Graph Algorithms - Lecture 7

November 13, 2020

Table of contents

Analytical Formulation of the Maximum Matching Problem

Let $G = (V, E)$ be a graph and $\mathcal{M}_G$ the family of its matchings.

Maximum matching problem.

P$_1$ given a graph $G = (V, E)$, find $M^* \in \mathcal{M}_G$ such that

$$|M^*| = \max_{M \in \mathcal{M}_G} |M|.$$

Let $B = (b_{ij})_{n \times m}$ be the incidency matrix of G (a fixed ordering of the n vertices of G and a fixed ordering of the m edges of G are considered) with

$$b_{ij} = \begin{cases} 1, & \text{if the edge } j \text{ is incident with the vertex } i \\ 0, & \text{otherwise} \end{cases}$$

If 1_p is the p-dimensional vector with all components 1, then the problem P$_1$ can be written equivalently:

Analytical Formulation of the Maximum Matching Problem

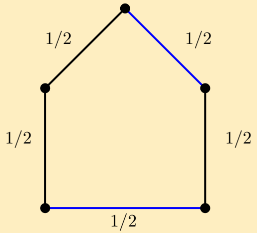
$$\mathrm{P}'_1 \ \max\left\{1_m^T\mathrm{x} \,:\, B\mathrm{x} \leqslant 1_n, \mathrm{x} \geqslant 0, x_i \in \{0,1\}, i=\overline{1,m}\right\}.$$

P'_1 is an ILP (Integer Linear Programming) problem, which is NP-hard. We can try to solve P'_1, using the associated (relaxed) LP (Linear Programming) problem

$$\mathrm{LP}'_1 \ \max\left\{1_m^T\mathrm{x} \,:\, B\mathrm{x} \leqslant 1_n, \mathrm{x} \geqslant 0\right\}.$$

The optimal solutions of LP'_1 can have non-integer components and also their optimal values can be larger than $\nu(G)$. For example, if $G = C_{2n+1}$, then $\nu(G) = n$, but the solution $x_i = 1/2$, $\forall 1 \leqslant i \leqslant 2n+1$ is optimal for the corresponding LP'_1 with the optimal value $n + 1/2 > n$.

Analytical Formulation of the Maximum Matching Problem



It follows that if the graph G has odd circuits, the problem P_1 can not be solved using LP_1'. More precisely,

Theorem 1

(Balinski, 1971) The extreme points of the polytope $B\mathrm{x} \leqslant 1_n$, $\mathrm{x} \geqslant 0$, $\mathrm{x} \in \mathbb{R}^m$, have components in $\{0, 1/2, 1\}$. The components 1/2 arises if and only if G has odd circuits.

Analytical Formulation of the Maximum Matching Problem

- Hence the problem P_1 is easy if the graph is bipartite: solve the problem LP_1' and the (integer) optimal solution found is an optimal solution for P_1.
- Combinatorial adaptation of the simplex algorithm for solving linear programming problems led to the so called hungarian method for solving P_1 for bipartite graphs.
- We will discuss a faster solution given by Hopcroft and Karp (1973).
- However, the theorem of duality in linear programming and the integrity of the optimal solution can be used obtain and explain the (already proven) results on mathchings in bipartite graphs:

Perfect Matchings - Tutte's Theorem

Theorem 2

(Hall, 1935) Let $G = (S, T; E)$ be a bipartite graph. There exists a mathching in G which saturates all the vertices in S if and only if

$$|N_G(A)| \geqslant |A|, \forall A \subseteq S.$$

Theorem 3

(Konig, 1930) Let $G = (S, T; E)$ be a bipartite graph. The maximum cardinality of a mathching in G is equal with the minimum cardinality of a vertex cover $\nu(G) = n - \alpha(G)$.

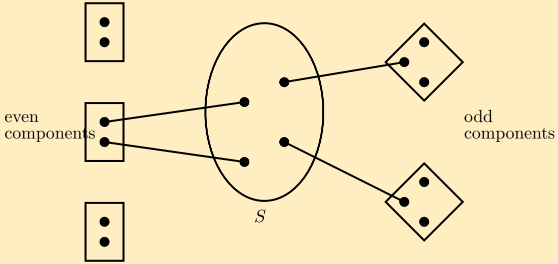
Perfect Matchings - Tutte's Theorem

Let G be a graph. Clearly, $|M| \leqslant |G|/2, \forall M \in \mathcal{M}_G$. A perfect matching (or 1-factor) in G is a matching M with the property $|M| = |G|/2$ (i. e., $S(M) = V(G)$).
A connected component of graph G is even (odd) if the number of its vertices is even (odd). We denote by $q(G)$ the number of odd connected components of G.

Remark

Let G be a graph having a perfect matching. Clearly, each connected component of G is even. Hence, $q(G) = 0$. Moreover, if $S \subseteq V(G)$ then for each odd connected component of the graph $G - S$ there exists an edge in the perfect matching of the graph with an extremity in this connected component and the other in S. Since the extremities of different edges are distinct, it follows that $|S| \geqslant q(G - S)$. For $S = \varnothing$, we obtain $q(G - \varnothing) \leqslant 0$, hence $q(G) = 0$.

Perfect Matchings - Tutte's Theorem



Theorem 4

(Tutte, 1947) A graph G has a perfect matching if and only if

(T) $$q(G-S) \leqslant |S|, \forall S \subseteq V(G).$$

Perfect Matchings - Tutte's Theorem

Proof. The necessity of condition (T) has been already observed in the above discussion. We prove by induction on $n = |G|$ that if a graph $G = (V, E)$ satisfies (T), then G has a perfect matching.

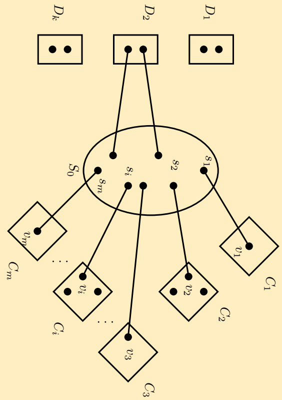
For $n = 1, 2$ the theorem obviously holds. In the inductive step, let G be graph with $n \geqslant 3$ vertices which satisfies (T) and suppose that any graph G' with $|G'| < n$ and which satisfies (T) has a perfect matching.

Let $S_0 \subseteq V(G)$ such that $q(G - S_0) = |S_0|$ and maximal with the property that we have equality in (T) (i. e., for any superset S of S_0 we have $q(G - S) < |S|$).

Note that the family of subsets of $V(G)$ for which (T) is satisfied with equality is non-empty, and therefore S_0 exists (for each vertex v_0 which is not vertex-cut in its component, we have $q(G - v_0) = 1 = |\{v_0\}|$, since each connected component is even, and in each connected component with at least one vertex, there is such a vertex v_0).

Perfect Matchings - Tutte's Theorem

Let $m = |S_0| > 0$, $C_1, C_2, \ldots, C_m$ the odd components of $G - S_0$ and $D_1, D_2, \ldots, D_k$ the even components of $G - S_0$ ($k \geqslant 0$):



We will build a perfect matching composed by

Perfect Matchings - Tutte's Theorem

a) a perfect matching in each even connected component D_i;

b) a matching with m edges, $\{e_1, \ldots, e_m\}$, the edge e_i having an end $s_i \in S_0$ and the other $v_i \in C_i$ $(i = 1, \overline{1, m})$;

c) a perfect matching in each subgraph $C_i - v_i$ $(i = 1, \overline{1, m})$.

a) For each $1 \leqslant i \leqslant k$ the graph $[D_i]_G$ has a perfect matching. Indeed, since $m > 0$, it follows that $|D_i| < n$ and by the induction hypothesis it is sufficiently to show that $G' = [D_i]_G$ satisfies (T).
Let $S' \subseteq D_i$. If $q(G' - S') > |S'|$, then we get the following contradiction:

$$q(G-(S_0 \cup S')) = q(G-S_0) + q(G'-S') = |S_0| + q(G'-S') > |S_0 \cup S'|.$$

Hence, $q(G'-S') \leqslant |S'|$, $\forall S' \subseteq D_i$, i. e., G' satisfies (T).

Perfect Matchings - Tutte's Theorem

b) Let $H = (S_0, \{C_1, \ldots, C_m\}; E')$ be the bipartite graph with a class of the bipartition S_0, the other class being set of odd connected components of $G - S_0$, and the edges of H are $\{s, C_i\}$ where $s \in S_0$ such that there exists $v \in C_i$ with $sv \in E(G)$.
H has a perfect matching. Indeed, we show that H satisfies Hall's condition for the existence of matching M_0 that saturates $\{C_1, \ldots, C_m\}$. Let $A \subseteq \{C_1, \ldots, C_m\}$. Then $B = N_H(A) \subseteq S_0$, and by the construction of H, in the graph G we have no edge from a vertex $v \in S_0 - B$ to a vertex $w \in C_i \in A$. Hence the odd connected components from A remains odd connected components in $G - B$; hence $q(G - B) \geqslant |A|$. Since G satisfies Tutte's condition (T), we have $|B| \geqslant q(G - B)$. We obtained that $|N_H(A)| = |B| \geqslant |A|$.

Perfect Matchings - Tutte's Theorem

By Hall's theorem H has a matching M_0, which saturates $\{C_1, \ldots, C_m\}$; since $|S_0| = m$, M_0 is perfect.

$$M_0 = \{s_1v_1, s_2v_2, \ldots, s_mv_m\}, S_0 = \{s_1, \ldots, s_m\}, v_i \in C_i, \forall i = \overline{1, m}.$$

c) $\forall i \in \{1, \ldots, m\}$ the graph $G' = [C_i - v_i]_G$ has a perfect matching.

Using the induction hypothesis, it is sufficiently to prove that G' satisfies (T).

Let $S' \subseteq C_i - v_i$. If $q(G' - S') > |S'|$, then, since $q(G' - S') + |S'| \equiv 0\ (mod\ 2)$ (because $|G'|$ is even), it follows that $q(G' - S') \geqslant |S'| + 2$.

If $S'' = S_0 \cup \{v_i\} \cup S'$, we have

$$|S''| \geqslant q(G - S'') = q(G - S_0) - 1 + q(G' - S') = |S_0| - 1 + q(G' - S') \geqslant$$

$$\geqslant |S_0| - 1 + |S'| + 2 = |S''|,$$

i. e., $q(G - S'') = |S''|$, contradicting the choice of S_0 (since $S_0 \subsetneq S''$).

Perfect Matchings - Tutte's Theorem

Therefore $\forall S' \subseteq C_i - v_i$, $q(G' - S') \leqslant |S'|$ and G' has a perfect matching (by the induction hypothesis).
Obviously the matching of G obtained by taking the union of the matchings in a), b), and c) above saturates all the vertices of G, and the theorem follows by induction.

Maximum Cardinality Matchings - Berge's Theorem

Let $G = (V, E)$ be a graph and $M \in \mathcal{M}_G$ be a matching in G.

Definition

An alternating path *of* G with respect to the matching M *is any path*

$$P : v_0, v_0v_1, v_1, \ldots, v_{k-1}, v_{k-1}v_k, v_k$$

such that $\{v_{i-1}v_i, v_iv_{i+1}\} \cap M \neq \varnothing, \forall i = \overline{1, k-1}$.

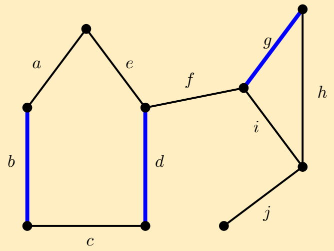
Note that, since M is a matching, if P is an alternating path w.r.t. M, then from any two consecutive edges of P exactly one belongs to M (the edges belong alternatively to M and $E \setminus M$).
In the following, when we will refer to a path P we understand its set of edges.

Maximum Cardinality Matchings - Berge's Theorem

Definition

An augmenting path of G with respect to the matching M is an alternating path joining two distinct exposed vertices with respect to M.

Note that, from the above definition, it follows that if P is an augmenting path w.r.t. M, then $|P \setminus M| = |P \cap M| + 1$.



a, b, c, d - alternating even path

f - alternating odd path

j - augmenting path

g, f, d - alternating odd path

a, b, c, d, e - closed alternating path

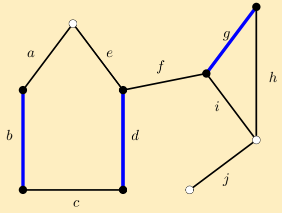
a, b, c, d, f, g, h - augmenting path

Maximum Cardinality Matchings - Berge's Theorem

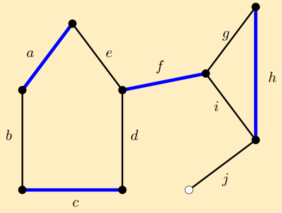
Theorem 5

(Berge, 1959) M is a maximum cardinality matching in the graph G if and only if there is no augmenting path in G w.r.t. M.

Proof. "$\Rightarrow$" Let M be a maximum cardinality matching in G. Suppose that P is an augmenting path in G with respect to M.
Then, $M' = M \Delta P = (P \setminus M) \cup (M \setminus P) \in \mathcal{M}_G$. Indeed, M' can be obtained by interchanging the edges in M with those not in M on the path P:



$P = a, b, c, d, f, g, h$ - augmenting path



$M \Delta P$

Maximum Cardinality Matchings - Berge's Theorem

Furthermore, $|M'| = |P \cap M| + 1 + |M \setminus P| = |M| + 1$, contradicting the choice of M.

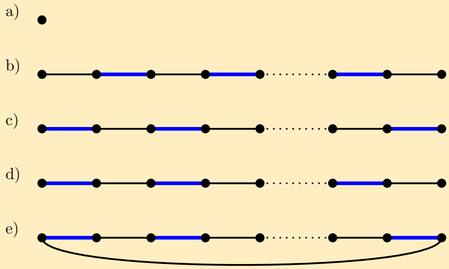
"$\Leftarrow$" Let M be a matching in G with the property that there are no augmenting paths in G with respect to M.

Let M^* a maximum cardinality matching in G. We will prove that $|M^*| = |M|$. Let G' the subgraph spanned by $M \Delta M^*$ in G ($G' = (V, M \Delta M^*)$).

Note that $d_{G'}(v) \leqslant 2, \forall v \in V$ and therefore the connecting components of G' are isolated vertices, paths of length at least one, or cycles. We have the five possibilities (see the following figure; blue edges are from M^*, black edges are from M).

Case b) does not occur since it is an augmenting path with respect to M^*, which is a maximum cardinality matching. Case c) does not appear since it is an augmenting path with respect to M.

Maximum Cardinality Matchings - Berge's Theorem



If we denote by $m_M(C)$ the number of edges from M in the connected component C of G' and by $m_{M^*}(C)$ the number of edges from M^* in the same connected component of G', we obtained that $m_M(C) = m_{M^*}(C)$. Hence,

$$|M \setminus M^*| = \sum_{C \text{ conn. comp. of } G'} m_M(C) =$$

$$= \sum_{C \text{ conn. comp. of } G'} m_{M^*}(C) = |M^* \setminus M|$$

Maximum Cardinality Matchings - Berge's Theorem

Therefore $|M| = |M^*|$. □

We get a strategy to find a maximum cardinality matching:

```
let $M$ be a matching in $G$ (e. g., $M = \varnothing$);
while ($\exists P$ augmenting path w. r. t. $M$) do
    $M \leftarrow M \Delta P$;
end while
```

At each while iteration the cardinality of M increases by 1, therefore in at most $n/2$ iterations we obtain a matching without augmenting paths, that is of maximum cardinality. It remains to implement the while loop in polynomial time complexity. This was firstly done by Edmonds (1965).

Maximum Cardinality Matchings - Hopcroft-Karp Algorithm

Lemma 1

Let $M, N \in \mathcal{M}_G$, $|M| = r$, $|N| = s$ and $s > r$. Then in $M \Delta N$ there are at least $s - r$ vertex-disjoint augmenting paths with respect to M.

Proof. Let $G' = (V, M \Delta N)$ and C_i $(i = \overline{1,p})$ be the connected components of G'. For each $1 \leqslant i \leqslant p$, we denote by $\delta(C_i)$ the difference between the number of edges of N in C_i and the number of edges of M in C_i:

$$\delta(C_i) = |E(C_i) \cap N| - |E(C_i) \cap M|.$$

Note that, since M and N are matchings, C_i are paths or circuits. Hence $\delta(C_i) \in \{-1, 0, 1\}$. $\delta(C_i) = 1$ if and only if C_i is an augmenting path with respect to M.

Maximum Cardinality Matchings - Hopcroft-Karp Algorithm

Proof cont'd. Since

$$\sum_{i=1}^{p} \delta(C_i) = |N \setminus M| - |M \setminus N| = s - r,$$

it follows that there are at least $s - r$ connected components of G' with $\delta(C_i) = 1$, that is, there are at least $s - r$ vertex-disjoint augmenting paths contained in $M \Delta N$. $\square$

Lemma 2

If $\nu(G) = s$ and $M \in \mathcal{M}_G$ with $|M| = r < s$, then there exists in G an augmenting path with respect to M of length at most $2\lceil r/(s-r)\rceil + 1$.

Proof. Let $N \in \mathcal{M}_G$ with $|N| = s = \nu(G)$.

Maximum Cardinality Matchings - Hopcroft-Karp Algorithm

Proof cont'd. By Lemma 1, there are $s - r$ edge-disjoint augmenting paths (vertex-disjoint paths are also edge-disjoint) contained in $M \Delta N$. It follows that at least one of them has at most $\lceil r/(s-r) \rceil$ edges from M. The length of this augmenting path is at most $2\lceil r/(s-r) \rceil + 1$. □

Definition

Let $M \in \mathcal{M}_G$. A minimum augmenting path *with respect to M in G is an augmenting path of minimum length over all augmenting paths with respect to M in G.*

Lemma 3

Let $M \in \mathcal{M}_G$, P a minimum augmenting path w. r. t. M, and P' an augmenting path w. r. t. $M \Delta P$. Then $|P'| \geqslant |P| + 2|P \cap P'|$.

Maximum Cardinality Matchings - Hopcroft-Karp Algorithm

Proof. Let $N = (M\Delta P)\Delta P'$. Then $M\Delta N = P\Delta P'$ and $|N| = |M| + 2$. By Lemma 1, there are two edge-disjoint augmenting paths w. r. t. M, P_1 and P_2, contained in $M\Delta N$. Since P is a minimum augmenting path w.r.t. M, we have $|P\Delta P'| \geqslant |P_1| + |P_2| \geqslant 2|P|$ and therefore $|P| + |P'| - 2|P \cap P'| \geqslant 2|P|$. $\square$

Let us consider the following algorithm:

```
$M_0 \leftarrow \varnothing$; $i = 0$;
while ($\exists$ augmenting paths w. r. t. $M$) do
    let $P_i$ a minimum augmenting path w.r.t. $M_i$;
    $M_{i+1} \leftarrow M_i \Delta P_i$;
    $i++$;
end while
```

Let $P_0, P_1, \ldots, P_{\nu(G)-1}$ be the sequence of minimum augmenting paths constructed by this algorithm.

Maximum Cardinality Matchings - Hopcroft-Karp Algorithm

Lemma 4

a) $|P_i| \leqslant |P_{i+1}|$ and $|P_i| = |P_{i+1}|$ if and only if P_i and P_{i+1} are vertex-disjoint, $\forall 1 \leqslant i \leqslant \nu(G) - 2$.
b) $\forall i < j < \nu(G) - 1)$, if $|P_i| = |P_j|$, then P_i and P_j are vertex-disjoint.

Proof. a) By taking $P = P_i$ and $P' = P_{i+1}$ in Lemma 3, we get $|P_{i+1}| \geqslant |P_i| + 2|P_i \cap P_{i+1}| \geqslant |P_i|$. The equality holds if and only if P_i and P_{i+1} are edge-disjoint, which implies that they are vertex-disjoint (since they are alternating paths).
b) follows by successively applying a) $\square$

Maximum Cardinality Matchings - Hopcroft-Karp Algorithm

Theorem 6

(Hopcroft, Karp, 1973) Let G be a graph and $\nu(G) = s$. The number of distinct integers in the sequence $|P_0|, |P_1|, \ldots, |P_{s-1}|$ (P_i are the minimum augmenting paths constructed by the above algorithm) is not greater than $2\lfloor\sqrt{s}\rfloor + 2$.

Proof. Let $r = \lceil s - \sqrt{s}\rceil$. Then $|M_r| = r$ and

$$|P_r| \leqslant 2\lceil r/(s-r)\rceil + 1 = 2\lceil\lceil s - \sqrt{s}\rceil/(s - \lceil s - \sqrt{s}\rceil)\rceil + 1 < 2\lfloor\sqrt{s}\rfloor + 3.$$

Hence, for every $i < r$, $|P_i|$ is one of $\lfloor\sqrt{s}\rfloor + 1$ odd integers not greater than $2\lfloor\sqrt{s}\rfloor + 1$.
In the sub-sequence $|P_r|, \ldots, |P_{s-1}|$ there are at most $s - r \leqslant \lfloor\sqrt{s}\rfloor + 1$ distinct integers. It follows that in the sequence $|P_0|, |P_1|, \ldots, |P_{s-1}|$ there are no more than $2\lfloor\sqrt{s}\rfloor + 2$ distinct integers. $\square$

Maximum Cardinality Matchings - Hopcroft-Karp Algorithm

$$2\lceil\lceil s-\sqrt{s}\rceil/(s-\lceil s-\sqrt{s}\rceil)\rceil+1<2\lfloor\sqrt{s}\rfloor+3 \Leftrightarrow$$

$$2\lceil\lceil s-\sqrt{s}\rceil/(s-\lceil s-\sqrt{s}\rceil)\rceil<2\lfloor\sqrt{s}\rfloor+2 \Leftrightarrow$$

$$\lceil s-\sqrt{s}\rceil/(s-\lceil s-\sqrt{s}\rceil)\leqslant\lfloor\sqrt{s}\rfloor+1 \Leftrightarrow$$

$$(s-\lfloor\sqrt{s}\rfloor)/\lfloor\sqrt{s}\rfloor\leqslant\lfloor\sqrt{s}\rfloor+1 \Leftrightarrow s\leqslant\lfloor\sqrt{s}\rfloor^2+2\lfloor\sqrt{s}\rfloor$$

If $\lfloor\sqrt{s}\rfloor=k\in\mathbb{N}$, then $k\leqslant\sqrt{s}<k+1$, hence $s=(\sqrt{s})^2<k^2+2k+1$ which is equivalent with $s=(\sqrt{s})^2\leqslant k^2+2k$ - the last inequality from above.

If the above algorithm is decomposed in phases such that in each phase a maximal (w.r.t. inclusion) set of vertex-disjoint minimum augmenting paths is found, then - by Lemma 4 - the length of the minimum augmenting paths in the next phase will not decrease (otherwise, the set of minimum augmenting paths constructed in the current stage is not maximal). By Theorem 6, the number of phases is not greater than $2\lfloor\sqrt{\nu(G)}\rfloor+2$.

Maximum Cardinality Matchings - Hopcroft-Karp Algorithm

Hence we have the following algorithm to find a maximum cardinality matching in G:

```
$M \leftarrow \varnothing$;
repeat
   find $\mathcal{P}$ a maximal set of vertex-disjoint minimum augmenting paths
   w.r.t. $M$;
   for ($P \in \mathcal{P}$) do
      $M \leftarrow M \Delta P$;
   end for
until ($\mathcal{P} = \varnothing$)
```

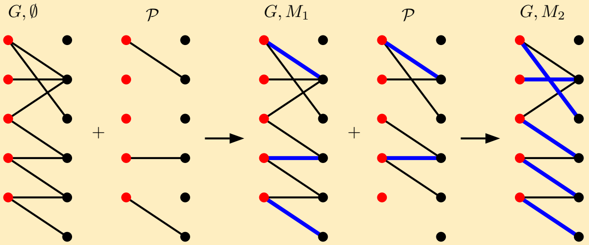
The time complexity of the above algorithm is $\mathcal{O}(\sqrt{n}A)$, where A is the time complexity of the determination of the set $\mathcal{P}$.

In the case of bipartite graphs, Hopcroft and Karp shown that this can be obtained in $\mathcal{O}(n+m)$ time and therefore the entire algorithm has $\mathcal{O}(m\sqrt{n})$ running time.

Maximum Cardinality Matchings - Hopcroft-Karp Algorithm

This result has been extended to arbitrary graphs by Micali and Vazirani (1980) using an elaborate data structure to maintain the labels associated to vertices in order to construct minimum augmenting paths.
Consider the case of bipartite graphs: $G = (S, T; E)$ and $M \in \mathcal{M}_G$. Starting from one of the classes, say S, we consider the set of initial extremities of augmenting paths $S \cap E(M)$. From each such vertex we start, in parallel, the construction of alternating paths in a bfs manner. The first augmenting path obtained stop the construction, giving the minimum length of an augmenting path. The set $\mathcal{P}$ is obtained using the labels and the adjacency lists in $\mathcal{O}(n + m)$.
Details are omitted, an example is given on the next slide.

Maximum Cardinality Matchings - Hopcroft-Karp Algorithm



Flow network

A flow network (transportation network) with source s and sink t is a tuple $R = (G, s, t, c)$ where:

- $G = (V, E)$ is a digraph,
- $s, t \in V$; $s \neq t$; $d^+_G(s) > 0$; $d^-_G(t) > 0$,
- $c : E \to \mathbb{R}_+$; $c(e)$ is the capacity of the arc e.

We will suppose that $V = \{1, 2, \ldots, n\}$ $(n \in \mathbb{N}^*)$ and $|E| = m$. We extend the function c to $c : V \times V \to \mathbb{R}_+$ by

$$c((i,j)) = \begin{cases} c(ij), & \text{if } ij \in E \\ 0, & \text{otherwise} \end{cases}$$

and will denote $c((i,j)) = c_{ij}$.

Flow

Definition

A flow in $R = (G, s, t, c)$ *is a function* $x : V \times V \to \mathbb{R}$ *s. t.*

(i) $0 \leqslant x_{ij} \leqslant c_{ij}, \forall (i,j) \in V \times V$,

(ii) $\sum\limits_{j \in V} x_{ji} - \sum\limits_{j \in V} x_{ij} = 0, \forall i \in V \setminus \{s, t\}$.

Remarks

- If $ij \in E$ then x_{ij} is referred as the flow (transported) on ij.
- Constraints (i) require that the flow on each arc is non-negative and does not exceed the capacity.
- Constraints (ii), (equilibrium constraints), require that the sum of flows on the arcs entering vertex i is equal to the sum of flows on the arcs leaving i (flow in equals flow out).

Flow value

- Note that, by the way the function c have been extended to $V \times V$, the constraints (i) implies that $x_{ij} = 0$ whenever $ij \notin E$. Hence, a flow is in fact a function on E. We prefer its extension on $V \times V$ to simplify the notations.

Let x be a flow in $R = (G, s, t, c)$. If we add all constraints (ii) (for $i \in V \setminus \{s, t\}$) we get

$$0 = \sum_{i \neq s,t} \left(\sum_{j \in V} x_{ji} - \sum_{j \in V} x_{ij} \right) = \sum_{i \neq s,t} \sum_{j \neq s,t} x_{ji} - \sum_{i \neq s,t} \sum_{j \neq s,t} x_{ij} +$$

$$+ \sum_{i \neq s,t} x_{si} + \sum_{i \neq s,t} x_{ti} - \sum_{i \neq s,t} x_{is} - \sum_{i \neq s,t} x_{it} =$$

Flow value

$$= \left(\sum_{i\in V} x_{si} - \sum_{i\in V} x_{is}\right) - \left(\sum_{i\in V} x_{it} - \sum_{i\in V} x_{ti}\right),$$

Definition

The value of the flow x in $R = (G, s, t, c)$ is

$$v(x) = \sum_{i\in V} x_{it} - \sum_{i\in V} x_{ti}.$$

In words, $v(x)$ is the net flow reaching the sink of the network or, as we proved above, the net flow leaving the source of the network.
Note that in any network $R = (G, s, t, c)$ there is a flow: x^0, the null flow, with $x^0_{ij} = 0$, $\forall ij \in E$ and $v(x^0) = 0$.

Maximum flow problem

Maximum Flow Problem: Given $R = (G, s, t, c)$ a flow network, find a flow of maximum value.

The maximum flow problem can be viewed as an LP problem:

$$\begin{aligned}
\max \quad & v \\
& \sum_{j \in V} x_{ji} - \sum_{j \in V} x_{ij} = 0, \forall i \neq s, t \\
& \sum_{j \in V} x_{js} - \sum_{j \in V} x_{sj} = -v \\
& \sum_{j \in V} x_{jt} - \sum_{j \in V} x_{tj} = v \\
& 0 \leqslant x_{ij} \leqslant c_{ij}, \forall ij \in E
\end{aligned}$$

However, we will consider a direct combinatorial approach which is important when some integrality constraints are imposed to the variables (flows on the directed edges).

Exercises for the next seminar

Exercise 1. Let X be a finite set, $X_1, \ldots, X_n \subseteq X$, and $d_1, d_2, \ldots d_n \in \mathbb{N}$. Prove that there are n disjoint subsets $Y_i \subseteq X_i$, $|Y_i| = d_i$, $\forall i = \overline{1, n}$ if and only if

$$\left|\bigcup_{i \in I} X_i\right| \geqslant \sum_{i \in I} d_i,$$

for all $I \subseteq \{1, \ldots, n\}$.

Exercise 2. Every p-regular bipartite graph has a perfect matching ($p \geqslant 1$).

Exercise 3. Let $G = (S, T; E)$ a bipartite graph. Use Hall's theorem to prove that, for every $0 \leqslant k \leqslant |S|$, G has a matching of cardinality at least $|S| - k$ if and only if $|N_G(A)| \geqslant |A| - k$, $\forall A \subseteq S$.

Exercises for the next seminar

Exercise 4. Using Tutte's theorem show that a 2-edge connected, 3-regular graph has perfect matching.

Exercise 5. Let $G = (V, E)$ a graph. A subset $A \subseteq V$ is called *m-independent* if G has a matching M which saturates all the vertices from A. Show that, for any two *m-independent* sets A and B with $|A| < |B|$, we can find a vertex $b \in B \setminus A$ such that $A \cup \{b\}$ is also *m-independent* ($\Rightarrow$ all maximal *m-independent* sets have the same cardinality).

Exercise 6. Let $T = (V, E)$ a rooted tree; we denote by r its root and by $parent(v)$ the direct ancestor of any $v \neq r$. A matching M of T is called *proper* if any exposed vertex $v \neq r$ has a brother w such that $w\, parent(v) \in M$.

(a) Show that any *proper* matching is a maximum cardinality matching.

(b) Devise an algorithm for finding a *proper* matching in $\mathcal{O}(n)$ time

Exercises for the next seminar

Exercise 7. Let $G = (V, E)$ be a p-regular bipartite graph ($p \geqslant 1$) and consider the following algorithm

```
for ($e \in E$) do
    $a(e) \leftarrow 1$;
end for
$E^+ \leftarrow \{e \in E \,:\, a(e) > 0\}$;
while ($G^+ = (V, E^+)$ contains a cycle $C$) do
    let $C = M_1 \cup M_2$, where $M_1$ and $M_2$ are matchings with $a(M_1) \geqslant a(M_2)$;
    // for any $F \subseteq E$, $a(F) = \sum_{e \in F} a(e)$;
    for ($e \in E(C)$) do
        if ($e \in M_1$) then
            $a(e)++$;
        else
            $a(e)--$;
        end if
    end for
    $E^+ \leftarrow \{e \in E \,:\, a(e) > 0\}$;
end while
return $E^+$;
```

Exercises for the next seminar

Exercise 7. (cont'd) Let $f(E^+) = \sum_{e \in E^+} a^2(e)$. Prove that

(a) after each while iteration $f(E^+)$ is an integer number and increases with at least $|C|$;

(b) after each while iteration, for every $u \in V$, $\sum_{uv \in E^+} a(uv) = p$;

(c) the algorithm doesn't end as long as there are edges e with $0 < a(e) < p$; at the end of the algorithm $a(e) = p$, $\forall e \in E^+$, and E^+ is a perfect matching in G;

(d) the while loop ends, at the end of the algorithm $f(E^+) = np^2/2 = pm$, and the total length of all cycles is at most pm;

(e) all the cycles can be found in $\mathcal{O}(\sum_{C \text{ cycle}} |C|)$ time complexity using dfs traversals;

(f) the overall time complexity of the algorithm is $\mathcal{O}(pm)$.

Exercises for the next seminar

Exercise 8. Let $G = (S, T; E)$ be a non-null bipartite graph. Prove that the following are equivalent:

(i) $G - \{x, y\}$ has a perfect matching, $\forall x \in S, \forall y \in T$.

(ii) G is connected and every edge of G belongs to a perfect matching.

(iii) $|S| = |T|$, and $\varnothing \neq A \subsetneq S$, $|N_G(A)| > |A|$.

Exercise 9. Let $G = (V, E)$ be a connected graph which has a perfect matching. Devise (and prove its correctness) an $\mathcal{O}(|V| + |E|)$ time complexity algorithm that constructs a spanning tree T of G such that $V(T)$ admits a bipartition in two stable sets of maximum cardinality in T.

(Partially) solved exercises

Exercise 1. Solution.

- Let $G = (S, X, E)$ be the following bipartite graph: S is composed of d_i copies of subsets X_i, for each $i = \overline{1, n}$; for each $x \in X$ and for each $X_i' \in S$ (a copy of X_i) we add xX_i' to E if and only if $x \in X_i$.
- There exist the sets $(Y_i)_{1 \leqslant i \leqslant n}$ if and only if there exists a matching that saturates S (can you explain why?), which, by Hall's theorem is equivalent with

$$\forall A \subseteq S, |A| \leqslant |N_G(A)| \iff$$

$$\iff \forall (d_i')_{1 \leqslant i \leqslant n} \subseteq \mathbb{N}, d_i' \leqslant d_i, \forall i = \overline{1, n} \text{ and}$$

$$\sum_{i=1}^{n} d_i' \leqslant \left| \bigcup_{i=\overline{1,n}, d_i' \neq 0} X_i \right|.$$

Taking $I = \{i \ : \ d_i' \neq 0\}$ we get the required inequality.

(Partially) solved exercises

Exercise 2. Solution. Let $G = (S, T; E)$ be a bipartite, p-regular graph.

- It easy to prove that $|S| = |T|$ (how?).
- Thus will be enough to prove that there exists a matching that saturates S.
- Let $A \subseteq S$; there are $k|A|$ edges between A and $N_G(A)$, hence $k|A| \leqslant k|N_G(A)|$ which is the total number of edges going outside $N_G(A)$.
- We get $|A| \leqslant |N_G(A)|$ and, by using the Hall's theorem we have the required conclusion.

(Partially) solved exercises

Exercise 3. Solution.

- Let's define a new graph $G' = (S, T \cup W; E')$, be adding k new nodes to T ($W = \{x_1, x_2, \dots x_k\}$); each of these nodes will be adjacent with all the nodes from S: $E' = E \cup \{ux_i : u \in S, 1 \leqslant i \leqslant k\}$).
- Obviously, G contains a matching of cardinal at least $|S| - k$ if and only if, G' contains a matching which saturates all the vertices from S (why?).
- In G' there exists a matching M', with $|M'| = |S|$ if and only if (why?):

  $$|N_{G'}(A)| \geqslant |A|, \forall A \subseteq S$$

  which is equivalent with

  $$|N_G(A)| + k \geqslant |A|, \forall A \subseteq S,$$

  since $N_{G'}(A) = N_G(A) \cup W$ and $|N_{G'}(A)| = |N_G(A)| + k$.

(Partially) solved exercises

Exercise 4. Solution. Let G be a 2-edge connected, 3-regular graph

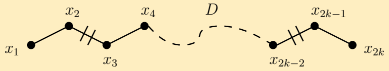
- We will show that $q(G-S) \leqslant |S|, \forall S \subseteq V$.
- G has even order as (why?); we may assume $S \neq \varnothing$.
- Let H_i be an odd connected component of $G-S$; since G is connected, we have an edge $e_1 = u_1v_1$ with $u_1 \in S$ and $v_1 \in V(H_i)$.
- $G - e_1$ is also connected (why?), hence we have another edge $e_2 = u_2v_2 \neq e_1$, with $u_2 \in S, v_2 \in V(H_i)$.
- Let us suppose that e_1 and e_2 are the only edges that link S with H_i, then:
  $$2|E(_iH)| = \sum_{v \in V(H_i)} d_{H_i}(v) \overset{\text{why?}}{=} 3|H_i| - 2 \equiv 1 (mod\ 2), \text{ contradiction.}$$
- Thus, every odd connected component of $G-S$ is linked by at least three edges with S:
  $$3|S| = \sum_{x \in S} d_G(x) \geqslant 3q(G-S).$$

(Partially) solved exercises

Exercise 6. Solution.

(a) Let M be a proper matching and suppose on the contrary that $|M| < \nu(T)$.

- By Berge's theorem, T contains an M- augmenting path D.
- Such a path has an odd length $\geqslant 3$ (why?).
- Suppose that $V(D) = \{x_1, x_2, \ldots x_{2k}\}$, $x_1, x_{2k} \in E(M)$ and $x_2 = parent(x_1)$, $x_3 = parent(x_2)$.
- Since $x_1 \in S(M)$, there exists y a brother of x_1 and $x_2 y \in M$ - contradiction (why?).



(Partially) solved exercises

(b) We use an in-depth traversal (why?) of the tree, which constructs a proper matching:

```
$M \leftarrow \varnothing$;
$E(M) \leftarrow V(T)$;
$ProperMatching(r)$;
```

- where *ProperMatching* is recursively described below:

```
$ProperMatching(v)$; {
   for($x \in \mathcal{A}(v)$) {
      $ProperMatching(x)$;
      if ($v, x \in E(M)$) {
         $M \leftarrow M \cup \{vx\}$;
         $E(M) \leftarrow E(M) \setminus \{x, v\}$;
      }
   }
}
```

(Partially) solved exercises

Exercise 7. Solution.

(a) Initially $f(E^+) = m \in \mathbb{Z}_+$; after an while iteration $f(E^+)$ increases by:

$$\sum_{e \in M_1} \left[(a(e)+1)^2 - a^2(e)\right] + \sum_{e \in M_2} \left[(a(e)-1)^2 - a^2(e)\right] =$$

$$\sum_{e \in M_1} [2a(e)+1] + \sum_{e \in M_2} [-2a(e)+1] =$$

$$= 2\left[a(M_1) - a(M_2)\right] + |C| \geqslant |C|, a(M_i) \in \mathbb{Z}.$$

(b) Obvious. Why?

(c) Having an edge e_1 with $0 < a(e_1) < p$ we can perform the following process: suppose that $e_1 = u_1u_2 \in E^+$; we know that there exists an edge $e_2 = u_2u_3 \in E^+$ with $0 < a(e_2) < p$ (why?), and so on until we reach a vertex, u_i, already visited and we close a cycle with edges from E^+.

(Partially) solved exercises

(d) At the end of the algorithm $|E^+| = n/2$ and $f(E^+) = np^2/2 = pm$ (why?). In each step $f(E^+)$ increases by at least $|C|$ hence the total increasing in $f(E^+)$ is $(pm - m)$ and the total length of all cycles is less than pm.

(e)

```
for $v \in V$ do
    $parent(u) \leftarrow -1$;
end for
let $e = uv \in E^+$ with $0 < a(e) < p$; $parent(u) \leftarrow 0$; $parent(v) \leftarrow u$;
create stack $S$ containing $v$;
while $S \neq \varnothing$ do
    $x \leftarrow top(S)$;
    if $((y \leftarrow next[A(u)]) \neq NULL)$ then
        if $parent(y) < 0$ then
            $parent(v) \leftarrow u$; $push(S, v)$;
        else
            return $y$;
        end if
    else
        $delete(S, x)$;
    end if
end while
```

- Using the returned y and the $parent(\cdot)$ array we can find the wanted cycle. Note that if there is no edge $e \in E^+$ with $0 < a(e) < p$, it means that $G^+ = (V, E^+)$ is acyclic (why?).

(Partially) solved exercises

Exercise 8. Solution.

- (i) $\Rightarrow$ (ii) If G is not connected then for one of its connected components, G', $0 < |V(G') \cap S| \leqslant |V(G') \cap T|$. Take $u \in V(G') \cap S$, $v \in T \setminus V(G')$, and M, a perfect matching in $G - \{u, v\}$; $M \cap E(G')$ cannot saturate all the vertices in G' (why?).
- (ii) $\Rightarrow$ (iii) Since G has a perfect matching, $|S| = |T|$ (why?). From Hall's theorem $|N_G(A)| \geqslant |A|$.
- Suppose on the contrary that $|N_G(A)| = |A|$, for a certain $\varnothing \neq A \subsetneq S$.
- $S \cup T \neq A \cup N_G(A)$ (why?), hence there exist two adjacent vertices $u \in N_G(A)$ and $v \in S \setminus A$.

(Partially) solved exercises

- Let M be a perfect matching that contains uv; M contains also $|A|$ edges from A to $N_G(A)$, therefore v it's saturated in M by an edge different from uv (why?) - contradiction.
- (iii) $\Rightarrow$ (i) We use the Hall's theorem for the bipartite graph $G - \{x, y\}$. For every $A \subseteq A - \{x\}$

$$|N_{G-\{x,y\}}(A)| \geqslant |N_G(A)| - 1 \geqslant |A|.$$

(Partially) solved exercises

Exercise 9. Solution.

- Let M be a perfect matching in $G = (V, E)$.
- First reorder the neighbors in the adjacency lists such that the edges from M come first. Let the following *dfs* traversal procedure be

```
dfs($v$, $i$)
$visited[v] \leftarrow i$;
for ($u \in \mathcal{A}(v)$) do
    if ($visited[u] = -1$) then
        $E' \leftarrow E' \cup \{uv\}$; dfs($u$, $1 - i$);
    end if
end for
```

(Partially) solved exercises

Exercise 9. Solution.

- The main algorithm is:

```
for ($v \in V$) do
    $visited[v] \leftarrow -1$;
end for
let $v_0 \in V$;
dfs($v_0$, 0);
```

- Define $A = \{v \in V : visited[v] = 0\}$ and $B = \{v \in V : visited[v] = 1\}$; A and B are stable sets (in $T = (V, E'))$ of equal cardinality (why?).
- (In other words, any path from from the root to a leaf is an alternating path: one of them starts and ends with edges from M and the remaining path starts with edges from outside M and ends with edges from M.)