Lecture Notes - Week IV

Matching

Fernando Dias, Philine Schiewe and Piyalee Pattanaik

January 29, 2024

CHAPTER 1
Matching

As always, a definition at first:

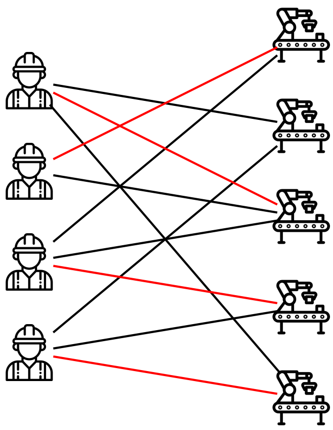
Definition 1 ***Matching*** *in an undirected graph is a set of edges without common vertices.*

Also known as **independent edge set**, this problem goal is to find a subset of the edges as a matching if each node appears in at most one edge of that matching.

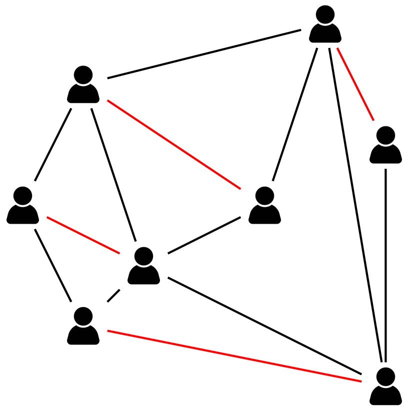
From an undirected graph $G = (V, E)$, $M \subset E$ is called *matching* if all $e \in M$ are pairwise disjoint, i.e., if the endpoints are different. In addition, $M \subset E$ is a *maximum matching* in G if M is a matching with highest cardinality, i.e.,

$$|M'| \leq |M| \quad \text{for all matchings } M'$$

Some illustrations as example:



Assignment different workers to different tasks in order that there is no conflict or overlapping.



Setting pairs for homework assignments.

For this problem, a simple integer linear programming formulation can be calculated:

$$\begin{aligned}
&\text{Maximize} \quad \sum_{e \in E} x_e \\
&\text{Subject to:} \\
&\sum_{e \in \delta(v)} x_e \leq 1 && \forall v \in V \\
&x_{ij} \in \{0, 1\} && \forall e \in E
\end{aligned}$$

where $\delta(v)$ is the set of incident edges of $v \in V$, such that:

$$\delta(v) = \{e \in E \colon e = \{v, w\}\}$$

Like flow problems, we can also define ***M*-augmenting paths**. Let $G = (V, E)$ be an undirected graph and $M \subseteq E$ matching. A node $v \in V$ is said to be **covered** by M if $v \in e$ for some $e \in M$ and it is **exposed** by M if $v \notin e$ for all $e \in M$.

With those, two types of paths can be defined *M-alternating path P*, where edges $E(P)$ are alternately in M and not in M (or not in M and in M) and *M-augmenting path P* that is a special type of M-alternating path, where the first and last vertex exposed.

Remark: M-augmenting paths have odd number of edges.

According to Berge's Theorem:

Theorem 1 (Petersen (1891), Berge (1957)) *Let G be a graph with some matching M. Then M is the maximum if and only if there is no M-augmenting path.*

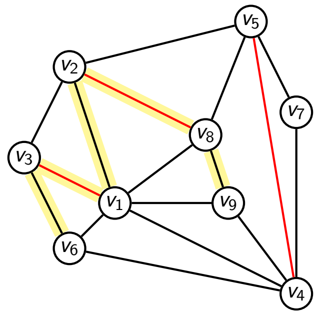
Proof 1 *Proof idea $\Rightarrow$: By contraposition: Let $P = (v_0, e_1, \dots, e_k, v_k)$ be an M-augmenting path.*

- *by definition: v_0, v_k exposed*

$\Rightarrow \ |E(P) \setminus M| = |E(P) \cap M| + 1$

$\Rightarrow$ *$M' = (M \setminus E(P)) \cup (E(P) \setminus M)$ is matching with $|M'| = |M| + 1$*

$\Rightarrow$ *M not maximum*



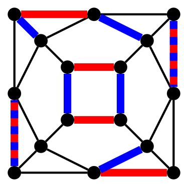
From this theorem, we can derive a few lemmas, such as

Lemma 1 *Let G be a graph with two matchings M, M'. Let $G' = (V, E' = M \blacksquare M')$, with symmetric difference*

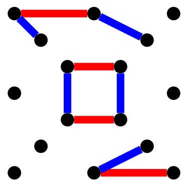
$$M \blacksquare M' = (M \cup M') \setminus (M \cap M').$$

Then, the connected components of G' are

- *isolated vertices*
- *cycles C with $|E(C)| \in 2\mathbb{N}$ where edges in C are alternately in M and M'*
- *paths $P = (v_0, e_1, \ldots, e_k, v_k)$ where edges are alternately in M and M'*



graph G



graph G'

Proof 2 *Proof idea: Let M, M' matchings:*

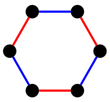
$$\begin{aligned}
&|\{e \in M \colon v \in e\}| \leq 1, v \in V \\
&|\{e \in M' \colon v \in e\}| \leq 1, v \in V \\
\Rightarrow &|\{e \in E' \colon v \in e\}| \leq 2, v \in V
\end{aligned}$$

If $g_{G'}(v) = |\{e \in E' \colon v \in e\}| = 2$: $\exists! e \in M \colon v \in e$ and $\exists! e \in M' \colon v \in e$.

- isolated vertices $v \rightsquigarrow g_{G'}(v) = 0$

•

- cycles C with $|E(C)| \in 2\mathbb{N} \rightsquigarrow g_{G'}(v) = 2$



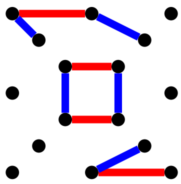
- paths $P = (v_0, e_1, \ldots, e_k, v_k) \rightsquigarrow g_{G'}(v_0) = 0 = g_{G'}(v_k) = 1$, $g_{G'}(v_i) = 2$, $1 \leq i \leq k-1$

Another way to prove the same theorem is listed below:

Theorem 2 (Petersen (1891), Berge (1957)) *Let G be a graph with some matching M. Then M is the maximum if and only if there is no M-augmenting path.*

Proof 3 *Proof idea:*
By contraposition: Let M' be a matching with $|M'| > |M|$.
Construct G'.

$$\begin{aligned} |M'| > |M| &\Rightarrow |E' \cap M'| > |E' \cap M| \\ &\Rightarrow \exists P = (v_0, e_1, \ldots, e_k, v_k) \text{ with } e_1 \in M', e_k \in M' \\ &\Rightarrow v_0, v_k \text{ exposed by } M \\ &\Rightarrow P \ M\text{-augmenting path} \end{aligned}$$



graph G'

CHAPTER 2

Maximum Matching

With all of this in mind, the **resulting algorithm** can be expressed:

Algorithm: MAXIMUM MATCHING

Input: undirected graph $G = (V, E)$
Output: maximum matching M

set $M = \emptyset$
while *there exists M-augmenting path in G* **do**
  choose M-augmenting path P
  set $M = (M \setminus E(P)) \cup (E(P) \setminus M)$
return M

In this algorithm, up to $\frac{|V|}{2}$ iterations are required. There is no obvious way to find an M-augmenting path. However, for bipartite graphs, the easier way is to find s-t-path in auxiliary graphs, while in general graphs, **Edmond's blossom algorithm** is the best approach. Nevertheless, such an algorithm is highly **complex** and has a **polynomial runtime**.

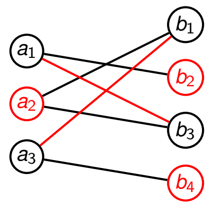
However, the challenge still remains on **finding M-alternating paths**. For bipartite graph $G = (V, E)$ with:

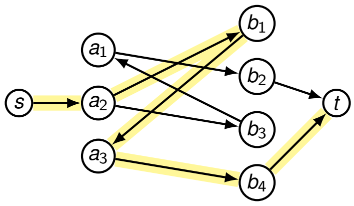
- $V = A \cup B$, $A \cap B = \emptyset$
- $E \subseteq \{\{a, b\} \colon a \in A, b \in B\}$

The easier approach is to construct **auxiliary directed graph** $G' = (V', E')$ with:

$$\begin{aligned}
V' =& V \cup \{s, t\}, \quad s, t \notin V \\
E' =& \{(b, a) \colon \{a, b\} \in M, a \in A, b \in B\} \\
& \cup \{(a, b) \colon \{a, b\} \in E \setminus M, a \in A, b \in B\} \\
& \cup \{(s, a) \colon a \text{ exposed}, a \in A\} \\
& \cup \{(b, t) \colon b \text{ exposed}, b \in B\}
\end{aligned}$$

Then, $\exists$ M-augmenting path in G if and only if $\exists$ s-t-path in G'.





The resulting **algorithm** encapsulates this procedure:

Algorithm: MAXIMUM MATCHING BIPARTITE GRAPHS

Input: undirected bipartite graph $G = (V, E)$
Output: maximum matching M

set $M = \emptyset$
construct G'
while *there exists* s-t*-path in* G' **do**
    choose s-t-path P
    set $M = (M \setminus E(P)) \cup (E(P) \setminus M)$
    update G'
return M

In order to construct G', it takes up to $O(n + m)$, where $n = |V|$ and $m = |E|$, due to no isolated nodes in G . The remaining $\frac{n}{2}$ iterations are divided into:

- finding P: $O(m)$
- updating M: $O(n)$
- updating G': $O(n)$

The final runtime is $O(nm)$.

2.1 CONNECTION TO MAXFLOW

Solving matching can also be formulated as solving maximum flow. By constructing an auxiliary directed graph $G'' = (V'', E'')$ with:

$$\begin{aligned} V'' =& V \cup \{s, t\}, \quad s, t \notin V \\ E'' =& \{(a, b) \colon \{a, b\} \in E, a \in A, b \in B\} \\ & \cup \{(s, a) \colon a \in A\} \\ & \cup \{(b, t) \colon b \in B\} \end{aligned}$$

and capacity $u(e) = 1$ for all $e \in E''$. With that, G'' has maximal flow with value k if and only if G has a maximum matching of cardinality k.