Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 349 (2005) 52–66
Theoretical Computer Science
www.elsevier.com/locate/tcs

List matrix partitions of chordal graphs

Tomás Feder[a], Pavol Hell[b,*], Sulamita Klein[c], Loana Tito Nogueira[d], Fábio Protti[e]

[a]268 Waverley St., Palo Alto, CA 94301, USA
[b]School of Computing Science, Simon Fraser University, Burnaby, BC, Canada V5A 1S6
[c]IM and COPPE-Sistemas, Universidade Federal do Rio de Janeiro, Caixa Postal 68511, 21945-970, Rio de Janeiro, RJ, Brasil
[d]COPPE-Sistemas, Universidade Federal do Rio de Janeiro, Caixa Postal 68511, 21945-970, Rio de Janeiro, RJ, Brasil
[e]IM and NCE, Universidade Federal do Rio de Janeiro, Caixa Postal 2324, 20001-970, Rio de Janeiro, RJ, Brasil

Abstract

It is well known that a clique with $k+1$ vertices is the only minimal obstruction to k-colourability of chordal graphs. A similar result is known for the existence of a cover by ℓ cliques. Both of these problems are in fact partition problems, restricted to chordal graphs. The first seeks partitions into k independent sets, and the second is equivalent to finding partitions into ℓ cliques. In an earlier paper we proved that a chordal graph can be partitioned into k independent sets and ℓ cliques if and only if it does not contain an induced disjoint union of $\ell+1$ cliques of size $k+1$. (A linear time algorithm for finding such partitions can be derived from the proof.)

In this paper we expand our focus and consider more general partitions of chordal graphs. For each symmetric matrix M over 0, 1, $*$, the M-partition problem seeks a partition of the input graph into independent sets, cliques, or arbitrary sets, with certain pairs of sets being required to have no edges, or to have all edges joining them, as encoded in the matrix M. Moreover, the vertices of the input chordal graph can be equipped with lists, restricting the parts to which a vertex can be placed. Such (list) partitions generalize (list) colourings and (list) homomorphisms, and arise frequently in the theory of graph perfection. We show that many M-partition problems that are NP-complete in general become solvable in polynomial time for chordal graphs, even in the presence of lists. On the other hand, we show that there are M-partition problems (without lists) that remain NP-complete for chordal graphs. It is not known whether or not each list M-partition problem is NP-complete or polynomial, but it has been shown that each is NP-complete or quasi-polynomial ($n^{O(\log n)}$). For chordal graphs even this 'quasi-dichotomy' is not known, but we do identify large families of matrices M for which dichotomy, or at least quasi-dichotomy, holds.

We also discuss forbidden subgraph characterizations of graphs admitting an M-partition. Such characterizations have recently been investigated for partitions of perfect graphs, and we focus on highlighting the improvements one can obtain for the class of chordal, rather than just perfect, graphs.

Keywords: Chordal graphs; List-colourings; List homomorphisms; Matrix partitions; Forbidden subgraph characterizations; Dichotomy

* Corresponding author.
E-mail addresses: tomas@theory.stanford.edu (T. Feder), pavol@cs.sfu.ca (P. Hell), sula@cos.ufrj.br (S. Klein), loana@cos.ufrj.br (L.T. Nogueira), fabiop@nce.ufrj.br (F. Protti).

0304-3975/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2005.09.030

1. Introduction

The *M-partition problem* was introduced in [13,14]. Let M be a fixed symmetric $m \times m$ matrix with entries $M(i, j) \in \{0, 1, *\}$. An M-partition of an input graph G is a partition of vertices in G into m *parts*, corresponding to the rows (and columns) of the matrix M, such that for distinct vertices x and y of the graph G, placed in parts i and j (possibly with $i = j$), respectively, we have the following:

- if $M(i, j) = 0$, then xy is not an edge of G;
- if $M(i, j) = 1$, then xy is an edge of G.

(If $M(i, j) = *$, then xy may or may not be an edge in G.)

Note that the diagonal entries of M describe the parts of an M-partition ($M(i, i) = 0$ means the ith part is independent, $M(i, i) = 1$ means the ith part is a clique, and $M(i, i) = *$ means there is no restriction on the ith part), while the off-diagonal entries of M describe the connections between the parts ($M(i, j) = 0$ means there are no edges between the ith and jth parts, $M(i, j) = 1$ means there are all edges between them, and $M(i, j) = *$ means there is no restriction).

The *list M-partition problem* assumes that the input graph G is equipped with a collection of *lists* $L(x), x \in V(G)$, each list being a set of parts. A list M-partition of such input graph G with lists $L(x), x \in V(G)$, is an M-partition of G, such that each vertex x of G is placed in a part $i \in L(x)$.

The *complementary matrix* to a matrix M is the matrix M' obtained from M by replacing all 0's by 1's and conversely. The M-partition and M'-partition problems are equivalent, since a graph G is M-partitionable if and only if its complement $\overline{G}$ is M'-partitionable, and similarly for the list M-partition and list M'-partition problems. (We note, however, that chordal graphs are not closed under complementation, so the chordal restrictions of these problems, introduced below, are not equivalent.)

Suppose H is a graph with m vertices and M is obtained from the adjacency matrix of H by replacing each 1 by $*$. Then each homomorphism (edge-preserving vertex mapping) f of G to H corresponds to an M-partition of G, where the parts are $f^{-1}(h), h \in V(H)$. In particular, when $H = K_m$, the matrix M is the matrix with all diagonal entries 0 and all off-diagonal entries $*$, and an M-partition of G is simply an m-colouring of G. Thus M-partitions generalize colourings and homomorphisms, and list M-partitions generalize list-colourings and list-homomorphisms [12,22].

With this in mind, we may define a *trigraph* H to consists of a set of vertices, any two of which may either form a non-edge, a weak edge, or a strong edge. The *adjacency matrix* of a trigraph H with m vertices is the symmetric $m \times m$ matrix M, with rows (and columns) indexed by the vertices of H, which has $M(i, j) = 0$ if ij is a non-edge, $M(i, j) = *$ if ij is a weak edge, and $M(i, j) = 1$ if ij is a strong edge. A *homomorphism* of a graph G to a trigraph H is a mapping f of the vertices of G to the vertices of H such that the partition formed by parts $f^{-1}(h)$, over all vertices h of H, is an M-partition of G. This point of view is further explored in [18,22].

List matrix partitions are also useful in unifying many partition problems arising in the study of perfect graphs. Often these problems are not stated in terms of partitions, but are in fact equivalent to partition problems. For instance, it is evident that G is a *split graph* (admits a partition into a stable set and a clique [19]), if and only if it admits an M-partition where M is the matrix

$$\begin{pmatrix} 0 & * \\ * & 1 \end{pmatrix}.$$

Less obviously, a graph G has a clique cutset [25,27], if it admits an M-partition, into non-empty parts, where M is the matrix

$$\begin{pmatrix} 1 & * & * \\ * & * & 0 \\ * & 0 & * \end{pmatrix}.$$

A similar approach allows us to model by M-partitions problems such as having an independent cutset [26], a skew cutset [6,7], a homogeneous set [23], or being a join of various kinds [5]. These connections are explored in more detail in [14,22], where it is in particular explained how to model restrictions on the size of the parts (for instance requiring parts to be non-empty) by introducing lists. Lists are especially useful because they allow recursing to smaller subproblems; thus the introduction of lists resulted in the solution of certain M-partition problems that were previously open [4,6,7], cf. [22].

In [13,14] the authors have given polynomial time algorithms for many list M-partition problems, and quasi-polynomial ($n^{O(\log n)}$) time algorithms for certain others. In [9] the authors have shown that all list M-partition problems are solvable in quasi-polynomial time, or are NP-complete. We call such a result a *quasi-dichotomy*. Many of our quasi-polynomial time algorithms from [14] were improved to polynomial time algorithms in [4,7], but it is not known whether all list M-partition problems are polynomial time solvable or NP-complete, even for matrices of size four [4,17]. This is known as the *Dichotomy Problem* for list M-partitions [9,22].

Recall that a graph is called *chordal* if it does not have an induced cycle of length greater than three. Equivalently [19], a graph is chordal if and only if its vertices can be enumerated as $v, v_2, \dots, v_n$ so that any two neighbours v_j, v_k of a v_i with $i < j,\ i < k$ are adjacent; such an enumeration is called a *perfect elimination ordering*. A graph G is called *perfect* if G and all its induced subgraphs have chromatic number equal to their maximum clique size. It is known that each chordal graph is perfect [19].

In this paper, we consider the restrictions of both the M-partition and the list M-partition problems to chordal input graphs G. We call these restricted problems the *chordal M-partition* and *chordal list M-partition problems*. (Clearly, the chordal M-partition problem is a restriction of the chordal list M-partition problem.) A preliminary version with some of these results has appeared in [15].

There are several classical examples to suggest that M-partitions of chordal graphs can be found in polynomial time. For instance, k-colourability of chordal graphs (M is the $k \times k$ matrix with 0 on the diagonal and $*$ everywhere else) can be decided in time $O(m + n)$ using a perfect elimination ordering [19]; in fact, the algorithm either produces a k-colouring of the input graph or produces the unique forbidden subgraph K_{k+1}. A similar result is known about clique covering (M is the $\ell \times \ell$ matrix with 1 on the diagonal and $*$ elsewhere). In [20,21] we have given, more generally, a linear time recognition algorithm, and a forbidden subgraph characterization, of chordal graphs that can be partitioned into k independent sets and ℓ cliques (M has k zeros and ℓ ones on the diagonal, $*$ everywhere else). Partionability into k independent sets and ℓ cliques has first been studied by [2], and is a natural generalization of the problem of recognizing split graphs (cf. also [3]). This partition problem is NP-complete for graphs in general, unless $k \leqslant 2$ and $\ell \leqslant 2$, and polynomial time solvable in these cases [2,13,14]. (Split graphs have $k = \ell = 1$.)

We now expand our attention to the general M-partition and list M-partition problems for chordal graphs. We find many classes of matrices M for which these problems can be solved in polynomial time for chordal graphs. However, we also find M-partition problems that remain NP-complete for chordal graphs, even in the absence of lists. Certain dichotomy and quasi-dichotomy results will also be proved. Finally, we will discuss forbidden subgraph characterizations of M-partitionability.

We focus on a particular kind of matrices M. For the most part, they will be matrices without $*$ on the diagonal. Note that the M-partition problem without lists is trivial if M contains a diagonal $*$, as all vertices of the input graph G can be placed to the corresponding (unrestricted) part. If M is a matrix without diagonal $*$, we may simultaneously permute its rows and columns so the diagonal has first k zeros and then ℓ ones (with $k + \ell = m$). Thus M consists of a $k \times k$ diagonal matrix A with zero diagonal, and an $\ell \times \ell$ diagonal matrix B with a diagonal of ones, and an off-diagonal $k \times \ell$ matrix C (and its $\ell \times k$ transpose). In this case, we say that M is an (A, B, C)-*block matrix*. In one exceptional case, we shall also admit the diagonal to contain $*$'s, both in the diagonal block A, and the diagonal block B. We will indicate this by calling the matrix M an (A, B, C)-block matrix *with diagonal $*$'s allowed*. (Of course any M can be put in the form of an (A, B, C)-block matrix with diagonal $*$'s allowed, by simultaneous row/column permutations; but we will still find this terminology useful.) In the last section, we shall consider a further restriction. Let $E(A)$ denote the set of entries (0, 1, or $*$) which appear in the off-diagonal positions of A, let $E(B)$ be the set of all entries which appear in the off-diagonal positions of B, and let $E(C)$ be the set of all entries which appear in C. We shall say that a subset of $\{0, 1, *\}$ is *normal* if it does not contain both an $*$ and another element. Thus every normal set is either $\{*\}$ or a subset of $\{0, 1\}$. We shall say that a matrix M is *normal* if all of $E(A)$, $E(B)$, $E(C)$ are normal sets. Note that a matrix in which $E(A)$, $E(B)$, $E(C)$ are all singletons, say $\{a\}$, $\{b\}$, $\{c\}$, respectively, is always normal. Such a matrix will be called an (a, b, c)-*block matrix*.

A final bit of general notation: If S, T are sets of parts for an m by m matrix M (i.e., subsets of $\{1, 2, \dots, m\}$), we denote by $M(S, T)$ the set of all entries $(0, 1, *)$ which occur as $M(s, t)$, $s \in S$, $t \in T$.

2. Algorithms for chordal list matrix partitions

Consider first the case when M is a $k \times k$ matrix with zero diagonal, i.e., when M is an (A, B, C)-block matrix with $\ell = 0$.

Theorem 2.1. *If all diagonal entries of M are zero, then the chordal list M-partition problem can be solved in time $O(nk(2k)^k)$, linear in n.*

Proof. A chordal graph G which admits an M-partition with such a matrix M cannot have a clique with $k+1$ vertices; hence it must have treewidth at most $k-1$. We can test whether G has treewidth at most $k-1$ in polynomial time, since k is fixed. For graphs of bounded treewidth, the existence of a list M-partition can be tested by standard dynamic programming techniques [1,8,24]. Recall that a tree decomposition of a graph G is a pair (X, U) where U is a tree and $X = (X_i)_{i \in V(U)}$ is a collection of subsets of $V(G)$ whose union equals $V(G)$, such that each edge xy of G is included in some X_i, and such that for each vertex x of G, the set of all X_i containing x forms a subtree of U. The treewidth of a decomposition is the maximum value of $|X_i| - 1$, and the treewidth of a graph is the minimum treewidth of a decomposition.

A tree decomposition in which U has a fixed root r is called *nice* [1] if each node of the rooted tree U has at most two children, and the following conditions are satisfied: If i has two children (a *join node*), say j and h, then $X_i = X_j = X_h$; if i has one child j then X_i is obtained from X_j by adding (an *introduce node*) or deleting (a *forget node*) a single vertex of G, and if $|X_i| = 1$ for each leaf (*start node*) i of U. It is known that, for a chordal graph of treewidth $k-1$, a nice tree decomposition, also of treewidth $k-1$, can be obtained in linear time [1].

Given a nice tree decomposition (X, U) of G with root r, we denote by G_i the subgraph of G induced by the union of X_i and all X_j where j is a descendant of i. Let $F(i)$ be the set of all pairs (Π, S), where Π is an assignment of the vertices in X_i to parts, obtained by restricting a list M-partition Σ of G_i, and S is the set of those parts in the partition Σ which contain vertices of $G_i - X_i$. Note that each $F(i)$ has at most $k^k 2^k$ elements.

We can compute the set $F(i)$ for any node, once all its descendants j have had their values $F(j)$ calculated. This is not hard to see, considered separately the start, introduce, forget, and join nodes. For instance, suppose i is a forget node, with the unique child j, and $X_i = X_j - x$. For each $(\Pi, S) \in F(j)$ we add to $F(i)$ the pair (Π', S'), where Π' is Π restricted to X_i and S' equals either S, if the part a that x was assigned in Π was already present in S, or equals $S \cup a$. On the other hand, if i is an introduce node, with the unique child j and $X_j = X_i - x$, then for each $(\Pi, S) \in F(j)$ we consider all possible values x can take with the current assignment Π, because of the adjacencies of x in X_j, and also because of the non-adjacencies of x in $G_i - X_i$; it is for this purpose that we keep track of the set S.

The above proof yields in fact an algorithm for the list M-partition problem for graphs of treewidth at most $k-1$, of complexity $O(nk(2k)^k)$. (The complexity analysis is easily adapted from that of [8].) Recall that M, and hence k, is fixed, so this is a linear time algorithm. (A similar remark applies to the other algorithms in this paper.) □

We next consider the case when M is an $\ell \times \ell$ matrix with all diagonal entries 1, i.e., an (A, B, C)-block matrix with $k = 0$. Let G be a chordal graph with lists $L(x), x \in V(G)$. A *rectangle* in G is a collection of sublists $L_x \subseteq L(x), x \in V(G)$, such that any choice of parts from L_x for each x constitutes a list M-partition of G. Note that a rectangle can be the empty set.

Theorem 2.2. *Let M be an (A, B, C)-block matrix with $k = 0$, and G a chordal graph with lists $L(x), x \in V(G)$. The set of all list M-partitions of G is the union of $n^{2\ell}$ rectangles, and can be found in time $O(n^{2\ell+d})$, for some constant d. Thus the chordal list M-partition problem can be solved in polynomial time.*

Proof. Consider a perfect elimination ordering τ of the graph G, and a particular list M-partition of G. Let the ith part be non-empty, and let x_i, y_i denote the first and last vertices, in the ordering τ, which belong to the ith part. (Note that $x_i = y_i$ is possible, if the i-part consist of a single vertex.) Thus for each i we can choose the i-part to be empty, or to have just one vertex, or to have the first vertex x_i and the last vertex y_i; altogether $1 + n + \binom{n}{2}$ choices. We simplify this to $1 + n\binom{n}{2} \leqslant n^2$, since we may take $n > 1$. We shall show that each such choice, for $i = 1, \ldots, \ell$, corresponds to a rectangle of M-partitions.

For all empty parts i, we remove i from the lists of all vertices. For all other parts i, we have pairs x_i, y_i (possibly equal): we remove part i from the list of any vertex that occurs either before x_i or after y_i in the ordering τ. We also remove from the list of each vertex z those parts j which are forbidden by the adjacency or non-adjacency of z to the vertices x_i, y_i. That is, we remove from $L(z)$ the part j if there is an edge zx_i or an edge zy_i in G and $M(i, j) = 0$, or if there is no edge zx_i or no edge zy_i in G and $M(i, j) = 1$. We denote the remaining sublists of $L(x)$ by L_x and claim they form a rectangle, i.e., that any assignment of parts from the lists L_x consitutes an M-partition of G. Indeed, suppose that adjacent vertices z, t were assigned parts i, j, respectively, but $M(i, j) = 0$. Say z occurs before t in the perfect elimination ordering. Then z is adjacent to y_i, since $M(i, i) = 1$. Thus y_i and t are both neighbours of z, and both occur after z, so y_i is adjacent to t by the definition of a perfect elimination ordering. Since $M(i, j) = 0$, part j would have been removed from the list of t, a contradiction. On the other hand, suppose non-adjacent vertices z, t were assigned parts i, j, respectively, but $M(i, j) = 1$. Say z occurs before t. Then x_i is adjacent to z since $M(i, i) = 1$. Also x_i is adjacent to t since $M(i, j) = 1$. Thus x_i is adjacent to both z_i and z_j, and both occur after x_i, so z is adjacent to t by the definition of a perfect elimination ordering, a contradiction. □

Feder et al. [13,14] introduced the following technique. Let $\mathcal{A}$ and $\mathcal{B}$ be two classes of graphs that are closed under taking induced subgraphs, and for which membership can be tested in polynomial time. Suppose further that there exists a constant c such that any graph both in $\mathcal{A}$ and in $\mathcal{B}$ has at most c vertices. Consider the problem of partitioning the vertices of a graph G into two induced subgraphs G_A and G_B so that G_A is in $\mathcal{A}$ and G_B is in $\mathcal{B}$. It is shown in [13,14] that there are at most n^{2c} such partitions, and that all such partitions can be found in polynomial time.

We shall apply this technique to chordal list M-partition problems for certain (A, B, C)-block matrices M. Consider first the case when all entries of the block C are $*$'s. Let $\mathcal{A}$ denote the class of chordal graphs that admit an A-partition, and let $\mathcal{B}$ denote the class of chordal graphs that admit a B-partition. Clearly, both these classes are closed under taking induced subgraphs. Furthermore, the membership problems for $\mathcal{A}$, $\mathcal{B}$, i.e., the chordal A-partition and B-partition problems are polynomial time solvable by Theorems 2.1 and 2.2, respectively. Finally, we note that a graph in $\mathcal{A}$ is k-colourable, and a graph in $\mathcal{B}$ can be covered by ℓ cliques. Thus a graph in both $\mathcal{A}$ and $\mathcal{B}$ can be covered by ℓ cliques, each of size at most k, and hence has at most $c = k\ell$ vertices. Since C has all entries $*$, a chordal graph G admits an M-partition if and only if it can be partitioned into induced subgraphs G_A and G_B where G_A is in $\mathcal{A}$ (i.e., G_A admits an A-partition), and G_B is in $\mathcal{B}$ (i.e., G_B admits a B-partition). The above result from [13,14] assures that the chordal list M-partition problem can be solved in polynomial time, in this case.

More generally, using the same technique, we shall solve the chordal list M-partition problem for (A, B, C)-block matrices, in which C has the following special form: Call a matrix C *crossed* if each non-$*$ entry belongs to a row or a column of non-$*$ entries. (We remark that this notion of a crossed matrix generalizes that given in [15], and hence our Theorem 2.3 is significantly more general than the result in [15].)

Theorem 2.3. *Suppose M is an (A, B, C)-block matrix.*
If C is crossed, then the chordal list M-partition can be solved in time polynomial in n.

Proof. Recall that $c = k\ell$. For each of the at most n^{2c} choices for a partition of G into induced subgraphs G_A, G_B where G_A admits an A-partition and G_B admits a B-partition, we choose at most one vertex of G_A for each part in A (a *representative* of the part), and at most one vertex of G_B for each part in B. This choice of representatives involves at most $(n + 1)^{k+\ell}$ additional possibilities (some parts may be left empty), still a number of choices polynomial in n. Now we modify the lists of vertices in G as follows: Vertices of G_A have all parts from B removed from their lists, and similarly vertices of G_B have all parts from A removed from their lists. If a part i was chosen empty, we remove i from all the lists. On the other hand, suppose a vertex x was chosen to represent part i. We remove all elements different from i from the list of x; we also remove from the list of any vertex non-adjacent to x all parts j such that $M(i, j) = 0$, and remove from the list of any vertex non-adjacent to x all j such that $M(i, j) = 1$. If furthermore all entries in row i of the matrix C are non-$*$'s, we let J_0 denote the set of parts j in B such that $C(i, j) = 0$, and let J_1 denote the set of parts j in B such that $C(i, j) = 1$. All vertices of G_B adjacent to x have all parts in J_0 removed from their lists, and all vertices of G_B nonadjacent to x have all parts from J_1 removed from their lists. (At this point each list is contained in J_0 or J_1.) Moreover, we remove i from the list of any vertex in G_A for which either some neighbour in G_B has its list contained in J_0, or some non-neighbour has its list contained in J_1. We proceed similarly if all entries in column j of C are non-$*$'s, reversing the roles of A and B, and rows and columns. Since C is crossed, we have now reflected all constraints between parts G_A and G_B in the lists, i.e., there exists a list M-partition of G (with the modified lists) if and only if there is a list M^*-partition of G (with the same modified lists), where M^* is obtained from M by replacing all entries of C by $*$. This is the problem solved above the theorem. $\square$

In the special case when C has all rows the same, the complexity can be improved to $n^{2\ell+\mathrm{O}(1)}(2k)^k$. We proceed as in the proof of Theorem 2.2, fixing a perfect elimination ordering of G, choosing ℓ pairs x_i, y_i for the parts i, and removing parts from lists of vertices they cannot be placed in, as explained there. Exactly as in that proof, it follows that any assignment of vertices to remaining parts of B on their lists, is a list B-partition. Each vertex x for which the remaining list L_x only contains parts of A, will be assigned to G_A (which concrete part it will be assigned to will be decided later). We then remove from the list of each neighbour of x in G_B all parts j such that the jth column of C consists of 0's, and remove from the list of each non-neighbour of x in G_B all parts j such that the jth column of C consists of 1's. (Note that the columns of C are constant, since all the rows of C are the same.) At this point, we may have created more vertices x with lists containing only parts of A, and we repeat the process, as long as possible. Since each iteration decides to place at least one vertex in G_A, we only repeat this process at most n times, and the procedure takes only polynomial time. At the end, we solve the list A-partition problem for G_A, in time $\mathrm{O}(nk(2k)^k)$.

More generally, we have the following result:

Theorem 2.4. *Suppose M is an (A, B, C)-block matrix in which A and B themselves consist of diagonal blocks A_i and B_j, respectively, with C being partitioned correspondingly into blocks $C_{i,j}$.*

If all entries of A outside of the diagonal blocks A_i are 1, *all entries of B outside of the diagonal blocks are zero, and all block matrices $C_{i,j}$ are crossed, then the chordal list M-partition problem can be solved in time polynomial in n.*

Proof. Since G_B does not have an independent set of size $\ell + 1$, it follows that G_B has at most ℓ components. Each component must be placed in a single B_i. If there are q diagonal blocks B_i, then there are at most ℓ^q ways of choosing which component is placed in which G_{B_i}. Similarly, the complement $\overline{G_A}$ does not have an independent set of size $k + 1$, so $\overline{G_A}$ has at most k components, each of which is placed to a single A_i. If there are p blocks A_i, then there are at most k^p ways of choosing which component is placed in which G_{A_i}. The problem is thus reduced, after $\ell^q k^p$ choices, to a problem involving just a single A_i, a single B_j, and C_{ij}, which is solved as in the previous theorem. $\square$

3. Dichotomies for list matrix partitions

In the previous section we have seen general classes of matrices M for which the chordal list M-partition problems are polynomial time solvable. In this section we examine some cases of NP-complete chordal list M-partition problems. (There are even NP-complete chordal M-partition problems without lists, as we show in the next section.)

It is not known whether or not every chordal list M-partition problem is polynomial time solvable or NP-complete. Such *dichotomy* is not known for general list M-partition problems either, even for matrices of size four [4], and is referred to as the *Dichotomy Problem* for list M-partitions. In [12] (see [14]), dichotomy is shown for matrices M which have no 1's, or have no 0's, or have no $*$'s. In [9], the authors prove a *quasi-dichotomy* of all list M-partition problems, i.e., prove that for each matrix M the list M-partition problem is solvable in quasi-polynomial time (time $n^{\mathrm{O}(\log n)}$) or is NP-complete.

More generally, we have the following theorem [9]. Suppose M is a fixed m by m matrix, and we also fix a set Λ of subsets of $\{1, 2, \ldots, m\}$ which is closed under taking subsets (i.e., if $L \in \Lambda$, $L' \subseteq L$, then $L' \in \Lambda$). The *Λ-restricted list M-partition problem* is a restriction of the list M-partition problem to instances which are graphs G with lists that are elements of Λ. We further say that a matrix M is *Λ-compatible* if the sets $M(L, L')$ for L, L' in Λ never contain both 0 and 1.

Theorem 3.1 (*Feder and Hell [9]*). *Each Λ-restricted list M-partition problem is NP-complete or solvable in quasi-polynomial time.*

Moreover, if M is Λ-compatible, then the Λ-restricted list M-partition problem is NP-complete or solvable in polynomial time.

We first explain how to obtain an NP-complete chordal list M-partition problem. Let H be a fixed graph, and let M be obtained from the adjacency matrix of H by replacing all 1's with $*$'s. Recall from the Introduction that list homomorphisms of G to H can be viewed as list M-partitions of G. A list homomorphism of G to H is also called a *list H-colouring of G*. The *list H-colouring problem* asks whether or not a given G has a list H-colouring. Thus the list H-colouring problem is equivalent to the list M-partition problem.

When H is bipartite, we may assume that the input G is also bipartite, and that white vertices of G have lists consisting of white vertices of H, and similarly for black vertices. The matrix M obtained as above is an block matrix, with a zero diagonal block matrix X corresponding to the white vertices of H, a zero diagonal block matrix Y corresponding to the black vertices of H, and the off-diagonal matrix Z, whose rows correspond to the white vertices and columns to the black vertices of H. The (i, j)th entry of Z is $*$ if the white vertex i and the black vertex j are adjacent in H, and is 0 otherwise. We call this matrix Z the *matrix corresponding to H*.

For bipartite graphs H, Feder et al. [11] showed that the list H-colouring problem is polynomial time solvable if H is the complement of a circular arc graph (we shall say that in this case H is a *a cocircular graph*), and is NP-complete otherwise. Based on this result, it will be possible to find NP-complete chordal list M-partition problems.

Theorem 3.2. *Let M be an (A, B, C)-block matrix, with diagonal $*$'s allowed. Let H be a bipartite graph that is not cocircular, and let Z be the matrix corresponding to H.*

If A does not contain any 1's, B does not contain any 0's, and C is the matrix Z or its complementary matrix, then the chordal list M-partition problem is NP-complete.

Proof. We reduce the list H-colouring problem to the chordal list M-partition problem. We may assume that C is the matrix Z, otherwise we replace the input G by its bipartite complement (exchanging edges and non-edges between the white and black vertices). Given input G, obtain the graph G' by adding all edges between pairs of black vertices. (The lists of G' remain the same as in G.) It is easy to see that G has a list H-colouring if and only if G' has a list M-partition. Since G' is a split graph (it can be partitioned into a clique and an independent set), it is also chordal [19]. □

The proof implies that the list M-partition problems obtained from bipartite graphs H that are not cocircular are NP-complete even when restricted to split graphs.

If we further restrict the matrices A and B to be the all-zero and all-one matrices, we can actually prove dichotomy.

Theorem 3.3. *Let M be an (A, B, C)-block matrix, where A is the all-zero matrix, B is the all-one matrix, and C or its complement corresponds to a bipartite graph H.*

The chordal list M-partition problem is polynomial if H is a cocircular graph and is NP-complete otherwise.

Proof. By complementation, we may again assume that C is the matrix corresponding to H. If H is not a cocircular graph, the result follows from Theorem 3.2. Suppose now that H is a cocircular graph. Given a chordal graph G with lists, if G is not a split graph then it does not have any M-partition. Otherwise, we can generate all $O(n^2)$ split partitions of G (a clique and an independent set have at most one vertex in common, so the technique from [14] discussed above applies). For each such partition of the vertices of G into white vertices (forming an independent set) and black vertices (forming a clique), we can remove the edges joining the black vertices, obtaining a bipartite instance G' of the list H-colouring problem, which can be solved in polynomial time by [11]. □

When A and B are as above, the all-zero and all-one matrices, we obtain quasi-dichotomy even in the case when C is any matrix.

Theorem 3.4. *Let M be an (A, B, C)-block matrix, where A is the all-zero matrix and B is the all-one matrix.*

Then the chordal list M-partition problem is quasi-polynomial or NP-complete.

Proof. Theorem 3.1 claims that every list M-partition problem is quasi-polynomial or NP-complete. Those list M-partition problems that are quasi-polynomial for general graphs, remain (at most) quasi-polynomial for chordal graphs. It is easy to see that those list M-partition problems that are NP-complete for general graphs remain NP-complete for chordal graphs, since inputs that are not chordal are not split, and hence do not have an M-partition. (Testing whether a graph is split is polynomial, as noted above, or see [19].) □

We have not proved quasi-dichotomy for all chordal list M-partition problems. Some general classes for which we have quasi-dichotomy are discussed below.

We begin by observing that the previous two theorems extend to matrices B which have $*$'s off the diagonal. Suppose M is an (A, B, C)-block matrix where A is the all-zero matrix and B has no 0's off the diagonal, and M' is obtained from M by changing all entries of B to be 1. Then the chordal list M- and M'-partition problems are polynomial time equivalent. Indeed, given an instance G for M, we may consider each choice of possible G_A and G_B. For each G_B, we may consider each of the possible rectangles from Theorem 2.2. With these restricted lists, we may replace G_B with a complete graph, yielding a split (and hence chordal) instance G' for M'. Conversely, given an instance G' for M', we may consider each choice of possible G_A and G_B as an independent set and a clique, and then treat the resulting problem as an instance for M. We have proved the following extension.

Corollary 3.5. *The dichotomy in Theorem* 3.3 *and the quasi-dichotomy in Theorem* 3.4 *hold even if the requirement on B is weakened to allow $*$'s off the diagonal.*

In other words, A is assumed to be the all-zero matrix and B is assumed to have no 0's and to have a diagonal of 1's. Of course, we could similarly keep B as an all-one matrix and correspondingly weaken the assumption on A.

Our broadest dichotomy and quasi-dichotomy result deals with the situation where A (or, similarly, B) has no $*$'s.

Theorem 3.6. *Let M be an (A, B, C)-block matrix, where A has no $*$'s.*
Then the chordal list M-partition problem is quasi-polynomial or NP-complete.
If, additionally, C has no 1*'s, or no* 0*'s, then the chordal list M-partition problem is polynomial or NP-complete.*

Proof. The proof proceeds by repeatedly reducing the problem to polynomial sized families of subproblems. (The original problem has a solution, if and only if all subproblems have a solution.) Since the reductions are polynomial, the existence of quasi-polynomial algorithms for all these subproblems will imply a quasi-polynomial time algorithm for the whole problem. On the other hand, if even one of the subproblems is NP-complete, then the whole problem is also NP-complete. At the end of the process, we will obtain problems for which the chordality of the input graph G is necessary, i.e., matrices M such that graphs that are not chordal do not admit an M-partition. Consider each such list M-partition problem, in both the general version and the chordal restriction. If the general list M-partition problem is quasi-polynomial, then so is the chordal restriction. Otherwise, by Theorem 3.1, the general list M-partition problem is NP-complete, and we can polynomially reduce it to the chordal list M-partition problem as follows. Given an instance G with lists, we first test whether G is chordal. (This can be done in linear time [19].) If G is not chordal, we associate it with some fixed chordal graph G_0 (with lists) which does not admit an M-partition. (Such a G_0 must exist, see below; this assures that we obtain a negative answer about the existence of a list M-partition.) If G is chordal, we simply associate it with G (and the same lists). It now follows that G (with the lists) has an M-partition if and only if the associated chordal graph (with lists) has an M-partition. Thus also the chordal list M-partition problem is NP-complete. Hence we obtain quasi-dichotomy (and dichotomy) of the corresponding chordal list M-partition problems, by Theorem 3.1.

To see that such a chordal graph G_0 with lists exits, suppose that the list M-partition problem is NP-complete but every chordal graph G with lists admits a list M-partition. Recall that we also assume that nonchordal graphs with lists do not admit M-partitions. Thus we can test whether or not a given graph G with lists admits an M-partition by testing whether or not G is chordal, which is polynomial [19]. Thus the general list M-partition problem is quasi-polynomial, and treated earlier.

We illustrate this idea to reduce the original problem to problems in which the matrix A has only zero entries. This will be useful at several points in the present proof (especially at the end), and will allow us to introduce the technique in a simple context.

For $i \neq j$, if $A(i, j) = 1$, then at most one vertex can be placed in part i or at most one vertex is placed in part j, otherwise we would have a chordless four-cycle in G, contrary to chordality. We can thus choose zero or one vertices x to be placed in i or in j in every possible way. This defines $2(n + 1)$ subproblems with the above properties. In each subproblem, we remove i (or j) from all other lists, remove all parts i' with $M(i, i') = 0$ from all neighbours of x and all parts i' with $M(i, i') = 1$ from all non-neighbours of x, and delete x from the input graph. This removes part i from the consideration. We then do the same step for another entry 1 in the new matrix A, if any. At the end of this process we will have a polynomial family of subproblems in which the resulting diagonal matrices A will contain only zeros. Thus in the following we assume that A is the all-zero matrix.

We will again use the technique from [14], with $\mathcal{A}$ being the class of A-partitionable chordal graphs, and $\mathcal{B}$ the class of B-partitionable chordal graphs. Note that, since $A = 0$, the A-partitionable graphs are precisely graphs without edges. As before, the membership problems for these graph classes can be solved in polynomial time. Moreover, a graph that is both in $\mathcal{A}$ and $\mathcal{B}$ has at most $c = \ell$ vertices. Therefore, there are only $O(n^{\ell})$ partitions of the input chordal graph G into two subgraphs G_A and G_B with G_A in $\mathcal{A}$ and G_B in $\mathcal{B}$, and the problem is reduced to solving this polynomial sized family of subproblems. Moreover, for each such partition of G, there are at most $n^{2\ell}$ rectangles describing all the possible B-partitions of G_B, as stated in Theorem 2.2, and the problem with this partition is reduced to the family of subproblems with the lists as given in the rectangle. Thus it suffices to focus on a particular rectangle. It is described by lists L_x, $x \in V(G_B)$, where any assignment of vertices $x \in V(G_B)$ to members of their lists L_x is a list B-partition of G_B.

Note that two vertices x, y of G_B with lists $L_x = S$, $L_y = T$, are either adjacent, and then $M(S, T)$ does not contain 0, or non-adjacent, in which case $M(S, T)$ does not contain 1. Thus $M(S, T)$ never contains both 0 and 1 for sets of parts S, T which occur as lists of some vertices, and in particular, $M(S, S)$ never contains 0 if S occurs as a list, since B has a diagonal of 1's.

We shall modify the matrix B by replicating rows and columns. To replicate a row and column i means to replace it with a set of identical rows and columns; specifically, each replacing part i' has $M(i', j) = M(j', i) = M(i, j)$ for all j, including $j = i$. (In particular, $M(i', i) = M(i, i') = M(i', i') = M(i, i) = 1$.) Each list L_x, $x \in V(G)$, is a set S of parts, corresponding to a set of rows (and columns) of B. If a part i belongs to f_i such sets S, we shall replicate the row and column i by f_i rows and columns. This way we can ensure that the lists are either equal or disjoint. Note that $f_i < 2^m$ where m is the (fixed) size of M.

We may assume that all parts of B are actually used in the lists L_x (since we may always simplify the matrix M by eliminating rows and columns corresponding to parts which do not occur lists). The parts in B are now partitioned into subsets each of which is the list L_x of at least one vertex x of the input graph G. This gives the matrix B a block structure—we may assume that its rows and columns are partitioned into sets $T_1, T_2, \ldots, T_p$ where each block determined by a pair of these sets has $B(T_a, T_b)$ subset of $\{0, *\}$ or of $\{1, *\}$.

Suppose T_a, T_b are such that no pair of adjacent vertices x, y have $L_x = T_a$, $L_y = T_b$. (This must happen, in particular, if $B(T_a, T_b)$ contains 0.) Then we can replace all entries in the block corresponding to the pair T_a, T_b by zeros. Now the non-zero blocks do not contain any zeros. We may define a graph H whose vertices are the parts of B and in which parts i, j are adjacent just if $B(i, j)$ contains no 0's. We note for future reference that the graph H is chordal. (Any chordless cycle in H induces a chordless cycle in G.) By symmetry, we may assume that each block of B is constant, $B(T_a, T_b)$ is $\{0\}$, $\{1\}$, or $\{*\}$.

If $B(T_a, T_b) = \{0\}$, we claim that there is at most one vertex u in G_A adjacent to a vertex x in G_B with list $L_x = T_a$ and a vertex y in G_B with list $L_y = T_b$. Indeed, if $u_1 \in G_A$ has such neighbours $x_1, y_1 \in G_B$ and $u_2 \in G_A$ has such neighbours $x_2, y_2 \in G_B$, then $y_1u_1x_1x_2u_2y_2y_1$ induce a chordless cycle of length between four and six (since x_1 may coincide with x_2 and y_1 may coincide with y_2), contrary to chordality. If such a vertex u exists, we assign it in every possible way to the members of its list, and remove it from the graph, after having reflected its assignment, in the usual way, in the lists of its neighbours and non-neighbours. Since there are at most ℓ^2 blocks in B, we can do that for each block with $B(T_a, T_b) = \{0\}$, and reduce to $O(n^{\ell^2})$ subproblems. Therefore, we may assume that if $B(T_a, T_b) = \{0\}$ then there is no vertex of G_A adjacent to some x in G_B with $L_x = T_a$ and some y in G_B with $L_y = T_b$.

For each vertex x in G_A, we now consider the set $f(x)$ consisting of all T_a which occur as lists L_y of neighbours y of x in G_B. We now define sets $S_1, S_2, \ldots, S_q$ of parts of A, where each subscript $r = 1, 2, \ldots, q$ is a possible value of $f(x)$, i.e., a set of T_a's. We place $i \in S_r$ just if i occurs in some list $L(x)$ of a vertex x with $f(x) = r$. Thus each list of a vertex in G_A is included in some S_r. We note that there are only $2^p \leqslant 2^{\ell}$ possible values $f(x)$. We may thus replicate rows and columns of A to ensure that $f(x) \neq f(y)$ implies that the lists $L(x)$ and $L(y)$ belong to different sets S_r. (This is similar to the replication we did in B.) Therefore $S_1, S_2, \ldots, S_q$ is a partition of the parts of A. Now A also has a block structure, each block of A corresponding to a pair of sets S_r, S_t.

Since both A and B have a block structure, we also obtain a block structure on C—each pair S_r, T_a defines a block of C. We now make a modification of the matrix C. Suppose T_a is not in the set r: If $C(S_r, T_a)$ contains a 1 in some position $C(i, j)$, then we cannot have both the part i and part j non-empty. We thus replace the current problem with the two subproblems obtained by removing part i and by removing part j from M. On the other hand (still supposing T_a is not in the set r), if $C(S_r, T_a)$ contains an $*$, we may simply replace it by 0. Thus we may assume that all blocks with T_a not in the set r have $C(S_r, T_a) = \{0\}$.

Finally, we shall modify both the matrix M and the graph G, so that the modified G is chordal and has a modified list M-partition if and only if the original G has an original list M-partition. We replace each block of B with $B(T_a, T_b) = \{*\}$ by an all-one block, and add to G all edges xy (if not present) with $L_x = T_a$ and $L_y = T_b$. It is easy to deduce, from the fact that H is chordal, that the new graph G is also chordal.

Now we have a matrix M in which B has a 0, 1 block structure, corresponding to a chordal graph H, A is an all-zero matrix, and for each part i of A the parts j of B with $C(i, j) \neq 0$ form a clique in H (because of our assumption on the vertices of G_A and blocks of B with $B(T_a, T_b) = \{0\}$). It is now easy to check that any graph G with an M-partition must be chordal. We have completed the promised reduction to a polynomial family of subproblems in which chordality is necessary; this proves the quasi-dichotomy.

For the dichotomy, we only need to observe that at the end of the reductions we have a Λ-restricted M-partition problem, where Λ consists of the sets S_r and T_a, and all their subsets. If C has no 1's, or no 0's, the matrix M is Λ-compatible; thus the dichotomy follows from Theorem 3.1. $\square$

4. NP-complete matrix partition problems

We now focus on constructing NP-complete M-partition problems (without lists). Let H again be a bipartite graph. The H*-retraction* problem is the restriction of the list H-colouring problem to instances G containing H as a subgraph, and with lists either $L(g) = g$, if $g \in V(H)$, or $L(g) = V(H)$, otherwise. A list H-colouring of G is called an H*-retraction* of G, in this situation. Many bipartite graphs H are known to yield NP-complete H-retraction problems, although a complete classification of complexity is not known, and dichotomy has not been proved, for H-retractions. In particular, it is known that if H is an even cycle of length greater than four, the H-retraction problem is NP-complete [11].

Theorem 4.1. *For every bipartite graph H such that the H-retraction problem is NP-complete, there exists a matrix M_H such that the chordal M_H-partition problem (without lists) is also NP-complete.*

Proof. Let H be a bipartite graph such that the H-retraction problem is NP-complete. We first extend the graph H to a larger bipartite graph H', by attaching to each white vertex of H a path of length five and to each black vertex of H a path of length four. Note that all the leaves (vertices of degree one) of H' are black.

We now introduce an auxiliary problem, which we shall call the *weak H'-retraction problem*. Suppose that the bipartite graph H' has k black vertices, forming the set V_B, and let L denote the set of all black leaves of H'. An instance of the weak H-retraction problem is a bipartite graph G with a specified set X of k black vertices, such that each vertex of G not in X has at most one neighbour in X. A solution to the instance is an edge-preserving and colour-preserving mapping of the vertices of G to the vertices of H such that X is mapped bijectively to V_B. We now show that the H-retraction problem reduces to the weak H'-retraction problem.

Suppose G is an instance of the H-retraction problem, i.e., a bipartite graph containing H. We transform G to an instance G' (with a set X) of the weak H'-retraction problem as follows: Let X be another copy of the set V_B, disjoint from G. Consider the union of G and X, and identify each vertex of L in X with the corresponding vertex of L in G. Finally, add internally disjoint paths of length four joining all pairs of vertices of X which correspond to vertices in V_B of distance two or four in H'. Call the resulting graph G'. We now argue that G admits an H-retraction if and only if G' admits a weak H'-retraction.

On the one hand, suppose f is an H-retraction of G. Then f, extended by taking each vertex of $X - L$ to the corresponding vertex of V_B, is a weak H'-retraction of G'. For the other direction, we note that any bijection between X and V_B has to map vertices of L to vertices of L, since leaves in H' have exactly two vertices in H' at distance two or four, while black vertices of H' that are not leaves have at least three vertices in H' at distance two or four. Therefore, any weak H'-retraction of G' which maps the vertices of X bijectively to the vertices of V_B must map the copy of H' in G' isomorphically to H'. It follows that G admits an H'-retraction, which can easily be modified to an H-retraction by mapping all the added paths of H' into H.

Next, we define a matrix M_H such that the chordal M_H-partition problem (without lists) is NP-complete, as claimed in the theorem. The matrix M_H will be an (A, B, C)-block matrix in which the diagonal matrix A is an all zero matrix; the diagonal matrix B has all diagonal entries 1 and all other entries $*$; and finally, the matrix C will be the matrix corresponding to the bipartite graph H'.

We now reduce the weak H'-retraction problem to the M_H-partition problem. Given an instance G' for the weak H'-retraction problem, we construct an instance G'' of the M_H-partition problem as follows. We replace each white vertex a of G' by a set $I(a)$ of $k+1$ independent vertices (where $k = |V_B|$), and each black vertex b of G' by a clique $K(b)$ of two vertices. Whenever a and b are adjacent in G', all vertices of I_a are adjacent to all vertices of K_b in G''. Finally, we add all edges between K_b and $K_{b'}$ unless both b and b' are in X. Note that each vertex every $I(a)$ is adjacent to at most one $K(b)$ with $b \in X$.

We claim that G' admits a weak H'-retraction if and only if G'' admits an M_H-partition. Indeed, if f is a weak H'-retraction of G', all vertices of a set $I(a)$ can be placed in the part $f(a)$ and all vertices of a set $K(b)$ can be placed in the part $f(b)$. Conversely, each M_H-partition of G'' must place at least one of the two vertices in any $K(b)$ to a part in B, since A is an all-zero matrix. Also, if b, b' are both in X, these vertices must be placed in distinct parts of B. By a similar argument, at least one vertex of each $I(a)$ must be placed in a part in A, since the vertices placed to parts in B are covered by k cliques. This way we deduce an H'-retraction of G'.

It remains to argue that the instance G'' is a chordal graph. We first note that each vertex of every $I(a)$ is only adjacent to vertices in $K(b)$ with $b \notin X$ except possibly in one $K(b)$ with $b \in X$. According to the definition of G'', these vertices are all mutually adjacent, i.e., a clique. Thus we can repeatedly remove simplicial vertices (vertices whose neighbours form a clique) from the sets $I(a)$, until G'' is reduced to the union of the $K(b)$, which is clearly chordal. □

Note that the matrices M_H constructed in the proof have $E(A) = \{0\}$, $E(B) = \{*\}$, and $E(C) = \{0, *\}$, and hence are not normal.

5. Forbidden subgraphs in matrix partition

In this section, we discuss forbidden induced subgraph characterizations of M-partitionable chordal graphs. We phrase our results in terms of minimal obstructions. Given a matrix M, a graph G is a *minimal obstruction* for M-partitionability, if G has no M-partition, but each induced subgraph of G has an M-partition. Clearly, for a matrix M, the size of chordal minimal obstructions is bounded if and only if M-partitionability of chordal graphs can be characterized by a finite set of forbidden induced subgraphs.

It should be clear that a bound on the size of chordal minimal obstructions implies a polynomial time algorithm for the corresponding partition problem. (In fact, frequently the algorithms can be made linear time [10].) Thus we cannot expect such a bound for the matrices M discussed in the last section. In fact, as remarked in [10], there are polynomial time solvable (list) M-partition problems which still admit infinitely many minimal obstructions.

In [10], we have studied minimal obstructions that are *perfect* graphs, and have proved that the size of such obstructions is bounded whenever the matrix M is normal. Since chordal graphs are perfect, this is also the case for chordal minimal obstructions. It follows from these results that if M has no $*$'s at all, or if $E(C)$ is $\{0\}$ or $\{1\}$, the size of chordal minimal obstructions is bounded by $(k+1)(\ell+1)$. The same bound applies to the matrices with $E(A) = E(B) = E(C) = \{*\}$, by the result of [20,21]. Most of the other bounds for the size of minimal obstructions to M-partitionability given in [10] for perfect graphs, can be improved for chordal graphs (with a similar improvement in the complexity of the corresponding algorithms). In particular,

- the bound of $2(\ell+1)^{2k\ell+1}$ for normal matrices with $E(C) = \{*\}$ can be improved to $2^{(6\ell+3)k+1}k^k$ when A does not contain $*$, and to $2(k+1)^{(4k+2)\ell+2}$ when $E(A) = \{*\}$.
- For the first bound, we obtain a further improvement when $E(A) = \{1\}$ and $E(B)$ is $\{*\}$ or $\{0\}$. The bound becomes $2(2\ell+2)^k$ for $(1, *, *)$-block matrices, and $2(8\ell^2 + 25\ell + 5)^k$ for $(1, 0, *)$- block matrices.
- For the second bound, we obtain a further improvement when $E(B) = \{0\}$. The bound becomes $2(k+1)^{(k+2)\ell+1}$, for these $(*, 0, *)$-block matrices.

(We assume $k \leqslant \ell$ in these bounds.) We will prove these bounds in a separate note [16]. In this paper we focus on a case where an exponential bound for perfect graphs, can be improved to a polynomial bound for chordal graphs. This is the case of $(a, 0, *)$-block matrices with $k = 1$. (Note that the value of a is irrelevant.) For perfect graphs we only have the exponential upper bound of $2(\ell+1)^{2k\ell+1}$ mentioned above. In [10], we have shown that there are minimal obstructions to M-partitionability (for $(a, 0, *)$-block matrices M) that are trees (hence chordal) and have $(\ell/3)^2$ vertices. Here we give an $O(\ell^2)$ upper bound on the size of chordal minimal obstructions, for these matrices M.

Note that we are discussing partitions of input graphs G into one independent set A and ℓ independent cliques $B_1, B_2, \ldots, B_\ell$ (i.e., the cliques $B_1, B_2, \ldots, B_\ell$ have no edges joining them).

We first give a simple structural property of instances that have such an M-partition, regardless of the parameter ℓ.

Lemma 5.1. *Let M be an $(a, 0, *)$-block matrix with $k = 1$.*

If G has an M-partition, then each biconnected component of G is a split graph. If a chordal graph G has some biconnected component that is not a split graph, then this can be witnessed with an induced subgraph R of G with at most $6\ell + 2$ vertices.

Proof. Suppose a biconnected component involves at least two of the ℓ cliques. Then a shortest cycle in G going through vertices in both cliques has at least four vertices, contrary to chordality. Thus every biconnected component involves only one of the ℓ cliques and the independent set, and is therefore a split graph.

A biconnected component of the chordal graph G is not a split graph if and only if it contains two edges xy and zt such that no edge joins either of x, y to either of z, t. By biconnectivity, there exist two paths starting at x, y and ending at z, t. Say one path P goes from x to z, and a disjoint path Q goes from y to t. Assume P and Q are shortest paths, of respective lengths l_P, l_Q. It follows that P contains $\lceil l_P/3 \rceil$ independent edges, and thus it contains an obstruction for $\ell \leqslant \lceil l_P/3 \rceil - 1$ involving only $\ell + 1$ edges. Assume thus that $\ell \geqslant \lceil l_P/3 \rceil, \lceil l_Q/3 \rceil$. Then the obstruction consisting of the subgraph induced by the two paths P and Q has at most $l_P + l_Q + 2 \leqslant 6\ell + 2$ vertices. $\square$

We now bound the size of a minimal obstruction that has an M-partition for some $\ell' > \ell$.

Lemma 5.2. *Let M and M' be $(a, 0, *)$-block matrices with $k = 1$, and ℓ, respectively, ℓ' diagonal 1's. Let R be a minimal chordal obstruction to M-partition, and suppose R has an M'-partition. (Thus $\ell' > \ell$.)*

Then R has at most $8\ell^2 + 25\ell + 5$ vertices.

Proof. By Lemma 5.1, all blocks of R are split graphs. Consider an M'-partition of R as it looks in the block-cutpoint forest of R. If we shrink each of the ℓ' cliques of the M'-partition into a single vertex, we obtain a forest F. We claim F has $O(\ell)$ internal nodes and $O(\ell^2)$ leaves. We may hang each tree T in F with r vertices from a leaf as root. Suppose an M-partition for the connected component of R corresponding to T has s cliques. Then starting from the root of T and going down to the leafs, we may charge each vertex in the independent set encountered to the clique at a child, so there are at most s internal nodes in the independent set, and at most $2s$ clique nodes, for a total of at most $3s$ internal nodes. Therefore, the forest has no more than $3\ell + 1$ internal nodes, or else the obstruction would not be minimal.

The number of cliques that are not single vertices is at most ℓ, otherwise R has $\ell+1$ independent edges that constitute an obstruction. For every clique K with r vertices in the M'-partition, at least $r - 1$ of these vertices must occur in a clique in every M-partition, omitting at most one vertex x of K from the clique. Putting x in the independent set may only reduce the value ℓ' if this affects the solution for some neighbour y of x that is currently in the independent set with y an internal vertex of the forest. If y has at least two neighbours and two non-neighbours in K, then y must necessarily remain in the independent set. If y has exactly one neighbour in K, and K has at least three vertices, then the solution for y is only affected if the only neighbour of y in K is x. In that case we may charge x to the edge xy, and there are at most $(3\ell + 1) + \ell = 4\ell + 1$ such edges xy joining internal vertices of the forest to cliques. If y has exactly $r - 1$ neighbours in K, and K has at least three vertices, then the solution for y is only affected if the only non-neighbour of y in K is x. Again we charge x to the edge xy. The remaining vertices x in K do not affect the value of the solution and may be removed, provided that this does not cause a vertex y to end up with exactly one or $r - 1$ neighbours in K, so in that case xy is charged at most twice if K has at least five vertices. This leaves only the $2(4\ell+1)$ charged vertices in cliques of size at least five, and 4ℓ vertices in the at most ℓ cliques of size between two and four.

Combining this with the $3\ell + 1$ internal nodes of the forest gives $15\ell + 3$ vertices plus the number of vertices that are single vertex leaves.

Suppose a clique K has some single vertex leaf neighbours z. If z has at least two neighbours x in K, then it prevents x from being removed from the independent set if this affects some y, and may be charged to xy. If z has only one neighbour x in K, then removing x from the independent set so that some y may be included in the clique causes z to form a clique, so we may assume that at most $\ell + 1$ such z are only adjacent to such x, and they may be charged to the $4\ell + 1$ edges xy, for a total charge of $(\ell + 1)(4\ell + 1)$. If a single vertex leaf x is a chosen clique adjacent to y in the independent set, then we may assume there are at most $\ell + 1$ such x adjacent to the internal vertex y, and there are at most $2(\ell + 1)$ such internal vertices, giving an additional charge of $(\ell + 1)(4\ell + 1)$. The total bound is thus $2(\ell + 1)(4\ell + 1) + 15\ell + 3 = 8\ell^2 + 25\ell + 5$. □

We finally bound the size of a chordal minimal obstruction that does not have an M'-partition for any $\ell' > \ell$.

Lemma 5.3. *Let M, M' be as above. Let R be a chordal minimal obstruction to M-partition, and suppose R does not have an M'-partition for any ℓ'. Then R has at most $17\ell + 1$ vertices.*

Proof. The biconnected components of R may be assumed to be split graphs by Lemma 5.1. We bound the size of each of these biconnected components B. In general, B consists of a clique K and an independent set I such that K contains a clique K' that has neighbours in I. Let K'' be the vertices in K not in K'. If K'' contains a single vertex v, let I' be the set of vertices in $I \cup \{v\}$ that have the same neighbourhood as v, and let $I'' = I \setminus I'$.

There are then three cases:

1. If K'' has no vertices, then the vertices in K must be in the chosen clique, and the vertices in I must not be in the chosen clique;
2. if K'' has at least two vertices, then the vertices in K' must be in the chosen clique, the vertices in I must not be in the chosen clique, and at most one vertex in K'' may not be in the chosen clique;
3. if K'' has exactly one vertex, then the vertices in K' must be in the chosen clique, the vertices in I'' may not be in the chosen clique, and at most one vertex in I' may be in the chosen clique.

If B has a vertex x that must not be in the chosen clique that is attached to other vertices, then we may assume that B is attached just at x. This holds because the component attached at x must not have a solution with x in the independent set, else this component could be removed. We may thus remove the graph attached to the remaining vertices of x. Furthermore, at most one component attached at x may be present, since just one such component is needed to show that there is no solution with x independent. In all three cases above, there must exist either a clique on four vertices y, z, t, u such that x is adjacent to y, z but not to t, u, or a clique on three vertices y, z, t such that x is adjacent y, z but not to t, and some vertex u is adjacent to z, t but not to x, and possibly to y. In both cases, the clique chosen must contain y, z, t and therefore not x, and the five vertices x, y, z, t, u suffice to witness this. We may thus assume B has at most five vertices in this case.

If B has a vertex x that must be in the chosen clique that is attached to other vertices, then we may assume again that B is attached at x, and furthermore, at most one component attached at x may be present. In all three cases above, there must exist three vertices y, z, t with edges yz, zt, yx, xt, xz, and the clique chosen must contain x, y, in particular x, so four vertices x, y, z, t suffice to witness this. We may thus assume B has at most four vertices in this case.

If B is attached only at a set of vertices S contained in K'', then S forms a clique and at most one vertex in S may not be chosen for a clique. If the components attached at x in S forbid both that x be chosen and that x not be chosen, then we may assume B is attached only at x and a triangle with three vertices x, y, z suffices to witness the fact that x may not belong to another clique outside of B. We may thus assume B has at most three vertices in this case, that B is attached only at x and to at most two other components through x. The other possibility is that there are two vertices x, y in S such that a single component attached at x forbids that x be chosen, and a single component attached at y forbids that y be chosen, so we may again assume B has at most three vertices in a triangle x, y, z.

If B is attached only at vertices in I' of which at most one may be chosen for a clique, then again in one case two components attached at x in S forbid both that x be chosen and that x not be chosen, in which case a triangle with three vertices x, y, z gives B only three vertices, with B attached only at x and to at most two components. The other possibility is that there are two vertices x, y in S such that a single component attached at x requires that x be chosen, and a single component attached at y requires that y be chosen, in which case two adjacent vertices z, t both attached to x, y suffice to witness this, so we may assume B has at most four vertices.

If R has s biconnected components, then the fact that each one is connected to at most two other components shows that at least $\lfloor s/3 \rfloor$ such components are independent, and thus obtain that many independent edges. By minimality of the obstacle, it follows that $\lfloor s/3 \rfloor \leqslant \ell$, and so $s \leqslant 3\ell$. Since each biconnected component B has at most 5 vertices, this shows that there are at most 15ℓ vertices in biconnected components.

A vertex x in R may not have just a single neighbour, since in that case removing it would maintain the fact that there is no M'-partition for R with any ℓ'. The vertices of R not belonging to biconnected components are thus internal vertices of trees. After removing a single vertex, the resulting R' will have an M-partition, and so we may use the argument in Lemma 5.2 to show that the number of internal independent vertices is at most ℓ. The number of new leaves adjacent to the removed vertex is at most ℓ, since $\ell + 1$ such leaves produce $\ell + 1$ independent edges forming a chordal minimal obstruction. Combining this with the at most 15ℓ vertices in biconnected components and the single vertex removed gives a total of $15\ell + 1 + \ell + l = 17\ell + 1$ vertices and completes the proof of the lemma and the theorem. □

Theorem 5.4. *Consider all $(a, 0, *)$-block matrices $M = M_\ell$ with $k = 1$.*

Any chordal minimal obstruction to M_ℓ-partitionability has at most $8\ell^2 + 25\ell + 5$ vertices.

There is an algorithm that finds, for a chordal graph G with n vertices and m edges, the smallest integer ℓ such that G has an M_ℓ-partition, or shows that there is no such M_ℓ-partition for any ℓ. The algorithm is time $O(m + n)$.

For a given ℓ, the algorithm finds an M_ℓ-partition or an obstruction of size at most $8\ell^2+25\ell+5$, in time $O((m+n)\ell^2)$.

Proof. The approach is to first find the biconnected components, which can be done in time $O(m + n)$ with standard techniques. Then verify that every biconnected component is a split graph (in time $O(m + n)$, again by standard techniques, cf. e.g., [20]), otherwise there is no M-partition. Once every biconnected component is a split graph, the block-cutpoint tree gives rise to a tree with vertices corresponding to the cliques of the M-partition, as explained above. Hang the tree by a leaf as root, and then proceed from the other leafs towards the root, maintaining information about partial solutions found, namely for every connection vertex x joining two biconnected components, whether there are solutions for the subgraph G_x from x towards the leaves (1) with x in the independent set, (2) with x in a clique containing no other vertex in G_x, or (3) with x in a clique containing another vertex with x in the clique, and for each case the minimum number of cliques used for G_x in a solution. This information gets combined at each vertex of the tree, which is a clique K in time $O(m + n)$, since all but at most one of the vertices in K must be in a chosen clique and new information is passed on to the parent. This completes the algorithm for finding the smallest feasible ℓ or showing that no ℓ is feasible.

The algorithm for testing whether a graph is a split graph [20] produces the two independent edges if the graph is not a split graph. Finding two disjoint paths joining two independent edges in a biconnected graph can be done in time $O(m+n)$ by a flow algorithm that sends two units of flow, thus producing the obstruction of Lemma 5.1 if a biconnected component is not a split graph. Given an ℓ, the algorithm can determine for each x as above with corresponding G_x, each $\ell' \leqslant \ell$, and each of the above three situations (1), (2), (3), or the situation (4), if x is removed from the graph, the following information: the number of vertices in G_x, and the maximum number of vertices that can be removed from G_x to obtain G'_x not be able to achieve the chosen situation out of the four situations using at most ℓ' cliques in G'_x. This information gets combined at a clique K, where combining the information for two connected components of the subgraph of G_x joined at x, or combining the information for a vertex x in a clique K with the information of the clique of vertices preceding x in an ordering of the vertices of K, takes time $O(\ell^2)$, since each pair of values $\ell' \leqslant \ell$ is considered. The information for different connected components of the graph can be combined similarly in $O(\ell^2)$ time per connected component. Thus finding the maximum number of vertices that can be removed and obtain an obstruction takes is obtained as a combined answer for the whole graph for the case $\ell' = \ell$, in time $O((m + n)\ell^2)$. □

References

[1] H.L. Bodlaender, T. Kloks, Efficient and constructive algorithms for the pathwidth and treewidth of graphs, J. Algorithms 21 (1996) 358–402.

[2] A. Brandstädt, Partitions of graphs into one or two stable sets and cliques, Discrete Math. 152 (1996) 47–54.

[3] A. Brandstädt, P.L. Hammer, V.B. Le, V.V. Lozin, Bisplit graphs, DIMACS TR 2002-44.

[4] K. Cameron, E.M. Eschen, C.T. Hoang, R. Sritharan, The list partition problem for graphs, SODA 2004, 2004.
[5] M. Chudnovsky, N. Robertson, P. Seymour, R. Thomas, The strong perfect graph theorem, eprint arXiv:math/0212070.
[6] V. Chvátal, Star-cutsets and perfect graphs, J. Combin. Theory B 39 (1985) 189–199.
[7] C.M.H. de Figueiredo, S. Klein, Y. Kohayakawa, B.A. Reed, Finding skew partitions efficiently, J. Algorithms 37 (2000) 505–521.
[8] J. Diaz, M. Serna, D.M. Thilikos, Recent results on parametrized H-colourings, in: J. Nešetřil, P. Winkler (Eds.), DIMACS series in Discrete Mathematics and Theoretical Computer Science, Vol. 63, pp. 65–85, American Math. Society, 2004.
[9] T. Feder, P. Hell, Full constraint satisfaction problems, manuscript.
[10] T. Feder, P. Hell, Matrix partitions of perfect graphs, Special Issue of Discrete Math., in Memory of Claude Berge, 2005.
[11] T. Feder, P. Hell, J. Huang, List homomorphisms and circular arc graphs, Combinatorica 19 (1999) 487–505.
[12] T. Feder, P. Hell, J. Huang, Bi-arc graphs and the complexity of list homomorphisms, J. Graph Theory 42 (2003) 61–80.
[13] T. Feder, P. Hell, S. Klein, R. Motwani, Complexity of list partitions, in: Proc. 31st Annu. ACM Symp. on Theory of Computing, 1999, pp. 464–472.
[14] T. Feder, P. Hell, S. Klein, R. Motwani, List partitions, SIAM J. Discrete Math. 16 (2003) 449–478.
[15] T. Feder, P. Hell, S. Klein, L. Tito Nogueira, F. Protti, List matrix partitions of chordal graphs, LATIN 2004, Lecture Notes in Computer Science, Vol. 2976, 2004, pp. 100–108.
[16] T. Feder, P. Hell, S. Klein, L. Tito Nogueira, F. Protti, Chordal obstructions to M-partitions, manuscript. www.cs.sfu.ca/∼pavol/chordalOBST.ps.
[17] T. Feder, P. Hell, D. Král', J. Sgall, Two algorithms for general list matrix partitions, SODA 2005, 2005.
[18] T. Feder, P. Hell, K.M. Tucker-Nally, Digraph matrix partitions and trigraph homomorphisms, manuscript.
[19] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980.
[20] P. Hell, S. Klein, L.T. Nogueira, F. Protti, Partitioning chordal graphs into independent sets and cliques, Discrete Appl. Math. 141 (2004) 185–194.
[21] P. Hell, S. Klein, L.T. Nogueira, F. Protti, Packing r-cliques in weighted chordal graphs, Ann. Oper. Res. 138 (2005) 179–187.
[22] P. Hell, J. Nešetřil, Graphs and Homomorphisms, Oxford University Press, Oxford, 2004.
[23] R.M. McConnel, J.P. Spinrad, Linear time modular decomposition and efficient transitive orientation of comparability graphs, in: Proc. ACM-SIAM Symp. on Discrete Algorithms, 1994, pp. 536–545.
[24] A. Proskurowski, S. Arnborg, Linear time algorithms for NP-hard problems restricted to partial k-trees, Discrete Appl. Math. (23) (1989) 11–24.
[25] R.E. Tarjan, Decomposition by clique separators, Discrete Math. 55 (1985) 221–232.
[26] A. Tucker, Coloring graphs with stable sets, J. Combin. Theory Ser. B 34 (1983) 258–267.
[27] S. Whitesides, An algorithm for finding clique cutsets, Inform. Process. Lett. 12 (1981) 31–32.