# Bipartite Matching: Framework and Solutions

## A path to the Hungarian Algorithm and beyond

Giancola Simone Maria[1]

[1]Bocconi University, Milan, Italy

Computer Science (Algorithms), Bocconi University, June 2022

# Quote slide

*We see the world in terms of our theories.*

Thomas Kuhn

*Perhaps even more than to the interaction between mankind and nature, graph theory is based on the interaction of human beings with each other.*

Dénes Kőnig

# Presentation Contents

# Presentation Path

# Notation

**This is a definition**

Here I define something

**This is a theorem**

Something is gnihtemoS backwards

**Proof**

This is a proof

**A remark an observation or an example**

for example, I observe or remark that this is an observation

# Presentation Path

# Motivating Example

## Resource allocation problem

An employer wants to fill three positions with three candidates. Costs are known. The aim is minimizing the total cost.

| | **President** | **CEO** | **CFO** |
|---|---|---|---|
| **George** | 40 | 2 | 45 |
| **Paul** | 1 | 20 | 30 |
| **Kristine** | 50 | 62 | 3 |

Table: Problem representation

The optimal allocation has cost $C = 6$ where:

$$(Paul, President)\ (George, CEO)\ (Kristine, CFO)$$

# Problem Nature

- Balanced Assignment Problem $\mathfrak{P}$ part of Combinatorial Optimization Problems family $\mathfrak{P} \subset \mathfrak{C}$

# Problem Nature

- Balanced Assignment Problem $\mathfrak{P}$ part of Combinatorial Optimization Problems family $\mathfrak{P} \subset \mathfrak{C}$
- Enumeration is highly inefficient, in this specific case of order $T(n) \in O(n!)$

# Problem Nature

- Balanced Assignment Problem $\mathfrak{P}$ part of Combinatorial Optimization Problems family $\mathfrak{P} \subset \mathfrak{C}$
- Enumeration is highly inefficient, in this specific case of order $T(n) \in O(n!)$
- Highlights nice properties joining matrices and graphs

# Integer Programming Permutation Formulation I

## $\mathfrak{P}$ permutation form

$$find \;\; X^* \in \mathcal{X} \;\;:\;\; X^* = argmin_{X \in \mathcal{X}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij} \right\} \tag{2.1}$$

Where we define $X$ as a permutation matrix such that:

$$X := \begin{cases} x_{ij} = 1 \;\; if \;\; \pi(a_i) = t_j \;\; a_i \in \mathscr{A}, t_j \in \mathscr{T} \\ x_{ij} = 0 \;\; otherwise \end{cases} \tag{2.2}$$

Subject to the constraints:

$$\begin{cases} \sum_{i=1}^{n} x_{ij} = 1 \\ \sum_{j=1}^{n} x_{ij} = 1 \end{cases} \tag{2.3}$$

# Graph objects I

## Bipartite Graphs $\mathscr{B}$

Bipartite Graphs in $\mathscr{B}$ present two disjoint sets of vertices that do not have any inner edge.

$$\mathscr{B} := \left\{ B \in \mathfrak{G} \mid B = \{(\mathscr{V} \cup \mathscr{W}), \mathscr{E}\} \right\}$$

## Matching $\mathscr{M}$

Given $G = (\mathscr{V}, \mathscr{E}) \in \mathfrak{G}$ a matching is a collection of edges such that all the vertices are reached only once: A vertex $v \in \mathscr{V}$ is said to be **exposed** if no edge in $\mathscr{M}$ is incident to it.
A matching has **maximum cardinality** if it contains the maximum possible number of edges in $\mathscr{E}$. Its size is denoted as $\nu(B)$
A matching is **perfect** if it assigns an edge to each vertex in $\mathscr{V}$. It is trivially also of maximum cardinality, and no vertex in $G$ is exposed.

# $\mathfrak{P}$ Graph Formulation

**$\mathfrak{P}$ graph form**

Given a Bipartite graph $B \in \mathscr{B}$ where $B = \{(\mathscr{A} \cup \mathscr{T}), C\}$ find the minimum cost perfect matching.

- Assign to each vertex one edge
- equivalently find a one to one agents-tasks matching through edges
- minimize the total cost

# Graph objects II

## Incidence Matrix $A$

For a graph $G = \{\mathscr{V}, \mathscr{E}\} \in \mathfrak{G}$ the incidence matrix $A \in \mathfrak{M}_{|\mathscr{V}|,|\mathscr{E}|}$ is:

$$A := \begin{cases} a_{ij} = 1 & \textit{if } e_j \in \mathscr{E} \ : \ e = (v_i, v), v_i, v \in \mathscr{V} \\ a_{ij} = 0 & \textit{otherwise} \end{cases} \tag{2.4}$$

## Neighbors set generator $\delta(\cdot)$

$$\delta(\cdot) \ : \ \forall v \in \mathscr{V} \ \delta(v) = \{e_v\} \ : e_v = (v, v') \text{ for some } v' \in \mathscr{V} \tag{2.5}$$

Both definitions encapsulate the proximity property that a graph presents.

# Integer Programming Formulation II

## $\mathfrak{P}$ incidence matrix form

For a given Bipartite graph $B \in \mathscr{B}$ where $B = \{(\mathscr{V} \cup \mathscr{W}), \mathscr{E}\}$ solve the following integer program:

$$\begin{cases} x^* = argmin_{x \in \mathbb{R}^{|\mathscr{E}|}} \left\{ x \cdot \mathbf{c} \right\} \\ \sum_{e \in \delta(v)} x_e = 1 \forall v \implies Ax = \mathbf{1} \\ x_e \in \{0, 1\} \forall e \end{cases} \tag{2.6}$$

Where $\mathbf{c}$ is a cost vector identifying the cost of each edge, and $\mathbf{1} = [1, \ldots, 1]^T \in \mathbb{R}^{|\mathscr{E}|}$.
The second constraint ensures that for each vertex only one edge is chosen.
The third constraint ensures that for each edge it is either included or not in the matching $\mathscr{M}$.

# Assumptions on $\mathfrak{P}$

- No isolated vertices
- $|\mathscr{A}| = |\mathscr{T}| = n$ equal size of bipartite sets
- $|\mathscr{E}| = n^2$ fully bipartitely connected graph

## Remark, transforming imbalanced instances

- If the Cost function is not defined for some pair $\nexists C(a_i, t_j)$ assign infinite cost to the missing edges:

$$C(a_i, t_j) = \infty$$

- If the number of agents and tasks is different $|\mathscr{A}| \neq |\mathscr{T}|$, take the least one of the two and adjust the formulation adding infinite costs.
- $B \in \mathscr{B} \rightsquigarrow B \in \mathscr{K}_{n,m} \rightsquigarrow B \in \mathscr{K}_{n,n}$ complete bipartite graphs

# Presentation Path

# Linear Programming Relaxations

## $\mathfrak{P}$ relaxed form(s)

$$X^* = argmin_{X \in \mathscr{X}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_{ij} \right\} s.t. \quad \mathscr{X} := \begin{cases} \sum_{i=1}^{n} x_{ij} = 1 \\ \sum_{j=1}^{n} x_{ij} = 1 \\ x_{ij} \geq 0 \end{cases} \tag{3.1}$$

Similarly:

$$\begin{cases} x^* = argmin_{x \in \mathbb{R}^{|\mathscr{E}|}} \left\{ x \cdot \mathbf{c} \right\} \\ Ax \leq \mathbf{1} \\ x_e \geq 0 \forall e \end{cases} \tag{3.2}$$

# Total Unimodularity

### Totally Unimodular (TU) Matrices $\mathcal{A}$

Totally unimodular matrices are such that every square submatrix has determinant either $0, 1, -1$.

### Hoffman and Kruskal Theorem [7]

$$A \in \mathcal{A} \iff Ext(\{x \ : \ Ax \leq b, \ x \geq 0, \ b \in \mathbb{Z}^n\}) \in \mathbb{Z}^E \tag{3.3}$$

Easier proof by Veinott and Dantzig as a corollary to a more general statement[14].

### Total Unimodularity of Bipartite Graphs

The incidence matrix of a bipartite graph is Totally Unimodular.

$$B \in \mathscr{B} \ : \ A \ incidence \iff A \in \mathcal{A} \tag{3.4}$$

# Proof of $B \in \mathscr{B} \iff A \in \mathcal{A}$

**( $\implies$ direction)**

Since edges are in the columns and each edge joins two vertices, each column has either two ones or all zeros.
By induction on a general graph $B \in \mathscr{B}$, consider all its square submatrices $A'$ of size $k$.
**Base case:** for $k = 1$ the determinant is trivially either 0 or 1.
**Induction Hypothesis:** Assume it is true $\forall A' \ : \ A' \in \mathfrak{M}_{k-1,k-1}$
**Conclusion:** let $A' \in \mathfrak{M}_{k,k}$ be a submatrix of $A$. Since it is a submatrix, then it must be the case that for each column $j'$ there are either all zeros, a single non zero entry, or two non zero entries.

# Proof of $B \in \mathscr{B} \iff A \in \mathcal{A}$

## ( $\implies$ direction)

- If $A'$ has a column with all zeros, then the determinant is zero.
- If $A'$ has a column with one non zero entry at coordinates $i'j'$, then $det(A') = \pm det(A'')$ where $A''$ is the submatrix obtained removing row $i'$ and column $j'$. By $A'' \in \mathfrak{M}_{k-1,k-1}$, the induction hypothesis holds.
- Lastly, if $A'$ has all entries with two ones in the columns, we have a sub bipartite graph originated from $B$. Consider a partition of the rows of $A'$ into the two disjoint vertex sets. Both have column sum equal to one, as any edge joins vertices of the two sets. Summing up all the rows in one vertex set and subtracting those of the other the result is zero, thus $det(A'') = 0$.

# Proof of $B \in \mathscr{B} \iff A \in \mathcal{A}$

## ($\Longleftarrow$ direction)

Let $A \in \mathcal{A}$, $B$ is not bipartite.
$B$ must contain a cycle of odd length $k$. Taking the submatrix $A'$ of this path indexed by $\{v_1, \dots v_k\} \times \{e_1, \dots, e_k\}$, after some column transformations:

$$A' = \begin{bmatrix} 1 & 1 & 0 & \dots & \dots & 0 & 0 \\ 0 & 1 & 1 & \dots & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & \dots & 0 & 0 \\ \dots & \dots & \dots & \ddots & & \dots & \dots \\ \dots & \dots & \dots & & \ddots & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots & 1 & 1 \\ 1 & 0 & 0 & \dots & \dots & 0 & 1 \end{bmatrix} \implies det(A') = 2 \implies A \notin \mathcal{A} \tag{3.5}$$

# Results so far

It holds that:

$$B \in \mathscr{B} \iff A \in \mathcal{A} \iff Ext(\{x \ : \ Ax \leq b, \ x \geq 0 \ b \in \mathbb{Z}^n\}) \in \mathbb{Z}^E \tag{3.6}$$

So the extreme points of the bipartite matching problem are attained at integer (valid) matchings.

$$\implies X^*_{rel} \equiv X^* \tag{3.7}$$

# Covers

## Vertex Cover $\mathscr{C}$

Given $G = (\mathscr{V}, \mathscr{E}) \in \mathfrak{G}$, a cover $\mathscr{C}$ is a set of vertices such that each edge has at least one endpoint included in it. Its minimum size is denoted as $\tau(G)$. A cover of minimum size is said to be **perfect**.

## Fractional Vertex Cover

Given $G = \{\mathscr{V}, \mathscr{E}\} \in \mathfrak{G}$ and a weight function $C : \mathscr{E} \to \mathbb{Z}_+$ a fractional vertex cover is a function $\kappa(\cdot)$ where:

$$\kappa : \mathscr{V} \to \mathbb{Z}_+ \tag{3.8}$$

$$\forall e \in \mathscr{E} \; \kappa(v) + \kappa(w) \leq C(v, w) \tag{3.9}$$

$$K(G) = \sum_{v \in \mathscr{V}} \kappa(v) \text{ is the weight} \tag{3.10}$$

## Min size fractional cost Vertex Cover problem $\mathfrak{D}$

# Earlier results

Duality generalizes the results of two Hungarian Mathematicians : Kőnig and Egerváry. For reference, their results are reported:

**Kőnig Theorem**

Given a Bipartite graph $B$ the size of its maximum matching $\nu(B)$ is equal to the size of its minimum cover $\tau(B)$.

**Kőnig-Egerváry Theorem**

Let $B = \{(\mathscr{V}, \mathscr{W}), \mathscr{E}\} \in \mathscr{B}$ and consider an edge cost function $C$. Then: the maximum weight of a matching is equivalent to the minimum size of valid fractional covers

$$max\{\sum_{e\in\mathscr{M}} c_e x_e\} = min\{\sum_{v} f(v)\} \tag{3.12}$$

$$f \in \{g : \mathscr{V} \to \mathbb{Z}_+ \;\; g(u) + g(v) \geq c_{uv} \; \forall (u, v) \in \mathscr{E}\} \tag{3.13}$$

# Equivalent Statements in Combinatorics

It can be proved ([2] for an incomplete but interesting reference) that 7 major results in Combinatorics are equivalent:

# Equivalent Statements in Combinatorics

It can be proved ([2] for an incomplete but interesting reference) that 7 major results in Combinatorics are equivalent:

- Kőnig's
- Kőnig-Egerváry
- Hall's (marriage)
- Birkhoff-Von Neumann
- Menger's
- Dilworth's
- Ford-Fulkerson (maxflow-mincut)

# Presentation Path

# History

- The original modern formulation of the Hungarian method was published by Harold Kuhn in 1955 [8].

# History

- The original modern formulation of the Hungarian method was published by Harold Kuhn in 1955 [8].
- It was not until 2016 that it was found that also Jacobi had proposed an equivalent method [9].

Main idea:

*If done correctly, decreasing the cost of a row or a column does not contaminate the optimal arrangement.*

# History

- The original modern formulation of the Hungarian method was published by Harold Kuhn in 1955 [8].
- It was not until 2016 that it was found that also Jacobi had proposed an equivalent method [9].
- The term *Hungarian* is a reference to the research contributions of Kőnig and Egerváry
- Computational time given by Munkres [12]

Main idea:

*If done correctly, decreasing the cost of a row or a column does not contaminate the optimal arrangement.*

# Paths and Matchings

## Augmenting Alternating Path $\mathcal{R}_{\mathscr{M}}$

Given a graph $G \in \mathfrak{G}$ and a matching $\mathscr{M} \subseteq \mathscr{E}$ an augmenting alternating path is a path where the first and the last vertices are exposed, so no edges in $\mathscr{M}$ are incident to the endpoints of the path, and the edges alternate between $e \in \mathscr{M}$ and $e \in \mathscr{V} \setminus \mathscr{M}$.

## Augmenting paths improve matchings

Given a bipartite graph $B$ and a matching $\mathscr{M}$ with augmenting $\mathcal{R}_{\mathscr{M}}$:

$$\mathscr{M} \Delta \mathcal{R}_{\mathscr{M}} = \mathscr{M}' \text{ where } |\mathscr{M}'| = |\mathscr{M}| + 1 \tag{4.1}$$

$$\mathscr{M} \ : \ |\mathscr{M}| = \nu(B) \iff \nexists \mathcal{R}_{\mathscr{M}} \tag{4.2}$$

# Directed Graphs

Given a matching $\mathscr{M}$ it is build a directed graph $Q$ as follows:

- If the edge is in the matching $e \in \mathscr{M}$, then it goes from $\mathscr{W} \longrightarrow \mathscr{V}$
- If the edge is not in the matching $e \notin \mathscr{M}$, then it goes from $\mathscr{V} \longrightarrow \mathscr{W}$

**Augmenting path and directed graphs**

Given a graph $G \in \mathfrak{G}$ and a matching $\mathscr{M}$ an augmenting path $\mathcal{R}_{\mathscr{M}}$ for the matching exists if and only if there is a directed path going from one exposed vertex in $\mathscr{W}$ to another one in $\mathscr{V}$.

# Maximum Cardinality Matching Algorithm

**Input:** $P$ with equal negligible costs $C$, and a matching $\mathscr{M}$

**Output:** $|\mathscr{M}'| = |\mathscr{M}| + 1$ in $T(n) \in O(n^2)$

1: Build $Q \in \mathscr{Q}$ by assigning:

$$\begin{cases} e^+ & \textit{if } e \in \mathscr{M} \\ e^- & \textit{if } e \notin \mathscr{M} \end{cases} \tag{4.3}$$

2: Find an augmenting path (a directed path) $\mathcal{R}_{\mathscr{M}}$ between:

$$V = \mathscr{V} \setminus \mathscr{V}_{\mathscr{M}} \quad W = \mathscr{W} \setminus \mathscr{W}_{\mathscr{M}} \tag{4.4}$$

3: $\mathscr{M}' \leftarrow \mathscr{M} \Delta \mathcal{R}_{\mathscr{M}}$

4: **return** $\mathscr{M}'$

# Labelings

### Labeling set $\mathcal{L}_{\mathscr{M}}$

Given a bipartite graph $B$ with matching $\mathscr{M}$, build its directed graph $Q \in \mathscr{Q}$. A labeling set is the set of vertices which can be reached with a directed path from an exposed vertex $v \in \mathscr{V} \setminus \mathscr{V}_{\mathscr{M}}$.

### Induced minimum size vertex cover

For a bipartite graph $B \in \mathscr{B}$, using the subroutine of maximum cardinality matching until convergence returns a matching $\mathscr{M}$. Considering its linked labeling set $\mathcal{L}_{\mathscr{M}}$ it holds that:

- $\mathscr{C} = (\mathscr{V} \setminus \mathcal{L}_{\mathscr{M}}) \cup (\mathscr{W} \cap \mathcal{L}_{\mathscr{M}})$ is a cover
- $|\mathscr{C}| = |\mathscr{M}|$ Kőnig

However, this is holds for the unweighted version!

# Dual as a lower bound

By construction, it is clear that for a fractional cover and a perfect matching $\mathscr{M}$ it holds that:

$$\sum_{e\in\mathscr{M}} \mathbf{c}_e = \sum_{i,j} c_{ij} \geq \sum_{i\in\mathscr{V}} \kappa(i) + \sum_{j\in\mathscr{W}} \kappa(j) \tag{4.5}$$

Which is the dual lower bound. If maximized, it would lead to the desired solution, attained exactly by strong duality. Considering a linear programming formulation for an instance $P \in \mathfrak{P}$:

$$\min_{\mathscr{M}:|\mathscr{M}|=\nu(B)} \left\{ \sum_{e=(i,j)\in\mathscr{M}} \mathbf{c}_e \right\} \geq \min_{x_{ij}\geq 0} \left\{ \mathbf{c}_{ij} x_{ij} \right\} \tag{4.6}$$

$$\geq \max_{\kappa:\mathscr{V} \text{ or } \mathscr{W}\to\mathbb{Z}_+} \left\{ \sum_{i\mathscr{V}} \kappa(i) + \sum_{j\in\mathscr{W}} \kappa(j) \right\} \tag{4.7}$$

# Complementary Slackness

Provided that a feasible solution to $D \in \mathfrak{D}$ is found, linked to a perfect matching $\mathscr{M}$ would lead to the $\geq$ signs to become $=$. This condition is a case of **Complementary Slackness**:

$$\begin{cases} \sum_{e \in \mathscr{M}} \mathbf{c}_e = \sum_{i \in \mathscr{V}} \kappa(i) + \sum_{j \in \mathscr{W}} \kappa(j) \\ \mathbf{c}_e = \mathbf{c}_{ij} \geq 0 \quad \kappa(i) \geq 0 \quad \kappa(j) \geq 0 \end{cases} \tag{4.8}$$

Which by element-wise positivity implies:

$$\implies \omega_{ij} = \mathbf{c}_{ij} - \kappa(i) - \kappa(j) = 0 \tag{4.9}$$

Whenever a dual solution such that $\omega_{ij} = 0 \forall (i,j) \in \mathscr{M}$ is found, the algorithm can return with an optimal matching. If this is not perfect, iterations to improve the proposed feasible dual are implemented until the perfect size is reached.

# Elements of the Hungarian Graph Algorithm

- Starting feasible dual

$$D_{start} = \begin{cases} \kappa(i) = 0 & \forall i \in \mathscr{V} \\ \kappa(j) = \min_{i \in \mathscr{V}} \{\mathbf{c}_{ij}\} & \forall j \in \mathscr{W} \end{cases} \tag{4.10}$$

- Complementary subgraph $B_\omega$

$$B_\omega := \{(\mathscr{V} \cup \mathscr{W}), \mathscr{E}_\omega\} \quad where \quad \mathscr{E}_\omega = \{(i,j) \in \mathscr{E} \; : \; \omega_{ij} = 0\} \tag{4.11}$$

- Labeling $\mathcal{L}_{\mathscr{M}}$ and induced cover $\mathscr{C}$

# Hungarian Algorithm

**Input:** Instance $P = (\mathscr{P}, C) \in \mathfrak{P}$
**Output:** Minimum cost maximum cardinality bipartite matching $\mathscr{M}$

1: Build $B \in \mathscr{B}, D_{start}$ from $P$
2: Evaluate $w_{ij} \forall e = (i, j) \in \mathscr{E}$ ▷ Complementary slackness
3: Build the subgraph $B_\omega$
4: $\mathscr{M} \leftarrow$ max cardinality matching of $B_\omega$ ▷ subroutine Algorithm
5: **if** $|\mathscr{M}| == n$ **then** ▷ Perfect matching found
6: **return** $\mathscr{M}$
7: **end if**
8: Build $\mathcal{L}_{\mathscr{M}}, \mathscr{C} \leftarrow (\mathscr{V} \setminus \mathcal{L}_{\mathscr{M}}) \cup (\mathscr{W} \cap \mathcal{L}_{\mathscr{M}})$ ▷ induced cover
9: Find $\omega^* \leftarrow \underset{i \in (\mathscr{V} \cap \mathcal{L}_{\mathscr{M}})\, j \in (\mathscr{W} \setminus \mathcal{L}_{\mathscr{M}})}{min} \{\omega_{ij}\}$ ▷ minimum non zero weight
10: Update $\kappa(i) \leftarrow \kappa(i) + \omega^* \quad \forall \quad i \in (\mathscr{V} \cap \mathcal{L}_{\mathscr{M}})$
Update $\kappa(j) \leftarrow \kappa(j) - \omega^* \quad \forall \quad j \in (\mathscr{W} \cap \mathcal{L}_{\mathscr{M}})$ ▷ update rules
11: Go back to **Step** 3 ▷ Go back with increased size matching

# Guarantees, Properties

It has a primal dual formulation, equivalent to the one proposed [13]

# Guarantees, Properties

It has a primal dual formulation, equivalent to the one proposed [13]

**Progress**

For each iteration, the algorithm finds a bigger matching

**Finiteness**

The algorithm terminates in finite time

**Polynomial time**

The computational time is polynomially bounded in the size of the problem

$$T(n) \in O(n^3) \tag{4.12}$$

# Proof I

## Progress

Added weight minus removed weight

$$\Delta(D) = \sum_{new\ dual} \kappa - \sum_{old\ dual} \kappa = \omega^* \left( |\mathscr{V} \cap \mathcal{L}_{\mathscr{M}}| - |\mathscr{W} \cap \mathcal{L}_{\mathscr{M}}| \right)$$

$$\begin{aligned}
\Delta(D) =& \omega^* \left( |\mathscr{V} \cap \mathcal{L}_{\mathscr{M}}| + |\mathscr{V} \setminus \mathcal{L}_{\mathscr{M}}| - |\mathscr{V} \setminus \mathcal{L}_{\mathscr{M}}| - |\mathscr{W} \cap \mathcal{L}_{\mathscr{M}}| \right) \\
=& \omega^* \left( |\mathscr{V} \cap \mathcal{L}_{\mathscr{M}}| + |\mathscr{V} \cap \overline{\mathcal{L}_{\mathscr{M}}}| - |\mathscr{V} \setminus \mathcal{L}_{\mathscr{M}}| - |\mathscr{W} \cap \mathcal{L}_{\mathscr{M}}| \right) \\
=& \omega^* (|\mathscr{V}| - |\mathscr{C}|) \\
=& \omega^* (n - |\mathscr{C}|) \geq 0
\end{aligned}$$

# Proof II

## Finiteness [8]

Consider a graph $B \in \mathscr{B}$ with current matching $\mathscr{M}$ for a given iteration. If a vertex $v \in \mathscr{V}$ is exposed, then no edge in $\mathscr{M}$ is incident to it. Thus, there is also a vertex $w \in \mathscr{W}$ which is reachable from $v$, otherwise, the edge joining them would be part of the matching. Now the cover can be constructed, and $\omega^*$ is necessarily found. This also implies that an edge between the cover induced sets is adjusted such that $\omega_{ij} = 0$ for some $e = (i, j)$ where $w_{ij} \equiv \omega^*$. This adds 1 vertex from $\mathscr{W}$ to those reachable from $v \in \mathscr{V}$ for each iteration. By the finiteness of vertices, the algorithm is finite, and the algorithm increases the size of $\mathscr{M}$ by at least one unit every $n$ loops.

# Proof III

## Computational time [12]

In terms of graph operations, it could be argued that:

- In $O(n)$ outer iterations reach a perfect matching

$$T(n) \in O(n^3) \tag{4.13}$$

# Proof III

## Computational time [12]

In terms of graph operations, it could be argued that:

- In $O(n)$ outer iterations reach a perfect matching
- for the inner loop either of the two hold:
  - matching $O(n)$ to find, $O(n)$ to update
  - labeling $O(n)$ to find $\omega$ and update dual, at most $O(n)$ times.
- $\implies O(n^2)$ inner time in the worst case with 1 increase each $n$ iterations.

$$T(n) \in O(n^3) \tag{4.13}$$

# Presentation Path

# Extensions

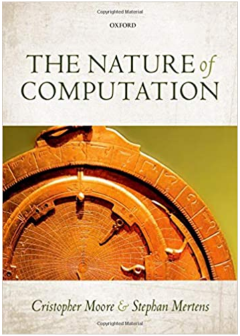
- Counting Hardness [11][6]
- Other objective functions [5][4][3]
- Multidimensional Assignment problem [5]
- Random cost statistical physics analysis [10][1]

# Wrap up

- The Assignment problem has been largerly studied, expanded, analyzed, solved
- Exhibits properties from linear algebra and graph theory
- The Hungarian Algorithm is an efficient solving method which exploits linear programming duality

# Concluding

A suggestion for counting matchings of a graph (Ch. 13-14):




Any question/discussion, let me know!

## Thank you!

simonegiancola09@gmail.com

personal webpage

# References I

[1] David J. Aldous. *The zeta(2) limit in the random assignment problem*. arXiv:math/0010063. type: article. arXiv, Oct. 6, 2000. arXiv: math/0010063. URL: http://arxiv.org/abs/math/0010063 (visited on 06/07/2022).

[2] Robert D Borgersen. "Equivalence of seven major theorems in combinatorics". In: (), p. 18.

[3] R. E. Burkard and U. Zimmermann. "Weakly admissible transformations for solving algebraic assignment and transportation problems". In: *Combinatorial Optimization*. Ed. by M. W. Padberg. Mathematical Programming Studies. Berlin, Heidelberg: Springer, 1980, pp. 1–18. ISBN: 978-3-642-00802-3. DOI: 10.1007/BFb0120884. URL: https://doi.org/10.1007/BFb0120884 (visited on 06/08/2022).

# References II

[4] Rainer E Burkard. "Admissible transformations and assignment problems". In: (), p. 15.

[5] Rainer E. Burkard and Eranda Çela. "Linear Assignment Problems and Extensions". In: *Handbook of Combinatorial Optimization*. Ed. by Ding-Zhu Du and Panos M. Pardalos. Boston, MA: Springer US, 1999, pp. 75–149. ISBN: 978-1-4419-4813-7 978-1-4757-3023-4. DOI: 10.1007/978-1-4757-3023-4_2. URL: http://link.springer.com/10.1007/978-1-4757-3023-4_2 (visited on 06/06/2022).

[6] Michel Goemans. *Advanced Combinatorial Optimization Lecture Notes*. URL: https://math.mit.edu/~goemans/18438S14/lec4-algmat.pdf (visited on 06/12/2022).

# References III

[7] Alan J. Hoffman and Joseph B. Kruskal. "Integral Boundary Points of Convex Polyhedra". In: *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*. Ed. by Michael Jünger et al. Berlin, Heidelberg: Springer, 2010, pp. 49–76. ISBN: 978-3-540-68279-0. DOI: 10.1007/978-3-540-68279-0_3. URL: https://doi.org/10.1007/978-3-540-68279-0_3 (visited on 06/06/2022).

[8] H. W. Kuhn. "The Hungarian method for the assignment problem". In: *Naval Research Logistics Quarterly* 2.1 (Mar. 1955), pp. 83–97. ISSN: 00281441, 19319193. DOI: 10.1002/nav.3800020109. URL: https://onlinelibrary.wiley.com/doi/10.1002/nav.3800020109 (visited on 06/06/2022).

# References IV

[9] Harold W. Kuhn. "A tale of three eras: The discovery and rediscovery of the Hungarian Method". In: *European Journal of Operational Research* 219.3 (June 2012), pp. 641–651. ISSN: 03772217. DOI: 10.1016/j.ejor.2011.11.008. URL: https://linkinghub.elsevier.com/retrieve/pii/S0377221711009957 (visited on 06/06/2022).

[10] M. Mézard and G. Parisi. "On the solution of the random link matching problems". In: *Journal de Physique* 48.9 (1987), pp. 1451–1459. DOI: 10.1051/jphys:019870048090145100. URL: https://hal.archives-ouvertes.fr/jpa-00210575 (visited on 06/07/2022).

# References V

[11] Cristopher Moore and Stephan Mertens. *The Nature of Computation*. Oxford University Press, Aug. 11, 2011. ISBN: 978-0-19-923321-2. DOI: 10.1093/acprof:oso/9780199233212.001.0001. URL: https://oxford.universitypressscholarship.com/view/10.1093/acprof:oso/9780199233212.001.0001/acprof-9780199233212 (visited on 06/07/2022).

[12] James Munkres. "Algorithms for the Assignment and Transportation Problems". In: *Journal of the Society for Industrial and Applied Mathematics* 5.1 (1957). Publisher: Society for Industrial and Applied Mathematics, pp. 32–38. ISSN: 0368-4245. URL: https://www.jstor.org/stable/2098689 (visited on 06/06/2022).

# References VI

[13] Giovanni Righini. "Complements of Operations Research". Type: presentation. Aug. 2, 2018. DOI: 10.1287/fa0f35b9-6bcb-4f07-ab71-78a7c9014fb2. URL: http://pubsonline.informs.org/do/10.1287/fa0f35b9-6bcb-4f07-ab71-78a7c9014fb2/full/ (visited on 06/06/2022).

[14] Arthur F. Veinott Jr. and George B. Dantzig. "Integral Extreme Points". In: *SIAM Review* 10.3 (July 1968), pp. 371–372. ISSN: 0036-1445, 1095-7200. DOI: 10.1137/1010063. URL: http://epubs.siam.org/doi/10.1137/1010063 (visited on 06/06/2022).