Combinatorial Optimization and Graph Theory ORCO Matchings in bipartite graphs

Zoltán Szigeti

Planning

- Bipartite graphs
- Matchings
- Maximum cardinality matchings in bipartite graphs
- Matchings in bipartite graphs by flows
- Perfect matchings in bipartite graphs
- Maximum weight matchings in bipartite graphs
  - Linear Programming background
  - Hungarian method
    - Execution
    - Justification
  - Applications

Definition

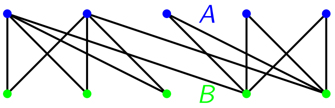
Bipartite graph: if there exists a partition of $V(G)$ into two sets A and B such that every edge of G connects a vertex of A to a vertex of B.

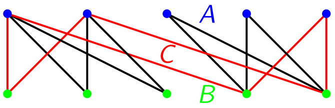
Theorem 1

G is bipartite $\Longleftrightarrow$ G contains no odd cycle.

Proof of necessity

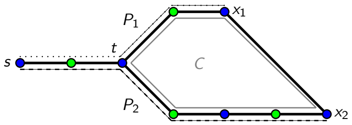
1. Let $G = (A, B; E)$ be bipartite and C an elementary cycle of G.
2. $(A, B; E(C))$ is bipartite and $d_C(v) = 0$ or 2 for all $v \in A$.
3. $|E(C)| = \sum_{v \in A} d_C(v) \equiv \sum_{v \in A} 0 = 0 \bmod 2.$





Proof of sufficiency

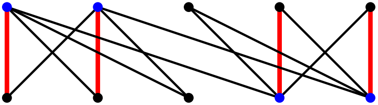
1. Let $G = (V, E)$ be a connected graph.
2. Choose a vertex s of V, $A := \{s\}$, $B := \emptyset$.
   While there exist $u \in A \cup B, v \notin A \cup B, uv \in E$ do:
   $B := B \cup \{v\}$ if $u \in A$ and $A := A \cup \{v\}$ if $u \in B$; $p(v) := u$.
3. If there exists $x_1x_2 \in E$ st x_1 and x_2 have the same color then do :
   1. Let P_i be the (s, x_i)-path obtained using $p(.)$ $(i = 1, 2)$.
   2. Let t be the last common vertex in P_1 and P_2 starting from s.
   3. $C := P_1[x_1, t] + P_2[t, x_2] + x_2x_1$ is an elementary odd cycle of G, contradiction.
4. G is hence bipartite.



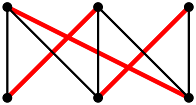
Basic definitions

Definitions : $G = (V, E)$

1. Matching : $M \subseteq E$ such that $d_M(v) \leq 1\ \forall v \in V$.
2. Perfect matching : $M \subseteq E$ such that $d_M(v) = 1\ \forall v \in V$.
3. Transversal : $T \subseteq V$ such that $T \cap \{u, v\} \neq \emptyset\ \forall uv \in E$.
4. $\nu(G)$ $:= \max\{|M| : M$ matching of $G\}$.
5. $\tau(G)$ $:= \min\{|T| : T$ transversal of $G\}$.



$\nu(G_1) = 4 = \tau(G_1)$



$\nu(G_2) = 3 = \tau(G_2)$

Relation between $\nu(G)$ and $\tau(G)$ in general

Lemma 1

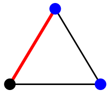
For every graph G, $\nu(G) \leq \tau(G)$.

Proof

1. Let M be a maximum matching and T a minimum transversal of G.
2. Since T is a transversal, T contains at least one end-vertex of every edge e of M, say v_e.
3. Since M is a matching, $v_e \neq v_f$ if $e, f \in M$ and $e \neq f$.
4. $|M| = |\{v_e \in T : e \in M\}| \leq |T|$.
5. $\nu(G) = |M| \leq |T| = \tau(G)$.

Example

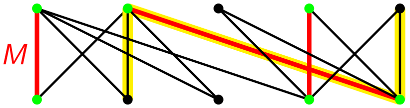
$\nu(K_3) = 1 < 2 = \tau(K_3)$.



Basic definitions

Definitions : $G = (V, E)$, M matching of G

1. M-satured vertex: $v \in V$ such that $d_M(v) = 1$.
2. M-unsatured vertex: $v \in V$ such that $d_M(v) = 0$.
3. M-alternating path: if its edges are alternating in M and in $E \setminus M$.
4. M-augmentanting path: if M-alternating with M-unsaturated end-vertices.
5. $G = (U, W; E)$: bipartite graph with color classes U and W.

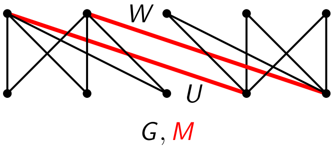


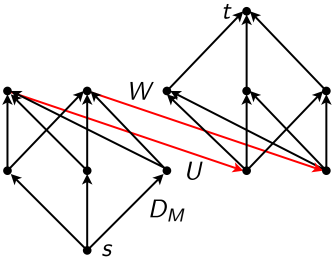
Construction of an auxiliary directed graph

Definition

Given a bipartite graph $G = (U, W; E)$ and a matching M of G, we construct a directed graph $D_M = (V, A)$ as follows:

1. $V := U \cup W \cup \{s, t\}$,
2. $A := \{su : u\ M\text{-unsaturated in } U\} \cup \{wt : w\ M\text{-unsaturated in } W\} \cup \{wu : uw \in M\} \cup \{uw : uw \in E \setminus M\}$.



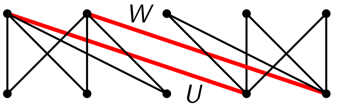


Characterization of a matching of maximum cardinality

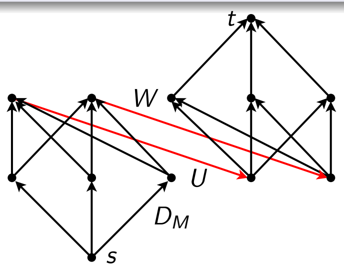
Theorem 2

Given a bipartite graph $G = (U, W; E)$ and a matching M of G, the following conditions are equivalent:

1. $\nu(G) = |M|$,
2. no M-augmenting path exists in G,
3. no (s, t)-path exists in D_M,
4. $\tau(G) \leq |M|$.



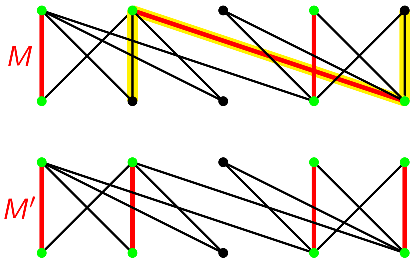
G, M



Characterization of a matching of maximum cardinality

$(1) \implies (2) : \nu(G) = |M| \implies$ no M-augmenting path

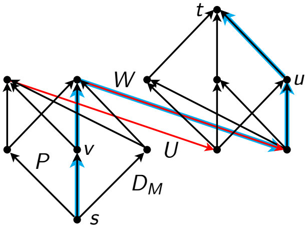
1. Suppose that an M-augmenting path P exists in G.
2. Let $M' := (M \setminus E(P)) \cup (E(P) \setminus M)$.
3. M' is a matching and $|M'| = |M| + 1$.
4. M is not of maximum cardinality, contradiction.

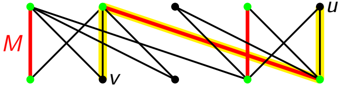


Characterization of a matching of maximum cardinality

$(2) \Longrightarrow (3)$: no M-augmenting path in G $\Longrightarrow$ no (s,t)-path in D_M

1. Suppose for a contradiction that an (s,t)-path P exists in D_M.
2. Let $P' := P - s - t$.
3. Since P starts with an arc sv and finishes with an arc ut, the end-vertices u and v of P' are M-unsaturated.
4. Since P is a directed path, P' is an M-alternating path.
5. P' is hence an M-augmenting path, contradiction.

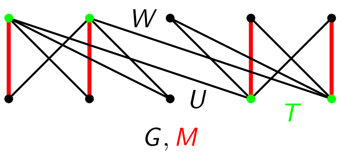


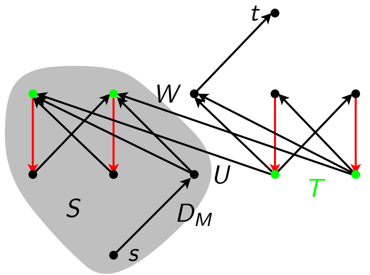


Characterization of a matching of maximum cardinality

(3) $\Longrightarrow$ (4) : no (s,t)-path in $D_M \Longrightarrow \tau(G) \leq |M|$

1. Suppose no (s,t)-path exists in D_M.
2. Let S be the set of vertices attainable from s in D_M.
3. Let $T := (U \setminus S) \cup (W \cap S)$.
4. Since no arc leaves S in D_M, T is a transversal of G and $|T| \leq |M|$.
5. $\tau(G) \leq |T| \leq |M|$.





Characterization of a matching of maximum cardinality

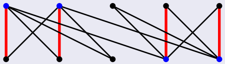
$(4) \Longrightarrow (1)$: $\tau(G) \leq |M| \Longrightarrow \nu(G) = |M|$

1. Suppose that $\tau(G) \leq |M|$.
2. Since M is a matching, $|M| \leq \nu(G)$.
3. By Lemma 1, $\nu(G) \leq \tau(G)$.
4. Thus, equality holds everywhere, in particular : $|M| = \nu(G)$.

Consequences

Theorem 3 (Kőnig)

For every bipartite graph G, $\nu(G) = \tau(G)$.



Maximum cardinality matching in a bipartite graph algorithm

INPUT: G bipartite graph.
OUTPUT: Maximum cardinality matching M of G.

Step 0. *Initialization.*
$M := \emptyset$.

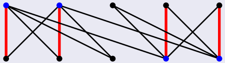
Step 1. *Matching augmentation.*
While an (s,t)-path P exists in D_M do
$M := (M \setminus E(P)) \cup (E(P - s - t) \setminus M)$.

Step 2. *End of algorithm.*
STOP.

Matchings by flows

Theorem 3 (Kőnig)

For every bipartite graph G, $\nu(G) = \tau(G)$.



Ford-Fulkerson $\Longrightarrow$ Kőnig

Let $(D := (W, A), g)$ be a network where $W := U \cup V \cup \{s, t\}$,
$A := \{su : u \in U\} \cup \{vt : v \in V\} \cup \{uv : u \in U, v \in V, uv \in E\}$,
$g(su) := 1\ \forall u \in U$, $g(vt) := 1\ \forall v \in V$ and $g(uv) := |U| + 1\ \forall uv \in E$,
x an integer feasible (s, t)-flow of max. value, Z an (s, t)-cut of min. capacity,
$M := \{uv \in E : x(uv) = 1\}$ and $T := (U \setminus Z) \cup (V \cap Z)$.

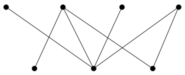
(a) Prove that M is a matching of G of size $val(x)$.

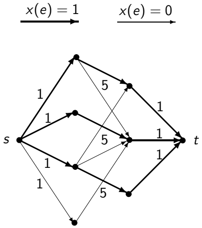
(b) Prove that T is a transversal of G of size $cap(Z)$.

(c) Deduce Kőnig Theorem from (a), (b) and Ford-Fulkerson Theorem.

Proof of (a)

1. There exists an integer g-feasible (s,t)-flow x of maximum value.
2. Since $d_x^+(u) = d_x^-(u) = x(su) \leq g(su) = 1 \ \forall u \in U$,
   and $d_x^-(v) = d_x^+(v) = x(vt) \leq g(vt) = 1 \ \forall v \in V$, we have
   1. $x(e) = 0$ or $1 \ \forall e \in A$ and
   2. at most one edge of M is incident to $w \in U \cup V$.
3. Thus M is a matching of G.
4. $val(x) = d_x^+(U \cup s) - d_x^-(U \cup s) = d_x^+(U \cup s) = \sum_{x(uv)=1, uv \in E} 1 = |M|$.



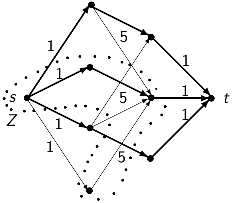


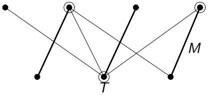
Proof of (b)

1. $cap(Z) \leq d_g^+(s) = |U|$ since Z is an (s,t)-cut of minimum capacity.
2. Let K be the set of arcs in D from $U \cap Z$ to $V \setminus Z$.

$$
\begin{aligned}
|U| \geq cap(Z) &= \sum_{u \in U \setminus Z} g(su) + \sum_{v \in V \cap Z} g(vt) + \sum_{uv \in K} g(uv) \\
&= \sum_{u \in U \setminus Z} 1 + \sum_{v \in V \cap Z} 1 + \sum_{uv \in K} (|U|+1) \\
&= |U \setminus Z| + |V \cap Z| + |K|(|U|+1) \\
&= |T| + |K|(|U|+1).
\end{aligned}
$$

3. Hence $K = \emptyset$, so T is a transversal and $cap(Z) = |T|$.

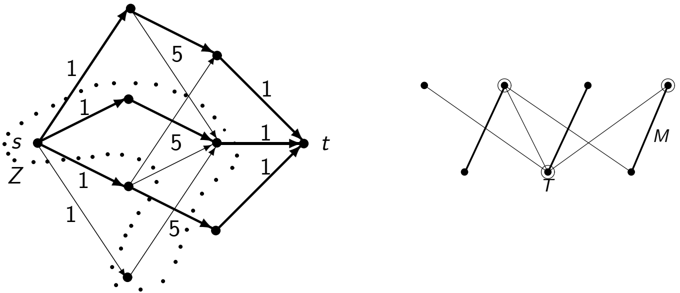




Matchings by flows

Proof of (c)

1. By Ford-Fulkerson's theorem, (a), Lemma 1 and (b), $cap(Z) = val(x) = |M| \leq \nu(G) \leq \tau(G) \leq |T| = cap(Z)$,
2. Hence equality holds everywhere, in particular $\nu(G) = \tau(G)$.



Perfect matchings in bipartite graphs

Notation

Given a bipartite graph $G = (U, W; E)$ and $X \subseteq U$,
$\Gamma_G(X)$: set of neighbors of X.

Theorem 4 (Hall)

A bipartite graph $G = (U, W; E)$ has a perfect matching $\Longleftrightarrow$

(a) $|U| = |W|$,

(b) $|\Gamma_G(X)| \geq |X| \ \forall X \subseteq U$.

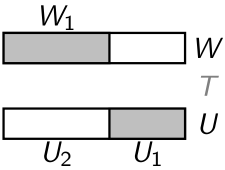
Proof of necessity :

1. If G has a perfect matching M then $|\Gamma_M(X)| = |X| \ \forall X \subseteq U$.
2. In particular, $|U| = |\Gamma_M(U)| = |W|$ and hence (a) is satisfied.
3. Since $|\Gamma_G(X)| \geq |\Gamma_M(X)|$, (b) is satisfied.

Perfect matchings in bipartite graphs

Proof of sufficiency :

1. By Theorem 3, $\exists$ matching M and transversal T of G st $|M| = |T|$.
2. $U_1 := T \cap U$, $W_1 := T \cap W$ et $U_2 := U - U_1$.
3. Since T is a transversal, $\Gamma(U_2) \subseteq W_1$; and hence $|W_1| \geq |\Gamma(U_2)|$.
4. By (b), $|\Gamma(U_2)| \geq |U_2|$ and by (a), $|U| = |W|$.
5. $|M| = |T| = |U_1 \cup W_1| = |U_1| + |W_1| \geq |U_1| + |U_2| = |U| = |W|$.
6. The vertices of U and those of W are hence M-saturated.
7. Thus M is a perfect matching of G.



Maximum weight matchings in bipartite graphs

Problem

P_1 : Given a bipartite graph $G = (U, V; E)$ and a weight function c on E, find a matching M of maximum weight ($\sum_{e \in M} c(e)$) of G.

- P_2 maximum weight matching in a bipartite graph with $c \geq 0$: delete the edges of negative weight as they are not in a maximum weight matching.
- P_3 maximum weight perfect matching in a complete bipartite graph $K_{n,n}$: we add new vertices and new edges of weight zero.
- P_4 minimum weight perfect matching in $K_{n,n}$: $c' := -c$.
- P_5 minimum weight perfect matching in $K_{n,n}$ with $c \geq 0$: $c' := c + L$ where $L := \max\{|c(e)| : e \in E\}$. The new weighting is non-negative and the weight of each perfect matching increased by constant ($n \cdot L$).
- P_6 minimum weight perfect matching in a bipartite graph having a perfect matching with $c \geq 0$: more general than P_5.

Linear Programming

Linear Program (Primal)

The problem of finding a c-minimum weight perfect matching in a bipartite graph $G = (U, V; E)$ can be formulated as a linear program.

$$\begin{aligned} \sum_{e \in \delta(w)} x(e) &= 1 \quad \forall w \in U \cup V, \\ x(e) &\geq 0 \quad \forall e \in E, \\ \sum_{e \in E} c(e)x(e) &= w(\min). \end{aligned}$$

Linear Program (Primal) for a bipartite graph $G = (U, V; E)$

$$
\begin{array}{rcll}
\sum_{e \in \delta(w)} x(e) & = & 1 & \forall w \in U \cup V, \\
x(e) & \geq & 0 & \forall e \in E, \\
\sum_{e \in E} c(e)x(e) & = & w(\min). &
\end{array}
$$

Remark

1. The characteristic vector of a perfect matching of G is a feasible solution of (P).
2. A basic solution of (P) is the characteristic vector of a perfect matching of G since the polyhedron of (P) is integer by Cramer's rule and since each square submatrix of the incidence matrix of a bipartite graph is of determinant $0, 1$ or -1.

Linear Programming

Dual of (P)

$$y(u) + y(v) \leq c(uv) \quad \forall uv \in E,$$
$$\sum_{w \in U \cup V} y(w) = z(\max).$$

Complementary slackness theorem

1. If x and y are feasible solutions of (P) and (D) and
2. the complementary slackness conditions are satisfied:
   $x(uv) > 0 \Longrightarrow y(u) + y(v) = c(uv)$ (uv is y-tight).
3. then x and y are optimal solutions of (P) and (D).

Theorem (Egerváry)

The minimum weight of a perfect matching in a bipartite graph is equal to the optimal value of (D).

- At that time, linear programming didn't exist!

How to find a minimum weight perfect matching in a bipartite graph

Algorithm Hungarian method (Kuhn)

Input: Bipartite graph $G = (U, V; E)$ that has a perfect matching and non-negative weighting c on E.

Output: Minimum c-weight perfect matching of G.

Idea : We will have in each step:

1. a vector x (the characteristic vector of a matching M),
2. a feasible solution y of (D),
3. such that the complementary slackness conditions are satisfied:
   - $x(e) > 0 \Longrightarrow e$ is y-tight, that is
   - M is a matching of the subgraph induced by y-tight edges.

Algorithm Hungarian method (Kuhn)

Step 0. *Initialization.*

$M_0 := \emptyset, i := 1.$

$$y_1(w) := \begin{cases} \min\{c(wv) : wv \in E\} & \text{if } w \in U, \\ 0 & \text{if } w \in V, \end{cases}$$

Step 1. *Construction of subgraph G_i of tight edges.*

$G_i := (U, V; E_i)$ where $E_i = \{uv \in E : y_i(u) + y_i(v) = c(uv)\}$.

Step 2. *Construction of maximum matching and of minimum transversal of G_i.*

Starting from M_{i-1} and using flows, find a maximum cardinality matching M_i and a minimum cardinality transversal T_i of G_i.

Step 3. *Stopping rule.*

If M_i is a perfect matching of G_i, then STOP with M_i.

Step 4. *Modification of dual solution.*

$\varepsilon_i := \min\{c(uv) - y_i(u) - y_i(v) : uv \in E(G - T_i)\}$

$$y_{i+1}(w) := \begin{cases} y_i(w) + \varepsilon_i & \text{if } w \in U \setminus T_i, \\ y_i(w) - \varepsilon_i & \text{if } w \in V \cap T_i, \\ y_i(w) & \text{otherwise.} \end{cases}$$

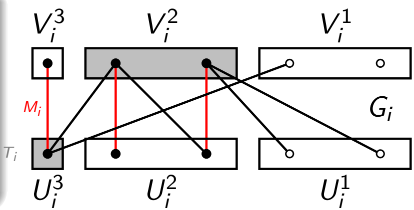
$i := i + 1$ and go to Step 1.

Construction of partitions of U and V.

$U_i^1 = \{u \in U : M_i\text{-unsaturated}\}$, $U_i^2 = U \setminus (T_i \cup U_i^1)$, $U_i^3 = U \setminus (U_i^1 \cup U_i^2)$,
$V_i^1 = \{v \in V : M_i\text{-unsaturated}\}$, $V_i^2 = V \cap T_i$, $V_i^3 = V \setminus (V_i^1 \cup V_i^2)$.

Reminder

- $U_i^2 = \{u \in U \setminus U_i^1 : \exists u' \in U_i^1$ and an M_i-alternating (u', u)-path$\}$,
- $V_i^2 = \Gamma_{M_i}(U_i^2)$,
- $V_i^3 = \Gamma_{M_i}(U_i^3)$.



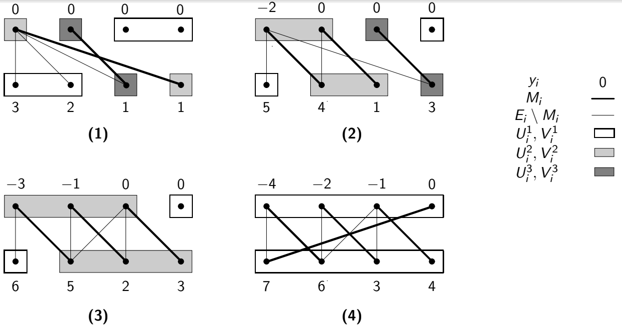
Remark

$\varepsilon_i := \min\{c(uv) - y_i(u) - y_i(v) : uv \in E(G - T_i)\}$

$$y_{i+1}(w) := \begin{cases} y_i(w) + \varepsilon_i & \text{if } w \in U \setminus T_i, \\ y_i(w) - \varepsilon_i & \text{if } w \in V \cap T_i, \\ y_i(w) & \text{otherwise.} \end{cases}$$

Theorem

1. y_i is a feasible solution of (D) $\forall i$.
2. E_{i+1} contains M_i and an edge of G from $U_i^1 \cup U_i^2$ to $V_i^1 \cup V_i^3$ $\forall i$.
3. After each execution of Step 2, $|M_{i+1}| > |M_i|$ or $|U_{i+1}^2| > |U_i^2|$.
4. The algorithm stops in polynomial time
5. with a minium weight perfect matching M_i of G.



Justification of the Hungarian method

1. y_i is a feasible solution of (D)
$y_i(u) + y_i(v) \leq c(uv)\ \forall uv \in E$.

Proof: By induction on i. Let uv be an arbitrary edge of G.

1. $i = 1 : y_1(u) + y_1(v) = \min\{c(uw) : uw \in E\} + 0 \leq c(uv)$.
2. Suppose it is true for i.
   1. If $u \in U_i^1 \cup U_i^2$ and $v \in V_i^2$, then
   $$y_{i+1}(u) + y_{i+1}(v) = (y_i(u) + \varepsilon_i) + (y_i(v) - \varepsilon_i) \leq c(uv).$$
   2. If $u \in U_i^1 \cup U_i^2$ and $v \in V_i^1 \cup V_i^3$, then, by definition of ε_i,
   $$\begin{aligned} y_{i+1}(u) + y_{i+1}(v) &= (y_i(u) + \varepsilon_i) + y_i(v) \\ &\leq y_i(u) + y_i(v) + (c(uv) - y_i(u) - y_i(v)) \\ &= c(uv). \end{aligned}$$
   3. If $u \in U_i^3$, then $y_{i+1}(u) + y_{i+1}(v) \leq y_i(u) + y_i(v) \leq c(uv)$.
3. In each cases, y_{i+1} is a feasible solution of (D).

Justification of the Hungarian method

2. E_{i+1} contains M_i and an edge of G from $U_i^1 \cup U_i^2$ to $V_i^1 \cup V_i^3$.

Proof :

1. y_i-tight edges from $U_i^1 \cup U_i^2$ to V_i^2 and from U_i^3 to V_i^3 are y_{i+1}-tight:
   1. $y_{i+1}(u) + y_{i+1}(v) = (y_i(u) + \varepsilon_i) + (y_i(v) - \varepsilon_i) = c(uv)$.
   2. $y_{i+1}(u) + y_{i+1}(v) = y_i(u) + y_i(v) = c(uv)$.

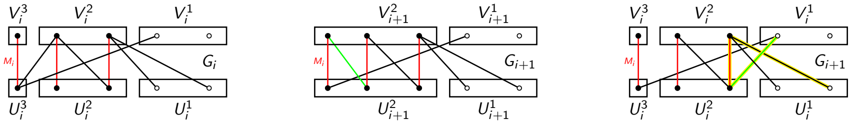
   In particular, $M_i \subseteq E_{i+1}$.
2. Since G has a perfect matching, by definition of ε_i, $uv \in E$ exists:
   1. $\varepsilon_i = c(uv) - y_i(u) - y_i(v), u \in U_i^1 \cup U_i^2, v \in V_i^1 \cup V_i^3$. Then
   2. $$\begin{aligned} y_{i+1}(u) + y_{i+1}(v) &= (y_i(u) + \varepsilon_i) + y_i(v) \\ &= y_i(u) + (c(uv) - y_i(u) - y_i(v)) + y_i(v) \\ &= c(uv). \end{aligned}$$

   and so $uv \in E_{i+1}$.

3. $|M_{i+1}| > |M_i|$ or $|U^2_{i+1}| > |U^2_i|$.

Proof: Let $uv \in E_{i+1}$ such that $u \in U^1_i \cup U^2_i, v \in V^1_i \cup V^3_i$.

1. If $v \in V^3_i$, then $|U^2_{i+1}| > |U^2_i|$.
2. If $v \in V^1_i$, then an M_i-augmenting path exists in G_{i+1}, hence, by Theorem 2, $|M_{i+1}| > |M_i|$.



Justification of the Hungarian method

4. The algorithm stops in polynomial time.

Proof :

1. Each Step is polynomial.
2. We show that the loop is executed a polynomial number times:
   1. By 3, either $|U^2_{i+1}| > |U^2_i|$ or $|M_{i+1}| > |M_i|$.
   2. The first case can happen at most $\frac{n}{2}$ times so after at most $\frac{n}{2}$ executions of the loop, M_i is augmented.
   3. One can augment M_i at most $\frac{n}{2}$ times.
3. The algorithm is hence polynomial.

Justification of the Hungarian method

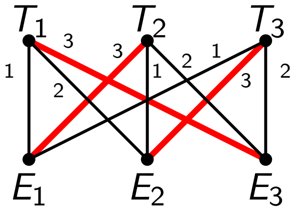
5. The algorithm stops with a min. weight perfect matching M_i of G.

Proof :

1. Let x_i be characteristic vector of the perfect matching M_i of G_i when the algorithm stops.
2. Then x_i is a feasible solution of (P).
3. By 1, y_i is a feasible solution of (D).
4. Since $M_i \subseteq E_i$, if $x_i(uv) > 0$ then uv is y_i-tight, thus the complementary slackness conditions are satisfied.
5. It follows that x_i and y_i are optimal solutions.
6. M_i is a minimum weight perfect matching of G.

Assignment

- A director must assign his n employees to n tasks to be executed.
  - Each employee will execute exactly one task and
  - each task will be executed by exactly one employee.
- Since the director knows his employees well, he knows the profit c_{ij} he can earn by assigning the employee E_i to the task T_j.
- He hires you to help him to find the assignment of maximum profit.
- How would you model this problem?



Applications

Locating objects in space

- We want to determine the exact positions of n objects in 3-dimensional space using two fixed infrared sensors.
- Each sensor provides us n straight lines, each containing one objects.
- These $2n$ lines give theoretically the exact positions of the n objects.
- Due to technical problems we only have approximately the lines.
- We know that two lines corresponding to the same object have a very small Euclidean distance.
- How would you model this problem?