LP relaxation of (IP) is given by

$$
\begin{aligned}
\text{minimize} \quad & \sum_{v \in V} y_v \\
\text{subject to} \quad & y_u + y_v \geq 1 \quad \text{for all } uv \in E, \\
& y_v \geq 0 \quad \text{for all } v \in V.
\end{aligned}
\tag{LP}
$$

LP formulation for vertex cover

Theorem

Let $G = (V, E)$ be a bipartite graph. Then the LP relaxation (LP) has an optimal solution y^* that satisfies $y_v^* \in \{0, 1\}$ for all $v \in V$. Moreover, one can find a minimum vertex cover for G by solving the linear program (LP).

- Let $\bar{y}$ be an optimal solution to (LP). By the nonnegativity constraint, we have $\bar{y}_v \geq 0$ for all $v \in V$.
- If $\bar{y}_v > 1$ for some $v \in V$, then one may replace $\bar{y}_v$ with 1 to improve the objective while keeping feasibility.
- This means that $\bar{y}_v \leq 1$ for all $v \in V$ because $\bar{y}$ is an optimal solution.

LP formulation for vertex cover

Theorem

Let $G = (V, E)$ be a bipartite graph. Then the LP relaxation (LP) has an optimal solution y^* that satisfies $y_v^* \in \{0, 1\}$ for all $v \in V$. Moreover, one can find a minimum vertex cover for G by solving the linear program (LP).

Randomized algorithm

1. Pick a random threshold $\theta \in (0, 1)$ uniformly at random.
2. Take $U_1 = \{v \in V_1 : \bar{y}_v \geq \theta\}$ and $U_2 = \{v \in V_2 : \bar{y}_v \geq 1 - \theta\}$.
3. Define $y^* \in \{0, 1\}^{|V|}$ as the incidence vector of $U_1 \cup U_2$.

LP formulation for vertex cover

LP formulation for vertex cover

LP-based algorithm for minimum vertex cover

Algorithm 1 LP-based algorithm for minimum vertex cover

The bipartition $V_1 \cup V_2$ of the vertex set V
Solve the linear program (LP) and get an optimal solution $\bar{y}$
Take $U_1 = \{v \in V_1 : \bar{y}_v \geq 1/2\}$ and $U_2 = \{v \in V_2 : \bar{y}_v \geq 1/2\}$
Return $U_1 \cup U_2$

LP-based proof for König's theorem

- The **strong duality theorem for linear programming** implies

$$\underbrace{\min\left\{\sum_{v\in V} y_v :\ y_u + y_v \geq 1 \quad \text{for all } uv \in E,\ y \in \{0,1\}^{|V|}\right\}}_{\text{the minimum size of a vertex cover}}$$

$$= \min\left\{\sum_{v\in V} y_v :\ y_u + y_v \geq 1 \quad \text{for all } uv \in E,\ y \in \mathbb{R}_+^{|V|}\right\}$$

$$\underbrace{=}_{\text{strong duality}}$$

$$\max\left\{\sum_{e\in E} w_e x_e :\ \sum_{v\in V: uv\in E} x_{uv} \leq 1 \quad \text{for all } u \in V,\ x \in \mathbb{R}_+^{|E|}\right\}$$

$$\underbrace{= \max\left\{\sum_{e\in E} w_e x_e :\ \sum_{v\in V: uv\in E} x_{uv} \leq 1 \quad \text{for all } u \in V,\ x \in \{0,1\}^{|E|}\right\}}_{\text{the maximum size of a matching}}$$

Combinatorial algorithm for maximum weight bipartite matching

- In Lecture 3, we learned an LP-based algorithm for maximum weight bipartite matching.
- Net we cover a combinatorial algorithm, that is known as the **Hungarian algorithm**.

Preprocessing step

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 $n \geq 1$.

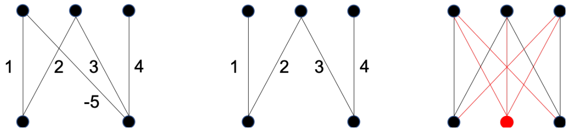


Figure: illustrating the preprocessing step

Hungarian algorithm

- After the preprocessing step, we may assume that $G = K_{n,n}$ for some $n \geq 1$ and $w \in \mathbb{R}_+^{|E|}$.
- Then 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

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

Hungarian algorithm

- Again, as $w_e \geq 0$ for all $e \in E$ and G is a complete bipartite graph, (1) has an optimal solution that corresponds to a perfect matching.
- Then it follows that (1) is equivalent to

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

- The dual of (Primal) is given by

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

- The following result is a direct consequence of the **complementary slackness condition** for linear programming.

Lemma

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

Hungarian algorithm

- Based on the lemma, the main idea behind the Hungarian algorithm is as follows.
  - (y, z) always remains feasible to (Dual), satisfying the constraints of (Dual).
  - Only an edge $uv \in E$ satisfying $y_u + z_v = w_{uv}$ can be added to our matching M.
- Once M becomes a perfect matching, then it will satisfy the conditions of the lemma, which guarantees that M is a maximum weight matching.

Hungarian algorithm

- To implement this idea, we introduce the notion of **equality subgraphs**.
- Given a feasible solution (y, z) to (Dual), 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 (Dual), we take a maximum matching M in $G_{y,z}$.

Hungarian algorithm

Algorithm 1 Hungarian algorithm for maximum weight bipartite matching

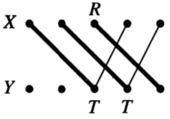
Input: complete bipartite graph $G = (V, E)$ with $V = V_1 \cup V_2$ and $w \in \mathbb{R}_+^{|E|}$
Initialize $y_u = \max_{v \in V_2} w_{uv}$ for $u \in V_1$, $z_v = 0$ for $v \in V_2$
Initialize $M = \emptyset$ and $B = \emptyset$
while M is not a perfect matching **do**
  Construct the equality subgraph $G_{y,z}$ 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

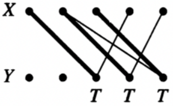
Example

Let us consider an example with $G = K_{5,5}$.

$$\begin{pmatrix} 4 & 1 & 6 & 2 & 3 \\ 5 & 0 & 3 & 7 & 6 \\ 2 & 3 & 4 & 5 & 8 \\ 3 & 4 & 6 & 3 & 4 \\ 4 & 6 & 5 & 8 & 6 \end{pmatrix} \rightarrow \begin{array}{cc} & \begin{matrix} 0 & 0 & 0 & 0 & 0 \end{matrix} \\ \begin{matrix} 6 \\ 7 \\ 8 \\ 6 \\ 8 \end{matrix} & \begin{pmatrix} 2 & 5 & \underline{0} & 4 & 3 \\ 2 & 7 & 4 & \underline{0} & 1 \\ 6 & 5 & 4 & 3 & \underline{0} \\ 3 & 2 & 0 & 3 & 2 \\ 4 & 2 & 3 & 0 & 2 \end{pmatrix} R \\ & \begin{matrix} & & T & T & \end{matrix} \end{array}$$



$$\begin{array}{cc} & \begin{matrix} 0 & 0 & 1 & 1 & 0 \end{matrix} \\ \begin{matrix} 5 \\ 6 \\ 8 \\ 5 \\ 7 \end{matrix} & \begin{pmatrix} 1 & 4 & \underline{0} & 4 & 2 \\ 1 & 6 & 4 & \underline{0} & 0 \\ 6 & 5 & 5 & 4 & \underline{0} \\ 2 & 1 & 0 & 3 & 1 \\ 3 & 1 & 3 & 0 & 1 \end{pmatrix} \\ & \begin{matrix} & & T & T & T \end{matrix} \end{array}$$



$$\rightarrow \begin{array}{cc} & \begin{matrix} 0 & 0 & 2 & 2 & 1 \end{matrix} \\ \begin{matrix} 4 \\ 5 \\ 7 \\ 4 \\ 6 \end{matrix} & \begin{pmatrix} 0 & 3 & \underline{0} & 4 & 2 \\ \underline{0} & 5 & 4 & 0 & 0 \\ 5 & 4 & 5 & 4 & \underline{0} \\ 1 & \underline{0} & 0 & 3 & 1 \\ 2 & 0 & 3 & \underline{0} & 1 \end{pmatrix} \end{array}$$

In each matrix, the rows correspond to the vertices in V_1, and the columns are for the vertices in V_2.

Correctness

Theorem

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.

Correctness