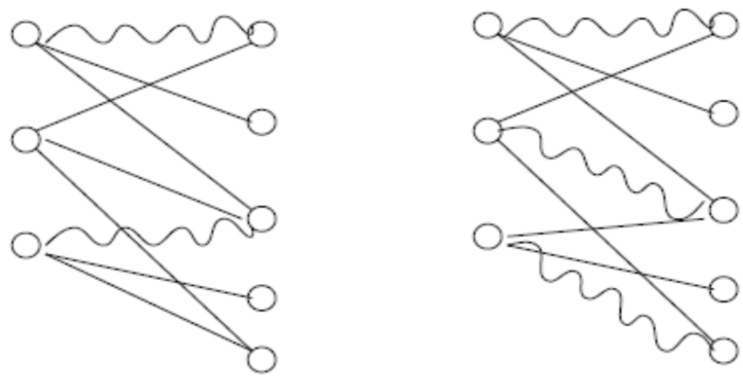
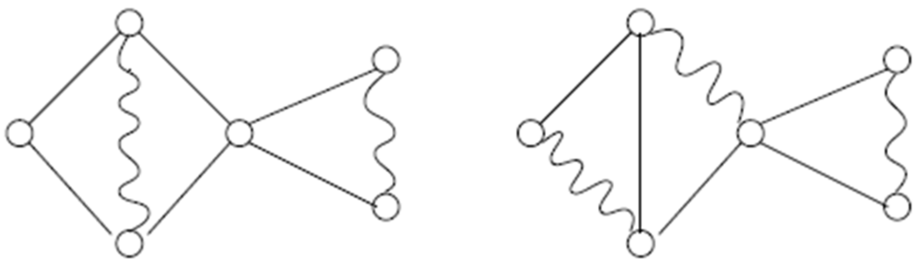
Lecture 7: Bipartite Matching

- Bipartite matching



- Non-bipartite matching



What is a Bipartite Matching?

- Let G=(N,A) be an unrestricted bipartite graph. A subset X of A is said to be a matching

  if no two arcs in X are incident to the same node.

- With respect to a given matching X, a node j is said to be *matched* or *covered* if there is an arc in X incident to j.
- If a node is not matched, it is said to be *unmatched* or *exposed*.
- A matching that leaves no nodes exposed is said to be *complete*.

How do I know a graph is bipartite?

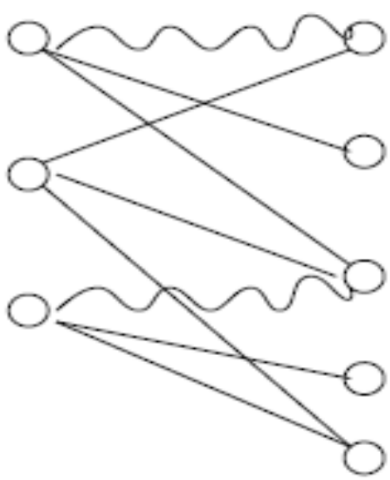
- Theorem: A undirected graph is bipartite if and only if its node-arc incidence matrix is totally unimodular.

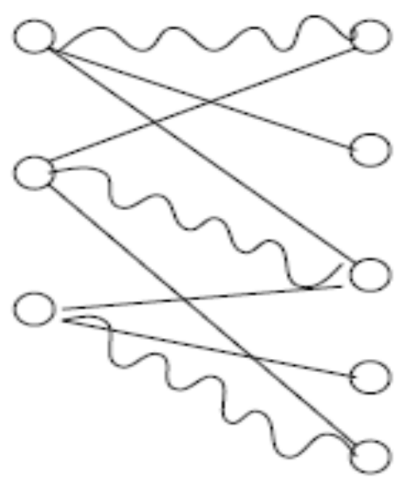
What types of problems we're interested in?

- (Maximum) cardinality matching problem
- (Maximum) weighted matching problem
- Max-min matching problem

(Maximum) Cardinality Matching Problem

- Given a bipartite graph, find a matching containing a maximum number of arcs.





Mathematical Model

maximize

$$\textstyle\sum_{i,j} \boxed{w_{ij}} x_{ij} \qquad (3.1)$$

subject to

$$\left.\begin{array}{l} \sum_j x_{ij} \le 1, \ (i = 1, 2, ..., m) \\ \sum_i x_{ij} \le 1, \ (j = 1, 2, ..., n) \\ x_{ij} \ge 0, \end{array}\right\} \qquad (3.2)$$

in which each variable x_{ij} takes on the value zero or one, regardless of the coefficients in the objective function (3.1).

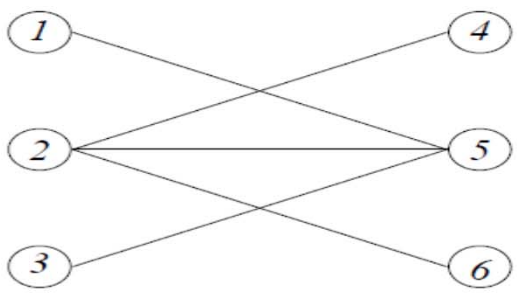
Potential Applications

- Match maker
- Roommate assignment
- Job assignment
- SDR

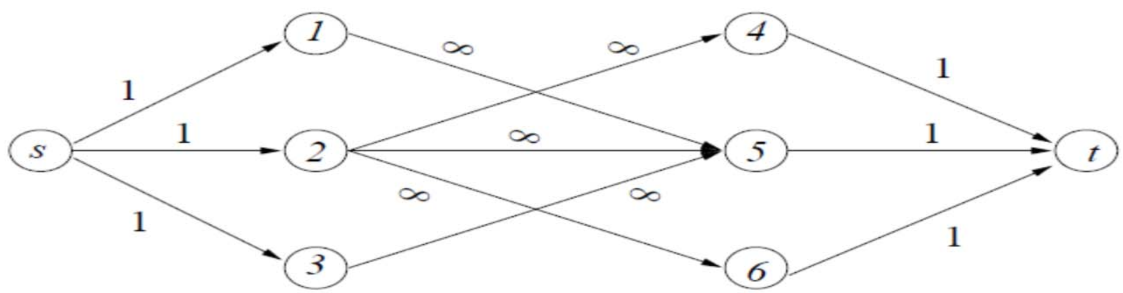
An important topic in combinatorial analysis is that of "systems of distinct representatives." Let $Q = \{q_i \ ; \ i = 1, 2, \ldots, m\}$ be a family of (not necessarily distinct) subsets of a set $E = \{e_j \ ; \ j = 1, 2, \ldots, n\}$. A set $T = \{e_{j(1)}, \ldots, e_{j(t)}\}$, $0 \le t \le n$, is called a *partial transversal* of Q if T consists of distinct elements in E and if there are distinct integers $i(1), \ldots, i(t)$, such that $e_{j(k)} \in q_{i(k)}$ for $k = 1, \ldots, t$. Such a set is called a *transversal* or a *system of distinct representatives* (SDR) of Q if $t = m$.

What's Special?

- Cardinality matching is a special case of the maximal flow problem.



Cardinality Matching



Maximal Flow Network

(Maximum) Weighted Matching Problem

- Given an arc-weighted bipartite graph, find a matching for which the sum of the weights of the arcs is maximum.

maximize

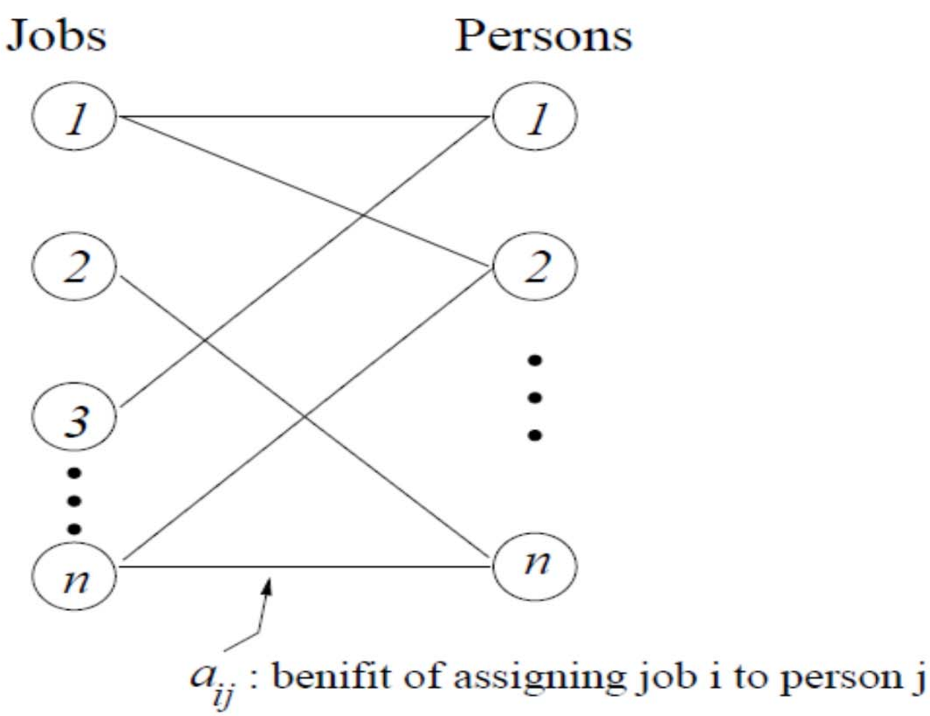
$$\textstyle\sum_{i,j} w_{ij} x_{ij} \qquad (3.1)$$

subject to

$$\left.\begin{array}{l} \sum_j x_{ij} \leq 1, \ (i = 1, 2, ..., m) \\ \sum_i x_{ij} \leq 1, \ (j = 1, 2, ..., n) \\ x_{ij} \geq 0, \end{array}\right\} \qquad (3.2)$$

Potential Applications

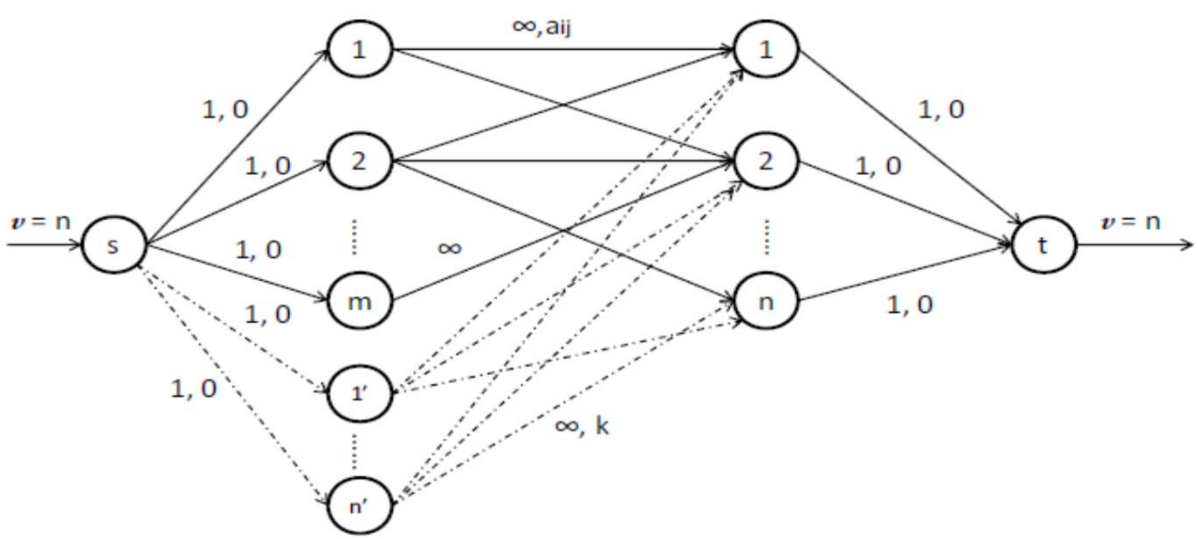
- Assignment Problem / Marriage Problem



Fact: A (maximum) weighted matching provides an optimal assignment.

What's Special?

- Weighted matching is a special case of the minimum cost flow problem.



Max-Min Matching Problem

- Given an arc-weighted bipartite graph, find a maximum-cardinality matching for which the minimum of weights of the arcs in the matching is maximum.

- Sometimes it is called the “bottleneck" problem.

Potential Applications

- "Bottleneck" matching
  - n workers assigned to n stations on a conveyorized production line.
  - w_{ij}: the rate of worker i working at station j.
  - Objective: to maximize the production rate.

Fact:

A max-min matching provides an optimal solution.

Network Flow and Bipartite Matching

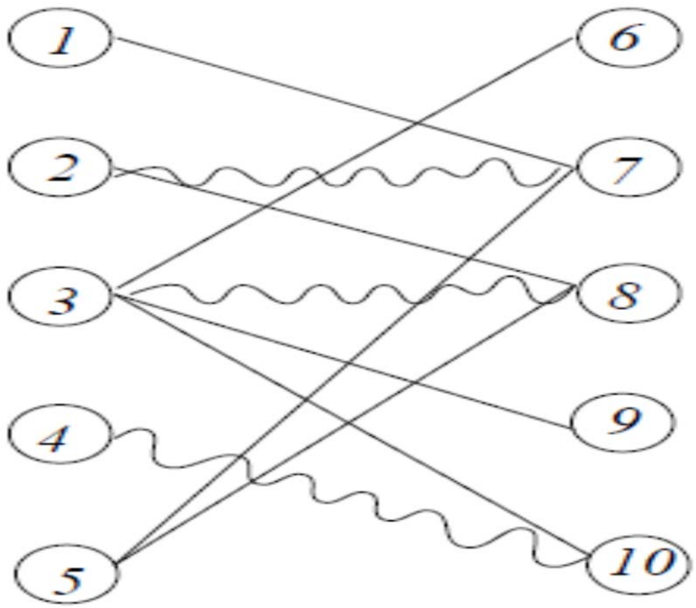
- For every cardinality matching problem on m + n nodes, there is a corresponding maximal flow problem in an (m + n + 2)-node network. Similarly, for every n x n assignment problem, there is a corresponding min-cost flow problem in a (2n + 2)-node flow network. Accordingly, there is a polynomial-bounded reduction of weighted matching problems to network flow problems and, indirectly, to the shortest path problem.

Network Flow and Bipartite Matching

- Conversely, we can also show that for every maximal flow problem there is a reduction to a cardinality matching problem, and a reduction of every min-cost flow problem to a weighted matching problem.
- Thus, network flow theory and bipartite matching theory are, for our purposes, essentially equivalent.
- We shall restate the essential theorems of network flow theory in the context of bipartite matchings.

Fundamental Concepts of Matching

- Is this bipartite matching of maximum cardinality?



- Why?

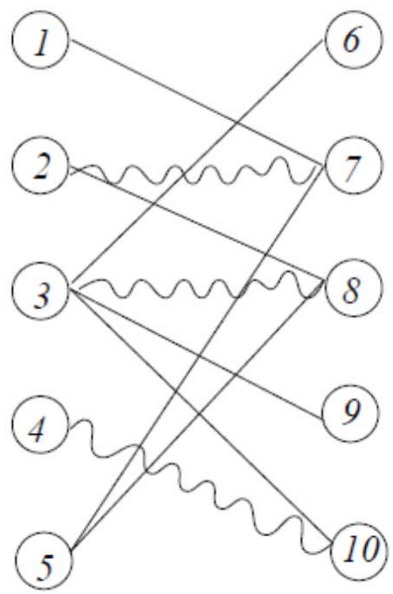
Two Basic Terminologies

- With respect to a given matching X, an alternating path is an (undirected) path of arcs which are alternately in X and not in X.

- An augmenting path is an alternating path between two exposed nodes.

Examples

- Alternating Path



$X = \{ (2, 7), (3, 8), (4, 10) \}$

$AP_1 = 1 - 7 - 2 - 8 - 3 - 10 - 4$

$AP_2 = 2 - 7 - 5$

$AP_3 = 7 - 2 - 8 - 3$

$AP_4 = 1 - 7 - 2 - 8 - 3 - 9$

$\vdots$

- Augmenting Path

$AP_4 = 1 - 7 - 2 - 8 - 3 - 9$

$AP_5 = 6 - 3 - 8 - 5$

$\vdots$

Optimality Condition

- Is X = {(2; 7); (3; 8); (4; 10)} a max-cardinality matching?

- Why?

- Augmenting Path Theorem (C. Berge 1957)

  “A matching X is of maximum cardinality if and only if it admits no augmenting path."

Proof

(only if part) If X admits an augmenting path, clearly X is not of maximum cardinality.

(if part) (Jack Edmond 1964)
If X is not of maximum cardinality, let $\bar{X}$ be a matching s.t. $|\bar{X}| > |X|$.

Consider the set of arcs

$$\bar{A} = (X - \bar{X}) \cup (\bar{X} - X).$$

Clearly, any node of G is incident to at most two arcs of $\bar{A}$, and if it is indeed incident to two, it must be incident to exactly one arc of X and one of $\bar{X}$ in $\bar{A}$.

Proof

Let H be a subgraph of G made up of $\bar{A}$ and all incident nodes. Since $|\bar{X}| > |X|$, H must have some components, say P, with more $\bar{X}$ than X arcs.

Now, since every node of P has degree ≤ 2, P must be a polygon (cycle) or a path. If P is a polygon, then since each of its nodes is incident to one arc of X and one of $\bar{X}$. So P has an equal number of arcs of each type.

This is a contradiction.

It follows that P is a path. But any internal nodes of P is incident to one arc of X and one of $\bar{X}$. Therefor, the terminal arcs of P are both in $\bar{X}$. Hence P is an augmenting path w.r.t. X.

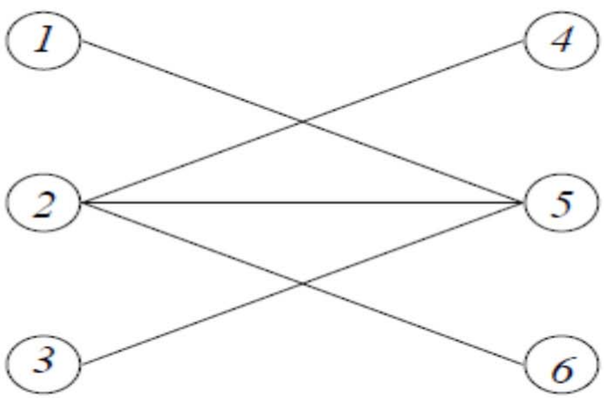
Dual of Max-Cardinality Matching

- Max-cardinality bipartite matching is a max-flow problem
- Dual of max-flow is min-cut
- What’s the dual of max-cardinality?

Basic Terminology

Definition: Let $G = (N, A)$ be an (undirected) graph. $C \subseteq N$ is said to *cover* A, if each arc in A is incident to at least one node in C.

How is covering related to matching?



$$C_1 = \{ 1, 2, 3 \}$$

$$C_2 = \{ 4, 5, 6 \}$$

$$C_3 = \{ 2, 5 \}$$

$$\vdots$$

max-cardinality matching

min-cardinality covering

Main Theorem

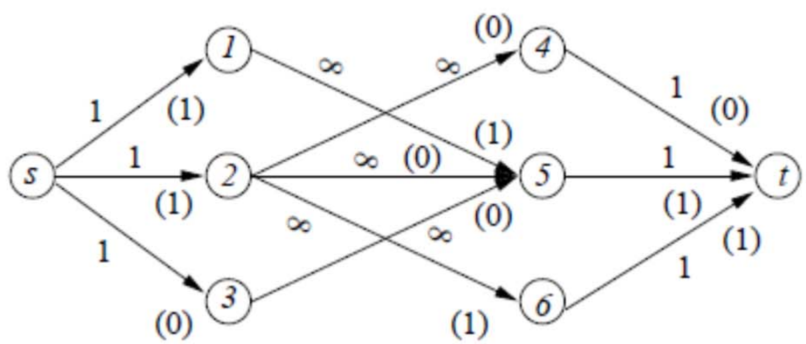
König-Egervary (Duality) Theorem

For any bipartite graph, the maximum # of arcs in a matching is equal to the minimum # of nodes in a covering of arcs by nodes.

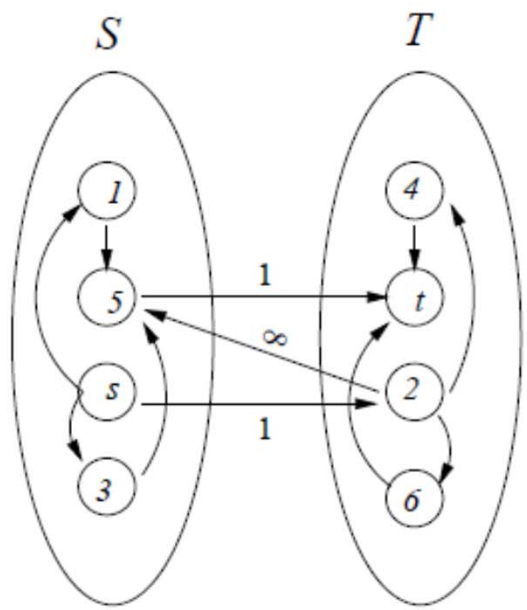
Proof

- Max-Flow Min-Cut Theorem

Example:



(max-flow)



(min-cut)

How to Find a Max-Cardinality Matching?

- Cardinality Matching Algorithm

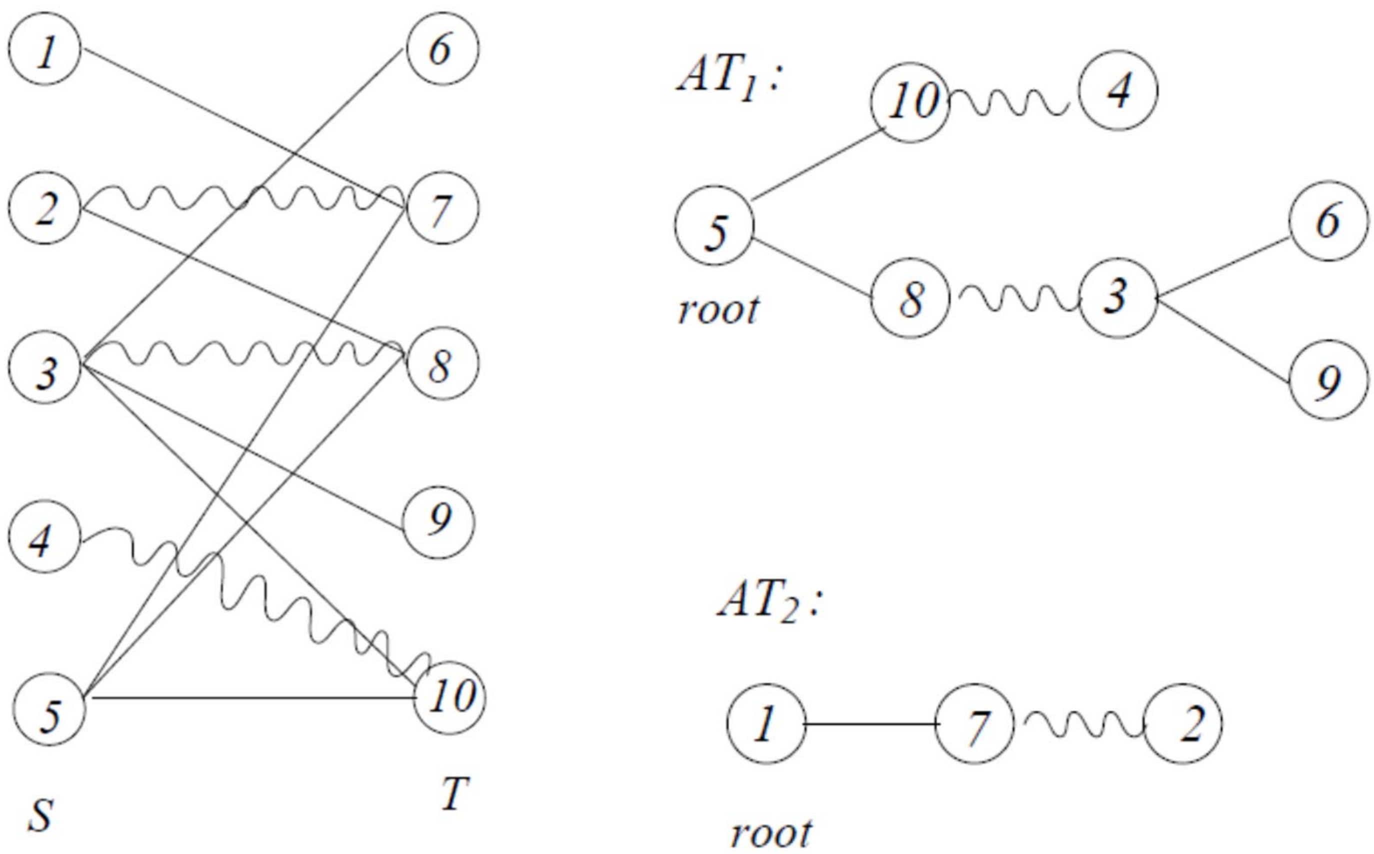
- Based on the concept of alternating tree

Alternating Tree

- For a given bipartite graph G = (S,T,A) and a given matching X in A, we define an alternating tree relative to the matching to be a tree which satisfies the following two conditions.
- First, the tree contains exactly one exposed node from S, which we call its root.
- Second, all paths between the root and any other node in the tree are alternating paths.

Basic Ideas

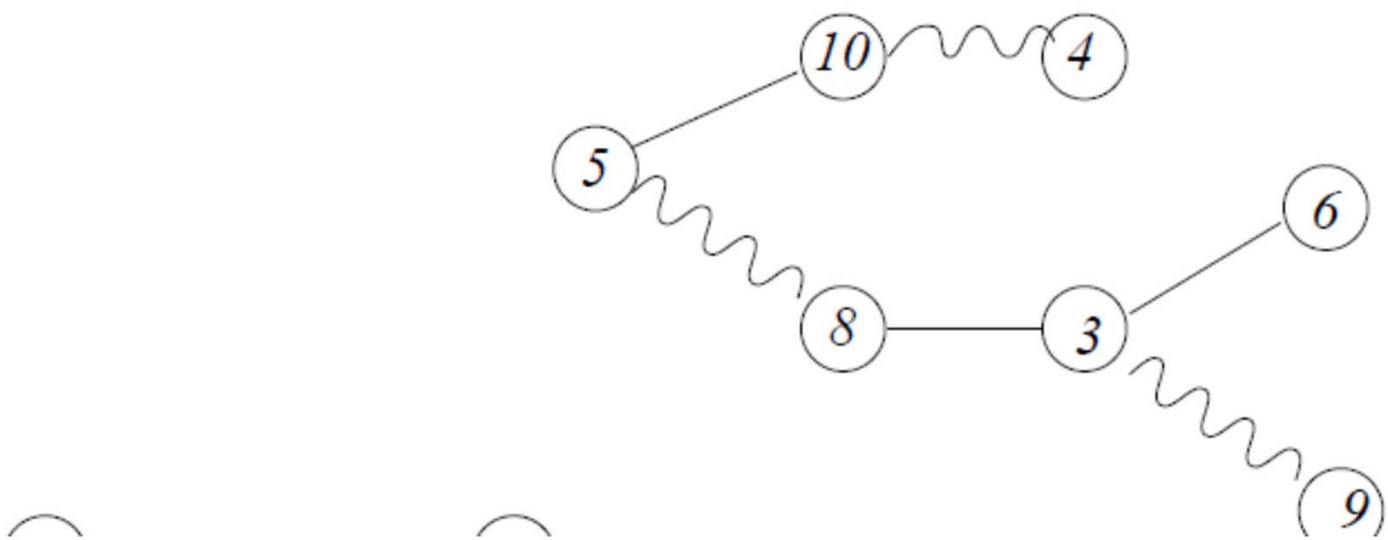
Example:



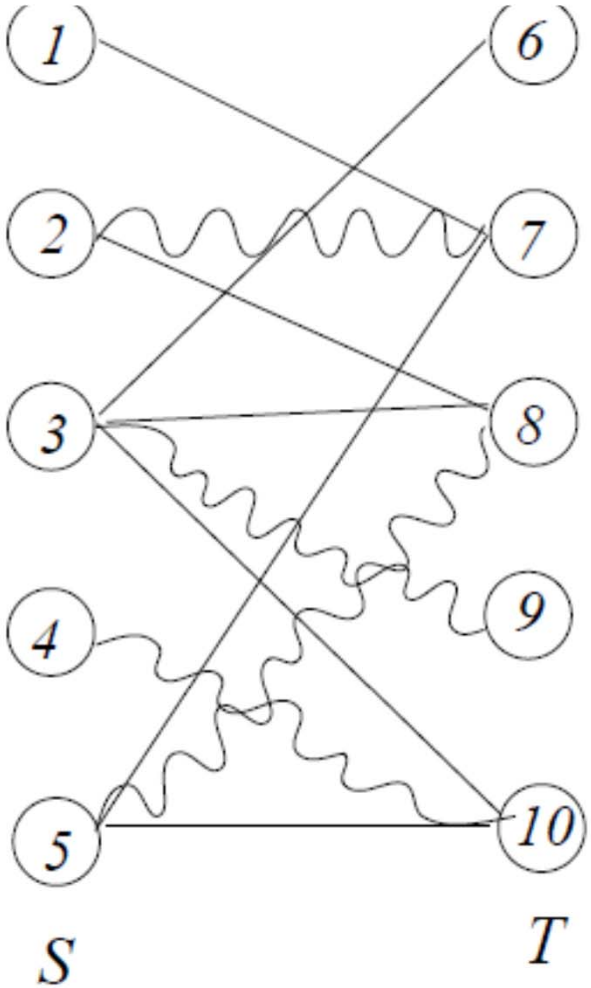
Basic Ideas

- An altrenating tree ends with an exposed node in T (like AT_1) has an augmenting path for the matching to be augmented.

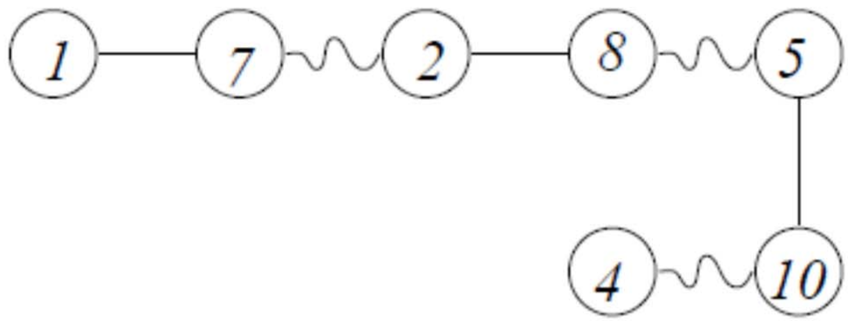
Example:



Basic Ideas



AT_3:



$\{\,3;\ \ 7,\ 8,\ 10\,\}$

Basic Ideas

- For a matching with some alternating trees (like AT_3) such that no more nodes and arcs can be added, the trees are said to be Hungarian.

- Hungarian trees can be used to construct a min-cardinality covering consisting of all out-of-tree nodes in S and all in-tree nodes in T.

Example:

$$C = \{3, 7, 8, 10\}$$

The current matching is a max-cardinality matching.

Outline of the Algorithm

- Begin with any matching, possibly empty matching.
- Each exposed node in S is made the root of an alternating tree.
- Nodes and arcs are added to the trees by a labeling technique.
- Eventually, either (i) an exposed node in T is added to one of the trees; or (ii) it is impossible to add more nodes and arcs to any of the trees.
- If (i) happens, augment the current matching and repeat the adding step. If (ii) occurs, the current matching is of maximum cardinality.

Bipartite Cardinality Matching Algorithm

Step 0 (*Start*) The bipartite graph $G = (S, T, A)$ is given. Let X be any matching, possibly the empty matching. No nodes are labeled.

Step 1 (*Labeling*)

(1.0) Give the label "$\emptyset$" to each exposed node in S.

(1.1) If there are no unscanned labels, go to Step 3. Otherwise, find a node i with an unscanned label. If $i \in S$, go to Step 1.2; if $i \in T$, go to Step 1.3.

(1.2) Scan the label on node i ($i \in S$) as follows. For each arc $(i,j) \notin X$ incident to node i, give node j the label "i," unless node j is already labeled. Return to Step 1.1.

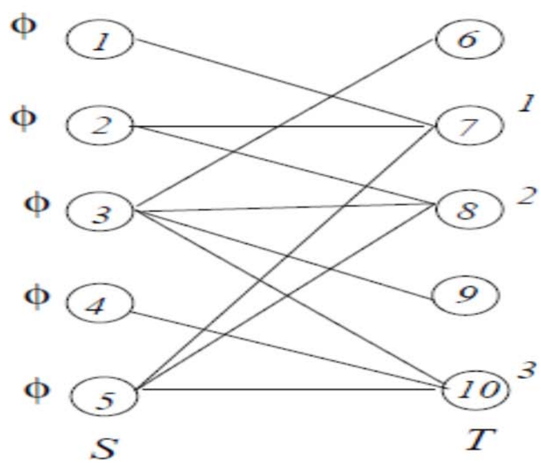
(1.3) Scan the label on node i ($i \in T$) as follows. If node i is exposed, go to Step 2. Otherwise, identify the unique arc $(i,j) \in X$ incident to node i and give node j the label "i." Return to Step 1.1.

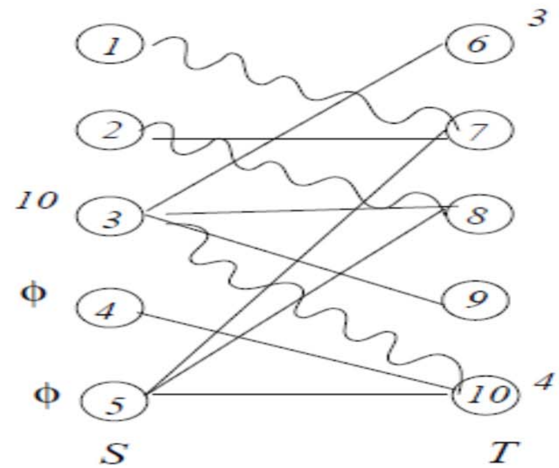
Bipartite Cardinality Matching Algorithm

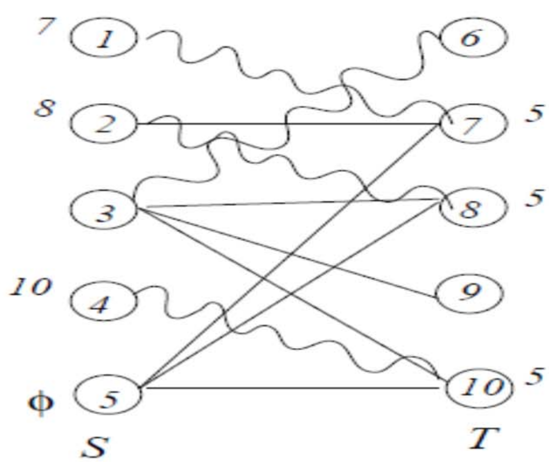
Step 2 (*Augmentation*) An augmenting path has been found, terminating at node i (identified in Step 1.3). The nodes preceding node i in the path are identified by "backtracing." That is, if the label on node i is "j," the second-to-last node in the path is j. If the label on node j is "k," the third-to-last node is k, and so on. The initial node in the path has the label "$\oslash$." Augment X by adding to X all arcs in the augmenting path that are not in X and removing from X those which are. Remove all labels from nodes. Return to Step 1.0.

Step 3 (*Hungarian Labeling*) The labeling is Hungarian, no augmenting path exists, and the matching X is of maximum cardinality. Let $L \subseteq S \cup T$ denote the set of labeled nodes. Then $C = (S - L) \cup (T \cap L)$ is a minimum cardinality covering of arcs by nodes, dual to X. Halt.//

Example







$C = \{ 3, 7, 8, 10 \}$

$X = \{ (1, 7), (2, 8), (3, 6), (4, 10) \}$

Complexity

- Let |S| = *m* and |T| = *n* with *m* < *n.*

- It is not hard to see that the algorithm can be implemented with a complexity of $O(m^2 n)$.

Related Results

Theorem 4.1 (*Mendelsohn-Dulmage*) Let $G = (S,\ T,\ A)$ be a bipartite graph and let X_1, X_2 be two matchings in G. Then there exists a matching $X \subseteq X_1 \cup X_2$, such that X covers all the nodes of S covered by X_1 and all the nodes of T covered by X_2.

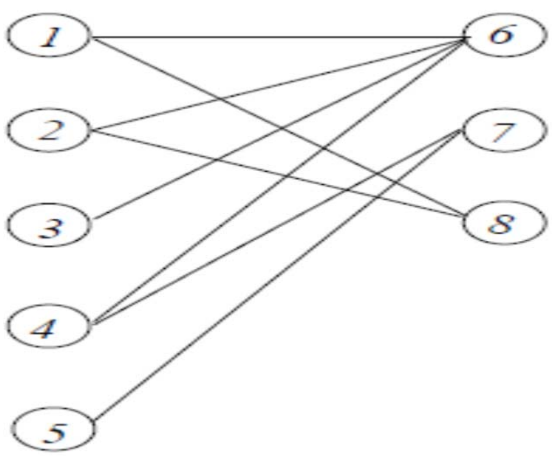
The following theorem and corollary follow directly from Theorem 4.1.

Theorem 4.2 Let X be any matching in $G = (S, T, A)$. Then there exists a maximum cardinality matching X^* which covers all the nodes of G covered by X.

Corollary 4.3 For any nonisolated node i (degree greater than zero), there exists a maximum cardinality matching which covers i.

Related Results

- The cardinality matching problem is particularly easy to solve for a special type of graph which F. Glover calls "convex."
- A bipartite graph G = (S,T,A) is said to be convex if it has the property that if (i,j) and (k,j) are arcs, where $i < k$, then $(i + 1, j), (i + 2, j),\ldots,(k - 1, j)$ are also arcs.



Cardinality Matching of Convex Graph (a tie breaker is occasionally needed)

The cardinality matching problem can be solved by the following procedure. For each node $j \in T$, let

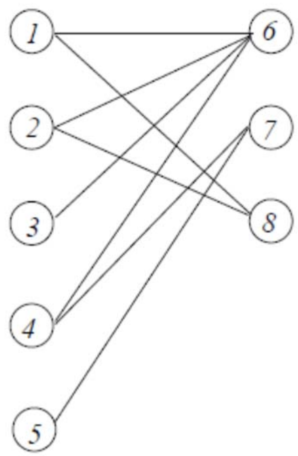
$$\pi_j = \max\{i \mid (i, j) \in A\}$$

.

Start with the empty matching and iterate over $i = 1,2,\ldots,m$. If there are any arcs (i,j), where j is an exposed node, add to the matching the arc (i,j) for which π_j is as small as possible.

- The complexity of the Glover's (1967) procedure is $O(mn)$ where $|S| = m$ and $|T| = n$.

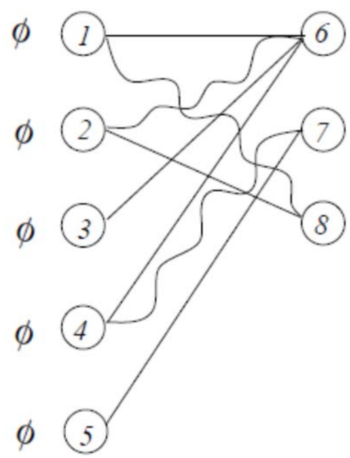
Example



$\pi_6 = \max\{1,2,3,4\} = 4$

$\pi_7 = \max\{4,5\} = 5$

$\pi_8 = \max\{1,2\} = 2$



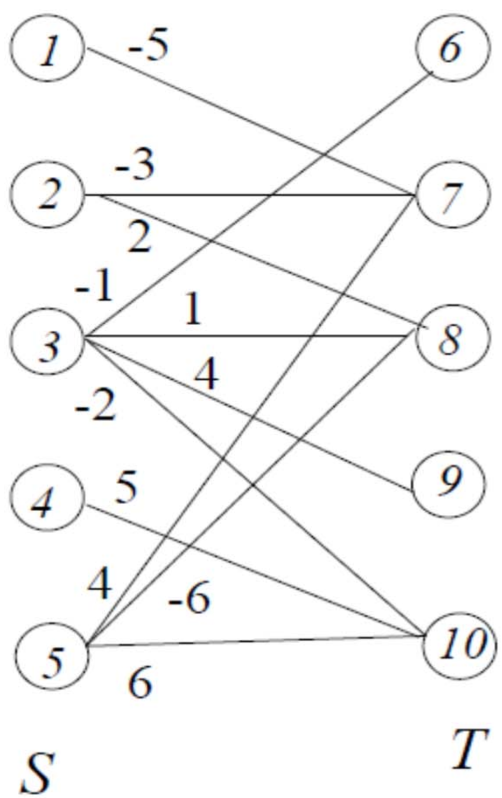
$\pi_6 = 4$

$\pi_7 = 5$

$\pi_8 = 2$

Max-Min Matching – Bottleneck Matching

- This problem calls for the computation of a maximum cardinality matching for which the minimum arc weight is maximum.
- Basic Idea



Cardinality	max-min
0	∞
1	6
2	4
3	4
4	2

Max-Min Matching

Let X_k denote any matching containing k arcs. Let H_{k-1} denote any subgraph obtained from G by deleting k - 1 nodes.

Theorem 7.1 (*Gross*) For any bipartite graph G,

$$\max_{x_k} \min\{w_{ij} \mid (i,j) \in X_k\} = \min_{H_{k-1}} \max\{w_{ij} \mid (i,j) \in H_{k-1}\}.$$

PROOF Let X_k^* be max-min optimal, with respect to matchings with k arcs. Let $(p, q) \in X_k$ be such that

$$w_{pq} = \min \{w_{ij} | (i, j) \in X_k^*\},$$

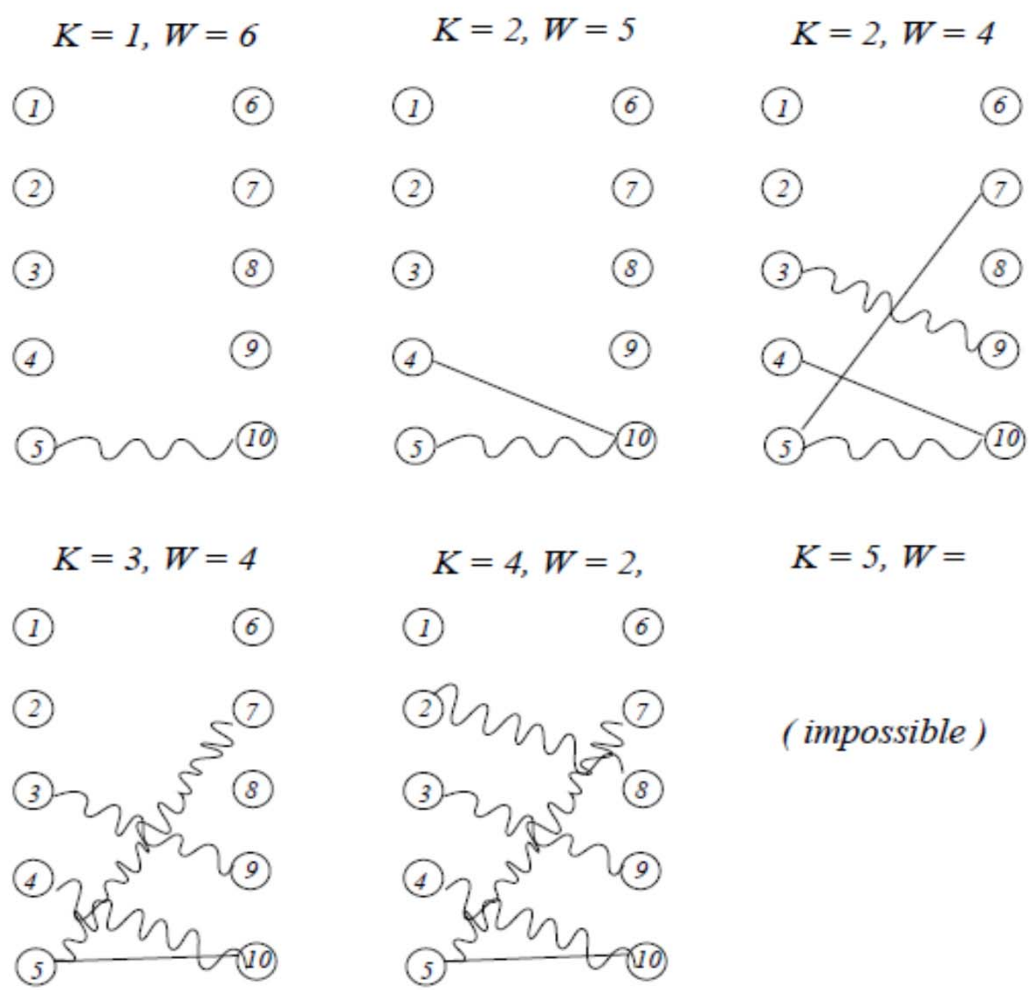
where the weights of the arcs are assumed to be distinct. Let G_{k-1}^* contain all arcs (i, j) such that $w_{ij} > w_{pq}$. Clearly a maximum cardinality matching in G_{k-1}^* contains at most $k - 1$ arcs, and G_{k-1}^* can be covered by an odd-set cover with capacity $k - 1$. Appropriate contraction and deletion operations with respect to this odd-set cover of G_{k-1}^* yields an H_{k-1} such that

$$w_{pq} = \max \{w_{ij} | (i, j) \in H_{k-1}\}. //$$

Outline of Max-Min Matching Algorithm

Step1:	Start with the empty matching and a suitably large "treshold" $W = \max\{w_{ij}\}$. Set $k = 0$.
Step2:	Set $k \leftarrow k + 1$.
Step3:	Find an augmenting path in the subgraph containing all arc (i, j) for which $w_{ij} \geq W$. If augmantation is possible, a max-min matching of cardinality k is obtained. Go to Step 2.
Step4	Reduce the threshold W just enough to permit augmentation to occur and go to Step 3. If not possible to lower W any more, STOP.

Example



Threshold Method for Max-Min Matching

Step 0 (*Start*) The bipartite graph $G = (S, T, A)$ and a weight w_{ij} for each arc $(i, j) \in A$ are given. Set $X = \emptyset, W = +\infty$, and $\pi_j = -\infty$ for each node $j \in T$. No nodes are labeled.

Step 1 (*Labeling*)

(1.0) Give the label "$\emptyset$" to each exposed node in S.

(1.1) If there are no unscanned labels, go to Step 3. If there are unscanned labels, but each unscanned label is on a node i in T for which $\pi_i < W$, then set $W = \max\{\pi_i \mid \pi_i < W\}$.

(1.2) Find a node i with an unscanned label, where either $i \in S$ or else $i \in T$ and $\pi_i \geq W$. If $i \in S$, go to Step 1.3; if $i \in T$, go to Step 1.4.

Threshold Method

(1.3) Scan the label on node $i (i \in S)$ as follows. For each arc $(i, j) \notin X$ incident to i, if $\pi_j < w_{ij}$ and $\pi_i < W$, then give node j the label "i" (replacing any existing label) and set $\pi_j = w_{ij}$. Return to Step 1.1.

(1.4) Scan the label on node $i (i \in T)$ as follows. If node i is exposed, go to Step 2. Otherwise, identify the unique arc $(i, j) \in X$ incident to node i and give node j the label "i." Return to Step 1.1.

Threshold Method

Step 2 (*Augmentation*) An augmenting path has been found, terminating at node i (identified in Step 1.4). The nodes preceding node i in the path are identified by "backtracing" from label to label. Augment X by adding to X all arcs in the augmenting path that are not in X, and removing from X those which are. Remove all labels from nodes. Set $\pi_j = -\infty$, for each node j in T. Return to Step 1.0.

Step 3 (*Hungarian Labeling*) No augmenting path exists, and the matching X is a max-min matching of maximum cardinality. Let $L \subseteq S \cup T$ denote the set of labeled nodes. Let $(i', j') \in X$ be such that

$$w_{i',j'} = \min \{w_{ij} \mid (i,j) \in X\}.$$

The subgraph obtained by deleting the nodes in $(S - L) \cup (T \cap L) - \{i', j'\}$ is a min-max solution dual to X. Halt. //

Complexity

- Complexity: $|S| = m,\ |T| = n.$
  - At most consider mn values for W.
  - For each value, the augmentation takes $O(mn)$ computations.
  - Total complexity $= O(m^2n^2)$.
- A careful labeling technique results in an implementation of $O(m^2n)$.

Max Weighted Bipartite Matching

- Given a bipartite graph $G(S, T; A)$ with weight w_{ij} on arc (i, j).

$$
\begin{array}{lll}
 & \max \sum_{i\in S} \sum_{j\in T} w_{ij} x_{ij} & \\
 & \text{s. t.} & \\
(\mathrm{P}) & \sum_{j\in T} x_{ij} \le 1, & \forall\ i \in S \\
 & \sum_{i\in S} x_{ij} \le 1, & \forall\ j \in T \\
 & x_{ij} \ge 0, & \forall\ i \in S, j \in T
\end{array}
$$

- Hungarian Algorithm
  - H. Kuhn (1955) in name of the Hungarian mathematician Egevary.
  - A primal-dual algorithm.

Optimality Conditions

The dual linear programming problem is:

$$
\begin{aligned}
&\text{minimize} \quad \textstyle\sum_i u_i + \sum_j v_j \\
&\text{subject to} \\
&\qquad u_i + v_j \geq w_{ij}, \\
&\qquad u_i \geq 0, \\
&\qquad v_j \geq 0.
\end{aligned}
$$

- Complementary Slackness Theorem

(a) If $x_{ij} > 0$ $(= 1)$, then $u_i + v_j = w_{ij}$.

(b) If $u_i > 0$, then $\sum_j x_{ij} = 1$. *(node i is covered)*

(c) If $v_j > 0$, then $\sum_i x_{ij} = 1$. *(node j is covered)*

Basic Idea of the Hungarian Method

- The Hungarian method maintains primal and dual feasibility at all times, and in addition maintains satisfaction of all orthogonality (complementary slackness) conditions, except conditions (b).

- The number of such unsatisfied conditions is decreased monotonically during the course of the computation.

Basic Approach

- Start with a primal feasible solution and a dual feasible solution such that conditions (a) and (c) are met.
- If condition (b) is also met, then the problem is solved. *(optimality check).*
- Otherwise, we attempt to find an augmenting path with the subgraph formed by the arcs for which $u_i + v_j = w_{ij}$.
- If such a path found, then the new matching will be feasible; while the dual solution remains the same, and conditions (a) and (c) remain valid. Moreover, few conditions (b) will be violated. (change x_{ij})
- If no such path can be found, the dual variables are adjusted so that at least one additional arc can be added to the subgraph in the next iteration. (change u_i, v_j)

Initial Conditions

$$x_{ij} = 0, \ \forall \ i, \ j$$

$$u_i = \max_j \{w_{ij}, 0\}, \ \forall \ i \in S$$

$$v_j = 0, \ \forall \ j \in T$$

Example

- Consider a fully connected bipartite graph

$$
\begin{array}{cc|cccc|c}
 & & & T & & & \\
 & w_{ij} & a & b & c & d & u_i \\
\hline
 & 1 & \underline{32} & 18 & \underline{32} & 26 & 32 \\
 & 2 & \underline{22} & \underline{24} & 12 & 16 & 22 \\
S & 3 & 24 & \underline{30} & 26 & 24 & 28 \\
 & 4 & 26 & \underline{30} & \underline{28} & 20 & 28 \\
\hline
 & v_j & 0 & 2 & 0 & 0 &
\end{array}
$$

<u>Primal Solution</u>: $x_{2a} = x_{4b} = 1$

Other $x_{ij} = 0$

<u>Dual Solution</u>: $u_1 = 32, u_2 = 22, u_3 = 28, u_4 = 28$

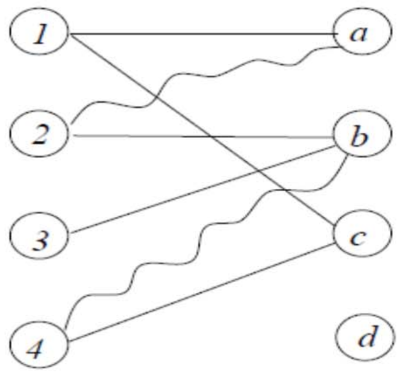
$v_a = 0, v_b = 2, v_c = 0, v_d = 0$

<u>Conditions</u>: (a) holds, (c) holds,

(b) violated for u_1 and u_3.

Example

- Find augmenting path with $u_i + v_j = w_{ij}$.



AP: 

Primal Solution: $x_{2a} = x_{3b} = x_{4c} = 1$

Other $x_{ij} = 0$

Dual Solution: Same as before

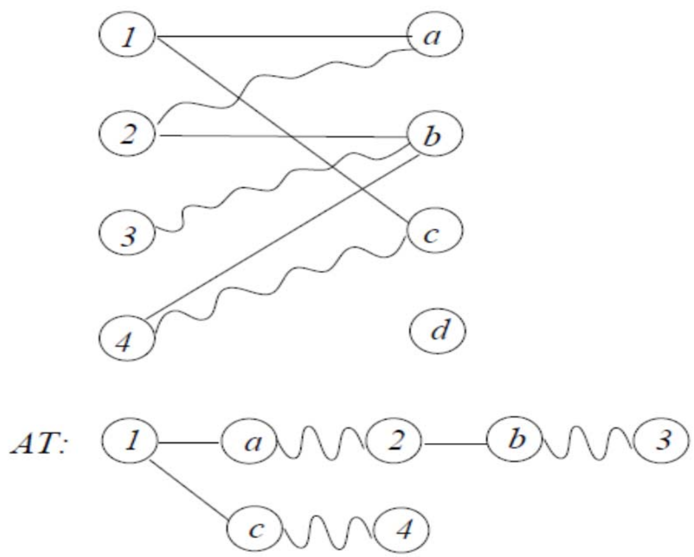
Conditions: (a) remains valid (why?)

(c) remains valid (why?)

(b) violated for u_1 only (why?)

Example

- Solution becomes



Hungarian !

- Change dual variables to add additional arcs.

Say,
$$u_i \longleftarrow u_i - 4, \quad \forall i = 1,2,3,4$$
$$v_j \longleftarrow v_j + 4, \quad \forall j = a,b,c$$

Example

	a	b	c	d	u_i
1	$\underline{32}$	18	$\underline{32}$	26	28
2	$\underline{22}$	$\underline{24}$	12	16	18
3	24	$\underline{30}$	26	$\underline{24}$	24
4	26	$\underline{30}$	$\underline{28}$	20	24
v_j	4	6	4	0	

<u>Primal Solution</u>: Same as before

<u>Dual Solution</u>: $u_1 = 28, u_2 = 18, u_3 = 24, u_4 = 24$

$v_a = 4, v_b = 6, v_c = 4, v_d = 0$

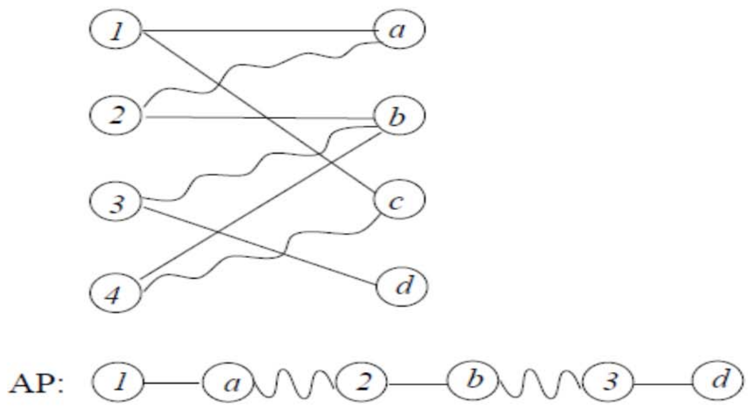
<u>Conditions</u>: (a) remains valid (why?)

(c) remains valid (why?)

(b) Not worse (why?)

A new arc $(3, d)$ is added!

Example



Primal Solution:	$x_{1a} = x_{2b} = x_{3d} = x_{4c} = 1$
	Other $x_{ij} = 0$
Dual Solution:	$u_1 = 28, u_2 = 18, u_3 = 24, u_4 = 24$
	$v_a = 4, v_b = 6, v_c = 4, v_d = 0$
Conditions:	(a), (c) remains valid
	(b) nodes 1, 2, 3, 4 all covered
$\Longrightarrow$	Optimality !
	Optimal value = 32+24+24+28 = 108

Bipartite Weighted Matching Algorithm

Step 0 (*Start*) The bipartite graph $G = (S,T,A)$ and a weight w_{ij} for each arc $(i,j) \in A$ are given. Set $X = \emptyset$. Set $u_i = \max\{w_{ij}\}$ for each node $i \in S$. Set $v_j = 0$ and $\pi_j = +\infty$ for each node $j \in T$. No nodes are labeled.

Step 1 (*Labeling*)

(1.0) Give the label "$\emptyset$" to each exposed node in S.

(1.1) If there are no unscanned labels, or if there are unscanned labels, but each unscanned label is on a node i in T for which $\pi_i > 0$, then go to Step 3.

(1.2) Find a node i with an unscanned label, where either $i \in S$ or else $i \in T$ and $\pi_i = 0$. If $i \in S$, go to Step 1.3; if $i \in T$, go to Step 1.4.

(1.3) Scan the label on node $i (i \in S)$ as follows. For each arc $(i,j) \notin X$ incident to node i, if $u_i + v_j - w_{ij} < \pi_j$, then give node j the label "i" (replacing any existing label) and set $\pi_j = u_i + v_j - w_{ij}$. Return to Step 1.1.

(1.4) Scan the label on node $i (i \in T)$ as follows. If node i is exposed, go to Step 2. Otherwise, identify the unique arc $(i,j) \in X$ incident to node i and give node j the label "i."
Return to Step 1.1.

Bipartite Weighted Matching Algorithm

Step 2 (*Augmentation*) An augmenting path has been found, terminating at node i (identified in Step 1.4). The nodes preceding node i in the path are identified by "backtracing" from label to label. Augment X by adding to X all arcs in the augmenting path that are not in X, and removing from X those which are. Set $\pi_j = +\infty$, for each node j in T. Remove all labels from nodes. Return to Step 1.0.

Step 3 (*Change in Dual Variables*) Find

$$\delta_1 = \min\{u_i \mid i \in S\},$$
$$\delta_2 = \min\{\pi_j \mid \pi_j > 0, j \in T\},$$
$$\delta = \min\{\delta_1, \delta_2\}.$$

Subtract δ from u_i, for each labeled node $i \in S$. Add δ to v_i for each $j \in T$ with $\pi_j = 0$. Subtract δ from π_j for each labeled node $j \in T$ with $\pi_j > 0$. If $\delta < \delta_1$ go to Step 1.1. Otherwise, X is a maximum weight matching and the u_i and v_j variables are an optimal dual solution. Halt. //

Example

	a	b	c	d	u_i
1	<u>32</u>	18	<u>32</u>	26	32
2	22	<u>24</u>	12	16	24
3	24	<u>30</u>	26	24	30
4	26	<u>30</u>	28	20	30
v_j	0	0	0	0	

<u>Initial Solution</u>:

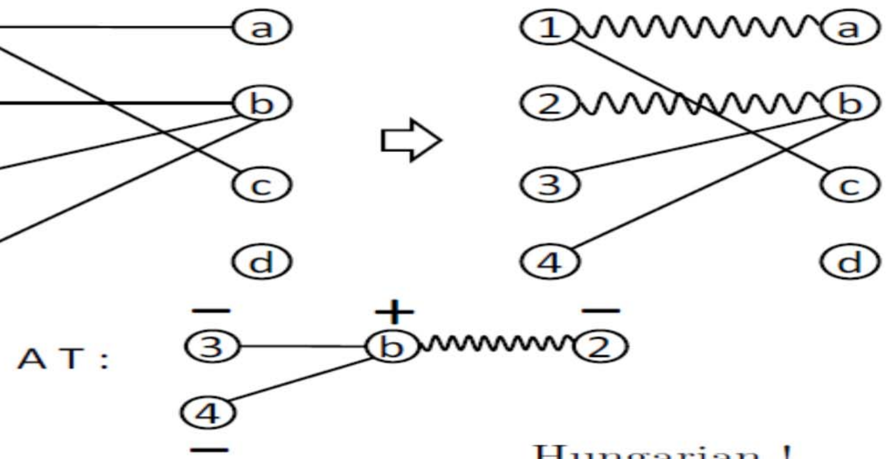
$$\begin{cases} x_{ij} = 0 & & \forall i, j & \\ u_1 = 32, & u_2 = 24, & u_3 = 30, & u_4 = 30 \\ v_a = 0, & v_b = 0, & v_c = 0, & v_d = 0 \end{cases}$$

Condition (a) holds

Condition (c) holds

Condition (b) violated at nodes 1, 2, 3, 4

Example



Hungarian !

	a	b	c	d	u_i	
1	~~32~~	~~18~~	~~32~~	~~26~~	32	
2	22	24	12	16	24	−
3	24	30	26	24	30	−
4	26	30	28	20	30	−
v_j	0	0 +	0	0		

$$\sigma_1 = \min\ \{24, 30, 30\} = 24$$
$$\sigma_2 = \min\ \{2, 2, 6\} = 2$$
$$\sigma = \min\ \{\sigma_1, \sigma_2\} = 2$$

Example

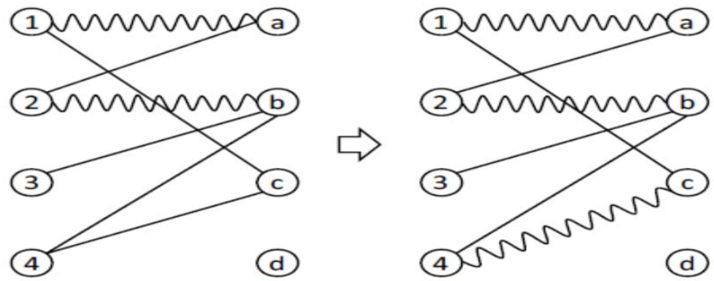
	a	b	c	d	u_i
1	$\underline{32}$	18	$\underline{32}$	26	32
2	$\underline{22}$	$\underline{24}$	12	16	22
3	24	$\underline{30}$	26	24	28
4	26	$\underline{30}$	$\underline{28}$	20	28
v_j	0	2	0	0	

Current Solution:

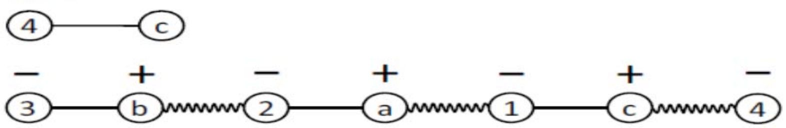
$$\begin{cases} x_{1a} = x_{2b} = 1 \\ u_1 = 32, u_2 = 22, u_3 = 28, u_4 = 28 \\ v_a = 0, v_b = 2, v_c = 0, v_d = 0 \end{cases}$$

Condition (a) holds
Condition (c) holds
Condition (b) violated at nodes 3, 4

Example



A P :



Hungarian !

	a	b	c	d	u_i	
1	32	18	32	26	32	−
2	22	24	12	16	22	−
3	24	30	26	24	28	−
4	26	30	28	20	28	−
v_j	0	2	0	0		
	+	+	+			

$$\sigma_1 = \min \{32, 22, 28, 28\} = 22$$

$$\sigma_2 = \min \{4\} = 4$$

$$\sigma = \min \{22, 4\} = 4$$

Example

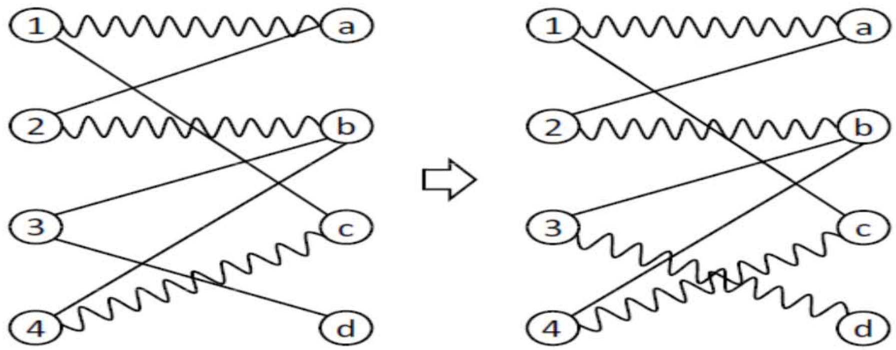
	a	b	c	d	u_i
1	$\underline{32}$	18	$\underline{32}$	26	28
2	$\underline{22}$	$\underline{24}$	12	16	18
3	24	$\underline{30}$	26	$\underline{24}$	24
4	26	$\underline{30}$	$\underline{28}$	20	24
v_j	4	6	4	0	

<u>Current Solution</u>:

$$\begin{cases} x_{1a} = x_{2b} = x_{4c} = 1 \\ u_1 = 28, u_2 = 18, u_3 = 24, u_4 = 24 \\ v_a = 4, v_b = 6, v_c = 4, v_d = 0 \end{cases}$$

Condition (a) holds
Condition (c) holds
Condition (b) violated at node 3

Example



A P : 

Hungarian ?

$$\underline{\text{Current Solution}}: \begin{cases} x_{1a} = x_{2b} = x_{3d} = x_{4c} = 1 \\ u_1 = 28, u_2 = 18, u_3 = 24, u_4 = 24 \\ v_a = 4, v_b = 6, v_c = 4, v_d = 0 \end{cases}$$

Condition (a) holds
Condition (c) holds
Condition (b) holds

$\Rightarrow$ Optimal solution!
Optimal value $= 32 + 24 + 24 + 28 = 108.$

Complexity

- Let |S| = *m* and |T| = *n* with *m* < *n.*

- It is not hard to see that the algorithm can be implemented with a complexity of $O(m^2 n)$.

Related Topics

- Gale-Shapley Matching: D. Gale and L. S Shapley have proposed a novel optimization criterion for matching which does not depend in any way on arc weights.

- Definition: A complete matching of men and women is said to be unstable if under it there are a man and a woman who are not married to each other but prefer each other to their assigned mates.
- Definition: A stable matching of men and women is said to be (man) optimal if every man is at least as well off under it as under any other stable matching.

Applications

- Match high school graduates with colleges
- Match NBA players with professional teams
- E-commerce two-way bidding
- Any matching with preference on both sides

Gale-Shapley Theorem

- Theorem 10.1: For any set of rankings, there exists a (man) optimal matching of men and women.

	A	B	C
alpha	1,3	2,2	3,1
beta	3,1	1,3	2,2
gamma	2,2	3,1	1,3

Man-Optimal Matching Algorithm

To start, let each boy propose to his favorite girl. Each girl who receives more than one proposal rejects all but her favorite from among those who have proposed to her. However, she does not accept him yet, but keeps him on a string to allow for the possibility that someone better may come along later.

We are now ready for the second stage. Those boys who are rejected now propose to their second choice. Each girl receiving proposals chooses her favorite from the group consisting of the new proposees and the boy on her string, if any. She rejects all the rest and again keeps the favorite in suspense.

We proceed in the same manner. Those who are rejected at the second stage propose to their second choices, and the girls again reject all but the best proposal they have had so far.

Eventually (in fact in at most $n^2 - 2n + 2$ stages) each girl will have received a proposal, for as long as any girl has not been proposed to there will be rejections and new proposals, but since no boy can propose to the same girl more than once, every girl is sure to get a proposal in due time. As soon as the last girl gets her proposal, the 'courtship' is declared over, and each girl is now required to accept the boy on her string.

Proof

- (Stable Matching)

Suppose that John and Mary are not married to each other, but John prefers Mary to his wife. Then John must have proposed to Mary before and get rejected. But Mary only keeps those she preferred on her list until she decides her husband. Hence Mary prefers her husband to John and there is no instability.

Proof

- (Optimal Matching)

We call a woman “possible” for a man if there is a stable matching that marries him to her. The proof is by induction.

Assume that up to a given point in the procedure no man has yet been rejected by a woman that is possible for him. Now, suppose a woman A, having received a proposal from a man *beta* she prefers, reject the man *alpha*. We have to show that A is impossible for *alpha*.

We know that *beta* prefers A to all the others, except for those who have previously rejected him, and, by assumption, are impossible for him.

Proof

- Consider a hypothetical matching in which *alpha* is married to A, and *beta* is married to a woman who is possible for him.
- Under such an arrangement *beta* is married to a woman who is less desirable to him than A. But such a hypothetical matching is unstable since *beta* and A could upset it to the benefit of both.
- The conclusion is that the algorithm rejects men only from women that they could not possibly be married to under any stable matching. The resulting matching is therefore optimal.

Extensions

- New matching models
- Nonlinear objective function bipartite matching
- Multi-objective bipartite matching

References

- A. J. Hoffman and H. M. Markowitz, "A Note on Shortest Path, Assignment, and Transportation Problems," *Naval Research Logistics Quarterly*, **10** (1963) 375-380.
- N. S. Mendelsohn and A. L. Dulmage, "Some Generalizations of the Problem of Distinct Representatives," *Canadian Journal of Mathematics*, **10** (1958) 230-241.
- M. Hall, Jr., "An Algorithm for Distinct Representatives," *American Mathematics Monthly*, **63**, (1956) 716-717.
- J. E. Hopcroft and R. M. Karp, "An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs," *SIAM Journal of Computing*, **2** (1973) 225-231.
- F. Glover, "Maximum Matching in a Convex Bipartite Graph," *Naval Research Logistics Quarterly*, **14** (1967) 313-316.
- D. R. Fulkerson, I. Glicksburg, and O. Gross, "A Production Line Assignment Problem," The RAND Corporation, RM-1102, May 1953.
- O. Gross, "The Bottleneck Assignment Problem: An Algorithm," *Proceedings, Rand Symposium on Mathematical Programming*, Rand Publication R-351, Philip Wolfe, editor, 1960, pp. 87-88.
- J. Edmonds and D. R. Fulkerson, "Bottleneck Extrema," *Journal of Combinatorial Theory*, **8** (1970) 299-306.

References

- M. Klein and H. Takamori, "Parallel Line Assignment Problems," *Management Science*, **19** (1972) 1-10.
- H. W. Kuhn, "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, **2** (1955) 83-97.
- H. W. Kuhn, "Variants of the Hungarian Method for Assignment Problems," *Naval Research Logistics Quarterly*, **3** (1956) 253-258.
- M. L. Balinski and R. E. Gomory, "A Primal Method for the Assignment and Transportation Problems," *Management Science*, **10** (1964) 578-593.
- D. Gale, *The Theory of Linear Economic Models*, McGraw-Hill, New York, 1958.
- P. C. Gilmore and R. E. Gomory, "Sequencing a One State-Variable Machine: A Solvable Case of the Traveling Salesman Problem," *Operations Research*, **12** (1964) 655-679.
- D. Gale and L. S. Shapley, "College Admissions and the Stability of Marriage," *American Mathematics Monthly*, **69** (1962) 9-14.