An $O(VE)$ Algorithm for Ear Decompositions of Matching-Covered Graphs

Marcelo H. de Carvalho * *Joseph Cheriyan* †

10 December, 2004

Abstract

Our main result is an $O(nm)$-time (deterministic) algorithm for constructing an ear decomposition of a matching-covered graph, where n and m denote the number of nodes and edges. The improvement in the running time comes from new structural results that give a sharpened version of Lovász and Plummer's Two-ear Theorem. Our algorithm is based on $O(nm)$-time algorithms for two other fundamental problems in matching theory, namely, finding all the allowed edges of a graph, and finding the canonical partition of an elementary graph. To the best of our knowledge, no faster deterministic algorithms are known for these two fundamental problems.

1 Introduction

Matching is a topic of central importance for graph theory, computer science, and combinatorial optimization. Problems from matching theory have acted as catalysts in the development of key topics within these areas, e.g., polynomial-time algorithms (Edmonds' maximum-matching algorithm [E 65] led him to the significance of polynomial-time algorithms), polyhedral combinatorics (Edmonds' characterization of the matching polytope inspired the development of this topic [Sc 03]), and the study of combinatorial lattices (which developed from Lovász's characterization of the matching lattice [Lo 87]), etc. Moreover, matching has many practical applications, such as the scheduling of parallel processors, determining chemical bonds, and the Ising model in statistical physics; see [AMO93, LP 86] for more applications.

In the study of graphs with perfect matchings, it is natural to focus on those edges that occur in at least one perfect matching — such an edge is called allowed. (The relevant definitions are in Sections 1 & 2.) Moreover, it can be seen that we may restrict our attention to a connected graph each of whose edges is allowed. Such graphs are called matching covered; they arise naturally in many graph-theoretical investigations. In particular, several well-known theorems and conjectures such as the Four-Colour Theorem, Tutte's 5-Flow Conjecture, and Seymour's Circuit Double Cover Conjecture, may be reduced to 2-connected cubic graphs and such graphs are matching covered. (For instance, the Four-Colour Theorem is equivalent to the statement that every planar 2-connected cubic graph is 3-edge-colourable.) Following the lead of Kotzig [Ko 59], Lovász and Plummer [LP 86] started the systematic study of matching covered graphs. These studies over a period of forty years have revealed a deep and rich structure. In a landmark paper, Lovász

*(mhc@dct.ufms.br) UFMS–Brazil. Supported by CNPq, Brasil, by PRONEX/CNPq (664107/1997-4), by FUNDECT-MS(0284/01), and by a Fellowship from the University of Waterloo, Canada.

†(jcheriyan@uwaterloo.ca) Department of Combinatorics & Optimization, University of Waterloo, Ontario, Canada. Supported by NSERC grant No. OGP0138432.

[Lo 87] gave a good characterization of the matching lattice and proposed a deep conjecture. This conjecture was recently settled by Carvalho et al. [CLM 02] based on some major new results on matching covered graphs. See [CLM 03, LP 86, Mu 94] for surveys on some of these topics.

Our focus is on computing an ear decomposition of a matching-covered graph. Ear decomposition techniques are the basis of some of the key advances in matching theory and algorithms. An early advance is Edmonds' maximum-matching algorithm [E 65], which constructs an ear decomposition of certain (factor-critical) subgraphs by repeatedly shrinking odd circuits. Hetyei [H 64] and Lovász & Plummer launched a systematic study of ear decompositions in matching-covered graphs. An important result from this research is Lovász & Plummer's Two-ear Theorem [LP 73, Theorem 5.4] (or see [LP 86, Theorem 5.4.6]). Ear decomposition techniques and the Two-ear Theorem have been instrumental in obtaining further advances in matching theory, such as Lovász's characterization of the matching lattice [Lo 87].

Let n and m denote the number of nodes and edges of the input graph. While discussing running times, we assume $m = \Omega(n)$ so we use $O(n+m) = O(m)$. Our main result is an $O(nm)$-time (deterministic) algorithm for constructing an ear decomposition of a matching-covered graph, improving on the previous best running time of $O(nm^2)$ due to Little & Rendl [LR 89]. An earlier algorithm due to Naddef & Pulleyblank [NP 82] runs in time $O(\sqrt{n} \cdot m^3)$. Our improvement in the running time comes from several things. One of our key contributions is a new structural result that gives a sharpened version of Lovász and Plummer's Two-ear Theorem (see Theorem 3.5). This enables us to quickly find "double ears" (which is a bottleneck step in algorithms for finding the ear decomposition). Another new feature of our algorithm is that it incrementally constructs *both* the ear decomposition of the edge set, and the canonical partition of the node set. These are the two main structures studied in the theory of matching-covered graphs. The canonical partition has been investigated by Kotzig [Ko 59] and Lovász, see [LP 86, Chapter 5.2].

Our algorithm implicitly solves (within the same time bound) two other fundamental problems in matching theory (1) finding all the allowed edges of a graph, and (2) finding the canonical partition of an elementary graph. For both problems, it is easy to design $O(nm)$-time algorithms, using well-known results on efficient implementations of Edmonds' maximum-matching algorithm (see [Ta 83]) and results from the matching folklore, but to the best of our knowledge, no faster deterministic algorithms are known. It may not be possible to improve on our running time of $O(nm)$ for ear decompositions, until faster algorithms are developed for problems (1), (2), though we do not have a proof for this claim. (Faster randomized algorithms are known for problems (1), (2), and these run in time $M(n)(\log n)^{O(1)} = O(n^{2.38})$ where $M(n)$ denotes the running time for multiplying two $n \times n$ matrices, see [RV 89, C 97].)

Ear decompositions of matching-covered graphs

We list a few central definitions, including that of an ear decomposition. Other definitions and preliminaries are in Section 2. Let $G = (V, E)$ be a graph. A *matching* of G is a subset M of the edges such that no two of the edges in M have an end node in common. A *perfect matching* is one with cardinality $|V|/2$. An edge is called *allowed* if it occurs in at least one perfect matching. A graph with a perfect matching is called *elementary* if its allowed edges form a connected subgraph, and the graph is called *matching-covered* if it is connected and each of its edges is allowed. (Thus, a matching-covered graph is elementary.) A subgraph G_0 of G is called *nice* if $G - V(G_0)$ has a perfect matching.

Let H be a subgraph of G. A *single ear* of G relative to H is a path of G of odd length that has both ends in H but no internal nodes in H. (For our purposes, a single ear has distinct end nodes.) Given any bipartite matching-covered graph G, there exists a sequence

$$G_1 \subset G_2 \subset \ldots \subset G_\ell = G$$

of nice matching-covered subgraphs of G such that (i) $G_1 = K_2$, and (ii) for $2 \le i \le \ell$, $G_i = G_{i-1} + P_i$, where P_i is a single ear of G relative to G_{i-1}, see [LP 86].

For a bipartite matching-covered graph $G = (V, E)$, with bipartition $V = (A, B)$, such a decomposition can be computed via the following folklore algorithm. Find a perfect matching M, then direct all edges of M from A to B and direct the remaining edges from B to A. Take G_1 to be any edge of M, and then for $i = 2, 3, \ldots$, take P_i to be any directed path that has only its start node and end node in G_{i-1}. It is easy to show that the method is correct by showing that each subgraph G_i is nice and matching-covered. Excluding the computation of M, this algorithm runs in linear time.

Such decompositions (via single ears) do not exist for non-bipartite matching-covered graphs. For example, K_4 has no such decomposition. To extend this type of decomposition to all matching-covered graphs, we need the notion of a double ear. A *double ear* P^* of G relative to a subgraph H is a pair $\{P', P''\}$, where P' and P'' are two node-disjoint single ears of G relative to H. We call P' and P'' the *members* of the double ear P^*. By an *ear* of G (relative to some subgraph H) we mean either a single ear or a double ear. For an ear P^* of G relative to a subgraph H, we use $H + P^*$ to denote the graph obtained from H by adding the edges and internal nodes of the constituent path(s) of P^*. An *ear decomposition* of a matching-covered graph G is a sequence

$$G_1 \subset G_2 \subset \ldots \subset G_\ell = G$$

of nice matching-covered subgraphs of G such that (i) $G_1 = K_2$, and (ii) for $2 \le i \le \ell$, $G_i = G_{i-1} + P^*$, where P^* is an ear (single or double) of G relative to G_{i-1}. The following fundamental theorem is due to Lovász and Plummer [LP 73] (see also [S 98]).

Theorem 1.1 (Ear Decomposition Theorem) *Every matching-covered graph has an ear decomposition.*

Whenever we use a double ear $P^* = \{P', P''\}$, we implicitly assume that adding either P' or P'' as a single ear does not give a matching-covered graph; thus, the ear decompositions in our paper are in fact what Lovász and Plummer call "nonrefinable graded ear decompositions" (see [LP 86, Section 5.4]). (We will mention this explicitly, where this is relevant.)

Most algorithms for computing such an ear decomposition have to deal with a bottleneck, namely, finding a double ear when it is impossible to add any single ear. If we try to find an appropriate pair of single ears by an exhaustive search, then this step alone may contribute a running time of $O(nm)$. In Section 3, we present a sharpened version of Lovász and Plummer's Two-ear Theorem that enables us to find a double ear in (essentially) linear running time.

Section 2 summarizes notation, definitions, and basic results on elementary graphs. Also, this section has deterministic algorithms (from the matching folklore) for the canonical partition, and for the allowed edges. Section 3 sharpens the Two-ear Theorem. Our ear decomposition algorithm and its analysis are presented in Section 4.

2 Preliminaries

2.1 Definitions and notation

We list some standard definitions from matching theory, see [LP 86]. Let $G = (V, E)$ be a graph. For subgraphs H and P of G, $H + P$ denotes the union, i.e., $H + P = (V(H) \cup V(P),\ E(H) \cup E(P))$. Given a matching M, a node is called *matched* if it is incident to an edge of M, and is called *exposed* (or, *M-exposed*) otherwise. An *M-alternating path* is a path whose edges are alternately in M and not in M. An *M-augmenting path* is an M-alternating path such that both end nodes are M-exposed.

A graph G is called *factor-critical* if, for every node v of G, the subgraph $G - v$ has a perfect matching. For any graph H, let $oc(H)$ denote the number of odd components of H, where a (connected) component of H is called *odd* (or *even*) if it has an odd (even) number of vertices. Let G be a graph with a perfect matching. A node set B of G is called a *barrier* if $oc(G - B) = |B|$. Clearly, the empty set is a barrier of G, but henceforth, by a barrier we shall mean a nonempty barrier. All singleton subsets of $V(G)$ are barriers of G. We refer to such barriers as *trivial* barriers.

For a graph $G = (V, E)$ and a subset S of V, $\partial_G(S)$ (or simply $\partial(S)$) denotes the set of edges that have exactly one end in S, and it is called an *(edge-) cut* of G with S and $\overline{S} = V - S$ as its *shores*. The graph obtained from G by contracting its shore S to a single node is called a *contraction* of G, denoted by G/S.

2.2 Elementary graphs

The following results are fundamental for our purposes and are (except for Corollary 2.7) proved in Lovász and Plummer's book [LP 86] (see also [Mu 94]). The next theorem characterizes the (inclusionwise) maximal barriers of an elementary graph.

Theorem 2.1 *Let G be an elementary graph, and let $\sim$ denote the binary relation on V where $u \sim v$ if $G - \{u, v\}$ has no perfect matching. Then, the relation $\sim$ is an equivalence relation on V and the equivalence classes are precisely the (inclusionwise) maximal barriers of G.*

The partition of V into maximal barriers is called the *canonical partition* of G, denoted by $\mathcal{P}(G)$. Suppose $\mathcal{P}(G) = \{S_1, S_2, \ldots, S_k\}$, where $S_1 \dot\cup S_2 \dot\cup \ldots \dot\cup S_k = V$, then each of the sets $S_i, 1 \le i \le k$, is called a *class* of $\mathcal{P}(G)$.

Proposition 2.2 *The following properties hold for an elementary graph G.*

(i) An edge e of G is allowed if and only if no barrier contains both ends of e.

(ii) If B is a (nonempty) barrier of G, then $G - B$ has no even components.

(iii) A barrier B of G is maximal if and only if all components of $G - B$ are factor-critical.

Proposition 2.3 *Let H be an elementary graph, and let G be obtained from H by adding some edges. Let B be a barrier of G, and let K be an odd component of $G - B$. Then B is a barrier of H, and $H - B$ has an odd component whose node set is $V(K)$.*

Proposition 2.4 *Let G be an elementary graph, and let e be an edge not in G but with both ends in $V(G)$. Then, $\mathcal{P}(G + e)$ is a refinement of $\mathcal{P}(G)$, that is, for each class S' of $\mathcal{P}(G + e)$ there exists a class S of $\mathcal{P}(G)$ such that $S' \subseteq S$.*

Proposition 2.5 *Let G be an elementary graph and let $e := xy$ be an allowed edge of G. If graph G' is obtained from G by subdividing e by the insertion of two new nodes u and v such that the path corresponding to e is x, u, v, y, then $\mathcal{P}(G')$ is the same as $\mathcal{P}(G)$ except that u is added to the class of y and v is added to the class of x.*

Proposition 2.6 *Suppose that G is an elementary graph and X, Y are barriers in G such that $X \cap Y \neq \emptyset$ and G has no edges between $X - Y$ and $Y - X$. Then $X \cap Y$ and $X \cup Y$ are barriers in G.*

Corollary 2.7 *Suppose that G is an elementary graph, and $S_1, S_2, \ldots, S_\ell$ are barriers in G such that $S_1 \cap S_2 \cap \cdots \cap S_\ell \neq \emptyset$ and for any i and j, $1 \le i < j \le \ell$, G has no edges between $S_i - S_j$ and $S_j - S_i$. Then $S_1 \cap S_2 \cap \cdots \cap S_\ell$ and $S_1 \cup S_2 \cup \cdots \cup S_\ell$ are barriers in G.*

2.3 Algorithmic preliminaries

The fastest known (deterministic) algorithms for finding a maximum matching are due to Micali & Vazirani [MV 80, V 94] and Goldberg & Karzanov [GK 04] (the running times are $O(\sqrt{n} \cdot m)$ and $O(\sqrt{n} \cdot m \cdot \frac{\log(n^2/m)}{\log n})$, respectively). We do not use these algorithms. Instead, we use the efficient implementation of Edmonds' maximum-matching algorithm as presented by Tarjan [Ta 83]; this implementation achieves linear running time for each iteration (augmentation) of Edmonds' algorithm by using Gabow and Tarjan's linear-time method for (a special case of) disjoint set union [GT 85].

Given any matching M, this algorithm assigns labels to the nodes as follows: a node v is labeled **0** (meaning, even) if there is an even-length alternating path from v to an M-exposed node, otherwise, v is labeled **1** (meaning, odd) if there is an odd-length alternating path from v to an exposed node, otherwise, v is unlabeled. Thus every exposed node is labeled **0**. (If M is a maximum matching, then this labeling corresponds to the Gallai-Edmonds decomposition, see [LP 86, Chapter 3.2]: The sets of nodes labeled **0** and **1** are the sets $D(G)$ and $A(G)$, respectively, and the set of unlabeled nodes is the set $C(G)$.) For a proof of the next result, see [Ta 83, pp. 115–122], [GT 85], [E 65].

Proposition 2.8 (Edmonds' Algorithm) *Let G be a graph, and let M be any matching of G (M need not be maximum).*

(i) Then the above labeling of the nodes can be computed in $O(m)$ time.

(ii) Given any node v labeled **0**, *an even-length M-alternating path from v to some M-exposed node can be computed in $O(m)$ time.*

(iii) If there exists an M-augmenting path, then one can be computed in $O(m)$ time. Moreover, we may start with an empty matching, and compute a maximum matching in $O(nm)$ time.

Proposition 2.9 *Let G be an elementary graph and let v be a node of G. Suppose that a perfect matching M of G is given.*

(i) The class of the canonical partition $\mathcal{P}(G)$ that contains v can be found in $O(m)$ time. Moreover, $\mathcal{P}(G)$ can be computed in $O(nm)$ time.

(ii) The allowed edges incident to v can be found in $O(m)$ time. Moreover, all the allowed edges can be found in $O(nm)$ time.

Proof: Suppose that v is matched to w in M, that is, $vw \in M$. Delete v from G and M, and apply one iteration of Edmonds' algorithm to the graph $G - v$ and matching $M - vw$ to obtain a labeling of the remaining nodes. By Proposition 2.8, this can be done in time $O(m)$.

Let S be the set of odd nodes (nodes labeled **1**). By basic results from matching theory, the components of $G - v - S$ are all factor-critical. (In more detail, S is the set $A(G - v)$ in the Gallai-Edmonds decomposition of $G - v$, see [LP 86, Exercise 9.1.2], hence, $S \cup \{v\}$ is a barrier of G; moreover, by Proposition 2.2,part(ii), $G - v - S$ has no even components.) Hence, by Proposition 2.2,part(iii), $S \cup \{v\}$ is a maximal barrier of G, that is, $S \cup \{v\}$ forms the class of $\mathcal{P}(G)$ that contains v. Also, note that each node in $G - v$ gets a label of **0** or **1** (there are no unlabeled nodes), because $G - v - S$ has no even components. It follows from Proposition 2.2,part(i) that an edge vx of G is allowed if and only if x is labeled **0**. ■

Remark: The linear-time method (in Proposition 2.9) for finding the allowed edges incident to a node v extends to any graph that has a perfect matching.

3 A sharpened Two-ear Theorem

This section develops our method for efficiently finding a double ear by sharpening some well-known matching theory results of Lovász and Plummer.

Let H be a matching-covered graph. Let $F = \{e_1, e_2, \ldots, e_k\}$, $k \geq 2$, be a set of edges not in H, but having both ends in H, such that the graph $H + F$ is matching-covered. In this section, we use G to denote the graph $H + F$ (rather than the input graph for our algorithm), and moreover, we assume the following:

> *Adding any of the edges $e_i \in F$ to H as a single ear gives a graph that is not matching-covered, that is, $H + e_i$ has no perfect matching containing e_i, for each $i = 1, \ldots, k$.*

Then by Proposition 2.2, each edge e_i has both ends in the same class of the canonical partition $\mathcal{P}(H)$. The next lemma gives an easy characterization of pairs of edges in F that form double ears. (Recall our convention: for every double ear $P^* = \{P', P''\}$, adding either P' or P'' as a single ear gives a graph that is not matching covered.)

Lemma 3.1 *For two distinct edges $e, f \in F$, the pair $\{e, f\}$ is a double ear relative to H if and only if the ends of f lie in distinct classes of the canonical partition $\mathcal{P}(H + e)$.*

Proof: Let $Q := \{e, f\}$ be a double ear relative to H. Then, $H + Q$ is a matching-covered graph, and so edge f is allowed in $H + Q$. It follows that the ends of f are in distinct classes of $\mathcal{P}(H + e)$.

Conversely, if the ends of f are in distinct classes of $\mathcal{P}(H + e)$, then f is allowed in $H + \{e, f\}$. But note that f is not allowed in $H + f$ (by the assumption at the start of this section). Similarly, e is not allowed in $H + e$. Thus, every perfect matching in $H + \{e, f\}$ containing one of e and f also contains the other edge, and so $\{e, f\}$ is a double ear relative to H. ■

The next result is the key one for our algorithm. It is inspired by the Two-ear Lemma and its proof in Lovász and Plummer's book (see [LP 86, Lemma 5.4.5]).

Lemma 3.2 *Let $S \in \mathcal{P}(H)$ and suppose that there is precisely one edge in F, say $e_k = xy$, with both ends in S. Then the edge e_k is a member of a double ear of $G = H + F$ relative to H (i.e., $\{e_k, e_j\}$ is a double ear for some $j = 1, \ldots, k - 1$).*

Proof: If $|F| = 2$, then obviously the lemma holds. Thus assume $|F| > 2$. Suppose to the contrary that for each $i = 1, 2, \ldots, k - 1$, there is no perfect matching in $H + e_i + e_k$ containing e_k. Thus, the ends x and y of e_k belong to the same class (or maximal barrier) of the canonical partition $\mathcal{P}(H + e_i)$, call this class $S(e_i)$. Recall that the addition of edges to an elementary graph refines its maximal barriers (by Proposition 2.4). As $\{x, y\} \subseteq S$, it follows that $S(e_i) \subseteq S$. Moreover, we claim that $S(e_i)$ is a barrier in H and both ends of e_i are in the same component of $H - S(e_i)$. To see this, note that e_i has no end in S (since e_i is not allowed in $H + e_i$ both its ends are in the same class of $\mathcal{P}(H)$ and this class differs from S by the choice of e_k), hence, both ends of e_i must be in one of the factor-critical components, say K, of $(H + e_i) - S(e_i)$. Then, by Proposition 2.3, $S(e_i)$ is a barrier of H and $H - S(e_i)$ has an odd component with node set $V(K)$. Our claim follows.

Let $I := S(e_1) \cap S(e_2) \cap \cdots \cap S(e_{k-1})$. Since $\{x, y\} \subseteq S(e_i)\ \forall i$, it follows that $\{x, y\} \subseteq I$. Thus, $I \neq \emptyset$. Moreover, H has no edge with one end in $S(e_i) - S(e_j)$ and the other end in $S(e_j) - S(e_i)$, for any i, j, $1 \leq i < j \leq k$, because any such edge would have both ends in $S \supseteq S(e_i) \cup S(e_j)$ but a matching-covered graph such as H cannot have an edge with both ends in one of its barriers. Then by Corollary 2.7, I is a barrier in H.

For $i = 1, 2, \ldots, k-1$, the above claim shows that e_i joins two nodes in the same odd component of $H - S(e_i)$, and $I \subseteq S(e_i)$, hence, e_i joins two nodes in the same component of $H - I$. Thus, I is a barrier in G and we conclude that e_k is not allowed in G. This is a contradiction. Thus there is a $j \in \{1, \ldots, k - 1\}$ such that e_k is allowed in $H + e_j + e_k$, so $\{e_j, e_k\}$ is a double ear. ■

Remark: In the above lemma, the condition that there is exactly one edge with both ends in S is crucial. Here is a counterexample (from [S 98]) to the weaker version of the lemma that omits this condition: Let H be a cycle $1, 2, \ldots, 8, 1$ on eight nodes, and let $F = \{15, 24, 37, 68\}$. Then the edge 15 is not a member of any double ear.

We now deal with the case where a class $S \in \mathcal{P}(H)$ contains both ends of two or more edges of F. This case reduces to the previous one via the following (technical) lemma whose proof may be skipped on first reading.

Lemma 3.3 *Let H be any matching-covered graph. Let $S \in \mathcal{P}(H)$ and let B be a nontrivial barrier of H such that $B \subseteq S$. Let K be a component of $H - B$. Let H_1 and H_2 be the graphs obtained from H by contracting $V(K)$ and $V(H) - V(K)$ to single nodes v_1 and v_2, respectively. Then*

(i) H_1 and H_2 are matching-covered;

(ii) B is a barrier of H_1 and $S - V(K)$ is the maximal barrier of H_1 that contains B.

Proof: Let $C := \partial(V(K))$; also, note that $|B| \geq 2$. Let e be any edge of H_1. Then e is also an edge of H, and as H is matching-covered, there is a perfect matching M of H containing e. Since B is a barrier of H, we have $|M \cap C| = 1$. It follows that the restriction of M to $E(H_1)$ is a perfect matching of H_1 containing e. Thus, H_1 is matching-covered. Similarly, H_2 is matching-covered. This proves part(i).

Every (odd) component of $H - B$ distinct from K is also a component of $H_1 - B$. Moreover, the contracted node v_1 is a trivial component of $H_1 - B$. Hence, $H_1 - B$ has precisely $|B|$ odd components, so B is a barrier of H_1. Let B_1 be the maximal barrier of H_1 containing B. We shall show that $B_1 = S - V(K)$. First, note that $v_1 \notin B_1$ because there are edges of H_1 joining v_1 to nodes of B and all these edges are allowed in H_1 (as H_1 is matching-covered).

Now, choose any node $u \in B$, and note that $u \in S \cap B_1$. We first show that $B_1 \subseteq (S - V(K))$. Suppose that there is a node $w \in B_1$ which is not in S. As $v_1 \notin B_1$, u and w are nodes of H. Let $B_1' := B_1 - \{u, w\}$. Note that $(H_1 - \{u, w\}) - B_1' = H_1 - B_1$ and this graph has $|B_1| = |B_1'| + 2$ odd components. Similarly, $(H - \{u, w\}) - B_1' = H - B_1$ and this graph has $|B_1| = |B_1'| + 2$ odd components (since $H_1 - B_1$ and $H - B_1$ have the same components except for v_1 and K). Therefore, $H - \{u, w\}$ has no perfect matching. On the other hand, as $u \in S$ and $w \notin S$, Theorem 2.1 implies that $H - \{u, w\}$ has a perfect matching. This contradiction shows that $B_1 \subseteq (S - V(K))$.

Now, assume that $w \in (S - V(K)) - B_1$. Then, $H_1 - \{u, w\}$ has a perfect matching M_1. Let e denote the edge of M_1 incident with v_1. As H_2 is matching-covered, there is a perfect matching M_2 of H_2 containing e. Then, $M_1 \cup M_2$ is a perfect matching of $H - \{u, w\}$. But $H - \{u, w\}$ cannot have a perfect matching because $\{u, w\} \subseteq S$. This contradiction shows that $(S - V(K)) - B_1 = \emptyset$, hence, $B_1 = (S - V(K))$. ∎

Let $F^* \subseteq F$ denote the set of edges with both ends in S. Consider the graph $G^* := H + (F - F^*) = G - F^*$. Observe that H is a spanning matching-covered subgraph of G^*.

Proposition 3.4 *Consider the graph $G^* = H + (F - F^*)$ and any edge $e \in F^*$. (Recall that $S \in \mathcal{P}(H)$, and $F^* \subseteq F$ consists of edges that have both ends in S.)*

(i) If e is allowed in $G^* + e$, then e is a member of a double ear relative to H.

(ii) If e is not allowed in $G^* + e$, then

(a) there is another edge $f \in F^*$ whose ends are in distinct odd components of $G^* - B$, where B is the maximal barrier of G^* that contains both ends of e, and

(b) any such edge f is a member of a double ear relative to H.

Proof: For part(i), observe that e is the only edge with both ends in S in the graph G^*+e. Let M be a perfect matching of G^*+e containing e. Let $F' := M \cap F$. Then $H+F'$ is matching-covered and e is the only edge of F' with both ends in S. By Lemma 3.2, e is a member of a double ear of $H+F'$ relative to H. This proves part(i).

For part(ii), assume that e is not allowed in G^*+e. Then both ends of e belong to a maximal barrier B of G^*. As e is allowed in $H+F = G^*+F^*$, there is another edge f of F^* with ends, say x and y, in distinct odd components of G^*-B. This proves part(a).

To prove part(b) via part(i), it suffices to show that f is allowed in the graph G^*+f. Observe that $\{x,y\} \subseteq S$, because each edge in F^* (including f) has both ends in S.

Let K be a component of G^*-B containing one end of f, say x. Since H is a spanning matching-covered subgraph of G^*, Proposition 2.3 implies that B is a barrier of H, and one of the odd components of $H-B$ has node set $V(K)$. Moreover, $B \subseteq S$, because both ends of e belong to S and the addition of edges to an elementary graph refines its maximal barriers (by Proposition 2.4).

Let H_1 be the graph obtained from H by contracting $V(K)$ to a single node v_1. By Lemma 3.3, H_1 is matching-covered and $B_1 = S - V(K)$ is a maximal barrier of H_1 containing B. Then $y \in B_1$ (since $\{x,y\} \subseteq S$ and $y \notin V(K)$) and $v_1 \notin B_1$ (since H_1, which is matching-covered, has edges joining v_1 to nodes in B, so any barrier of H_1 containing B must be disjoint from v_1). Hence, $H_1 - \{v_1, y\}$ contains a perfect matching M_1. Now, focus again on K (the component of G^*-B containing x). By Proposition 2.2, K is factor-critical, hence, $K-x$ has a perfect matching M_2. Then, $M_1 \cup M_2 \cup \{f\}$ is a perfect matching of G^*+f containing $f = xy$. Thus, part(b) follows by applying part(i) to G^* and f. ■

Algorithmic aspects of double ears

Theorem 3.5 *Let $S \in \mathcal{P}(H)$, and let F^* denote the set of edges of F with both ends in S. If $F^* \neq \emptyset$ then there is an edge of F^* that is a member of a double ear relative to H. Moreover, such an edge can be found in time $O(m)$.*

Proof: The first part of the theorem follows from Lemma 3.2 and Proposition 3.4.

For the running time, observe that if $|F^*| = 1$ then the unique edge of F^* is a member of a double ear, and we are done.

If $|F^*| > 1$, Proposition 3.4 gives the procedure for finding the right edge of F^*: Start with the graph $G^* = H + (F - F^*)$, take any edge e of F^* and find the class B of the canonical partition of G^* containing one of the ends of e. This can be done in time $O(m)$ by Proposition 2.9. If the other end of e does not lie in B then e is allowed in the graph G^*+e, and by Proposition 3.4,part(i) e is a member of a double ear relative to H. If both ends of e lie in B, then by Proposition 3.4,part(ii) there is another edge of F^* with ends in distinct odd components of G^*-B, and any such edge is a member of a double ear relative to H. We can find such an edge in time $O(m)$. Thus, the running time of this procedure is $O(m)$. ■

Suppose we have found an edge $e \in F$ that is a member of a double ear relative to H. Then we apply Lemma 3.1 to find a second edge $f \in F$ such that $\{e,f\}$ is a double ear relative to H. This step can be implemented in linear time by maintaining relevant information on the canonical partition, see Section 4 for details.

4 Finding an ear decomposition in $O(nm)$ time

> **input**: a matching-covered graph $G = (V, E)$
> **output**: an ear decomposition of G and the canonical partition $\mathcal{P}(G)$
>
> (0) start by finding a perfect matching M of G;
>
> (1) let xy be any edge of M, let subgraph H correspond to xy, and let the canonical partition be $\mathcal{P}(H) = \{\{x\}, \{y\}\}$;
>
> (2) while $H \neq G$ do
>
> (2.1) if H is a spanning subgraph of G then let $F = E(G) - E(H)$, else compute F using the detailed explanation of this step in the text; note that each edge $e_j \in F$ corresponds to a (single) ear P_j relative to H; finally, let $F_0 := F$;
>
> (2.2) repeat
>
> (2.2.1) let $H_0 := H$ and let $p_0 := |\mathcal{P}(H_0)|$; let F' be the set of edges in F that have their two ends in distinct classes of $\mathcal{P}(H)$; replace F by $F - F'$;
>
> (2.2.2) sequentially examine the edges of F' and add each edge to H as a single ear; update $\mathcal{P}(H_0)$ to $\mathcal{P}(H)$;
>
> (2.2.3) if $p_0 = |\mathcal{P}(H)|$ and $F \neq \emptyset$ then find a double ear $\{e, f\} \subseteq F$ by using the method in Theorem 3.5; remove e, f from F and add them to H; update $\mathcal{P}(H - \{e, f\})$ to get $\mathcal{P}(H)$;
>
> ($*$2.2) until $F = \emptyset$;
>
> (2.3) for each edge $e_j \in F_0$ take the corresponding path P_j of G (see step (2.1)), and insert the internal nodes of P_j (if any) into appropriate classes of $\mathcal{P}(H)$ (see Proposition 2.5);
>
> ($*$2) end (while loop); STOP.

Table 1: Ear-decomposition algorithm

Our algorithm is summarized in Table 1. The input consists of a matching-covered graph G, and the output is an ear decomposition of G, together with the canonical partition $\mathcal{P}(G)$. Throughout the computation, we maintain the canonical partition of the current graph H, $\mathcal{P}(H)$, and for each class $S \in \mathcal{P}(H)$, we maintain the node sets of the (connected) components of $H - S$ (see the proof of Lemma 4.2 for details). We represent $\mathcal{P}(H)$ explicitly, as well as by labeling each node v in H by the class in $\mathcal{P}(H)$ that contains v. The rest of this section is devoted to a proof of the following theorem.

Theorem 4.1 *An ear decomposition of a matching-covered graph can be computed in $O(nm)$ time.*

To prove Theorem 4.1, we discuss each nontrivial step of the algorithm in detail, and analyse its contribution to the running time. The following key lemma allows us to bound the total running time for all changes to the canonical partition (over the whole computation), hence, when we analyse the individual steps, we ignore the running time devoted to updates of the canonical partition. The proof of the lemma is deferred to the end.

Lemma 4.2 *Over the whole computation, the total number of changes to the canonical partition $\mathcal{P}(H)$ is $O(n)$, and each update takes $O(m)$ time. Thus the total running time for all changes to the canonical partition is $O(nm)$.*

STEP (0):
We use an efficient implementation of Edmonds' maximum-matching algorithm (see Proposition 2.8) to find a perfect matching M of G in time $O(nm)$.

STEP (2.1):
This step finds a set of edges F such that $H+F$ is matching covered. We claim that this step takes $O(m)$ time (for each iteration of the while loop), and this step contributes a total of $O(nm)$ to the running time (over the whole computation). If $V(H) = V(G)$, then we take $F = E(G) - E(H)$, and clearly, this takes linear time. Now, suppose that H is not a spanning subgraph of G. In this case, we take any edge joining a node v in H to a node w not in H. Then vw is contained in some perfect matching M^* of G and $vw \notin M$, therefore, vw is contained in an alternating circuit of $M \cup M^*$. Let vv' and ww' denote the edges of M incident with v and w, respectively. To find an M-alternating circuit C containing vw, we delete nodes v and w from G, and then we find an M'-alternating path P from v' to w' in $G-\{v,w\}$, where $M' = M - \{vv', ww'\}$; C consists of the edges $v'v, vw, ww'$ and the path P. The running time is $O(m)$, by Proposition 2.8. The edges of C not in H form one or more ears $P_1, \ldots, P_q$ relative to H. If some P_i has internal nodes, e.g., $P_i = v_0v_1\ldots v_\ell$, then we replace P_i by the single edge $e_i := v_0v_\ell$. We now have $F := \{e_1, e_2, \ldots, e_q\}$. In either case, $H+F$ is a matching-covered graph.

Over the whole computation, the number of executions of this step is $O(n)$, because every execution that finds an M-alternating circuit adds at least two nodes to H. Thus the total running time contributed by this step is $O(nm)$.

STEP (2.2.1):
We compute a list F' consisting of the edges in F whose ends are in distinct classes of $\mathcal{P}(H)$, either after each change of the canonical partition (in steps (2.2.2) or (2.2.3)), or when a new set F of edges is found (in step (2.1)). Each execution of step (2.2.1) takes time $O(1)$ per edge in F (by comparing the class labels of the two ends). If F' is nonempty, we remove edges one by one from F' and add them to H in single ear addition steps. (Also, we update $\mathcal{P}(H)$, but we ignore the time for this update since Lemma 4.2 handles this.) Thus the running time for step (2.2.1) is $O(m)$ per change of the canonical partition, and $O(m)$ per new set F. By Lemma 4.2, there are $O(n)$ changes of the canonical partition, and we have seen that a new set F is found $O(n)$ times. Thus, over the whole computation, the running time contributed by this step is $O(nm)$.

STEP (2.2.3):
If $H+F$ has no single ear relative to H, then Theorem 3.5 gives a linear-time procedure for finding an edge e that is a member of a double ear relative to H. We then add e to H, update the canonical partition, and then search for any edge $f \in F - \{e\}$ with ends in distinct classes of the canonical partition. By Lemma 3.1, $\{e, f\}$ is a double ear relative to H. Edge f can be found in time $O(m)$. We now add f to $H+e$, and then update the canonical partition. Thus, the time for finding a double ear is $O(m)$ (ignoring the time for updates to $\mathcal{P}(H)$ which is handled in Lemma 4.2).

The number of classes of the canonical partition increases by at least two whenever we add a double ear. (For every double ear $P^* = \{P', P''\}$, recall that adding either P' or P'' as a single ear gives a graph that is not matching covered.) It follows that the number of double ear addition steps is at most $|V|/2 = O(n)$. Thus, over the whole computation, the running time contributed by this step is $O(nm)$.

Proof: (of LEMMA 4.2) For the sake of exposition, let us consider updating the canonical partition $\mathcal{P}(H) = \{S_1, \ldots, S_k\}$ of H when the ear added to H is a single edge, say e (the procedure is analogous if a set F' of two or more single ears is added). By Proposition 2.4 the new canonical partition, $\mathcal{P}(H+e)$, is a refinement of the old one $\mathcal{P}(H)$, that is, for each class S'_j of $\mathcal{P}(H+e)$ there exists a class S_i of $\mathcal{P}(H)$ such that $S'_j \subseteq S_i$. A class S_i of $\mathcal{P}(H)$ is *not* a maximal barrier of $H+e$

if and only if the ends of e are in distinct components of $H - S_i$. For each class S_i of $\mathcal{P}(H)$, we maintain the node sets of the components of $H - S_i$; this enables us to verify in time $O(1)$ whether S_i is a maximal barrier of $H + e$. Hence, verifying whether every class S_i of $\mathcal{P}(H)$ is a maximal barrier of $H + e$ takes time $O(|\mathcal{P}(H)|) = O(n)$, given the above information. Note that the total time for such verification (over the whole computation) is $O(nm)$, because we add $O(m)$ ears, and each ear requires $O(n)$ time for the verification.

If S_i is a maximal barrier of $H + e$, then S_i is a class of $\mathcal{P}(H + e)$, otherwise, S_i has "split" into two or more classes of $\mathcal{P}(H + e)$. In the latter case, we repeatedly apply the algorithm of Proposition 2.9 to partition S_i into classes of $\mathcal{P}(H+e)$: we take an arbitrary node $v_1 \in S_i$ and find the class $S'(v_1)$ of $\mathcal{P}(H+e)$ containing v_1, by applying the algorithm of Proposition 2.9 to $(H+e)$, v_1, and the restriction of M to H; if $S_i - S'(v_1)$ has another node v_2, then we apply the algorithm to $(H + e)$, v_2, and the restriction of M to H, to find the class $S'(v_2)$ of $\mathcal{P}(H + e)$ containing v_2, and so on.

The next claim provides the main tool for bounding the total number of changes to the canonical partition. Define the "class splits" tree $\mathcal{T}$ as follows. (This tree represents all of the splits of the classes of the canonical partition, over the whole computation.) Each class of the canonical partition during the execution corresponds to a node in $\mathcal{T}$. Also, to make $\mathcal{T}$ connected, we introduce a root node that corresponds to the set $\{x, y\}$, where $xy \in M$ is the first edge added to H. An edge (S', S) is present in $\mathcal{T}$ if and only if S' is a class resulting from the splitting of class S in the execution, i.e., $S' \subset S$. In the execution, if a class S' is obtained by adding some of the internal nodes of an ear P_i to a previous class S (via Proposition 2.5), then the same node of $\mathcal{T}$ corresponds to both S' and S.

Claim: *Over the whole computation, the total number of "class splits" is $O(n)$, where each "class split" corresponds to a class of the new canonical partition that is properly contained in a class of the old canonical partition.*

To prove this claim, focus on the "class split" tree $\mathcal{T}$. Each nonleaf node of $\mathcal{T}$ has at least two children, and $\mathcal{T}$ has at most n leaf nodes. Hence, $\mathcal{T}$ has at most $n - 1$ nonleaf nodes and at most $2n$ edges. The claim follows since each "class split" in the execution corresponds to a distinct edge of $\mathcal{T}$.

Over the whole algorithm, the total time for updating the canonical partition, and the components of $H - S_i$, $S_i \in \mathcal{P}(H)$, is $O(nm)$ by the above claim. The lemma follows. ■

Acknowledgments. We thank U.S.R.Murty and J.F.Geelen for useful discussions.

References

[AMO93] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications.* Prentice-Hall, Englewood Cliffs, N. J., 1993.

[CLM 02] M. H. Carvalho, C. L. Lucchesi, and U. S. R. Murty, *On a conjecture of Lovasz concerning bricks I. The characteristic of a matching covered graph*, J. Combinatorial Theory B **85** (2002), 94–136.
On a conjecture of Lovasz concerning bricks II. Bricks of finite characteristic, J. Combinatorial Theory B **85** (2002), 137–180.

[CLM 03] M. H. Carvalho, C. L. Lucchesi, and U. S. R. Murty, *The matching lattice*, in *Recent Advances in Algorithms and Combinatorics*, edited by B. Reed and C. L. Sales, CMS Books in Mathematics, Springer, 2003.

[C 97] J. Cheriyan, *Randomized $\widetilde{O}(M(|V|))$ algorithms for problems in matching theory*, SIAM J. Computing **26** (1997), 1635–1655.

[E 65] J. Edmonds, *Paths, trees and flowers*, Canad. J. Math. **17** (1965), 449–467.

[GT 85] H. N. Gabow and R. E. Tarjan, *A linear time algorithm for a special case of disjoint set union*, J. Computer and System Sciences **30** (1985), 209–221.

[GK 04] A. V. Goldberg and A. V. Karzanov, *Maximum skew-symmetric flows and matchings*, Math. Program., Ser. A **100** (2004), 537–568.

[H 64] G. Hetyei, *2×1-es téglalapokkal lefedhetö idomokról*, Pécsi Tanárképzö Föiskola Tud. Közl. (1964), 351–368.

[Ko 59] A. Kotzig, *Ein Beitrag zur Theorie der endlichen Graphen I–II–III*, Mat. Fyz. Casopis **9** (1959), 73–91, 136–159 and **10** (1960) 205–215.

[LR 89] C. H. C. Little and F. Rendl, *An algorithm for the ear decomposition of a 1-factor covered graph*, J. Austral. Math. Soc. (Series A) **46** (1989), 296–301.

[Lo 83] L. Lovász, *Ear decompositions of matching-covered graphs*, Combinatorica **3** (1983), 105–117.

[Lo 87] L. Lovász, *Matching structure and the matching lattice*, J. Combinatorial Theory B **43** (1987), 187–222.

[LP 73] L. Lovász and M. D. Plummer, *On bicritical graphs*, Infinite and finite sets (Colloq. Keszthely, Hungary, 1973), II, Eds.: A. Hajnal, R. Rado and V. T. Sós, Colloq. Math. Soc. János Bolyai, **10**, North-Holland, Amsterdam, 1975, 1051–1079.

[LP 86] L. Lovász and M. D. Plummer, *Matching Theory*, Akadémiai Kiadó, Budapest, Hungary, 1986.

[MV 80] S. Micali and V. V. Vazirani, *An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs*, Proc. 21st IEEE F.O.C.S. (1980), 17–27.

[Mu 94] U. S. R. Murty, *The matching lattice and related topics*, preliminary report, Dept. of Combinatorics & Optimization, Univ. Waterloo, Ontario, Canada, 1994.

[NP 82] D. Naddef and W. R. Pulleyblank, *Ear decompositions of elementary graphs and GF(2)-rank of perfect matchings*, Bonn Workshop on Combinatorial Optimization, Eds.: A. Bachem, M. Grötschel and B. Korte, Ann. Discrete Math., **16**, North-Holland, Amsterdam, 1982, 241–260.

[RV 89] M. O. Rabin and V. V. Vazirani, *Maximum matchings in general graphs through randomization*, J. Algorithms **10** (1989), 557–567.

[Ta 83] R. E. Tarjan, *Data Structures and Network Algorithms*, SIAM Publications, Philadelphia, PA, 1983.

[V 94] V. V. Vazirani, *A theory of alternating paths and blossoms for proving correctness of the $O(\sqrt{V}E)$ general graph matching algorithm*, Combinatorica **14** (1994), 71–109.

[Sc 03] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Springer-Verlag, Berlin, 2003.

[S 98] Z. Szigeti, *The two ear theorem on matching-covered graphs*, J. Combinatorial Theory B **74** (1998), 104–109.