6

Forbidding Holes and Anti-Holes

Ryan Hayward[a] **and Bruce A. Reed**[b]

[a]*Department of Computing Science, University of Alberta, Canada.*
[b]*CNRS, Equipe Combinatoire, Université Pierre et Marie Curie, Paris 6, France.*

6.1 Introduction

In 1984, the second author made a bet with Manfred Padberg. He wagered that it is NP-complete to determine if a graph contains an odd chordless cycle of length at least five, but that determining if a graph or its complement contains such a cycle takes only polynomial time. If these questions were not settled by the turn of the century, the bet was to be called off. The second author is very glad that the new millennium has begun, as he now believes that both problems can be solved in polynomial time and that this fact will be proven soon. We expect that after finishing the chapter, the reader will share these beliefs.

This chapter discusses the structure of various classes of graphs obtained by forbidding induced cycles of specified lengths and their complements. Our main interest is decomposition theorems which yield efficient recognition algorithms.

By a *hole*, we mean an induced cycle of length at least four. A graph is an *anti-hole* if its complement is a hole. The *length* of a hole or anti-hole is the number of vertices on it. A hole or anti-hole is *long* if it has length at least five.

We focus mainly on three classes of graphs, those with no long holes or anti-holes, those with no even holes, and bipartite graphs without holes of length 2 mod 4.

We recall that the Strong Perfect Graph Conjecture states that a graph is perfect precisely if it has no odd holes and no odd anti-holes (i.e. if it is *Berge*). The study of graphs without long holes and anti-holes was motivated by this conjecture. As we shall see, the interest in the last two classes of graphs on which we focus stemmed from the study of β-perfect graphs and balanced matrices respectively. Nevertheless,

Perfect Graphs. Editor J.L. Ramírez Alfonsín and B.A. Reed

the approach used to characterize these classes may well lead to a polynomial time recognition algorithm for graphs without odd holes, and thus also for Berge graphs.

6.2 Graphs with no Holes

Graphs with no holes are called *chordal*, since in such graphs every cycle with four or more vertices is not an induced cycle and so has at least one chord. Chordal graphs are also known as *triangulated* graphs, although the name chordal is perhaps more descriptive and less likely to be confused with the unrelated geometric notion of triangulation.

The hole with five vertices is self-complementary, and every anti-hole with six or more vertices contains a hole with four vertices, so if a graph is chordal then it has no long anti-hole. Thus chordal graphs are Berge. Indeed, the knowledge that both chordal graphs [3] and complements of chordal graphs [44] have chromatic number equal to maximum clique size was instrumental in motivating Claude Berge to introduce the notion of perfection and to propose the two Perfect Graph Conjectures (see Chapter 1 and also [5]).

Chordal graphs are a well studied class of graphs; see for example Chapter 4 in [40] and Chapters 1 and 3 in [8]. The foundation on which the theory of chordal graphs rests is a theorem of Dirac, stated below.

For vertices x, y of a graph G, a vertex subset S is a *vertex separator* if x and y are in different components of $G - S$. We say a vertex separator S for x and y is *minimal* if no proper subset of S is a vertex separator for x and y.

Theorem 6.1 [29] *G is chordal if and only if all of the minimal vertex separators in G are cliques.*

Corollary 6.2 *G is chordal if and only if every induced subgraph of G is either a clique or has a clique cutset.*

These two results were proven in Chapter 2. Readers who do not remember the simple proof can either reproduce it themselves or flip back to that chapter.

As we now show, these results allow us to develop polynomial time algorithms for solving many problems on chordal graphs.

Gavril showed that decomposing a graph along clique cutsets yields a tree-like decomposition structure having only a polynomial number of leaves, each of which corresponds to a non-decomposable subgraph [39]. Whitesides showed that such decompositions can be obtained in polynomial time and lead to polynomial time algorithms for many optimization problems in many superclasses of chordal graphs [78]. By Corollary 6.2, if G is chordal then the non-decomposable graphs at the leaves are cliques. So to test if G is chordal we need only test if each leaf corresponds to a clique, which is easy to do efficiently. Thus clique cutsets do indeed lead to efficient algorithms for chordal graphs.

The schema sketched above generalizes to many other classes of graphs and is the template for many of the recognition algorithms we present. How it has evolved over time is the main subject of this chapter. For this reason, we give a formal definition and an example of a clique cutset tree below (at the very end of this section). Actually, chordal graphs have very strong structural properties which permit the development of recognition and optimization algorithms which are much simpler and faster than those obtained using this widely applicable approach. We now digress briefly from our main topic to discuss more fully the structure of chordal graphs.

A *simplicial vertex* of a graph is a vertex whose neighbours induce a clique. A *simplicial vertex ordering* is an ordering of the vertices of a graph such that every vertex is simplicial in the subgraph induced by the vertex and all previous vertices. These orderings are also known as (*perfect*) *vertex elimination orderings*, and sometimes defined in the reverse order (that is, so that each vertex is simplicial with respect to all following vertices).

From Theorem 6.1, it is easy to obtain:

Theorem 6.3 *A graph is chordal if and only if*

[35] *it has a simplicial ordering,*
[10] [38] [78] *it is the intersection graph of subtrees of a tree.*

The first of these properties leads to simple and efficient algorithms for many problems on chordal graphs. Specifically, Rose, Tarjan, and Leuker [69] used it to establish optimal[1] algorithms for recognition and the four standard perfect graph optimization problems, namely

- *maximum clique*: find a maximum size clique,
- *maximum independent set*: find a maximum size independent set,
- *minimum clique covering*: find a vertex partition into the minimum number of cliques, and
- *minimum colouring*: find a vertex partition into the minimum number of independent sets (known as *colour classes* in this context).

These algorithms are based on the graph traversal method known as lexicographic breadth-first search. A *Lexicorgraphic Breadth First Ordering* (LBFO) of a graph is obtained by continually refining an ordered partition of those vertices not yet ordered. The partition is initialized to be the single set consisting of all vertices. At each step, any vertex v is removed from the first partition set and placed next in LexBFS order; each partition set S is then replaced (so that its relative position among the other partition sets is maintained) with $S^+ = S \cap N(v)$ and $S^- = S - N(v)$, with S^+ immediately preceding S^-.

Also, for any simplicial ordering, assigning to each vertex in order the smallest positive integer not already assigned to any of its preceding neighbours yields an optimal colouring (this is partly what inspired Chvátal to introduce the superclass of chordal graphs known as perfectly orderable graphs, see Chapter 7 and [11]); thus colouring chordal graphs takes linear time.

The preceding recognition algorithm is easily modified to return a hole whenever the input graph is not chordal, so finding a hole in an arbitrary graph takes linear time.

---

[1] For an input graph with n vertices and m edges, a *linear time* algorithm is one which takes $\Theta(n+m)$ time. Assuming that graph input itself requires linear time, any algorithm which takes linear time is clearly (within a constant factor of) optimal.

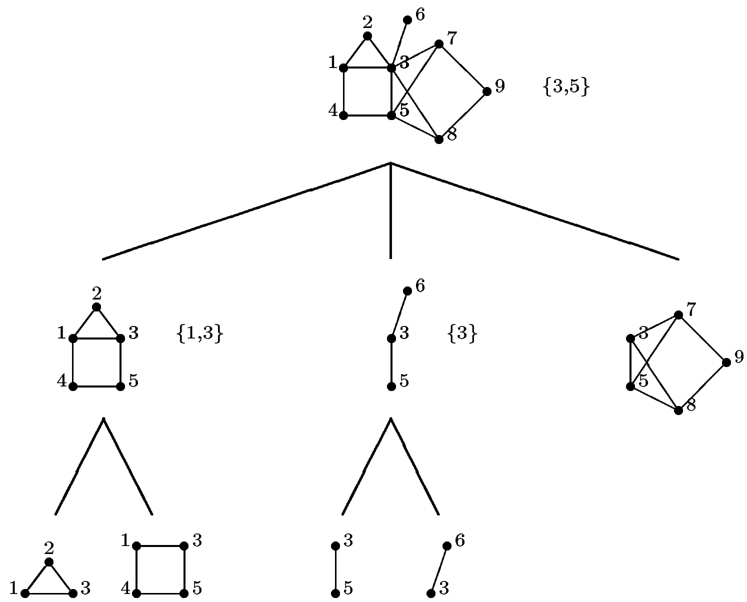

Figure 6.1 A clique cutset tree.

We close this section with a formal definition of a clique cutset tree $T(G)$ of a graph G:

- each node t of $T(G)$ is labelled with a subgraph H_t of G
- the root is labelled by G
- every leaf is labelled by a subgraph of G with no clique cutset,
- for every internal node t there is some clique cutset C of H_t such that if the componenents of $H_t - C$ are $U_1, ...U_k$ then the children of H_t are labelled with the subgraphs G induced by $C \cup U_1, ..., C \cup U_k$

Notice that clique cutset trees are not unique, since graphs can have multiple and/or overlapping clique cutsets. A clique cutset tree is shown in Figure 6.1. Note further that similar trees can be defined for any class of cutsets.

6.3 Graphs with no Discs

A *disc* is a long hole or a long anti-hole. Graphs with no disc, called *weakly chordal* or *weakly triangulated*, were introduced by Hayward as a natural generalization of chordal graphs within the class of Berge graphs [45]. Since the only difference in terms of forbidden holes and anti-holes between chordal graphs (no hole and no long anti-hole) and weakly chordal graphs (no long hole and no long anti-hole) is the hole with four vertices, it is not surprising that there are characterizations of weakly chordal graphs of similar nature to the above characterizations of chordal graphs. In order to present these characterizations we need to define some structures which generalize the notions of clique cutset and simplicial vertex.

A *star cutset* is a cutset which contains a vertex adjacent to all other vertices in the cutset; such a vertex is a *center* of the cutset. Star cutsets, introduced by Chvátal [12], generalize several kinds of cutset, including clique cutsets (see Chapter 8). As we now see, they come in two flavours.

A star cutset S centered at v is *full* if $S = \{v\} \cup N(v)$.

A vertex *dominates* another vertex if its closed neighbourhood contains the other vertex's open neighbourhood, i.e. x *dominates* y if $\{x\} \cup N(x)$ contains $N(y)$. Notice that if a vertex is simplicial then it is dominated by each of its neighbours. Also notice that if x dominates y then $x + N(y)$ is a star cutset unless $V = x + y + N(y)$.

Observation 6.4 *A graph with a star cutset has either a full star cutset or a dominating vertex.*

Proof. If S is a star cutset with center x and $x + N(x)$ is not a full star cutset then some component U of $G - S$ is completely contained in $N(x)$. But then x dominates every vertex of U. □

An *even pair* in a graph is a pair of non-adjacent vertices such that every induced path joining them has an even number of edges (see Chapter 4). A *two-pair* is an even pair where every such path has two edges. Two vertices form a two-pair if and only if their common neighbourhood is a vertex separator for them. Thus, a two-pair can be thought of as a special kind of cutset. Actually, the common neighbourhood will form a star-cutset with one of the two vertices unless the graph consists only of the two vertices and their common neighbours.

P_k denotes an induced path with k vertices. An edge is *simplicial* if it is not the middle edge of any P_4. Simplicial edges are analogous to simplicial vertices in that a vertex is simplicial if it is not the middle vertex of any P_3. The vertices of a two-pair are not the end vertices of any P_4 (or $P_{k \geq 5}$), so the vertices of a two-pair of a graph form a simplicial edge in the complement of the graph. The converse is not true, since the vertices of a simplicial edge in the complement of a graph can be the end vertices of a $P_{k \geq 5}$ in the graph. Analogous to a simplicial vertex ordering, a *simplicial edge ordering* is an ordering of the edges of a graph such that each edge is simplicial in the subgraph formed by the edge and all preceding edges.

Theorem 6.5 *A graph G is weakly chordal if and only if*

- [52] *every induced subgraph is a clique or has a two-pair,*
- [46] *for every induced subgraph H, H has a star cutset or is a clique or the complement of a perfect matching,*
- [45] *for every induced subgraph with three or more vertices, either the subgraph or its complement has a star cutset,*
- [49] *it has a simplicial edge ordering,*
- [45] *for every induced connected subgraph H with some minimal cutset C such that the subgraph of $\overline{H}$ induced by C is connected, every component of $H - C$ has some vertex which is adjacent to every vertex of C.*

Remark 6.6 *The first four characterizations are all consequences of the last characterization, which was obtained first. For a strengthening of the last characterization (describing conditions whenever the subgraph of $\overline{H}$ induced by C is disconnected) see [50].*

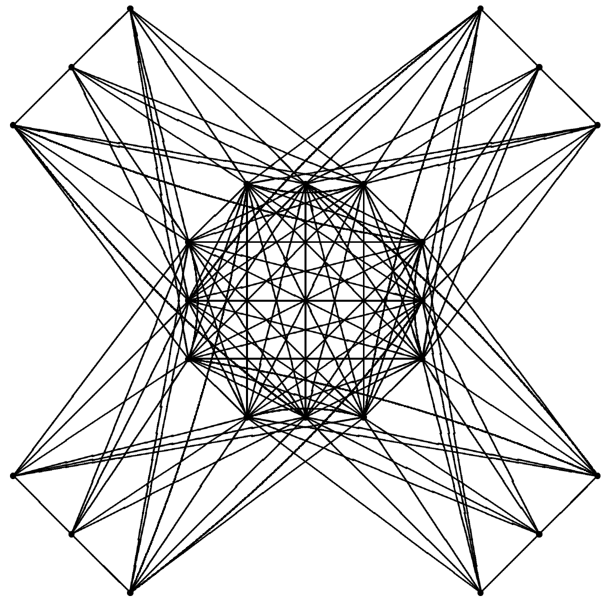

Figure 6.2 The smallest known domination-free weakly chordal graph.

One direction of the third characterization can be strengthened slightly: if a graph has some vertex in no disc, then either the graph or its complement has a star cutset [48].

A graph is a *domination graph* if every induced subgraph with at least two vertices has a dominating vertex. Domination graphs are weakly chordal (since long holes have no dominating vertex), and the previously described relationship between star cutsets and dominating vertices suggests the question of whether the converse holds. The answer is no, although the smallest known example, shown in Figure 6.2, has twenty-four vertices [45]. The complexity of recognizing domination graphs is open: Dalhaus et. al. have an algorithm which recognizes a large subclass of domination graphs [28], but Rusu and Spinrad show that this subclass is proper [71].

Since a minimal imperfect graph has a connected complement and has no star cutset [12], the second characterization above implies that weakly chordal graphs are perfect. However, it does not lead immmediately to efficient algorithms for weakly chordal graphs. One problem is that the tree-like star cutset decomposition analogous to the clique cutset decomposition, mentioned above, can yield an exponential number of non-decomposable graphs as leaves [12]. Thus some stronger structural property is needed to provide efficient algorithms for weakly chordal graphs.[2]

Fortunately, weakly chordal graphs do indeed have very strong structural properties which permit us to develop such algorithms. Once again, we digress briefly from our main focus to discuss these algorithms.

An even pair has the property that its *contraction* (namely, replacement of the pair of

[2] Actually, strictly speaking this is only true with respect to optimization, as the techniques we discuss in later sections could be used to develop recognition algorithms based on a simple variant of these star cutset decomposition trees.

vertices with a single vertex whose neighbourhood is the union of the neighbourhoods of the two vertices being replaced) does not change either the graph's largest clique size or its chromatic number (an optimal colouring of the original graph is obtained from an optimal colouring of the resulting graph by assigning the two-pair the same colour as the contracted vertex). Furthermore, contracting a two pair cannot create a disc, so contracting a two pair in a weakly chordal graph yields a new weakly chordal graph. For more on this topic, see chapter 4.

Hayward, Hoàng, and Maffray presented polynomial time algorithms for both the weighted and unweighted versions of our four optimization problems [52], which are based on repeatedly contracting two pairs. Arikati and Pandu Rangan obtained a speedup of these algorithms by reducing the time to find a two-pair [1]; Spinrad and Sritharan obtained further improvements [74]. The currently fastest unweighted optimization algorithms are due to Hayward, Spinrad, and Sritharan and take $O(nm)$ time; the algorithms exploit the so-called handle structures to reduce the total time spent finding two-pairs [53].

While two-pair contraction leads to efficient optimization algorithms for weakly chordal graphs, it does not lead to efficient recognition algorithms for these graphs, since such contractions can destroy discs. For example, consider a graph with some vertex x in a disc and some vertex y adjacent to every vertex in the graph except x; then $\{x, y\}$ is a two-pair whose contraction essentially deletes x, consequently destroying the disc which contained x. There is an operation on two-pairs which neither creates nor destroys discs, namely adding an edge between a two-pair [74]; this is the basis of the currently fastest recognition algorithm for weakly chordal graphs, which takes $O(n + m^2)$ time [53].

No analogue of the intersection tree characterization of chordal graphs is known for weakly chordal graphs; however, there are some graph classes between these classes which have natural intersection graph characterizations, for example, the neighbourhood subtree tolerance graphs, or NeST graphs [51], which includes among other classes interval graphs and tolerance graphs.

As well as chordal graphs and NeST graphs, other well known classes of graphs properly contained in the class of weakly chordal graphs include brittle graphs [54], chordal bipartite graphs [40], cographs [27], domination graphs [28], permutation graphs [40], split graphs [40], threshold graphs [40], tolerance graphs [41], classes of visibility graphs [34] [16], and also the classes consisting of the complements of graphs of these classes.

A *comparability graph*, also known as a *transitively orientable graph*, is a graph whose edges can be oriented such that if there is an arc (namely, an oriented edge) from x to y and an arc from y to z, then there is an arc from x to z. As with weakly chordal graphs, there are no known linear time recognition or optimization algorithms for comparability graphs. The bottleneck in these algorithms is a matrix multiplication subroutine used in finding a transitive orientation; for example, recognition takes $\Theta(m+n+X)$ time [64], where X is the time required for n by n matrix multiplication, currently $\Theta(n^{2.376\cdots})$ [26]. In fact, the problems of recognizing comparability graphs and recognizing triangle-free graphs are known to be linear-time equivalent: a linear time algorithm for either one would imply a linear time algorithm for the other [33]. On the other hand, there are linear time recognition and optimization algorithms for graphs which are both weakly chordal and comparability [31].

We could try and obtain a proof of the Strong Perfect Graph Conjecture by characterizing larger and larger superclasses of the weakly triangulated graphs defined in terms of constraints on the disc size. The most natural such superclass is the class of Berge graphs with no disc with seven or more vertices, namely graphs all of whose discs are of length six. While the perfection of this class is not known, it is known that the subclass of so-called murky graphs, namely graphs with no C_5, P_6 and $\overline{P}_6$, is perfect [47].

6.4 Graphs with no Long Holes

We now turn our attention to the superclass of weakly chordal graphs consisting of all graphs with no long hole. We remark that this is not a class of perfect graphs, as it contains all the odd anti-holes. More strongly, Hoàng and McDiarmid [56] and independently Randerath and Schiermeyer[67] have shown that for every c there exists a graph G without long holes such that $\chi(G) > c\omega(G)$. Gyárfas [43] conjectured that there is some function f such that every graph G with no long holes has chromatic number at most $f(\omega(G))$.

Although this conjecture remains open, we do know a fair bit about the structure of graphs without long holes. Consider for example the following characterization, conjectured by Sritharan and proved by Chvátal and Rusu [14]:

Theorem 6.7 *A graph has no long hole if and only if every induced subgraph with some edge has some edge which is simplicial in the subgraph.*

Remark 6.8 $\overline{C_6}$ *has three simplicial edges but deleting any one of them yields a graph containing a* C_5*. This is essentially the reason that weakly chordal graphs are characterized by the existence of simplicial edge orders while graphs without long holes are characterized by the existence of a simplicial edge.*

This theorem yields the following (new) characterization of graphs with no long holes:

A cutset S is a *double star cutset* with center (u, v) if u and v are adjacent nodes of S and $S \subseteq N(u) \cup N(v)$.

Theorem 6.9 *A graph has no long hole if and only if every induced subgraph either*

(i) has a star cutset,
(ii) has a double star cutset, or
(iii) is a clique or C_4 *or two non-adjacent vertices.*

Proof. Long holes satisfy none of (i)-(iii), so we need only prove that a graph G without long holes satisfies one of these properties. By Theorem 6.7, we may assume that G has a simplicial edge xy. If $V = \{x, y\} \cup N(x) \cup N(y)$ then we can assume $V - x - y$ is a clique as otherwise there is a double star cutset centered at xy containing all of V except two non-adjacent vertices. Now, it is easy to see that if $V - x - y$ is a clique then one of (i)-(iii) must hold.

So, we can assume that there is a vertex z in $V - x - y - N(x) - N(y)$. Now, if y dominates x then $y + N(x)$ is a star cutset separating x from z. So we can assume that there is a vertex v in $N(x) - N(y)$. Since xy is simplicial, v sees all of $N(y) - N(x)$. Thus, $x + v + N(y)$ is a double star cutset centered at xv separating y from z. □

An induced path $(v_1, \ldots, v_t)$ in a graph is *the middle of a* P_{t+2} if there are vertices v_0 and v_{t+1} such that $(v_0, v_1, \ldots, v_t, v_{t+1})$ is an induced path. Sritharan generalized Theorem 6.7 as follows [14]:

Theorem 6.10 *A graph has no hole with k or more vertices if and only if each induced subgraph with at least one P_{k-3} has some P_{k-3} which is not the middle of any P_{k-1} in the subgraph.*

The following characterization due to Eschen and Sritharan categorizes graphs with no long hole according to the size of admissible anti-holes by refining the definition of simplicial edge [32]. A simplicial edge is *k-simplicial* if in the complement of the graph the vertices are not the two end vertices of any induced path with at least four and at most k vertices. C_k denotes an induced cycle with k vertices. $2K_2$ is the graph with four vertices and two non-adjacent edges, namely the complement of C_4.

Theorem 6.11 *A graph has no long hole and no long anti-hole with $k+1$ or fewer vertices if and only if for every induced subgraph H with at least one edge,*
if H has no $2K_2$, then at least $|H|/2$ of the edges of H are k-simplicial, and
if H has some $2K_2$, then some $2K_2$ has both edges k-simplicial in H.

In a graph with n vertices, two vertices form a two-pair if and only if they form an n-simplicial edge in the the complement of the graph. Setting $k = n$, the above theorem implies that every weakly chordal but not chordal graph has two two-pairs whose four vertices induce a cycle.

Some subclasses of Berge graphs with no long hole but some long anti-holes are known to be perfect. For example, Rusu showed that for any fixed even $p \geq 6$, the class of graphs in which every hole is even and has at least p and at most $2p - 6$ vertices is perfect [70]; such graphs have no long anti-hole (since they have no C_4 or C_5), so their complements have no long hole. Also, Maffray and Preissmann showed that Berge graphs with no P_5 (and so no long hole) and no K_5 are perfect [60].

While finding a hole in a graph takes only linear time, the fastest known algorithm to find a long hole takes $\Theta(mn^{2.376\ldots})$ time [73], where again the exponent $2.376\ldots$ is due to matrix multiplication. This is perhaps not surprising, since there is no known algorithm for the simple problem of finding a triangle in a graph which is faster than matrix multiplication.

6.5 Balanced Matrices

In the next section we discuss *balanced graphs*, which are those bipartite graphs which have no hole of length 2 *mod* 4. As with perfect graphs, much of the interest in this class of graphs is due to their importance in the theory of integer programming. In this brief section, we explain the link between balanced graphs and integer programs.

This material is motivational in nature. The only fact from this section that we will need later is that there is a class of balanced graphs, the totally unimodular graphs, for which membership testing is in P.

We recall that a $0-1$ matrix A is perfect precisely if the polyhedron

$$\{x \geq 0, Ax \leq b\}$$

is integral for all 0-1 vectors b. A $0-1$ matrix A is balanced if the polyhedron

$$\{x \geq 0, Ax \geq b\}$$

is integral for all 0-1 vectors b.

In [4], Berge proved that a $0-1$ matrix A is balanced precisely if it contains no square submatrix of odd order with precisely two ones in each row and column (see [6] and [36] for extensions).

A 0-1 matrix A is *totally unimodular* if the polyhedron

$$\{x \geq 0, Ax \leq b\}$$

is integral for all integral b. Thus a totally unimodular 0-1 matrix is perfect by definition. In fact, as shown by Hoffman and Kruskal in [57], a 0-1 matrix is totally unimodular if and only if

$$\{x \geq 0, Ax \geq b\}$$

is integral for all integral b. Thus, totally unimodular 0-1 matrices are also balanced.

In [72], Seymour obtained a beautiful decomposition theorem for totally unimodular matrices which is the basis of a recognition algorithm for this class. In the next section, we describe a polynomial time algorithm due to Conforti, Cornuejols, and Rao for recognizing balanced matrices. To do so, we first transform our matrices into graphs.

For any $0-1$ matrix A, we define a bipartite graph $G(A)$ such that (i) the vertices on one side of the bipartition correspond to the columns of A, (ii) the remaining vertices correspond to the rows of A, and (iii) two vertices on opposite sides of the bipartition are adjacent precisely if the corresponding row and column intersect in a 1. We note that we have actually defined a bijection between bipartite graphs and $0-1$ matrices (this is a bijection to labelled graphs, permutations of the rows and columns correspond to relabellings of the nodes).

Remark 6.12 *This is a more natural and more general way of building graphs from matrices than using the matrix to represent clique-node incidence, for example each matrix gives rise to a unique graph.*

We say that a graph is totally unimodular if the corresponding matrix is. Since it is easy to construct the matrix corresponding to a given graph quickly, we can use Seymour's characterization of totally unimodular matrices to test if a graph is totally unimodular in polynomial time.

For the same reason, the characterization of Berge implies that to efficiently test if A is a balanced matrix we need only be able to determine in polynomial time, if $G(A)$ is a balanced graph.

6.6 Bipartite Graphs with no 4k+2 Hole

Recall that a cutset S is a *double star cutset* with center (u, v) if u and v are adjacent nodes of S and $S \subseteq N(u) \cup N(v)$. Conforti, Cornuejols, and Rao (Theorem 1.2 of [22]) proved:

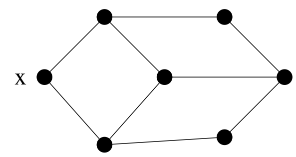

Figure 6.3

Theorem 6.13 *Every balanced graph which is not totally unimodular has a double star cutset.*

Remark 6.14 *Totally unimodular graphs do not seem natural from a graph theoretic viewpoint. We could actually decompose totally unimodular graphs further using another decomposition, the 2-join, obtaining a decomposition theorem which would be much more appealing to graph theorists. However, Theorem 6.13 yields a much simpler algorithm than this more graph-theoretic variant.*

We now explain how they applied this theorem to obtain a recognition algorithm for balanced graphs.

Just as is the case with star cutsets, double star cutsets come in two flavours:

Definition 6.15 *A double star cutset S centered at (u, v) is full if $S = N(u) \cup N(v)$.*

Observation 6.16 *A connected bipartite graph with a double star cutset either contains a dominated node or a full double star cutset.*

The recognition algorithm of Cornuejols, Conforti, and Rao proceeds as follows:

1. If G contains a dominated node x then to determine whether G contains an unbalanced hole, it tests whether $G - x$ contains an unbalanced hole.
2. If G contains a full double star cutset S then to determine whether G contains an unbalanced hole, it tests for each component U of $G - S$ whether $S + U$ contains an unbalanced hole.
3. If G contains no double star cutset then to determine whether G contains an unbalanced hole, it tests if G is totally unimodular.

We note that this algorithm essentially builds a double star cutset decomposition tree. However, rather than decomposing into many pieces when there is a dominated node it simply deletes the dominated node. As we shall see, this ensures that the resultant decomposition tree has only a polynomial number of nodes.

An astute reader will have noticed that simply applying the algorithm given above will not allow us to test if a graph is balanced. We would fail given a graph G which contains a dominated node x for which G contains an unbalanced hole but $G - x$ does not (see Figure 6.3 for an example of such a graph). Conforti, Cornuejols, and Rao use the subroutines above to test for the existence of unbalanced holes of a special type, which they call *clean*. They then show that for any graph G, there is a family $\mathcal{F}$ of subgraphs of G such that G contains an unbalanced hole precisely if one of the graphs in $\mathcal{F}$ contains a clean unbalanced hole. Combining these results yields the desired algorithm to test if a graph is balanced. Forthwith the details.

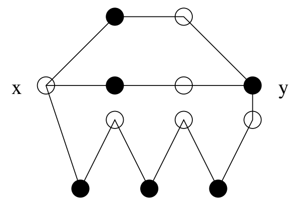

Figure 6.4

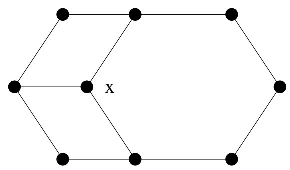

Figure 6.5

6.6.1 Some preliminary remarks

A graph is called *balanceable* if we can assign a label (or weight) of $+1$ or -1 to each of its edges so that he total weight on every cycle is a multiple of four (and such a labelling is called balanced). Clearly every balanced graph is balanceable.

A bipartite graph is a 3-path configuration (3PC) if it consists of three chordless paths between two vertices on opposite sides of the bipartition , such that every pair of paths forms a hole (see Figure 6.4). Note that in any labelling of a 3PC, each of the 3 paths has weight 1 or 3 mod 4 so some two of them will form a hole of total weight 2 mod 4. Thus no balanceable graph contains a 3PC, and hence neither does any balanced graph.

A bipartite graph is a *k-wheel*, for $k \geq 2$ if it consists of a hole H and a vertex x such that $|N(x) \cap H| = k$ (see Figure 6.5). A k-wheel is an *odd wheel* if k is odd. We now verify that no balanceable graph, and hence no balanced graph, contains an odd wheel (H, x). We do so by considering the hole H and all the holes formed by taking two consecutive neighbours of x on H and the subpath of $H - N(x)$ between them. Each edge of the odd wheel is in two of these holes and hence contributes 2 *mod* 4 to the sum of their weights. So, since the odd wheel has an odd number of edges, in any weighting the total weight of these holes must be 2 *mod* 4. Thus, clearly, they cannot all be of length 0 *mod* 4.

Truemper[76] proved that these are all the minimal unbalanceable graphs:

Theorem 6.17 *A bipartite graph is balanceable if and only if it it contains neither a 3PC nor an odd wheel.*

Remark 6.18 *See [24] for an easy proof of this theorem and a discussion of its consequences.*

By a *short* 3-wheel, we mean a 3-wheel (H, x) such that two of the components of $H - N(x)$ are vertices (the 3-wheel of Figure 6.5 is short). For a given choice of (i) x, (ii) u, v, w in $N(x)$, and (iii) an induced path of length five: $uyvzw$, we can test in polynomial time whether there is a short 3-wheel (H, x) with $uyvzw \subset H$ by checking

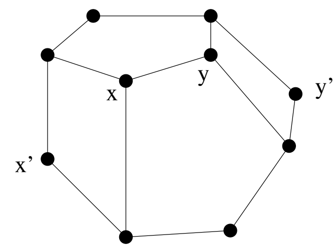

Figure 6.6

if u and w are in the same component of $G - N(v) - (N(y) - u) - (N(z) - w) - (N(x) - u - w)$. So, by running through all the possibilities for x, u, y, w, z, v we can determine if G has any short 3-wheel in polynomial time.

For two adjacent nodes u and v in a bipartite graph, a *simple* 3PC through (u, v) is a $3PC$ consisting of three paths between a neighbour u' of u and a neighbour v' of v such that (i) one path is $u'uvv'$, and (ii) the interiors of the other two paths are in different components of $G - N(u) - N(v)$. As the reader can verify, it is straightforward to determine if there is a simple 3PC through u and v in polynomial time.

We note that there are also polynomial time algorithms to (i) determine if G contains a dominated node and if so find one, and (ii) determine if G contains a full double star cutset and if so find one. Again, a keen reader may verify these facts herself.

6.6.2 Clean holes

With these preliminary remarks out of the way, we can now define the special type of unbalanced hole we focus on and indicate why they are easier to handle then arbitrary unbalanced holes.

Definition 6.19 *A tent over a hole H consists of two adjacent nodes x and y of $G - H$ such that (i) $N(x) \cap H = N(x') \cap (H)$ for some node x' of H, (ii) $N(y) \cap H = N(y') \cap (H)$ for some node y' of H, (iii) x' and y' are not adjacent.*

Definition 6.20 *A hole H is clean if*

(a) no vertex sees more than two vertices of H,
(b) for any vertex which sees two vertices of H, there is a vertex x' of H such that $N(x) \cap H = N(x') \cap H$, and
(c) there is no tent over H.

The following lemmas explain our interest in clean holes.

Lemma 6.21 *If for a bipartite graph G with no short 3-wheel, one of the shortest unbalanced holes in G is clean and x is a dominated node of G then one of the shortest unbalanced holes in $G - x$ is clean.*

Lemma 6.22 *If a bipartite graph G with no short 3-wheel contains a full double star cutset centered at (u, v) and one of the shortest unbalanced holes in G is clean then either,* *(i) For some component U of $G - S$, one of the shortest unbalanced hole in $U + S$ is clean, or*

(ii) there is a simple 3PC through (u, v).

Lemma 6.23 *For any graph G, we can find in polynomial time a family $\mathcal{F}$ of subgraphs of G such that if G contains an unbalanced hole then for some $F \in \mathcal{F}$ one of the shortest unbalanced holes in F is clean.*

We discuss the proofs of these lemmas below, after having shown how they are used in our algorithm for recognizing balanced graphs.

6.6.3 A recognition algorithm

By Lemma 6.23, to develop an efficient algorithm for recognizing balanced graphs, we need only provide an efficient algorithm which given a graph F returns either (i) F is unbalanced, or (ii) F is either balanced or an unbalanced graph none of whose shortest unbalanced holes is clean. We now present such an algorithm. We assume F is connected as we can test each component of F separately.

Algorithm: Short Unclean Hole Testing

Input: A connected graph F

Output: either (i) F is unbalanced, or
(ii) F is either balanced or an unbalanced graph none of whose shortest unbalanced holes is clean.

Data Structures: A list $\mathcal{L}$ of graphs.

Step 0: Determine if F is bipartite and contains no short 3-wheel, if not return (i) and stop. Otherwise, set $\mathcal{L} = \{F\}$

Step 1: If $\mathcal{L}$ is empty return (ii) and stop. Otherwise, let H be the first graph on $\mathcal{L}$.

Step 2: Determine if H has a dominated node. If it does, let x be such a node. Add $H - x$ to $\mathcal{L}$ and return to Step 1.

Step 3: Otherwise, determine if H has a full star cutset. If it does we let S be such a cutset. We check if there is a simple 3PC through the center of the star-cutset. If such an object exists we return (i) and stop. Otherwise, letting $U_1, ..., U_k$ be the components of $H - S$, we add the graphs $H_1 = U_1 \cup S, ..., H_k = U_k \cup S$ to $\mathcal{L}$.

Step 4: Otherwise, we test if H is totally unimodular. If it is not, we return (i) and stop. If it is, we return to Step 1.

Now, this algorithm clearly terminates as a simple induction argument shows that we add at most n^n graphs to $\mathcal{L}$ when applying the algorithm to a graph with n nodes.

Applying Lemmas 6.21 and 6.22 in conjunction with Theorem 6.13 and Observation 6.16, we see that if we do not stop in Step 0 then throughout the rest of the algorithm the following property holds: $\mathcal{L}$ is a list of connected (bipartite) subgraphs of F such that F is an unbalanced graph one of whose shortest unbalanced holes is clean if and only if one of the graphs on $\mathcal{L}$ also has this property. So, when the algorithm terminates, it returns the right answer.

We now consider the time complexity of the algorithm. Our previous remarks show that processing one graph by carrying out Steps 1-4 can be done efficiently, it remains only to bound the number of subgraphs to which we apply the algorithm, i.e. the number of graphs added to $\mathcal{L}$.

To this end, we focus on the induced paths of length three in F. We let $h(F)$ be the number of pairs of vertices $\{x, y\}$ of F such that neither x nor y is dominated and there is an induced path $xvwy$ in F. The key to our analysis is the following:

Observation 6.24 *If for a graph F, we add k graphs $F_1, ..., F_k$ to $\mathcal{L}$ upon applying the algorithm to F then $\sum_{i=1}^{k} h(F_i) \leq h(F)$.*

Proof. It is easy to verify that if a node is dominated in F and is in F_i then it is also dominated in F_i. So, for each F_i, any pair of vertices counted towards $h(F_i)$ also counts towards $h(F)$. Thus, we need only show that no pair of vertices counts towards $h(F_i)$ and $h(F_j)$ for distinct i and j. This implies that k is at least two and thus that we decomposed F using a full double star cutset. S, with centre (u, v) say. Now, suppose to the contrary that there is such a pair x, y. Then $x, y \in V(F_i) \cap V(F_j) = S$. Furthermore, x and y are on opposite sides of the bipartition, so, by symmetry, we can assume $x \in N(v)$ and $y \in N(u)$. Since x and y are non-adjacent, this yields $y \neq v$ and $x \neq u$. Now, since x is not dominated by u in F_i , it must see a vertex of U_i. Symmetrical arguments imply that y sees a vertex of U_i and so there is a path from x to y in $x + y + U_i$. Similarily, there is a path from x to y in $x + y + U_j$. But these two paths yield a simple 3PC through (u, v) in F. Hence, we should not have decomposed F in Step 2 rather we should have halted because F is not balanced. This contradiction yields the desired result. □

We call a graph F *trivial* if $h(F) = 0$. Using our observation, a simple inductive argument implies that when we apply our algorithm to a nontrivial graph F, at most $|V(F)|h(F)$ non-trivial graphs are added to $\mathcal{L}$. It is then straightforward to show that the total number of graphs added to $\mathcal{L}$ is at most $|V(F)|^2 h(F)$ (because a connected trivial bipartite graph without a dominated node is a vertex).

Thus, our algorithm does indeed run in polynomial time.

6.6.4 A fresh look at cleanliness

To complete our discussion of balanced graphs, we return to the three lemmas on clean holes stated earlier.

Proof of Lemma 6.21. Let G be a bipartite unbalanced graph one of whose shortest unbalanced holes H is clean. Let x be a vertex of G dominated by some other vertex x'. If x is not a vertex of H then H is a shortest unbalanced hole of $G - x$ which is clean, and we are done. Otherwise, since H is clean, $H' - x + x'$ is a hole. Clearly, H' is a shortest unbalanced hole of G and hence of $G - x$, we need only show that if G contains no short 3-wheel then H' is clean.

Now no vertex y of $G - x - H'$ is adjacent to more than three vertices of H', as otherwise y would be adjacent to at least three vertices of H, contradicting the fact that H is clean. Similar reasoning yields that if y is adjacent to three vertices of H' then y sees x' but not x and for some y' in H, $N(y) \cap H = N(y') \cap H$. This implies that x and y form a tent over H, contradicting the fact that it is clean.

If some vertex y of $G - x$ sees exactly two vertices of H' then one of the two cycles formed by y and a path of H' between the two vertices in $N(y) \cap H'$ is unbalanced.

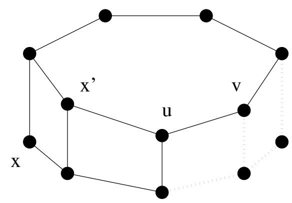

Figure 6.7

So, since H' is a shortest unbalanced hole of G, there must be a vertex y' of H' such that $N(y) \cap H' = N(y') \cap H'$.

Finally, if there is a tent over H' then either there is a tent over H or one of the vertices of the tent sees x' but not x. In this latter case, as shown in Figure 6.7, there is a short 3-wheel in G. □

Proof of Lemma 6.22. Let G be a bipartite unbalanced graph one of whose shortest unbalanced holes H is clean. Let S be a full double star cutset of G centered at (u, v). If H does not intersect two components of $G - S$ then for some component U of $G - S$, we have that H is a shortest unbalanced hole of $U + S$ which is clean, and we are done.

Now, since H is clean, $|N(u) \cap H| \leq 2$ and $|N(v) \cap H| \leq 2$, so $|H \cap S| \leq 4$.

If $|N(u) \cap H| = |N(v) \cap H| = 1$, then H along with the vertices u and v induces a simple 3PC through u and v, and we are done.

If $|N(u) \cap H| = 2$ and $|N(v) \cap H| = 0$ then since H is clean, there is a vertex u' of H such that $H' = H - u' + u$ is a hole. Note that H' is a shortest unbalanced hole of $U \cup S$ for some component U of $G - S$. Proceeding as in the proof of Lemma 6.21, we can show that either G contains a short 3-wheel or H' is clean.

By symmetry, if $|N(v) \cap H| = 2$ and $|N(u) \cap H| = 0$ then the desired result holds. So, if $|H \cap S| = 2$, we are done. If $|H \cap S| \in \{3, 4\}$, we proceed in a similar manner. For more details, we refer the reader to [22]. □

Proof of Lemma 6.23. This lemma builds on the following result of Conforti and Rao [25]

Lemma 6.25 *For any shortest unbalanced hole H in a bipartite graph, there are a pair of vertices a and b in H such that $N(a) \cup N(b)$ contains the set $N_H^* = \{z | N(z) \cap H \geq 3\}$.*

Proof (Sketch). We partition N_H^* into M_H^* and L_H^* in accordance with the bipartition of G, and show that there is a vertex of H seeing all of M_H^*. By symmetry, there is a vertex seeing all of L_H^*. Clearly, we can assume that H has at least eight vertices and $|M_H^*| \geq 2$. Note that the former implies that G contains no C_6.

We first observe that every vertex x in M_H^* sees an odd number of vertices of H as otherwise there would be an unbalanced hole shorter than H, induced by x and some path of H between two neighbours of x. Using this fact, it is straightforward to show that for every two vertices x and y in M_H^* there is a vertex of H seeing both x and y as otherwise there would be an unbalanced hole shorter than H induced by some subset of $V(H) + x + y$.

Now, it is also straightforward to show that, in a bipartite graph, if every pair of vertices on one side M have a common neighbour then either the graph contains a

C_6 or some vertex on the other side sees all of M. Combining these results yields the lemma. For further details the readers may consult ([25] pp. 37-39). □

Cournejols, Conforti, and Rao (see Lemma 3.6 and 3.7 in [22]) were able to strengthen this result proving:

Lemma 6.26 *For any shortest unbalanced hole H in a bipartite graph, there are a pair of edges bc and fg in H such that $N(b) \cup N(c) \cup N(f) \cup N(g)$ contains the set $N_H^* = \{z|N(z) \cap H \geq 3\}$ as well as a vertex from every tent over H.*

This implies that for every shortest unbalanced hole H there are two P_4s: $abcd$ and $efgh$ of H such that H is clean in the graph $H_{abcd,efgh} = G - ((N(b) \cup N(c) \cup N(e) \cup N(f)) - a - b - c - d - e - f - g - h)$.

Since there are at most n^8 choices for the two P_4s, the desired result follows.

6.6.5 Recognizing balanceable graphs

In [17], Conforti, Cornuejols, Kapoor, and Vušković extend the results of Conforti, Cornuejols, and Rao by presenting an algorithm to determine if a graph is balanceable.

We note that the algorithm for determining if a graph is balanceable can be used to determine if a graph is balanced, as follows. We first check if G is balanceable, if it is not then obviously it is not balanced. So, we will assume G is balanceable. Next we choose some spanning forest F and for each edge e of F check the length of the unique hole H_e through e in $e+F$. If any of these holes are unbalanced then obviously G is not balanced, so we assume the contrary. In this case, the only labelling of G which labels every edge of F with a 1 which could conceivably be balanced is the all 1s labelling. We claim that G has a balanced labelling in which all the edges of F are labelled 1, which implies that G is balanced.

To prove our claim we note that for any subset A of V, every cycle contains an even number of edges in the cut consisting of the edges between A and $V - A$. Hence, given a balanced labelling, swapping the signs of all the edges in a given cut yields a new balanced labelling. Applying this result once yields that for any edge e of G we can find a balanced labelling in which e is labelled 1. Applying it repeatedly we can obtain that there is a labelling in which all the edges on F are labelled 1 because for every edge e of F there is a cut containing e and no other edge of F.

We remark that for any graph G, by considering a spanning forest F and the holes H_e as above, we can find a unique candidate labelling of G in which all the edges of F are labelled 1, such that no other labelling with all the edges of F labelled 1 is balanced. Mimicking the proof of our claim, we see that G is balanceable if and only if our candidate labelling is balanced. In fact these remarks show that the problem of recognizing balanceable graphs is equivalent to the problem of determining if a labelling is balanced.

The algorithm for determining if a graph is balanceable is similar in many respects to that given above for recognizing balanced graphs. The only difference is that the analogue of Theorem 6.13 states that a balanceable graph which is not totally unimodular and has at least 11 vertices has a double star cutset or one of other two decompositions: a *2-join* or a *6-join.* Thus, the authors had to use these new decompositions in their algorithm and prove analogues of Lemmas 6.21 and 6.22 for them. Readers can refer to [17] for the details.

6.7 Graphs without Even Holes

The recognition algorithm for even-hole free graphs was motivated by and has a similar structure to that for balanceable graphs. Once again, the key is a decomposition theorem. To state this theorem we need some definitions:

Definition 6.27 *A graph is a clique-tree if each of its maximal 2-connected components is a clique. A graph is an extended clique-tree if it can be obtained from a clique tree by adding at most two nodes.*

Definition 6.28 *A star cutset S is a k-star cutset if for some clique C of size k in S: $S - C \subseteq \cup_{x \in C} N(x)$.*

Definition 6.29 *A connected graph G has a 2-join if its nodes can be partitioned into two sets V_1 and V_2 so that V_i contains disjoint non-empty subsets A_i and B_i such that:*

(a) every node of A_1 is adjacent to every node of A_2,
(b) every node of B_1 is adjacent to every node of B_2,
(c) there are no other edges between V_1 and V_2, and
(d) If $|A_i| = 1$ and $|B_i| = 1$ then V_i does not induce a chordless path.

Theorem 6.30 *Every connected even hole free graph which is not an extended clique tree has either a k-star cutset for $k \leq 3$ or a 2-join.*

Remark 6.31 *This theorem is really just a restatement of work of Conforti, Cornuejols, Kapoor, and Vušković.*

To apply this theorem to solve the recognition problem for even hole free graphs, we once again need to focus on a special class of *clean* even holes. The definitions are more complicated than in the balanceable case, we refer the reader to [19] for details.

In any event, using an algorithm similar to Short Unclean Hole Testing, Conforti et al. reduce the problem of testing if a graph is even hole free to determining if extended clique trees are even-hole free. Clearly, clique trees contain no holes. Moreover, it is not hard to see that in a clique tree there is at most one induced path between every pair of vertices. So, if $G - x$ is a clique tree, to determine if G is even hole free we need only test for every pair y, z of neighbours of G, whether there is a path from y to z in $G - x$ which has an even number of edges and contains no other neighbour of x. Since there is at most one path to test, this can be done in polynomial time. A similar algorithm allows us to test in polynomial time if an extended clique tree contains an even hole.

Thus, Theorem 6.30 can be used to show that recognizing even hole free graphs is in P. We believe that it can also be used to prove:

Conjecture 6.32 *Every even hole free graph contains a vertex whose neighbourhood can be partitioned into two cliques.*

Of course, for perfect graph theorists, the real interest of this decomposition theorem is the possibility that similar results might be obtained for odd hole free graphs.

6.7.1 The decomposition theorem

We now derive Theorem 6.30 using three decomposition theorems due to Conforti et al.

Definition 6.33 *A cap is a hole together with a node which is adjacent to two adjacent vertices but nothing else on the hole. G is cap-free if none of its induced subgraphs is a cap.*

Definition 6.34 *A gem is a P_4 together with a vertex which sees all of the P_4.*

Note that if x and y are the endpoints of a P_4 in a gem then they are joined through the gem by paths of both parities. Clearly, either (a) one of these paths can be extended to an even hole or (b) there is a 3-star cutset separating x and y contained in the union of the neighbourhoods of the other three vertices in the gem. So, every even hole free graph which contains a gem has a 3-star cutset.

Conforti et al. proved:

Theorem 6.35 *Every connected even hole free graph which is not an extended clique tree and contains no gem but contains a cap has either a k-star cutset for $k \leq 3$ or a 2-join.*

By our earlier remarks, this implies:

Corollary 6.36 *Every connected even hole free graph which is not an extended clique tree but contains a cap has either a k-star cutset for $k \leq 3$ or a 2-join.*

In [19], Conforti et al. define the class of *basic* graphs. Every basic graph is either (a) chordal, (b) a biconnected triangle-free graph, or (c) a biconnected triangle-free graph H along with a vertex adjacent to all of H. They also prove in [21]:

Theorem 6.37 *Every biconnected triangle-free even hole free graph is an odd hole or has a star cutset.*

Remark 6.38 *Actually they prove a much stronger result.*

Clearly, if some vertex in a graph sees all the others than either the graph is a clique or contains a star cutset. Furthermore, by Corollary 6.2, every chordal graph which is not a clique has a star cutset. Thus we have:

Corollary 6.39 *Every basic graph which does not have a star cutset is either a clique or an odd hole.*

An *amalgam* defined in [65] is a special kind of star cutset (see [12]). In [20], Conforti et al. prove:

Theorem 6.40 *Every cap-free graph which is not basic has an amalgam.*

Since both odd holes and cliques are clearly extended clique trees, combining this result with Corollary 6.39 and Corollary 6.36 yields Theorem 6.30.

We remark that the results of Conforti et al. easily imply (we omit the details):

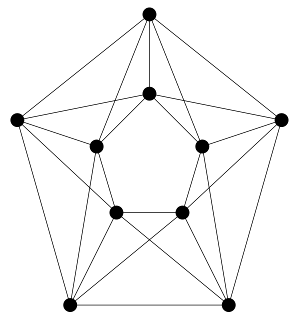

Figure 6.8

Theorem 6.41 *A cap-free even hole free graph G satifies one of the following:*

(a) G is triangle-free
(b) G has a clique cutset,
(c) G has two adjacent vertices with the same neighbour set,
(d) G has a vertex which is adjacent to all the other vertices.

This allows us to prove Conjecture 6.32 for cap free graphs. Indeed we obtain by induction that every vertex which is non-adjacent to some vertex is non-adjacent to a vertex whose neighbourhood splits into two cliques. We omit the details.

6.8 β-Perfect Graphs

For any ordering $\{v_1, ..., v_n\}$ of the vertices of a graph G, we can greedily colour G by considering the vertices in turn and colouring v_i with the lowest nonnegative integer not already used on a neighbour of v_i (i.e. not appearing on $N(v_i) \cap \{v_1, ..., v_{i-1}\}$). In this fashion, we can colour G with $\Delta + 1$ colours; where Δ is the maximum degree of a vertex in the graph. Actually, we can do much better.

We let $\beta(G) = max_{H \subseteq G}\{\delta(H) + 1\}$ where $\delta(H)$ is the minimum degree of H. We order the vertices by repeatedly removing a vertex of minimum degree in the subgraph of vertices not yet chosen and placing it after all the remaining vertices but before those already removed (see [63] for results on this order; in [13] this order is discussed in relation to perfect graphs). Applying our greedy colouring procedure to this order yields $\chi(G) \leq \beta(G)$ (as was noted in [63] and [75]).

We call a graph G *β-perfect* if for every induced subgraph H of G, $\chi(H) = \beta(H)$. We note that an even hole H satisfies $\chi(H) = 2$ and $\beta(H) = 3$ so β-perfect graphs are even hole free. On the other hand, there are even-hole free graphs which are not β-perfect, one of these is depicted in Figure 6.8. In fact, no exact characterization of β-perfect graphs is known.

There are some partial results on this question. In [62], Markossian, Gasparian, and Reed proved that if G is even hole free, contains no cap, and contains no C_4 with just one chord then G is β-perfect. De Figuereido and Vušković strengthened this result in [30], by proving that cap-free even hole free graphs are β-perfect. Furthermore, although even hole free graphs are not β-perfect, they do satisfy $\chi \geq \frac{\beta}{2} + 1$ (this result is obtained by noting that every pair of colour classes in a colouring of an even hole free graph induces a forest and then counting edges, see [62]). Note that Conjecture 6.32, if true, yields $\omega \geq \frac{\beta}{2} + 1$. Finally, Gasparian, Markossian, and Reed[62] proved:

Theorem 6.42 *G and $\overline{G}$ are β-perfect if and only if G and $\overline{G}$ are even-hole free.*

This is an interesting analogue of the strong perfect graph conjecture. It does not completely characterize β-perfect graphs as there are graphs which are β-perfect whose complements are not. The smallest such graph is $\overline{C_4}$.

The study of even-hole free graphs was motivated by the connections to β-perfection discussed above.

6.9 Graphs without Odd Holes

A *2-division* of a graph is a partition of its vertex set into two parts neither of which contains a maximum clique. Hoàng and McDiarmid[56] call a graph *2-divisible* if all of its induced subgraphs permit a 2-division.

Clearly, for any colour class S in an ω colouring of a graph, $(S, V-S)$ is a 2-division. Thus, every perfect graph is 2-divisible. On the other hand, an odd hole has chromatic number three and thus no 2-division. Thus, every 2-divisible graph is odd hole free.

In [56], Hoàng and McDiarmid made the following conjectures.

Conjecture 6.43 *G is 2-divisible if and only if it is odd hole free.*

Conjecture 6.44 *Both G and $\overline{G}$ are 2-divisible if and only if G is perfect.*

We propose the following conjecture.

Conjecture 6.45 *Both G and $\overline{G}$ are 2-divisible if and only if G is Berge.*

Note that the first two conjectures together trivially imply the SPGC. Actually, as remarked by Hoàng and McDiarmid, the second conjecture is equivalent to the SPGC. Thus, it implies the third conjecture. So the third conjecture is a common weakening of the first two.

Conjecture 6.43 provides some motivation for the study of odd hole free graphs independent of the SPGC.

Hoàng and McDiarmid proved Conjecture 6.43 for claw-free graphs; a graph is claw-free if it does not contain the graph consisting of a stable set of size three together with a vertex adjacent to all three vertices of the stable set. Their proof relies on the following two results.

Theorem 6.46 *[15] Every Berge claw-free graph is perfect.*

Theorem 6.47 *[2] Every connected odd hole free claw-free graph containing a stable set of size three is Berge.*

Combining these results we see that every claw-free odd hole free graph containing a stable set of size three is perfect and hence 2-divisible. Thus, Hoàng and McDiarmid only needed to prove Conjecture 6.43 for graphs with stability number at most two, which is straightforward.

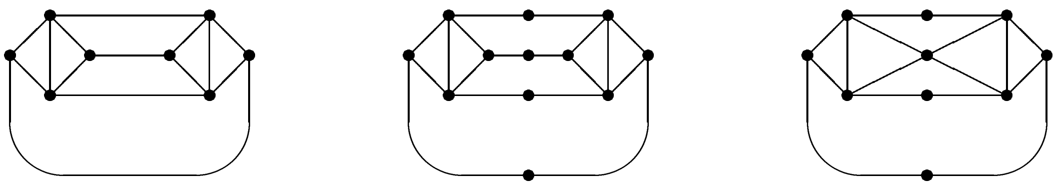

Figure 6.9 Three special graphs from Hsu's decomposition.

Conjecture 6.43 also holds for planar graphs because planar odd-hole free graphs are perfect and hence 2-divisible (see [77]).

Robertson, Seymour, and Thomas [68] proved that Conjecture 6.43 holds for graphs with $\omega \leq 3$. They actually proved that every odd hole free graph not containing a clique of size four is four colourable which, as noted by Hoàng and McDiarmid, amounts to the same thing.

Of course, these results all use structural properties of the graphs under consideration. As we now discuss, these properties also lead to recognition algorithms for some of these graph classes.

Chvatal and Sbihi [15] showed that every claw-free Berge graph can be decomposed using clique cutsets into graphs in two simple base classes thereby obtaining a polynomial time recognition algorithm for the class (see also [61]). By Lemma 6.47 this algorithm can be used to test if a claw free graph containing a stable set of size three contains an odd hole. Now, if a graph contains no stable set of size three then it is odd hole free if and only if it is C_5 free. So we can also deal with the remaining case efficiently, and have a polynomial time algorithm to test if claw-free graphs are odd hole free.

Clearly, a planar graph is odd hole free precisely if it is Berge.

In [58] Hsu proved that every planar Berge graph either has a cutset of size at most four or is either a comparability graph, a line graph of a bipartite graph, one of the three graphs in Figure 6.9, or a graph obtained from the leftmost graph in this figure by replacing one or more of the three horizontal edges with an odd path. This allowed him to reduce the problem of testing if a planar graph is Berge to testing if a graph is in one of these three classes. The result was an $O(n^3)$ algorithm for testing if a planar graph is Berge. For further details, see [58].

The result of Robertson, Seymour, and Thomas did not lead to a polynomial time recognition algorithm for testing if a K_4-free graph contains an odd hole. Indeed, this problem is still open, as is the problem of testing the perfection of a K_4-free graph.

Cap-free odd hole free graphs are called Meyniel because it was Meyniel who proved such graphs are perfect [65]. Burlet and Fonlupt [9] proved that every cap-free odd hole free graph which is not basic has an amalgam. They used this special case of Theorem 6.40 to develop a polynomial time recognition algorithm for cap-free odd hole free graphs.

Conforti et al. used Theorem 6.40 to obtain an algorithm to recognize cap-free even-signable graphs; a graph is *even signable* if its edges can be labelled with 1s and -1s so that the sum of the edges around every triangle is odd but the sum around every hole is even. They have also obtained recognition algorithms and decomposition theorems for other classes of odd hole free graphs. Most interestingly, they have recently shown that graphs without odd holes and holes of length four are perfect [23].

With all this interest in characterizing odd hole free graphs, and with weapons with proven track records being brought to bear, it appears that this problem will be resolved in the near future.

Acknowledgements

We thank Michele Conforti, Gerard Cornuejols, Celina de Figueiredo, Chinh Hoàng, Frederic Maffray, Colin McDiarmid, Bert Randerath, Jerry Spinrad, Sri Sitharan, Robin Thomas and Kristina Vušković for helpful discussions and pointers to references.

REFERENCES

[1] S. R. Arikati and C. Pandu Rangan, An efficient algorithm for finding a two-pair, and its applications, *Discrete Appl. Math.* **65** (1996), 5-20.
[2] A. Ben Rabea, manuscript, see also [15].
[3] C. Berge, Les problèmes de coloration en théorie des graphes, *Publ. Inst. Statist. Univ. Paris 9* (1960), 123-160.
[4] C. Berge, Balanced matrices, *Math Programming* **2** (1972), 19-31.
[5] C. Berge and V. Chvátal, ed., Topics on Perfect Graphs, *Ann. Discrete Math.* **21**, North Holland, Amsterdam, (1984)
[6] C. Berge and M. Las Vergnas, Sur une theoreme de type Konig pour les hypergraphes, *Ann. of the New York Academy of Science* **175** (1970), 32-40.
[7] D. Bienstock, On the complexity of testing for odd holes and odd induced paths, *Discrete Math* **90** (1991), 85-92.
[8] A. Brandstädt, V.B. Le, J.P. Spinrad, Graph Classes: A Survey, *SIAM*, Philadelphia, (1999).
[9] M. Burlet and J. Fonlupt A polynomial time algorithm to recognize a Meyniel Graph, in [5].
[10] P. Buneman, A characterization of rigid circuit graphs, *Discrete Math.* **9** (1974), 205-212.
[11] V. Chvátal, Perfectly ordered graphs, in: C. Berge and V. Chvátal, ed., *Ann. Discrete Math.* **21** (1984), 63-65.
[12] V. Chvátal, Star cutsets, *J. Comb. Theory B* **39** (1985), 189-199.
[13] V. Chvátal, C, Hoàng, N. Mahadev, D. De Werra, Four classes of perfectly orderable graphs, *Journal of Graph Theory* **11** (1987), 481-495.
[14] V. Chvátal, I. Rusu, R. Sritharan, Dirac-type characterizations of graphs without long chordless cycles, *Discrete Math*, to appear.
[15] V. Chvátal and N. Sbihi, Recognizing Claw-Free Berge Graphs, *J. Comb. Theory B* **44** (1988), 154-176.
[16] P. Colley, A. Lubiw, J. Spinrad, Visibility Graphs of Towers, *Computational Geometry Theory* **7** (1997), 161-172.
[17] M. Conforti, G. Cornuejols, A. Kapoor, K. Vušković, Balanced 0,+1,-1, Matrices, *Journal of Combinatorial Theory (B)*, to appear.
[18] M. Conforti, G. Cornuejols, A. Kapoor, K. Vušković, Even Holes I: A Decomposition Theorem, manuscript.
[19] M. Conforti, G. Cornuejols, A. Kapoor, K. Vušković, Even Holes II: A Recognition Algorithm, manuscript.
[20] M. Conforti, G. Cornuejols, A. Kapoor, K. Vušković, Even and Odd Holes in Cap-free Graphs, *Journal of Graph Theory* **30** (1999), 289-308.

[21] M. Conforti, G. Cornuejols, A. Kapoor, K. Vušković, Triangle-free graphs which are signable without even holes, *Journal of Graph Theory* **34** (2000), 289-308.
[22] M. Conforti, G. Cornuejols, R. Rao, Decomposition of Balanced Matrices, *Journal of Combinatorial Theory(B)* **77** (1999), 292-406.
[23] M. Conforti, G. Cornuejols, K. Vušković, private communication.
[24] M. Conforti, B. Gerards, A. Kapoor, A theorem of Truemper, *Combinatorica* **20** (2000), 15-27.
[25] M. Conforti and R. Rao Properties of Balanced and Perfect Matrices, *Math Programming* **55** (1992), 35-49.
[26] C. Coppersmith and S. Winograd, Matix multiplication via arithmetic progressions, *Proc. 19th Ann. Symp. on Theory of Computation* (1987), 1-6.
[27] D. G. Corneil, Y. Perl, L. K. Stewart, A linear recognition algorithm for cographs. *SIAM J. Comput.* **14** (1985), 926-934.
[28] E. Dalhaus, P. L. Hammer, F. Maffray, S. Olariu, On domination elimination orderings and domination graphs, *RUTCOR Res. Rpt.* (1994), 27-94.
[29] G. A. Dirac, On rigid circuit graphs, *Abh. Math. Sem. Univ. Hamburg* **25** (1961), 71-76.
[30] C. De Figuereido and K. Vušković, A class of β perfect graphs, *Discrete Mathematics* **216** (2000), 169-193.
[31] E. M. Eschen, R. B. Hayward, J. Spinrad, and R. Sritharan, Weakly triangulated comparability graphs, *SIAM Journal of Computing* **29**(2) (1999), 378-386.
[32] E. M. Eschen and R. Sritharan, A characterization of some graph classes with no long holes, *J. Comb. Theory, Series B* **65** (1995), 156-162.
[33] E. M. Eschen, R. Sritharan, J. Spinrad, private communication.
[34] H. Everett, D. G. Corneil, Recognizing visibility graphs of spiral polygons, *J. Algorithms* **11** (1990), 1-26.
[35] D. R. Fulkerson and O. A. Gross, Incidence matrices and interval graphs, *Pacific J. Math.* **15** (1965), 835-855.
[36] D. R. Fulkerson, A. Hoffmann, and R. Oppenheim, On Balanced Matrices, *Mathematical Programming Study* **1** (1974), 120-132.
[37] F. Gavril, Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph, *SIAM J. Comp.* **1** (1972), 180-187.
[38] F. Gavril, The intersection graphs of subtrees in trees are exactly the chordal graphs, *J. Comb. Theory B* **16** (1974), 47-56.
[39] F. Gavril, Algorithms on clique separable graphs, *Discrete Math.* **19** (1977), 159-165.
[40] M. C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, (1980).
[41] M. C. Golumbic, C. L. Monma, W. T. Trotter, Tolerance graphs, *Discrete Appl. Math* **9** (1984), 157-170.
[42] M. Grötschel, L. Lovász, and A. Schrijver, Polynomial algorithms for perfect graphs, in: C. Berge and V. Chvátal, ed., *Ann. Discrete Math.* **21** (1984), 325-356.
[43] A. Gyárfas, Problems from the World Surrounding Perfect Graphs, *Zastos. Math.* **XIX** (1987), 1-11.
[44] A. Hajnal and J. Surányi, Über die Auflösung von Graphen in vollständige Teilgraphen, *Ann. Univ. Sci. Budapest Eötvos. Sect. Math.* **1** (1958), 113-121.
[45] R. B. Hayward, Weakly triangulated graphs, *J. Comb. Theory, Series B* **39** (1985), 200-209.
[46] R. B. Hayward, Two classes of perfect graphs, Ph.D. thesis, McGill University (1987).
[47] R. B. Hayward, Murky graphs, *J. Comb. Theory, Series B* **49** (1990) 200-235.
[48] R. B. Hayward, Discs in Unbreakable Berge Graphs, *Graphs and Combinatorics* **11** (1995), 249-254.
[49] R. B. Hayward, Generating weakly triangulated graphs, *J. Graph Theory* **21** (1996), 67-70.

[50] R. B. Hayward, Meyniel weakly triangulated graphs I: co-perfect orderability, *Discrete Appl. Math* **73** (1997), 199-210.
[51] R. B. Hayward and P. Kearney, NeST Graphs, *Discrete Appl. Math.*, submitted
[52] R. B. Hayward, C. T. Hoàng, and F. Maffray, Optimizing weakly triangulated graphs, *Graphs and Combin.* **6** (1990), 33-35.
[53] R. B. Hayward, J. Spinrad, and R. Sritharan, Weakly chordal graph algorithms via handles, *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, CA (2000), 42-49.
[54] C. T. Hoàng and N. Khouzam, On brittle graphs, *J. Graph Theory* **12** (1988), 391-404.
[55] C. T. Hoàng and F. Maffray, Weakly triangulated graphs are strict quasi-parity graphs, *RUTCOR Research Rpt., Rutgers Univ.* (1998), 6-86.
[56] C. T. Hoàng and C. McDiarmid, On the divisibility of graphs, *Discrete Mathematics*, to appear.
[57] A. Hoffman and J. Kruskal, Integral Boundary Points of Convex Polyhedra, in Linear Inequalities and Related Systems (H. W. Kuhn and A. W. Tucker Eds.), Princeton University Press, Princeton N.J. (1956), 223-246.
[58] W. L. Hsu Recognizing planar perfect graphs, *Journal of the ACM* **34** (1987), 255-288.
[59] L. Lovász, Normal hypergraphs and the perfect graph conjecture, *Discrete Math.* **2** (1972), 253-267.
[60] F. Maffray and M. Preissmann, Perfect graphs with no P_5 and no K_5, *Graphs and Combinatorics* **10** (1994), 179-184.
[61] F. Maffray and B. Reed A Description of Claw-Free Perfect Graphs, *Journal of Combinatorial Theorey(B)* **75** (1999), 134-156.
[62] S.E. Markossian, G.S. Gasparian, B. Reed, β-Perfect Graphs, *Journal of Combinatorial Theory(B)* **47** (1996), 1-11.
[63] D. Matula, A min-max theorem with application ot graph colouring, *SIAM review* **10** (1968), 481-482.
[64] R. McConnell and J. Spinrad, Linear-time transitive orientation, *Proc. 8th Ann. ACM-SIAM Symposium on Discrete Alg.* (1997), 19-25.
[65] H. Meyniel, On the Perfect Graph Conjecture, *Discrete Mathematics* **16** (1976), 339-342.
[66] K.R. Parthasarathy and G. Ravindra, The Strong Perfect Graph Conjecture is true for $K_{1,3}$-free graphs, *Journal of Combinatorial Theory(B)* **21** (1976), 212-223.
[67] B. Randerath and I Schiermeyer, Colouring Graphs with Prescribed Induced Cycle Lengths, Tech. Report 98.341, Universitat Zu Koln, 1998 (an extended abstract appeared in the proceedings of SODA99).
[68] N. Robertson, P. Seymour, R. Thomas, Private Communication.
[69] D. J Rose, R. E Tarjan, G. S. Leuker, Algorithmic aspects of vertex elimination on graphs, *SIAM J. Comput.* **5** (1976), 266-283.
[70] I. Rusu, Graphs with chordless cycles of bounded length, *J. Graph Theory*, to appear.
[71] I. Rusu and J. Spinrad, *Domination graphs: examples and counterexamples*, manuscript.
[72] P. Seymour, Decomposition of Regular Matroids, *Journal of Combinatorial Theory(B)* **28** (1980), 305-359.
[73] J. Spinrad, Finding large holes, *Info. Proc. Letters* **39** (1991), 227-229.
[74] J. Spinrad and R. Sritharan, Algorithms for weakly triangulated graphs, *Discrete Math.* **19** (1995), 181-191.
[75] G. Szekeres and H. Wilf, An inequality for the chromatic number of a graph, *Journal of Combinatorial Theory* **4** (1968), 1-3.
[76] K. Truemper Alfa-balanced graphs and matrices and GF(3)-representability of matroids. *Journal of Combinatorial Theory(B)* **32** (1982), 112-139.
[77] A. Tucker, The Strong Perfect Graph Conjecture for Planar Graphs, *Canadian Journal of Mathematics* **25** (1973), 103-114.

[78] J. R. Walter, Representations of rigid cycle graphs, Ph.D. thesis, Wayne State Univ. (1972).

[79] S. H. Whitesides, An Algorithm for Finding Clique Cutsets, *Information Procesing Letters* **12** (1981), 31-32.