24.1a. Tutte's proof of his 1-factor theorem

The original proof of Tutte [1947b] of his 1-factor theorem (Corollary 24.1a), with a simplification of Maunsell [1952], and smoothed by Halton [1966] and Lovász [1975d], is as follows.

Suppose that there exist graphs $G = (V, E)$ satisfying the condition, but not having a perfect matching. Fixing V, take such a graph G with G simple and $|E|$ as large as possible. Let $U := \{v \in V \mid v \text{ is adjacent to every other vertex of } G\}$. We show that each component of $G - U$ is a complete graph.

Suppose to the contrary that there are distinct $a, b, c \notin U$ with $ab, bc \in E$ and $ac \notin E$. By the maximality of $|E|$, adding ac to E makes that G has a perfect matching (since the condition is maintained under adding edges). So G has a matching M missing precisely a and c. As $b \notin U$, there exists a vertex d with $bd \notin E$. Again by the maximality of $|E|$, G has a matching N missing precisely b and d. Now each component of $M \triangle N$ contains the same number of edges in M as in N — otherwise there would exist an M- or N-augmenting path, and hence a perfect matching in G, a contradiction. So the component P of $M \triangle N$ containing d is a path starting at d, with first edge in M and last edge in N, and hence ending at a or c; by symmetry we may assume that it ends at a. Moreover, P does not traverse b. Then extending P by the edge ab gives an N-augmenting path, and hence a perfect matching in G — a contradiction.

So each component of $G - U$ is a complete graph. Moreover, by the condition, $G - U$ has at most $|U|$ odd components. This implies that G has a perfect matching, contradicting our assumption.

More proofs were given by Gallai [1950,1963b], Edmonds [1965d], Balinski [1970], Anderson [1971], Brualdi [1971d], Hetyei [1972,1999], Mader [1973], and Lovász [1975a,1979b].

24.1b. Petersen's theorem

The following theorem of Petersen [1891] is a consequence of Tutte's 1-factor theorem (a graph is *cubic* if it is 3-regular):

Corollary 24.1b (Petersen's theorem)**.** *A bridgeless cubic graph has a perfect matching.*

Proof. Let $G = (V, E)$ be a bridgeless cubic graph. By Tutte's 1-factor theorem, we should show that $G - U$ has at most $|U|$ odd components, for each $U \subseteq V$.

Each odd component of $G - U$ is left by an odd number of edges (as G is cubic), and hence by at least three edges (as G is bridgeless). On the other hand, U is left by at most $3|U|$ edges, since G is cubic. Hence $G - U$ has at most $|U|$ odd components. ∎

24.2. Cardinality matching algorithm

The idea of finding an M-augmenting path to increase a matching M is fundamental in finding a maximum-size matching. However, the simple trick for bipartite graphs, of orienting the edges based on the colour classes of the graph, does not extend to the nonbipartite case. Yet one could try to find an M-augmenting path by finding an 'M-alternating walk', but such a walk can run into a loop that cannot simply be deleted. It was Edmonds [1965d] who found the trick to resolve this problem, namely by 'shrinking' the loop (for which he introduced the term 'blossom'). Then applying recursion to a smaller graph solves the problem[1].

Let $G = (V, E)$ be a graph, let M be a matching in G, and let X be the set of vertices missed by M. A walk $P = (v_0, v_1, \ldots, v_t)$ is called *M-alternating* if for each $i = 1, \ldots, t-1$ exactly one of the edges $v_{i-1}v_i$ and v_iv_{i+1} belongs to M. Note that one can find a shortest M-alternating $X - X$ walk of positive length, by considering the auxiliary directed graph $D = (V, A)$ with

$$(24.6) \qquad A := \{(u, v) \mid \exists x \in V : ux \in E, xv \in M\}.$$

Then each M-alternating $X - X$ walk of positive length yields a directed $X - N(X)$ path in D, and vice versa (where $N(X)$ denotes the set of neighbours of X).

An M-alternating walk $P = (v_0, v_1, \ldots, v_t)$ is called an *M-flower* if t is odd, $v_0, \ldots, v_{t-1}$ are distinct, $v_0 \in X$, and $v_t = v_i$ for some even $i < t$. Then the circuit $(v_i, v_{i+1}, \ldots, v_t)$ is called an *M-blossom* (associated with the M-flower).

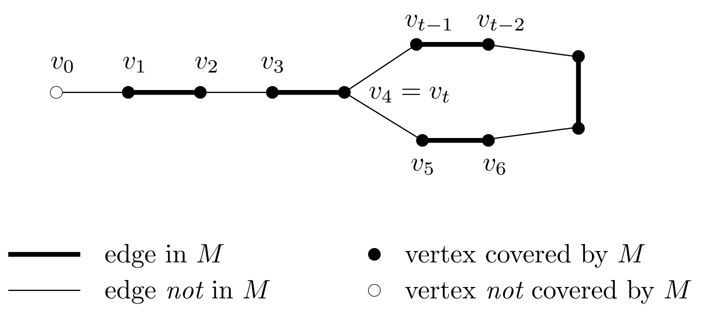

Figure 24.1
An M-flower

The core of the algorithm is the following observation. Let $G = (V, E)$ be a graph and let B be a subset of V. Denote by G/B the graph obtained by *contracting* (or *shrinking*) B to one new vertex, called B. That is, G/B has vertex set $(V \setminus B) \cup \{B\}$, and for each edge e of G an edge obtained from e by replacing any end vertex in B by the new vertex B. (We ignore loops that may arise.) We denote the new edge again by e. (So its ends are modified,

[1] The idea of applying shrinking recursively to matching problems was introduced by Petersen [1891], and was applied in an algorithmic way by Brahana [1917].

but not its name.) We say that the new edge is the *image* (or *projection*) of the original edge.

For any matching M, let M/B denote the set of edges in G/B that are images of edges in M not spanned by B. Obviously, if M intersects $\delta(B)$ in at most one edge, then M/B is a matching in G/B. In the following, we identify a blossom with its set of vertices.

Theorem 24.2. *Let B be an M-blossom in G. Then M is a maximum-size matching in G if and only if M/B is a maximum-size matching in G/B.*

Proof. Let $B = (v_i, v_{i+1}, \ldots, v_t)$.

First assume that M/B is not a maximum-size matching in G/B. Let P be an M/B-augmenting path in G/B. If P does not traverse vertex B of G/B, then P is also an M-augmenting path in G. If P traverses vertex B, we may assume that it enters B with some edge uB that is not in M/B. Then $uv_j \in E$ for some $j \in \{i, i+1, \ldots, t\}$.

(24.7) If j is odd, replace vertex B in P by $v_j, v_{j+1}, \ldots, v_t$.
If j is even, replace vertex B in P by $v_j, v_{j-1}, \ldots, v_i$.

In both cases we obtain an M-augmenting path in G. So M is not maximum-size.

Conversely, assume that M is not maximum-size. We may assume that $i = 0$, that is, $v_i \in X$, since replacing M by $M \triangle EQ$, where Q is the path $(v_0, v_1, \ldots, v_i)$, does not modify the theorem. Let $P = (u_0, u_1, \ldots, u_s)$ be an M-augmenting path in G. If P does not intersect B, then P is also an M/B-augmenting path in G/B. If P intersects B, we may assume that $u_0 \notin B$. (Otherwise replace P by its reverse.) Let u_j be the first vertex of P in B. Then $(u_0, u_1, \ldots, u_{j-1}, B)$ is an M/B-augmenting path in G/B. So M/B is not maximum-size. ▮

Another useful observation is:

Theorem 24.3. *Let $P = (v_0, v_1, \ldots, v_t)$ be a shortest M-alternating $X - X$ walk. Then either P is an M-augmenting path or $(v_0, v_1, \ldots, v_j)$ is an M-flower for some $j \leq t$.*

Proof. Assume that P is not a path. Choose $i < j$ with $v_j = v_i$ and with j as small as possible. So $v_0, \ldots, v_{j-1}$ are all distinct.

If $j - i$ would be even, we can delete $v_{i+1}, \ldots, v_j$ from P so as to obtain a shorter M-alternating $X - X$ walk. So $j - i$ is odd. If j is even and i is odd, then $v_{i+1} = v_{j-1}$ (as it is the vertex matched to $v_i = v_j$), contradicting the minimality of j.

Hence j is odd and i is even, and therefore $(v_0, v_1, \ldots, v_j)$ is an M-flower. ▮

We now describe an algorithm (the *matching-augmenting algorithm*) for the following problem:

(24.8) given: a matching M;
find: an M-augmenting path, if any.

Denote the set of vertices missed by M by X.

(24.9) If there is no M-alternating $X-X$ walk of positive length, there is no M-augmenting path.
If there exists an M-alternating $X-X$ walk of positive length, choose a shortest one, $P=(v_0,v_1,\ldots,v_t)$ say.
Case 1: *P* is a path. Then output P.
Case 2: *P* is not a path. Choose j such that $(v_0,\ldots,v_j)$ is an M-flower, with M-blossom B. Apply the algorithm (recursively) to G/B and M/B, giving an M/B-augmenting path P in G/B. Expand P to an M-augmenting path in G (cf. (24.7)).

The correctness of this algorithm follows from Theorems 24.2 and 24.3. It gives a polynomial-time algorithm to find a maximum-size matching, which is a basic result of Edmonds [1965d].

Theorem 24.4. *Given a graph, a maximum-size matching can be found in time $O(n^2m)$.*

Proof. The algorithm directly follows from algorithm (24.9), since, starting with $M=\emptyset$, one can iteratively apply it to find an M-augmenting path P and replace M by $M\triangle EP$. It terminates if there is no M-augmenting path, whence M is a maximum-size matching.

By using (24.6), path P in (24.9) can be found in time $O(m)$. Moreover, the graph G/B can be constructed in time $O(m)$. Since the recursion has depth at most n, an M-augmenting path can be found in time $O(nm)$. Since the number of augmentations is at most $\frac{1}{2}n$, the time bound follows. ∎

This implies for perfect matchings:

Corollary 24.4a. *A perfect matching in a graph (if any) can be found in time $O(n^2m)$.*

Proof. Directly from Theorem 24.4, as a perfect matching is a maximum-size matching. ∎

24.2a. An $O(n^3)$ algorithm

The matching algorithm described above consists of a series of matching augmentations. Each matching augmentation itself consists of a series of two steps performed alternatingly:

(24.10) finding an M-alternating walk, and
shrinking an M-blossom,

until the M-alternating walk is simple, that is, is an M-augmenting path.

Each of these two steps can be done in time $O(m)$. Since there are at most n shrinkings and at most n matching augmentations, we obtain the $O(n^2m)$ time bound.

If we want to save time we must consider speeding up both the walk-finding step and the shrinking step. In a sense, our description above gives a brute-force polynomial-time method. The $O(m)$ time bound for shrinking gives us time to construct the shrunk graph completely, by copying all vertices that are not in the blossom, by introducing a new vertex for the shrunk blossom, and by introducing for each original edge its 'image' in the shrunk graph. The $O(m)$ time bound for finding an M-alternating walk gives us time to find, after any shrinking, a walk starting just from scratch.

In fact, we cannot do much better if we explicitly construct the shrunk graph. But if we modify the graph only locally, by shrinking the M-blossom B and removing loops and parallel edges, this can be done in time $O(|B|n)$. Since the sum of $|B|$ over all M-blossoms B is $O(n)$, this yields a time bound of $O(n^2)$ for shrinking.

To reduce the $O(m)$ time for walk-finding, we keep data from the previous walk-search for the next walk-search, with the help of an M-alternating forest, defined as follows.

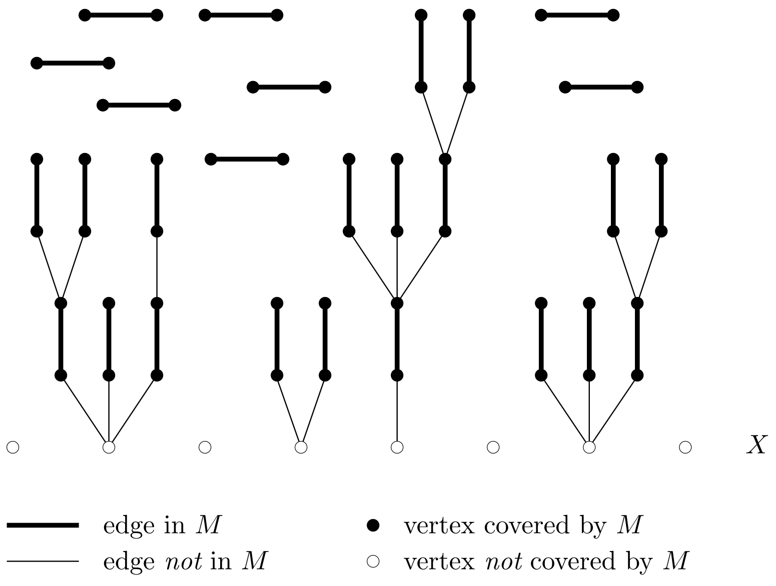

Figure 24.2
An M-alternating forest

Let $G = (V, E)$ be a simple graph and let M be a matching in G. Define X to be the set of vertices missed by M. An *M-alternating forest* is a subset F of E satisfying:

(24.11) F is a forest with $M \subseteq F$, each component of (V, F) contains either exactly one vertex in X or consists of one edge in M, and each path in F starting in X is M-alternating

(cf. Figure 24.2). For any M-alternating forest F, define

(24.12) $\text{even}(F) := \{v \in V \mid F \text{ contains an even-length } X - v \text{ path}\}$,
$\text{odd}(F) := \{v \in V \mid F \text{ contains an odd-length } X - v \text{ path}\}$,
$\text{free}(F) := \{v \in V \mid F \text{ contains no } X - v \text{ path}\}$.

Then each $u \in \text{odd}(F)$ is incident with a unique edge in $F \setminus M$ and a unique edge in M. Moreover:

(24.13) if there is no edge connecting $\text{even}(F)$ and $\text{even}(F) \cup \text{free}(F)$, then M is a maximum-size matching.

Indeed, if there is no such edge, $\text{even}(F)$ is a stable set in $G - \text{odd}(F)$. Hence, setting $U := \text{odd}(F)$:

(24.14) $o(G - U) \geq |\text{even}(F)| = |X| + |\text{odd}(F)| = (|V| - 2|M|) + |U|$,

and hence M has maximum size by (24.2).

Now algorithmically, we keep, next to E and M, an M-alternating forest F. We keep the set of vertices by a doubly linked list. We keep for each vertex v, the edges in E, M, and F, incident with v as doubly linked lists. We also keep the incidence functions $\chi^{\text{even}(F)}$ and $\chi^{\text{odd}(F)}$. Moreover, we keep for each vertex v of G one edge $e_v = vu$ with $u \in \text{even}(F)$, if such an edge exists.

Initially, $F := M$ and for each $v \in V$ we select an edge $e_v = vu$ with $u \in X$ (if any). The iteration is:

(24.15) Find a vertex $v \in \text{even}(F) \cup \text{free}(F)$ for which $e_v = vu$ exists.
Case 1: $v \in$ free(F). Add uv to F. Let vw be the edge in M incident with v. For each edge wx incident with w, set $e_x := wx$.
Case 2: $v \in$ even(F). Find the $X - u$ and $X - v$ paths P and Q in F.
Case 2a: P and Q are disjoint. Then P and Q form with uv an M-augmenting path.
Case 2b: P and Q are not disjoint. Then P and Q contain an M-blossom B. For each edge bx with $b \in B$ and $x \notin B$, set $e_x := Bx$. Replace G by G/B and remove all loops and parallel edges from E, M, and F.

The number of iterations is at most $|V|$, since, in each iteration, $|V| + |\text{free}(F)|$ decreases by at least 2 (one of these terms decreases by at least 2 and the other does not change). We end up either with a matching augmentation or with the situation that there is no edge connecting $\text{even}(F)$ and $\text{even}(F) \cup \text{free}(F)$, in which case M has maximum size by (24.13).

It is easy to update the data structure in Case 1 in time $O(n)$. In Case 2, the paths P and Q can be found in time $O(n)$, and hence in Case 2a, the M-augmenting path is found in time $O(n)$.

Finally, the data structure in Case 2b can be updated in $O(|B|n)$ time[2]. Also a matching augmentation in G/B can be transformed to a matching augmentation in G in time $O(|B|n)$. Since $|B|$ is bounded by twice the decrease in the number of vertices of the graph, this takes time $O(n^2)$ overall.

Hence a matching augmentation can be found in time $O(n^2)$, and therefore:

Theorem 24.5. *A maximum-size matching can be found in time $O(n^3)$.*

Proof. From the above. ∎

The first $O(n^3)$-time cardinality matching algorithm was published by Balinski [1969], and consists of a depth-first strategy to find an M-alternating forest, replacing shrinking by a clever labeling technique.

Bottleneck in a further speedup is storing the shrinking. With the disjoint set union data structure of Tarjan [1975] one can obtain an $O(nm\alpha(m,n))$-time algorithm (Gabow [1976a]). A special set union data structure of Gabow and Tarjan [1983,1985] gives an $O(nm)$-time algorithm. An $O(\sqrt{n}\,m)$-time algorithm was announced (with partial proof) by Micali and Vazirani [1980]. A proof was given by Blum [1990], Vazirani [1990,1994], and Gabow and Tarjan [1991] (cf. Peterson and Loui [1988]).

24.3. Matchings covering given vertices

Brualdi [1971d] derived from Tutte's 1-factor theorem the following extension of the Tutte-Berge formula:

Theorem 24.6. *Let $G = (V, E)$ be a graph and let $T \subseteq V$. Then the maximum size of a subset S of T for which there is a matching covering S is equal to the minimum value of*

$$(24.16) \qquad |T| + |U| - o_T(G - U)$$

over $U \subseteq V$. Here $o_T(G-U)$ denotes the number of odd components of $G-U$ contained in T.

Proof. For any matching M in G and any $U \subseteq V$, at most $|U|$ odd components of $G - U$ can be covered completely by M. So M misses at least $o_T(G - U) - |U|$ vertices in T. This shows that the minimum is not less than the maximum.

To see equality, let μ be equal to the minimum. Let C be a set disjoint from V with $|C| = |V|$ and let $C' \subseteq C$ with $|C'| = |T| - \mu$. Make a new graph H by extending G by C, in such a way that C is a clique, each vertex in C'

[2] For each $Z \in \{E, M, F\}$, we scan the vertices b in B, and for $b \in B$ we scan the Z-neighbours w of b. If w does not belong to B and was not met as a Z-neighbour of an earlier scanned vertex in B, we replace bw by Bw in Z. Otherwise, we delete bw from Z.