3. Matching in bipartite and general graphs

Let G be a graph. A **matching** M in G is a set of nonloop edges with no shared endpoints.

Let M be a matching in G. A vertex of G is **saturated** by M if it is incident to an edge in M; otherwise the vertex is **unsaturated** by M.

A matching that saturates all vertices in G is a **perfect matching**.

A matching in G is **maximal** if it is not a subset of a matching of larger cardinality. A **maximum** matching is a matching of the maximum size among all matchings in G.

In this lecture we study matchings in bipartite and general graphs. Our main topics of interest are:

1. Necessary and sufficient conditions for a graph to have a perfect matching (equivalently, a 1-factor).
- ▷ Hall's matching theorem for bipartite graphs.
- ▷ Tutte's 1-factor theorem for general graphs.

2. The size of a maximum matching in a graph.
- ▷ König-Egerváry theorem for bipartite graphs.
- ▷ Berge-Tutte formula for general graphs.

3. Efficient algorithms for finding a maximum matching in a given graph G.

4. A small digression: stable matchings.

Not considered: weighted matchings; see [Jun, Chapter 13].

Sources for this lecture

The material for this lecture has been prepared with the help of [Wes, Chapter 3], [Jun, Section 7.2 and Chapter 12] and [Die, Chapter 2].

[Die] R. Diestel, *Graph Theory, 2nd Edition*, Graduate Texts in Mathematics 173, Springer, New York NY, 2000.

[Lov] L. Lovász and M. D. Plummer, *Matching Theory*, North-Holland, Amsterdam, 1986.

The monograph [Lov] is a comprehensive reference to matching theory.

Symmetric difference

For convenience, we identify a matching M in G with the graph induced by the edges in M.

Let G and H be graphs. The **symmetric difference** $G \oplus H$ is the graph defined by

$$V(G \oplus H) := V(G) \cup V(H),$$
$$E(G \oplus H) := (E(G) - E(H)) \cup (E(H) - E(G)).$$

Lemma A.6 *Every component of the symmetric difference of two matchings is either a path or an even cycle.*

(Note that some of the paths above may have length zero.)

Augmenting paths and maximum matchings

Let G be a graph and let M be a matching in G.

An M**-alternating path** is a nontrivial path in G whose edges are alternately in M and not in M. An M-alternating path whose endpoints are unsaturated by M is an M**-augmenting path**.

The following theorem characterizing maximum matchings in an arbitrary graph is due to Berge (1957).

Theorem A.26 *A matching M in G is a maximum matching if and only if there exists no M-augmenting path.*

Proof: ($\Rightarrow$) Suppose that P is an M-augmenting path. Then, $M \oplus P$ is a matching with cardinality $|M| + 1$. Thus, M is not a maximum matching.

($\Leftarrow$) Suppose that M is not a maximum matching. We must prove that there exists an M-augmenting path P. Since M is not a maximum matching, there exists a matching M' with $|M'| = |M| + 1$. By Lemma A.6, every component of $M \oplus M'$ is either an even cycle or a path. Moreover, $M \oplus M'$ consists of an odd number of edges with one more edge from M' than from M. Since each even cycle in $M \oplus M'$ must use equally many edges from M and from M', we have that $M \oplus M'$ contains an odd-length path P, which is both M- and M'-alternating. Thus, P is an M-augmenting path since both of its endpoints must be incident with an edge in M'. $\square$

Matching in bipartite graphs

Recall the theorem of König (1931) and Egerváry (1931):

Theorem A.27 *For a bipartite graph G, the cardinality of a maximum matching in G is equal to the minimum size of a vertex cover of G.*

Thus, given a matching M in a bipartite graph G, we can prove that M is a maximum matching by exhibiting a vertex cover C that satisfies $|M| = |C|$.

Hall's theorem

Let G be a bipartite graph and let X, Y be a bipartition of its vertex set. Let $S \subseteq X$. We denote by $N(S)$ the vertices in G adjacent to a vertex in S.

The following theorem is due to P. Hall (1935).

Theorem A.28 *There exists a matching in G that saturates every vertex in X if and only if $|N(S)| \geq |S|$ for all $S \subseteq X$.*

Proof: ($\Rightarrow$) Let M be a matching that saturates X and let $S \subseteq X$. Since $N(S)$ must contain the vertices matched to the vertices S in M, we have $|N(S)| \geq |S|$.

Proof: ($\Leftarrow$) We apply the König-Egerváry theorem. Suppose G has no matching that saturates every vertex in X. Then, $\alpha'(G) < |X|$. Let $C \subseteq X \cup Y$ be a vertex cover of size $\alpha'(G)$ and put $S := X - C$. We have $|N(S)| \leq |C \cap Y|$ since all edges with one endpoint in S must have the other endpoint in $C \cap Y$ by definition of a vertex cover. Since $|C| = |C \cap X| + |C \cap Y|$ and $|C| < |X| = |C \cap X| + |S|$, we have $|N(S)| \leq |C| - |C \cap X| < |S|$. $\square$

The following is an example application of Hall's theorem.

Theorem A.29 *A regular bipartite graph has a perfect matching.*

Proof: Let G be a k-regular bipartite graph. Every bipartition X, Y of G satisfies $k|X| = k|Y|$, hence $|X| = |Y|$. Let $S \subseteq X$. Every vertex in $N(S)$ is adjacent to at most k vertices in S. Since there are $k|S|$ edges with one endpoint in S and the other in $N(S)$, we must have $k|N(S)| \geq k|S|$. Thus, $|N(S)| \geq |S|$. $\square$

Remarks

- The condition $|N(S)| \geq |S|$ is often called **Hall's matching condition** or simply **Hall's condition**.
- Hall's theorem is perhaps the most often applied result from graph theory. In the exercises we will consider some applications of Hall's theorem.
- Many proofs of Hall's theorem are known. For example, [Die, Chapter 2] contains three independent proofs (one of which was just given); in a subsequent lecture/exercise we will see a proof using network flow theory.

An algorithm for maximum matching

- Theorem A.26 gives us a way to determine a maximum matching in a graph: given a (possibly empty) matching M, search for a M-augmenting path. If an M-augmenting path P exists, we obtain a larger matching $M \oplus P$; otherwise M is a maximum matching.
- At most $n(G)/2$ iterations suffice to produce a maximum matching.
- Finding augmenting paths in a bipartite graph is straightforward; the general case is more subtle, we will sketch the procedure at the end of this lecture.

Augmenting paths: the bipartite case

Procedure BIPARTITE-AUGMENT takes as input a bipartite graph G and a matching M in G. The output is either an M-augmenting path or a vertex cover of G of size $|M|$.

The procedure uses the following local data structures:

X, Y	an arbitrary bipartition of G;
$S \subseteq X$; $T \subseteq Y$	contain vertices that are unsaturated or reachable from an unsaturated vertex in X by an M-alternating path;
$p[v]$	is either undefined or contains a vertex that follows $v \in V(G)$ in an M-alternating path terminating at an unsaturated vertex of X.

Procedure BIPARTITE-AUGMENT(G, M)

(1) $X, Y \leftarrow$ a bipartition of G;
(2) **for each** $v \in V(G)$ **do** $p[v] \leftarrow$ UNDEF **end for**
(3) $S \leftarrow$ the vertices in X unsaturated by M; $T \leftarrow \emptyset$;
(4) **for each** unmarked $x \in S$ **do**
(5)   **for each** $xy \in E(G) - M$ **do**
(6)   **if** y is unsaturated **then**
(7)   output $y, x, p[x], p[p[x]], \ldots$ as an M-augmenting path;
(8)   **halt**
(9)   **end if**
(10) $w \leftarrow$ the vertex for which $yw \in M$;
(11) $p[y] \leftarrow x$; $p[w] \leftarrow y$;
(12) $T \leftarrow T \cup \{y\}$; $S \leftarrow S \cup \{w\}$
(13)   **end for**
(14)   mark x;
(15) **end for**
(16) output $T \cup (X - S)$ as a vertex cover for G

Correctness sketch

Lemma A.7 *The **for**-loop on lines 4–15 maintains the invariants*

(i) the set S contains only vertices in X that are either unsaturated or reachable from an unsaturated vertex in X by an M-alternating path; and

(ii) the set T contains only saturated vertices in Y that are reachable from an unsaturated vertex in X by an M-alternating path; and

(iii) $v, p[v], p[p[v]], \ldots$ is an alternating path from v to an unsaturated vertex in X whenever $p[v] \neq$ UNDEF.

Whenever $T \cup (X - S)$ is output as a vertex cover, the sets S, T are maximal with respect to the conditions in (i) and (ii).

Theorem A.30 *A set $T \cup (X - S)$ output by Procedure* BIPARTITE-AUGMENT *is a vertex cover of cardinality $|M|$.*

Proof: To show that $T \cup (X - S)$ is a vertex cover it suffices to show that there is no edge joining S to $Y - T$. Let $e = uv \in E(G)$, where $u \in X$, $v \in Y$. Suppose $u \in S$. If $e \notin M$, then $v \in T$ after u is marked. If $e \in M$, then $u \in S$ implies that u is reachable from an unsaturated vertex in X by a nontrivial M-alternating path. The edge incident with u in such a path must be e. Hence, $v \in T$.

Because all vertices in T are saturated and no edge $uv \in M$ satisfies $u \in T$ and $v \in X - S$, we have that M contains precisely $|T|$ edges with one endpoint in S and the other in T. The remaining edges in M have one endpoint in $X - S$. Because all vertices in $X - S$ are saturated, $|M| = |T| + |X - S| = |T \cup (X - S)|$. $\square$

- Invariant (iii) in Lemma A.7 shows that any path output by BIPARTITE-AUGMENT is an M-augmenting path. Thus, by Theorem A.30 BIPARTITE-AUGMENT outputs either an M-augmenting path or a vertex cover of size $|M|$.
- The size of a vertex cover is an upper bound on the size of a maximum matching. Thus, M is a maximum matching if and only if BIPARTITE-AUGMENT outputs a vertex cover.
- The correctness proof for procedure BIPARTITE-AUGMENT constitutes an algorithmic proof of the König-Egerváry theorem.

Remarks

- The bipartite matching algorithm based on BIPARTITE-AUGMENT can be implemented to run in time $O(n(G)e(G))$.
- The algorithm of Hopcroft and Karp (1973) solves the maximum matching problem for bipartite graphs in time $O(\sqrt{n(G)}e(G))$; see [Wes, p. 132–134] and [Jun, p. 191, 502].
- In a subsequent lecture/exercise we will use a maximum flow algorithm to compute a maximum matching in a bipartite graph. (This is essentially the Hopcroft-Karp algorithm.)
- Balinsky and Gonzales (1991) describe an $O(n(G)e(G))$ time algorithm for maximum bipartite matching based on strong spanning trees.

Stable matchings

We briefly consider matching with preferences.

Consider the problem of establishing n "stable" marriages between n men and n women.

If man x and woman a are paired with other partners such that x prefers a over his current partner, and a prefers x over her current partner, then the pair x, a is **unstable**.

A perfect matching is **stable** subject to the preference rankings of each man and woman if there exists no unstable pair.

Gale-Shapley proposal algorithm

Stable matchings always exist. The following **proposal algorithm** due to Gale and Shapley (1962) always produces a stable matching.

Input: preference rankings for each of the n men and n women.

Iteration: each man proposes to the highest ranked woman (in his preference) who has not previously rejected him. If each woman receives exactly one proposal, stop and use the resulting matching. Otherwise, every woman proposed says "maybe" to the most attractive proposal and rejects the other proposals, if any.

Theorem A.31 *The Gale-Shapley proposal algorithm always produces a stable matching.*

Proof: Denote by A_i the set of women who receive at least one proposal during iteration i. By the structure of the algorithm, $A_i \subseteq A_{i+1}$ (equality can hold). The algorithm finds a matching when A_i is the set of all women. This must happen either on the round some man x proposes to the last woman on his preference list (because then x has proposed to all the women) or during some preceding round.

The matching output by the algorithm is always stable. To reach a contradiction, suppose that x, a is an unstable pair, i.e. x prefers a over his partner b, and a prefers x over her partner y. Then, by the structure of the algorithm, x has proposed to a before b, so a would have rejected y who she prefers less than x, a contradiction. $\square$

Remarks

- The matching output by the proposal algorithm has the property that, among all possible stable matchings, *every* man is happiest in the matching produced by the proposal algorithm (exercise).
- For women, no such property holds. (Unless the women do the proposing, of course.) For example, it can happen that *every* woman ends up with the least preferred partner.
- The proposal algorithm was actually devised and implemented already in 1952 in the United States to organize the chaos of matching the graduates of medical schools to residency positions.[a]

[a] In 2002, the National Resident Matching Program (www.nrmp.org) offered 22,916 positions to 31,083 applicants.

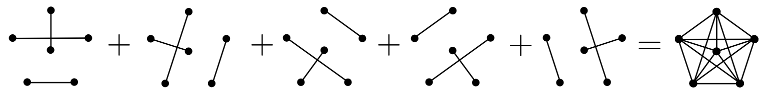
Decomposition and factors

Let G be a graph. A **decomposition** of G is a set of subgraphs of G such that every edge of G occurs in exactly one subgraph.

Let $k \geq 1$. A (k-)**factor** in G is a (k-regular) spanning subgraph. A (k-)**factorization** of G is a decomposition of G into (k-)factors.

A 1-factor in G corresponds to a perfect matching. A 1-factorization of G corresponds to a partition of $E(G)$ into perfect matchings.

Example. A 1-factorization of K_6:



Tutte's 1-factor theorem

Let G be a graph and let $S \subseteq V(G)$. Denote by $G - S$ the graph obtained by deleting the vertices in S from G. Denote by $o(G - S)$ the number of odd components in $G - S$.

The following theorem is due to Tutte (1947).

Theorem A.32 *A graph G has a 1-factor if and only if $o(G - S) \leq |S|$ for every $S \subseteq V(G)$.*

Proof: ($\Rightarrow$) Each odd component of $G - S$ has a vertex matched to a distinct vertex of S. Thus, $o(G - S) \leq |S|$.

($\Leftarrow$) See e.g. [Wes, p. 137–138] or [Jun, p. 356–357]. $\square$

The following theorem demonstrates the use of Tutte's theorem; the theorem is originally due to Petersen (1891).

Theorem A.33 *Every 3-regular graph G without a cut-edge has a 1-factor.*

Proof: We verify Tutte's condition for G. Let $S \subseteq V(G)$. For $S = \emptyset$ the claim holds since each component H satisfies $2e(H) = 3n(H)$, so $n(H)$ must be even. Otherwise, let H be an odd component in $G - S$ and denote by m the number of edges with exactly one endpoint in H. Clearly, $2e(H) = 3n(H) - m$ is even, so m must be odd. We cannot have $m = 1$ since then G would contain a cut-edge, so $m \geq 3$. Thus, each odd component of $G - S$ is connected to S by at least three edges. Since each vertex in S is incident to at most three such edges, $3o(G - S) \leq 3|S|$, which implies Tutte's condition. $\square$

The claim does not hold for an arbitrary 3-regular graph (exercise).

Matching in general graphs

The König-Egerváry theorem does not hold for an arbitrary graph G; consider an odd cycle. Nevertheless, we have $\alpha'(G) \le \beta(G) \le 2\alpha'(G)$ (exercise). Computing $\beta(G)$ for an arbitrary graph G is **NP**-hard.

The following formula is due to Berge (1958) and is known as the **Berge-Tutte formula**:

$$2\alpha'(G) = \min_{S \subseteq V(G)} \{n(G) - (o(G-S) - |S|)\}.$$

Thus, given a matching M in G, we can prove that M is a maximum matching by exhibiting a subset $S \subseteq V(G)$ such that $2|M| = n(G) - (o(G-S) - |S|)$.

Proof: (of the Berge-Tutte formula)

Put $d := \max_{S \subseteq V(G)}\{o(G-S) - |S|\}$. From $S = \emptyset$ we obtain $d \ge 0$. If $d = 0$, Tutte's condition holds for G and we are done. So, suppose $d \ge 1$. It suffices to show that G has a matching M that saturates $n(G) - d$ vertices. (Clearly, at least d vertices are left unsaturated by every matching in G.)

The **join** $G \vee H$ of two graphs G, H is the graph obtained from the disjoint union of G and H by adding the edges $\{uv \ : \ u \in V(G),\ v \in V(H)\}$.

Put $G' := G \vee K_d$. We show that G' has a 1-factor. By deleting from the 1-factor the at most d edges incident with vertices of K_d we obtain a matching in G that saturates at least $n(G) - d$ vertices.

Proof: (continued) We verify Tutte's condition for G'. Let $S' \subseteq V(G')$. Tutte's condition holds for $S' = \emptyset$ because G' is connected and $n(G') = n(G) + d \equiv 0 \pmod 2$. The last equality holds because

$$o(G-S) + |S| \equiv o(G-S) - |S| \equiv n(G) \pmod 2$$

for all $S \subseteq V(G)$.

Next, suppose $V(K_d) - S' \ne \emptyset$. Then, $G' - S'$ is connected. Hence, $o(G' - S') \le 1 \le |S'|$.

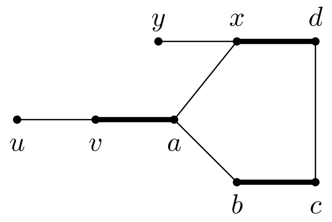
Otherwise we have $V(K_d) \subseteq S'$. Put $S := S' - V(K_d)$. Now,

$$o(G' - S') - |S'| = o(G-S) - (|S| + d) = o(G-S) - |S| - d \le 0.$$

Therefore, $o(G' - S') \le |S'|$ and Tutte's condition holds for G'. $\square$

Augmenting paths: the general case

Consider the graph G below. Edges in M are indicated in bold.



If we search for M-augmenting paths starting at u, then we reach x either using uv, va, ax or using uv, va, ab, bc, cd, dx. Only the latter of these M-alternating paths can be extended to an M-augmenting path. (This situation cannot arise in the bipartite case. Why?)

The previous observation points out the following obstacle:

> In searching for M-augmenting paths starting at an unsaturated vertex u, we may arrive at a vertex x both along an edge in M and along an edge not in M.

Both alternatives must be considered when continuing the search from x, or otherwise we may miss an M-augmenting path in the search.

The straightforward solution of searching through *all* M-alternating paths starting at u is very expensive since there are in general exponentially many such paths.

Edmonds (1965) gave an elegant solution to this problem (which we shall only sketch along the lines in [Wes, p. 142–145]). For a detailed exposition, see [Jun, Section 12.4].

Flower, stem, blossom

Let M be a matching in a graph G and let $u \in V(G)$ be unsaturated.

A **flower** is the union of two M-alternating paths from u to a vertex x such that (i) one path reaches x along an edge in M, the other along an edge not in M; and (ii) the union contains exactly one cycle.

The **stem** of the flower is the maximal common initial path starting at u, the **root** of the flower. The **blossom** of the flower is the odd cycle obtained by deleting the stem (apart from its last vertex). The **base** of the blossom is the vertex common to the stem and blossom.

Note that the base and root of a flower may be equal.

Edmonds' blossom algorithm

Edmonds' blossom algorithm searches for an M-augmenting path starting from an unsaturated vertex u. Whenever a flower is encountered, the corresponding blossom is *contracted* to a single vertex. The contracted vertex is then considered in the search as any other vertex. Repeated contractions can occur.

Let B be an odd cycle in a graph G. We denote by G/B the graph obtained by **contracting** B to a single vertex β, where $\beta \notin V(G)$. The graph G/B is defined by:

$$
\begin{aligned}
V(G/B) &:= (V(G) - V(B)) \cup \{\beta\}, \\
E(G/B) &:= \{uv \;:\; uv \in E(G),\ u, v \in V(G) - V(B)\} \cup \\
&\qquad \cup \{u\beta \;:\; uv \in E(G),\ u \in V(G) - V(B),\ v \in V(B)\}.
\end{aligned}
$$

Similarly, the matching M/B is obtained from M by removing the edges in $E(B)$ and replacing the endpoint in $V(B)$ by β for the edge in M with exactly one endpoint in $V(B)$.

Theorem A.34 *Let M be a matching in a graph G and let B be the blossom of a flower with root u. Then, G contains an M-augmenting path starting at u if and only if G/B contains an M/B-augmenting path starting at u; or, at β if $u \in V(B)$.*

(See [Jun, Lemma 12.4.3] and [Jun, Lemma 12.4.4].)

In what follows we sketch the augmenting path search procedure for a general graph G with a matching M, omitting details how the contraction and expansion of blossoms is done in practice.

We now search from a fixed unsaturated source vertex u (whereas BIPARTITE-AUGMENT searches from all unsaturated vertices at once). To settle the nonexistence of an M-augmenting path we need to consider every unsaturated vertex u.

The search procedure expands two sets $S, T \subseteq V(G)$ that contain vertices reachable from u along even- and odd-length M-alternating paths, respectively. Unless undefined, $p[v]$ contains a vertex that precedes v on an alternating path from u.

When a discovered blossom B is contracted, we put $G \leftarrow G/B$ and $M \leftarrow M/B$. Furthermore, we put $p[v] \leftarrow \beta$ for all $v \in V(G/B)$ such that $p[v] \in V(B)$.

Procedure (sketch) GENERAL-AUGMENT-FROM(G, M, u)

$S \leftarrow \{u\}$; $T \leftarrow \emptyset$;
for each unmarked $x \in S$ **do**
  for each $xy \in E(G) - M$ such that $y \notin T$ **do**
    if y is unsaturated **then**
      output $y, x, p[x], p[p[x]], \ldots$ as an M-augmenting path
        (expanding blossoms if necessary); **halt**
    end if
    if $y \in S$ **then**
      contract the blossom B of the flower defined by
        the paths $y, x, p[x], p[p[x]], \ldots$ and $y, p[y], p[p[y]], \ldots$;
      $p[\beta] \leftarrow p[$the base of $B]$; $S \leftarrow (S - V(B)) \cup \{\beta\}$; **break**
    end if
    $w \leftarrow$ the vertex for which $yw \in M$;
    $p[y] \leftarrow x$; $p[w] \leftarrow y$; $T \leftarrow T \cup \{y\}$; $S \leftarrow S \cup \{w\}$
  end for
  mark x;
end for
output "no M-augmenting path starting at u"

Remarks

- When appropriately implemented, Edmonds' blossom algorithm runs in time $O(n(G)^3)$, see [Jun, p. 380–386].
- Micali and Vazirani (1980) give an algorithm for maximum matching in an arbitrary graph that runs in time $O(\sqrt{n(G)}e(G))$. The analysis of the algorithm appears in Vazirani (1994); for discussion and implementation issues, see the references in [Jun, p. 363].