Proceedings Paper:
Vuskovic, K (2013) The world of hereditary graph classes viewed through Truemper configurations. In: Surveys in Combinatorics 2013. 24th British Combinatorial Conference, 30 June - 5 July 2013, Royal Holloway, University of London. Cambridge University Press , 265 - 325. ISBN 978-1-107-65195-1

The world of hereditary graph classes viewed through Truemper configurations

Kristina Vušković

Abstract

In 1982 Truemper gave a theorem that characterizes graphs whose edges can be labeled so that all chordless cycles have prescribed parities. The characterization states that this can be done for a graph G if and only if it can be done for all induced subgraphs of G that are of few specific types, that we will call Truemper configurations. Truemper was originally motivated by the problem of obtaining a co-NP characterization of bipartite graphs that are signable to be balanced (i.e. bipartite graphs whose node-node incidence matrices are balanceable matrices).

The configurations that Truemper identified in his theorem ended up playing a key role in understanding the structure of several seemingly diverse classes of objects, such as regular matroids, balanceable matrices and perfect graphs. In this survey we view all these classes, and more, through the excluded Truemper configurations, focusing on the algorithmic consequences, trying to understand what structurally enables efficient recognition and optimization algorithms.

1 Introduction

Optimization problems such as coloring a graph, or finding the size of a largest clique or stable set are NP-hard in general, but become polynomially solvable when some configurations are excluded. On the other hand they remain difficult even when seemingly quite a lot of structure is imposed on an input graph. For example, determining whether a graph is 3-colorable remains NP-complete for triangle-free graphs with maximum degree 4 [92]. The approximation approach to these problems does not help either, since for example unless P=NP, there does not exist a polynomial time algorithm that can find a $2\chi(G)$-coloring of a graph G [73]. So if $\mathcal{C}$ is a class of graphs for which there exist polynomial time algorithms that find the chromatic number, $\mathcal{C}$ must have some "strong structure". Understanding structural reasons that enable efficient algorithms is our primary interest in this survey.

In 1982 Truemper [121] gave a theorem (Theorem 2.1) that characterizes graphs whose edges can be labeled so that all chordless cycles have prescribed parities. The characterization states that this can be done for a graph G if an only if it can be done for all induced subgraphs of G that are of few specific types (depicted in Figure 1), that we will call *Truemper configurations*, and will describe precisely in Section 2. Truemper was originally motivated by the problem of obtaining a co-NP characterization of bipartite graphs that are signable to be balanced (i.e. bipartite graphs whose node-node incidence matrices are balanceable matrices, a class of matrices that have important polyhedral properties).

The configurations that Truemper identified in his theorem ended up playing a key role in understanding the structure of several seemingly diverse classes of objects, such as regular matroids, balanceable matrices and perfect graphs. A powerful technique called the decomposition method, which we describe in Section 3, was used in structural analysis of all these classes. In these decomposition theorems

Truemper configurations appear both as excluded structures that are convenient to work with, and as structures around which the actual decomposition takes place.

In this survey, in trying to understand what structurally enables efficient recognition and optimization algorithms, we will view different classes of objects and their associated decomposition theorems, through excluded Truemper configurations. We survey the above mentioned classes, as well as other classes closed under taking graph minors (such as cycle-free graphs, outerplanar graphs, series-parallel graphs, etc.) and those closed under taking induced subgraphs (such as hole-free graphs, claw-free graphs, bull-free graphs, even-hole-free graphs, odd-hole-free graphs, graphs that do not contain cycles with a unique chord, ISK4-free graphs, etc.).

Most generally all of the above mentioned classes of objects can be viewed as hereditary graph classes, i.e. classes of graphs closed under taking induced subgraphs. We say that a graph G *contains* a graph F, if F is isomorphic to an induced subgraph of G, and it is F*-free* if it does not contain F. For a family of graphs $\mathcal{F}$ we say that G is $\mathcal{F}$*-free* if G is F-free for every $F \in \mathcal{F}$. So for every hereditary graph class $\mathcal{C}$ there is a family $\mathcal{F}$ of graphs such that $\mathcal{C}$ is precisely the set of graphs that are $\mathcal{F}$-free.

Throughout the paper all graphs are finite and simple. A *hole* in a graph is an induced cycle of length at least 4, and it is *even* or *odd* depending on the parity of its length. A *clique* is a graph in which every pair of nodes are adjacent. A *stable set* (or *independent set*) S in a graph G is a subset of the vertex set of G such that no pair of vertices of S are adjacent. For $S \subseteq V(G)$, $G[S]$ denotes the subgraph of G induced by S, $N(S)$ denotes the set of nodes in $V(G) \setminus S$ with at least one neighbor in S, and $N[S] = N(S) \cup S$.

The remainder of this survey is organised in the following sections.

2 Truemper's Theorem

Theorem 2.1 (Truemper [121]) *Let β be a $\{0,1\}$ vector whose entries are in one-to-one correspondence with the chordless cycles of a graph G. Then there exists a subset F of the edge set of G such that $|F \cap C| \equiv \beta_C \pmod 2$ for all chordless cycles C of G, if and only if for every induced subgraph G' of G that is a Truemper configuration or K_4 (see Figure 1), there exists a subset F' of the edge set of G' such that $|F' \cap C| \equiv \beta_C \pmod 2$, for all chordless cycles C of G'.*

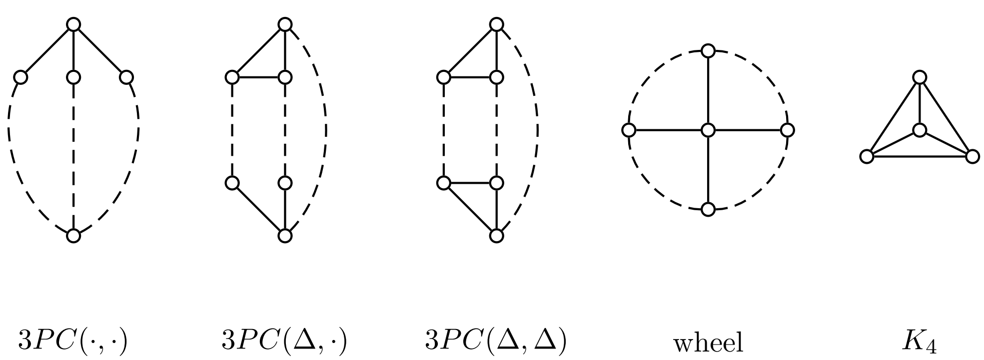

Figure 1: Truemper configurations and K_4

Truemper configurations are depicted in Figure 1, where a solid line denotes an edge and a dashed line denotes a chordless path containing one or more edges. We now define these configurations.

The first three configurations in Figure 1 are referred to as *3-path configurations* ($3PC$'s). They are structures induced by three paths P_1, P_2 and P_3, in such a way that the nodes of $P_i \cup P_j$, $i \neq j$, induce a hole. More specifically, a $3PC(x, y)$ is a structure induced by three paths that connect two nonadjacent nodes x and y; a $3PC(x_1x_2x_3, y)$, where $x_1x_2x_3$ is a triangle, is a structure induced by three paths having endnodes x_1, x_2 and x_3 respectively and a common endnode y; a $3PC(x_1x_2x_3, y_1y_2y_3)$, where $x_1x_2x_3$ and $y_1y_2y_3$ are two node-disjoint triangles, is a structure induced by three paths P_1, P_2 and P_3 such that, for $i = 1, 2, 3$, path P_i has endnodes x_i and y_i. We say that a graph G contains a $3PC(\cdot,\cdot)$ if it contains a $3PC(x, y)$ for some $x, y \in V(G)$, a $3PC(\Delta,\cdot)$ if it contains a $3PC(x_1x_2x_3, y)$ for some $x_1, x_2, x_3, y \in V(G)$, and it contains a $3PC(\Delta,\Delta)$ if it contains a $3PC(x_1x_2x_3, y_1y_2y_3)$ for some $x_1, x_2, x_3, y_1, y_2, y_3 \in V(G)$. Note that the condition that nodes of $P_i \cup P_j$, $i \neq j$, must induce a hole, implies that all paths of a $3PC(\cdot,\cdot)$ have length greater than one, and at most one path of a $3PC(\Delta,\cdot)$ has length one. In literature $3PC(\cdot,\cdot)$ is also referred to as *theta* [23], $3PC(\Delta,\cdot)$ as *pyramid* [22], and $3PC(\Delta,\Delta)$ as *prism* [23].

A *wheel* (H, x) consist of a hole H, called the *rim*, and a node x, called the *center*, that has at least three neighbors on H. Finally, a K_4 is a clique on four vertices. We note that in [121] K_4's are also referred to as wheels, but in this paper we choose to separate these two structures. In this survey we will refer to 3-path-configurations and wheels as *Truemper configurations.*

Truemper's interest in this theorem at the time was to obtain a co-NP characterization of balanceable matrices, that are a generalization of regular matrices. An alternative simple proof of Theorem 2.1 is given by Conforti, Gerards and Kapoor in [52], where they also give some of its consequences, such as an easy way to obtain Tutte's characterization of regular matrices.

2.1 Recognizing Truemper configurations

A natural question to ask is whether Truemper configurations can be recognized in polynomial time. These questions in fact arose when people were studying how to construct polynomial time recognition algorithms for even-hole-free graphs and perfect graphs. Observe that if a graph contains a $3PC(\Delta,\Delta)$ or a $3PC(\cdot,\cdot)$ then it must contain an even hole, and if it contains a $3PC(\Delta\cdot)$ then it must contain an odd hole. Even-hole-free graphs and perfect graphs are further discussed in Section 9. We now briefly describe different general techniques that were developed when trying to recognize whether a graph contains a particular Truemper configuration.

In [22] it is shown that detecting whether a graph contains a $3PC(\Delta,\cdot)$ can be done in $\mathcal{O}(n^9)$ time. This algorithm is based on the *shortest-paths detector* technique developed by Chudnovsky and Seymour. The idea of their algorithm is as follows. If G has a $3PC(\Delta,\cdot)$, then it has a $\Sigma = 3PC(\Delta,\cdot)$ with fewest number of nodes. The algorithm "guesses" some vertices of Σ, and then finds shortest paths in G between the guessed vertices that are joined by a path in Σ. If the graph induced by the union of these paths is a $3PC(\Delta,\cdot)$, then clearly G contains a $3PC(\Delta,\cdot)$. If it is not, then it turns out that G is $3PC(\Delta,\cdot)$-free.

Chudnovsky and Seymour [29] show that detecting whether a graph contains a $3PC(\cdot,\cdot)$ can be done in $\mathcal{O}(n^{11})$ time. For this detection problem, the shortest-paths detector technique does not work. The detection of $3PC(\cdot,\cdot)$'s relies on being able to solve a more general problem called the *three-in-a-tree problem* defined as follows: given a graph G and three specified vertices a, b and c, the question is whether G contains a tree that passes through a, b and c. It is shown in [29] that this problem can be solved in $\mathcal{O}(n^4)$ time. What is interesting is that the algorithm for the three-in-a-tree problem is based on an explicit construction of the cases when the desired tree does not exist, and that this construction can be directly converted into the algorithm. As we shall see in this survey, this direct connection between structure and algorithm does not occur so frequently for graph classes closed under taking induced subgraphs. The three-in-a-tree algorithm is quite general, and can be used to solve different detection problems, including the detection of a $3PC(\cdot,\cdot)$, and a $3PC(\Delta,\cdot)$ (this time in $\mathcal{O}(n^{10})$ time).

It turns out that detecting whether a graph contains a $3PC(\Delta,\Delta)$ is NP-complete, as shown by Maffray and Trotignon [94]. Detecting whether a graph contains a wheel remains an open problem.

A number of related detection problems will be looked at throughout this survey. The reader is also referred to [86] for more on detection of induced subgraphs problems.

3 The decomposition method

In the past few decades a number of important results were obtained through the use of decomposition theory, such as a polynomial time recognition algorithm for regular matroids [115] and the proof of the Strong Perfect Graph Conjecture (SPGC) [26] (discussed further in Sections 4 and 9). The power of decomposition is that it allows us to understand complex structures by breaking them down into simpler ones. Once these simpler structures are understood, this knowledge is propagated back to the original structure by understanding how their composition behaves. Decomposition is a general concept that applies to different classes of objects. Here we start by introducing the method in the context of graphs.

In a connected graph G, a subset S of nodes and/or edges is a *cutset* if its removal disconnects G. If S consists only of nodes then it is referred to as a *node cutset*, and if it consists only of edges then it is referred to as an *edge cutset*. A decomposition theorem for a class of graphs $\mathcal{C}$ is of the following form.

Decomposition Theorem: *If G belongs to $\mathcal{C}$ then G is either "basic" or G has a cutset S for $S \in \mathcal{S}$.*

Depending on what one wants to prove about the class of graphs $\mathcal{C}$ using the Decomposition Theorem, "basic" graphs and cutsets in $\mathcal{S}$ have to have adequate properties. For example, the SPGC was proved using the decomposition theorem for Berge graphs [26], by ensuring that "basic" graphs were simple in the sense that the SPGC could be easily proved for them directly, and the cutsets in $\mathcal{S}$ had the property that no minimal imperfect graph could contain them (or if it did it would have to be an odd hole or an odd antihole).

To use a decomposition theorem to recognize a class of graphs $\mathcal{C}$, "basic" graphs need to be simple in the sense that they can be easily recognized, and the cutsets $S \in \mathcal{S}$ need to have the following property. The removal of a cutset S from a graph G disconnects G into two or more connected components. From these components *blocks of decomposition* are constructed by adding some more nodes and edges. A decomposition is *$\mathcal{C}$-preserving* if it satisfies the following: G belongs to $\mathcal{C}$ if and only if all the blocks of decomposition belong to $\mathcal{C}$. A recognition algorithm takes a graph G as input and decomposes it using $\mathcal{C}$-preserving decompositions into undecomposable blocks, which are then checked whether they belong to $\mathcal{C}$ (which according to the decomposition theorem reduces to checking whether they are basic). The decomposition can be represented with a *decomposition tree* T, whose root is the input graph, and for every non-leaf node H of T, its children in T are the blocks of decomposition of H. In order for such an algorithm to have polynomial complexity we need to ensure that T can be constructed in polynomial time (which in particular means that we can find the cutsets in polynomial time and that we can ensure that the decomposition tree is polynomial in size) and that checking whether a graph is basic can be done in polynomial time.

This is an ideal scenario, which works, for example, for obtaining a recognition algorithm for regular matroids [115]. On the other hand, it does not work, for example, for obtaining a recognition algorithm for perfect graphs. The problem is that for the cutsets from the decomposition theorem in [26], one does not know how to construct the blocks of decomposition that would, at the same time, be class-preserving as well as guarantee polynomiality of the decomposition tree. This problem was first encountered when trying to construct a polynomial time recognition algorithm for balanced matrices. At that time a technique called "cleaning" (i.e. preprocessing the input graph, so that later when the decomposition is applied it would be class-preserving) was developed by Conforti and Rao [54] that enabled them to recognize, in polynomial time, linear balanced matrices. This technique was further developed and used in obtaining decomposition based polynomial time recognition algorithms for balanced matrices [47], balanced $0, \pm 1$ matrices [44], even-hole-free graphs [46, 62], and it was the key to obtaining a recognition algorithm for perfect (in fact Berge) graphs [22].

Decomposition can also be used to construct optimization algorithms. The general paradigm would be as follows: given a decomposition tree T for a graph G obtained by using S-decompositions, for $S \in \mathcal{S}$ (referring to the general decomposition theorem stated above), with the property that for every leaf L of T one can solve an optimization problem (such as coloring or finding the size of the largest clique or a stable set), can we construct an algorithm to solve the problem on G? This general paradigm sometimes works nicely, but most of the time it is difficult to apply to classes whose decomposition theorems use "powerful cutsets".

We next illustrate the ideal scenarios, discussed above, for using a decomposition theorem for constructing a recognition algorithm as well as for obtaining combinatorial optimization algorithms, on the class of triangulated graphs. We close this section by introducing some cutsets that commonly appear in the decomposition theorems we will discuss in this survey.

3.1 Triangulated graphs

A graph is *triangulated* (or *chordal* or *hole-free*) if it does not contain a hole. On this class we will illustrate different techniques for obtaining recognition and combinatorial optimization algorithms. Although, as we shall see, there are more efficient methods for obtaining algorithms for triangulated graphs, we will start by describing the use of the general decomposition method, because it is ideally illustrated on this class and it generalizes to other more complex classes of objects. First, it is simple to obtain the following decomposition theorem for triangulated graphs. A node set $S \subseteq V(G)$ is a *clique cutset* of G, if S is a node cutset of G and it induces a clique in G.

Theorem 3.1 (Dirac [65]) *If G is a connected triangulated graph, then G is either a clique or G has a clique cutset.*

Proof Suppose that G is not a clique. Then G clearly has a node cutset. Let S be a minimal node cutset of G, and let C_1 and C_2 be two connected components of $G \setminus S$. Suppose S is not a clique and let u and v be two non-adjacent vertices of S. Since S is minimal, both u and v have a neighbor in both C_1 and C_2. Hence, for $i = 1, 2$, there exists a chordless path P_i from u to v whose interior vertices belong to C_i. But then $P_1 \cup P_2$ induces a hole, a contradiction. □

Let S be a clique cutset of a graph G, and let $C_1, \ldots, C_k$ be the connected components of $G \setminus S$. We define the *blocks of decomposition by a clique cutset S* to be graphs $G_i = G[C_i \cup S]$, for $i = 1, \ldots, k$. It is now easy to see that this definition of blocks is class-preserving for the class of triangulated graphs.

Theorem 3.2 *G is triangulated if and only if all the blocks of decomposition by a clique cutset are triangulated.*

Proof Since the blocks of decomposition are all induced subgraphs of G, if G is triangulated then so are all the blocks. Now suppose that all the blocks $G_1, \ldots, G_k$ are triangulated, but that G contains a hole H. Since H cannot be contained in any of the blocks, it must contain nodes of at least two connected components $C_1, \ldots, C_k$. Consequently H contains at least two nodes of S that are not consecutive on H, which contradicts the assumption that S is a clique. □

Theorems 3.1 and 3.2 actually give us a complete structure theorem for the class of triangulated graphs, i.e. they show how (connected) triangulated graphs can be built starting from cliques, gluing them together through cliques (clique composition), and all graphs built this way are triangulated. Such structure theorems are stronger than the usual decomposition theorems, and are quite rare for classes of graphs closed under taking induced subgraphs.

We now turn to using a decomposition theorem to construct algorithms. We construct a *decomposition tree* T using clique cutsets as follows: the root of T is our input graph G; for every internal node G' of T, the children of G' are the blocks of decomposition of G' by some clique cutset; and the leaves of T are graphs that have no clique cutset. An $\mathcal{O}(nm)$ algorithm is given in [126] for finding a clique cutset in a graph, and a simple counting argument shows that the number of nodes in T is bounded by $\mathcal{O}(n^2)$, giving an $\mathcal{O}(n^3m)$ algorithm for constructing T. As we shall see one can actually do better than that.

First observe that, by Theorem 3.1 and Theorem 3.2, the input graph G is triangulated if and only if all the leaves of T are cliques. One can now recognize triangulated graphs (in the same time it takes to construct T) as follows: construct a decomposition tree T using clique cutsets, check whether all the leaves of T are cliques, if yes then G is triangulated, and otherwise it is not.

Clique cutsets have another interesting property, that is quite useful for constructing algorithms. We say that S is an *extreme clique cutset* if for some i, the block of decomposition $G_i = G[C_i \cup S]$ has no clique cutset. We say that G_i is an *extreme block*. It turns out that every graph that has a clique cutset has an extreme clique cutset. This is a very useful property, that not many types of cutsets have.

Lemma 3.3 *If a graph G has a clique cutset, then it has an extreme clique cutset.*

Proof Let S be a clique cutset of G such that out of all clique cutsets of G a connected component C of $G \setminus S$ is smallest possible. Suppose that $G' = G[C \cup S]$ has a clique cutset S'. Since S is a clique, there is a connected component C' of $G' \setminus S'$ such that $C' \cap S = \emptyset$. Clearly, $S' \cap C \neq \emptyset$, and hence C' is a proper subset of C. In particular $|C'| < |C|$. Also C' is a connected component of $G \setminus S'$, contradicting our choice of S and C. So G' has no clique cutset, and hence S is an extreme clique cutset of G. □

We will now use extreme clique cutsets to decompose. Suppose that S is an extreme clique cutset with G_i being an extreme block. This time we will construct only two blocks of decomposition: $G_B = G_i = G[C_i \cup S]$ and $G_A = G \setminus C_i$. We now construct an *extreme decomposition tree* T using clique cutsets as follows: the root of T is our input graph G; for every internal node G' of T, the children of G' are the blocks of decomposition G'_A and G'_B of G' by some extreme clique cutset; and the leaves of T are graphs that have no clique cutset. Note that every G'_B is a leaf, so T is a binary tree in which every internal node has a child that is a leaf.

It turns out that such an extreme decomposition tree using clique cutsets can be built in $\mathcal{O}(nm)$ time [117]. This relies on being able to find a particular ordering of vertices, called a minimal elimination ordering, in $\mathcal{O}(nm)$ time, which is done in [112] using lexicographic breadth-first search (Lex-BFS).

For a graph G, let T be an associated extreme decomposition tree using clique cutsets, and let $L_1, \ldots, L_t$ be the leaves of T. We now consider how T can be used to construct combinatorial optimization algorithms for maximum weight clique, vertex coloring and maximum weight independent set problems, assuming that these problems can be efficiently solved on the leaves of T (see [117]). For any graph G, let $\omega(G)$ denote the weight of a maximum weighted clique of G, $\chi(G)$ the chromatic number of G, and $\alpha(G)$ the weight of a maximum weighted stable set of G.

Since any clique of G is contained in one of the blocks of decomposition by a clique cutset, it follows that $\omega(G) = \max\{\omega(L_1), \ldots, \omega(L_t)\}$. And hence the problem of finding a maximum weight clique reduces to doing it on the leaves. Similarly, the coloring problem reduces to coloring the leaves, since any k-colorings of the blocks of decomposition by a clique cutset S can be combined into a k-coloring of the graph by renaming the colors in the blocks so that they agree on S. In particular $\chi(G) = \max\{\chi(L_1), \ldots, \chi(L_t)\}$. For both of these problems it is not essential that T is an extreme decomposition tree, but it gives a better time bounds if it is.

For solving the maximum weight independent set problem in polynomial time (assuming this is possible to do on the leaves of T) it actually does matter that T is an extreme decomposition tree. Let H be an interior node of T and let H_A and H_B be its children (where H_B is a leaf of T), obtained by decomposing H with clique cutset S. Let w be the weight function defined on the nodes of H. For every $u \in S$ redefine the weight of u in H_A to be $w(u) + \alpha(H[V(H_B) \setminus N_{H_B}(u)]) - \alpha(H_B \setminus S)$. Let H'_A be the resulting weighted graph. Then it is easy to see that $\alpha(H) = \alpha(H'_A) + \alpha(H_B \setminus S)$. So the independent set problem for H reduces to recursively solving the independent set problem on block H'_A (with newly defined weights). Note that computing the weights for H'_A and computing $\alpha(H_B \setminus S)$ amounts to solving $|S|+1$ independent set problems on H_B. Since H_B is a leaf this is not a problem since it requires no further recursion, but if we were not using an extreme decomposition tree this method could lead to an exponential explosion.

Note that if the input graph G is triangulated, then all the leaves of T are cliques, and hence all of the above mentioned problems can be solved on G in the same time it takes to construct T.

Triangulated graphs are in fact characterized by having very special types of minimal elimination orderings that can be found more efficiently. A *perfect elimination ordering* is an ordering of vertices $v_1, \dots, v_n$ such that v_i is simplicial (where a vertex is *simplicial* if its neighborhood induces a clique) in $G[v_i, \dots, v_n]$.

Theorem 3.4 (Dirac [65]) *G is triangulated if and only if G has a perfect elimination ordering.*

Proof Suppose G has a perfect elimination ordering $v_1, \dots, v_n$, but is not triangulated. Let H be a hole of G, and let v_i be a smallest indexed vertex of H. Then clearly v_i has two nonadjacent neighbors in $G[v_i, \dots, v_n]$, a contradiction. To prove the converse, assume G is triangulated. If G is a clique then any ordering of vertices is a perfect elimination ordering. Otherwise by Theorem 3.1, G has a clique cutset, and by Lemma 3.3 it has an extreme clique cutset S. So for some connected component C of $G \setminus S$, $G' = G[C \cup S]$ has no clique cutset. By Theorem 3.1, G' is a clique, and hence any vertex $u \in C$ is simplicial in G' and hence in G as well. Let $u = v_1$ and inductively construct the remainder of a perfect elimination ordering.
$\square$

Note that if $v_1, \dots, v_n$ is a perfect elimination ordering of G and G is not a clique, then $N(v_1)$ is an extreme clique cutset of G separating a single vertex v_1 from the rest of the graph.

In [112] it is shown that a perfect elimination ordering of a triangulated graph can be found in linear time using Lex-BFS, and more generally, by using Lex-BFS to construct this particular ordering and checking that in fact the ordering constructed is a perfect elimination ordering one gets a linear time recognition algorithm for triangulated graphs. It follows that all of the above mentioned optimization problems can be solved in linear time for triangulated graphs. Note that triangulated graphs can also be optimally colored, in linear time, by applying a greedy coloring algorithm to the vertices in the reverse of a perfect elimination ordering.

3.2 Common cutsets

Here we introduce some cutsets that commonly appear in decompositions of graph classes closed under taking induced subgraphs.

We start by introducing several edge cutsets. First observe that a disconnected graph can be defined as a graph that has a partition (X_1, X_2) of its vertex set satisfying: there are no edges between X_1 and X_2; and for $i = 1, 2$, $|X_i| \geq 1$. This concept can be generalized by controlling the kinds of edges that go across the partition.

A partition (X_1, X_2) of the vertex set of a graph G is a *general join* if, for $i = 1, 2$, there exist disjoint $A_i, B_i, C_i \subseteq X_i$ satisfying the following: every vertex of A_1 is adjacent to every vertex of A_2, every vertex of B_1 is adjacent to every vertex of B_2, every vertex of C_1 is adjacent to every vertex of $A_2 \cup B_2$, every vertex of C_2 is adjacent to every vertex of $A_1 \cup B_1$, and there are no other edges between X_1 and X_2. Sets X_1 and X_2 are the two *sides* of the general join. We say that $(X_1, X_2, A_1, B_1, C_1, A_2, B_2, C_2)$ is a *split* of a general join (X_1, X_2). For $i = 1, 2$ sets A_i, B_i, C_i are called the *special sets* of general join (X_1, X_2).

A *general k-join*, for $k = 0, 1$, is a general join with split $(X_1, X_2, A_1, B_1, C_1, A_2, B_2, C_2)$ such that for $i = 1, 2$ exactly k of the sets A_i, B_i, C_i are nonempty, there are at least k edges going from X_1 to X_2, and $|X_i| \geq k+1$. A *general 2-join* is a general join with split $(X_1, X_2, A_1, B_1, C_1, A_2, B_2, C_2)$ such that for $i = 1, 2$ at least 2 of the sets A_i, B_i, C_i are nonempty, and $|X_i|$ is greater than the number of nonempty sets among A_i, B_i, C_i. A general 2-join was first introduced in [41]. General joins generalize some of the previously introduced edge cutesets.

A general 0-join corresponds to a disconnected graph. A general 1-join is exactly the *1-join* (or *join* or *split decomposition*) as introduced by Cunningham and Edmonds [59]. A related notion is that of a *homogeneous set* (or *module*) of a graph G, that is a proper subset S of $V(G)$ of at least two vertices such that every vertex not in S is adjacent to either all or none of the vertices in S. Note that if $V(G) \setminus S \geq 2$, then homogeneous set corresponds to a 0-join, or 1-join with split $(X_1, X_2, A_1, \emptyset, \emptyset, A_2, \emptyset, \emptyset)$ such that $X_1 \setminus A_1 = \emptyset$.

A general 2-join with $C_1 = C_2 = \emptyset$ and all the other special sets nonempty is called a *2-join* and it was first introduced by Cornuéjols and Cunningham [56]. A general 2-join with $B_1 = C_2 = \emptyset$ (or equivalently $A_1 = A_2 = \emptyset$) and all the other special sets nonempty is called a *N-join*. A general 2-join with $C_2 = \emptyset$ (or equivalently $B_2 = \emptyset$) and all the other special sets nonempty is called a *M-join*. A general 2-join with all special sets nonempty is called a *6-join* and it was first introduced by Conforti, Cornuéjols, Kapoor and Vušković [44].

General joins also generalize the notion of a *homogeneous pair* introduced by Chvátal and Sbihi [39]. A homogeneous pair in a graph G is a pair $\{Q_1, Q_2\}$ of disjoint sets of vertices of G such that: every vertex of $V(G) \setminus (Q_1 \cup Q_2)$ is adjacent to either all vertices of Q_1 or to no vertex of Q_1; every vertex of $V(G) \setminus (Q_1 \cup Q_2)$ is adjacent to either all vertices of Q_2 or to no vertex of Q_2; $|Q_1| \geq 2$ or $|Q_2| \geq 2$; and $|V(G) \setminus (Q_1 \cup Q_2)| \geq 2$. Note that a homogeneous pair in a graph with no homogeneous set is a special case of a general 2-join, where $A_1, B_1 \neq \emptyset$ and $X_1 \setminus (A_1 \cup B_1) = \emptyset$.

Furthermore, there is a correspondence between k-separations, $k = 1, 2, 3$, in binary matroids and general joins. A 1-separations corresponds to a general 0-join, a 2-separation corresponds to a general 1-join and a 3-separation corresponds to a general 2-join. This correspondence is discussed in Section 4.

We now consider some commonly appearing node cutsets. Let S be a node cutset of a graph G. S is a *k-node cutset* if $|S| = k$. We say that S is a *small node cutset* if $|S|$ is bounded by some fixed integer k. Recall that S is a *clique cutset* if S induces a clique in G.

A node set $S \subseteq V(G)$ is a *k-star* if S is comprised of a clique C (the *clique center* of S) of size k and nodes with at least one neighbor in C, so $S \subseteq N[C]$. A *k-star cutset* is a k-star S that is a node cutset. A 1-star cutset is also referred to as a *star cutset*, a 2-star cutset as a *double star cutset*, and a 3-star cutset as a *triple star cutset*.

Here is another generalization of a star cutset, that is a special case of a double star cutset. A node cutset S is a *skew cutset* if there exists a partition (S_1, S_2) of S such that every node of S_1 is adjacent to every node of S_2. Star cutsets and skew cutsets were first introduced by Chvátal [38].

In trying to understand why these cutsets appear "naturally" in decomposition theorems, we first observe that with clique cutsets one can only separate vertices that are not contained in a hole. When we need to break a hole, we can either use a node that has neighbors on this hole as a center of a star cutset (or more generally a k-star cutset), or when no such node exists we can hope for example that two edges of this hole will extend to a 2-join that separates the hole.

4 Regular and balanced matrices

A matrix is *totally unimodular* if every square submatrix has determinant equal to $0, \pm 1$. In particular, all entries of a totally unimodular matrix are $0, \pm 1$. A $0,1$ matrix is *balanced* if it does not contain a square submatrix of odd order with two 1's per row and per column. This notion was introduced by Berge [6], and it was extended to $0, \pm 1$ matrices by Truemper [121]. A $0, \pm 1$ matrix is *balanced* if, in every square submatrix with exactly two nonzero entries per row and column, the sum of the entries is a multiple of 4. Note that the class of $0, \pm 1$ balanced matrices properly includes totally unimodular matrices. All these matrices have important polyhedral properties, see for example [55]. In this section we describe decomposition theorems that were the key to obtaining polynomial time recognition algorithms for all these classes of matrices.

4.1 Decomposition of regular matroids

What enabled the structural understanding of totally unimodular matrices, which led to their polynomial time recognition, was the translation of the property into the realm of matroids, and the use of existent powerful tools from matroid theory.

A $0,1$ matrix is *regular* if its nonzero entries can be signed $+1$ or -1 so that the resulting matrix is totally unimodular. Camion [10] observed that this signing is unique up to multiplying rows and columns by -1, and gave a simple signing algorithm, from which it follows that the recognition of totally unimodular matrices reduces to the recognition of regular matrices. This shift to regular matrices allows for the focus on the structure of the pattern of zero/nonzero entries.

Let M be a binary matroid and $X \subseteq V(M)$ a base of M. The *partial representation* of M with respect to X is the 0,1 matrix $A(M)$ with rows indexed by the elements of X, columns indexed by the elements of $Y = V(M) \setminus X$, and $a_{xy} = 1$ if and only if x belongs to the unique circuit contained in $X \cup \{y\}$. Note that if $A(M)$ is a partial representation of a binary matroid M, then $(I, A(M))$ is a binary representation of M.

A binary matroid is *regular* if all of its partial representation matrices are regular. Let A be a partial representation matrix of a binary matroid, i.e. rows of A are indexed by a base of the matroid. One can always go from one partial representation of a binary matroid to another by using GF(2)-pivoting and row and column permutations. Pivoting over GF(2) consist in replacing $A = \begin{pmatrix} 1 & y \\ x & D \end{pmatrix}$ by $\tilde{A} = \begin{pmatrix} 1 & y \\ x & D+xy \end{pmatrix}$. It can be shown that if A is a regular matrix, then so is $\tilde{A}$, and hence it is the case that for a binary matroid either all or none of its partial representation matrices are regular.

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

Figure 2: Partial representation matrix of the Fano matroid.

The matrix in Figure 2 is not regular, and is a partial representation matrix of the *Fano matroid* F_7. The transpose of this matrix is a partial representation of the dual of the Fano matroid F_7^*. Let M be a binary matroid and A a partial representation of M. Any submatrix of A is a partial representation of a binary matroid M'. A matroid M' obtained from M in this way is a *minor* of M. A convenient way to work with regular matroids is provided by the following excluded minors characterization.

Theorem 4.1 (Tutte [124]) *A binary matroid is regular if and only if it has no minor isomorphic to F_7 or F_7^*.*

What enabled the polynomial time recognition of regular matroids, and hence totally unimodular matrices, is the following decomposition theorem. Let M be a matroid defined by a finite ground set $V(M)$ and a family $E(M)$ of subsets of $V(M)$ that are the independent sets of M. The *rank* $r(U)$ of a set $U \subseteq V(M)$ is the maximum cardinality of an independent set contained in U. A *k-separation* of M is a partition (U_1, U_2) of $V(M)$ such that $|U_1| \geq k$, $|U_2| \geq k$ and $r(U_1) + r(U_2) \leq r(V(M)) + k - 1$.

Theorem 4.2 (Seymour [115]) *A regular matroid is either graphic, cographic or R_{10} (a certain 10-element matroid), or it has a k-separation, for $k = 1, 2, 3$.*

This theorem leads to a decomposition based polynomial time recognition algorithm for regular matroids in the following way. First of all, 1-, 2- and 3-separations can be found in polynomial time (see [123] Section 8.4). For 1-, 2- and 3-separation, blocks of decomposition can be constructed that are regularity-preserving and lead to a linear size of the decomposition tree (see for example [55]). Finally, by Theorem 4.2, it just remains to check whether the leaves of this decomposition tree are R_{10}, graphic or cographic matroids, which can be done in polynomial time (see for example [123] Section 10.6). By Camion's signing algorithm it follows that totally unimodular matrices can be recognized in polynomial time. We note that before Seymour's decomposition approach no polynomial time recognition algorithm for totally unimodular matrices was known. The fastest known algorithm for testing total unimodularity is the $\mathcal{O}(n+m)^3$ algorithm of Truemper [122], where the input is an $n \times m$ real matrix. The algorithm uses Seymour's decomposition theorem, but does not blindly search for 3-separations as described above. Instead it searches for 3-separations by starting with particular minors that have 3-separation that should extend to the entire matroid.

This decomposition approach was later extended to recognition algorithms for other classes of matrices and graphs (such as balanced matrices and perfect graphs), but as we shall see, with many more complications. To relate these results, we close this section by translating the work described above into graphs.

Let A be a $0,1$ matrix. A can be thought of as a node-node incidence matrix of a bipartite graph, which we denote with $G(A)$ and call the *bipartite graph representation of* A. We say that a bipartite graph $G(A)$ is *regular* if A is regular. Pivoting on an entry a_{ij} of A corresponds to the following operation on $G(A)$: let ij be the edge of $G(A)$ that corresponds to the pivot element, then $G(\tilde{A})$ is obtained from $G(A)$ by complementing the edges between $N(i) \setminus \{j\}$ and $N(j) \setminus \{i\}$. We refer to this operation as *pivoting on the edge* ij. Note that the bipartite representation of the matrix in Figure 2 is a wheel whose rim is of length 6 (in particular, the center of the wheel has three neighbors on the rim and they are all on the same side of the bipartition). Let us call this wheel *Fano wheel.* Theorem 4.1 now translates into the following:

A bipartite graph is regular if and only if it cannot be transformed into a Fano wheel by a sequence of edge pivots and/or node deletions.

Let G be a bipartite graph. Since G is bipartite, the only Truemper configurations that G can possibly have are $3PC(\cdot,\cdot)$'s and wheels. Let (H,x) be a wheel. Suppose that x has more than 3 neighbors on H. It can easily be seen that if we pivot on an edge xx_i, where x_i is a neighbor of x on H, we get a wheel (H',x) such that x has fewer neighbors on H' than on H. Now suppose that a sector S of (H,x) is of length greater than 2, and let uv be an interior edge of that sector. If we pivot on uv we get a wheel (H',x) that has all the sectors of (H,x) except for S, and the sector S' of (H',x) that corresponds to S in (H,x) is shorter than S. So clearly, a wheel (H,x) such that x has an odd number of neighbors on H, can be transformed into a Fano wheel by a sequence of edge pivots and node deletions. Let us call such a wheel in a bipartite graph an *odd bipartite wheel.* Similarly it can be seen that a $3PC(u,v)$ where u and v are on opposite sides of the bipartition, can be transformed into a Fano wheel. Let us call such a $3PC(u,v)$ a *3-odd-path configuration.* Therefore regular bipartite graphs cannot contain odd bipartite wheels nor 3-odd-path configurations (as well as all the other configurations that can be transformed into a Fano wheel). In other words, out of all the Truemper configurations, regular bipartite graphs may contain only wheels (H,x) such that x has an even number of neighbors on H, and $3PC(u,v)$'s such that u and v are on the same side of the bipartition.

Let M be a binary matroid, and consider a k-separation (U_1,U_2) where $r(U_1)+$

$r(U_2) = r(|V(M)|) + k - 1$. Let X_2 be a maximal independent subset of U_2, and enlarge X_2 by subset X_1 of U_1 to a base of M. The partial representation matrix A of M w.r.t. base $X_1 \cup X_2$ is $A = \begin{array}{c} X_1 \\ X_2 \end{array} \begin{pmatrix} A_1 & 0 \\ D & A_2 \end{pmatrix}$ where the sum of rows and columns of A_i is at least k, for $i = 1, 2$, and the rank of D over GF(2) is $k - 1$. Observe that when $k = 1$, then $D = 0$, and hence $G(A)$ corresponds to a disconnected graph, or a general 0-join. By similarly analyzing the possibilities for matrix D, it turns out that a 2-separation corresponds to a general 1-join in $G(A)$, and a 3-separation corresponds to a general 2-join in $G(A)$. (As mentioned in Section 3.2 different forms of general joins were introduced by different authors, interestingly without being aware of this correlation. They were notions that emerged naturally when dealing with different graph classes.) Therefore, Theorem 4.2 translated states that *regular bipartite graphs can be decomposed by general k-joins, for $k = 0, 1, 2$.*

4.2 Decomposition of balanced matrices

We immediately switch from $0, 1$ matrices to their bipartite graph representations. So a bipartite graph is *balanced* if it does not contain a hole of length 2 (mod 4). A *signed bipartite graph* is a bipartite graph with edge weights $+1$ and -1. A signed bipartite graph is *balanced*, if it does not contain a hole of weight 2 (mod 4). A bipartite graph is *balanceable* if there exists a signing of its edges so that the resulting signed bipartite graph is balanced.

If a graph is a balanceable bipartite graph, there exists an easy signing algorithm that signs it into a balanced signed bipartite graph (since if such a signing exists, it is essentially unique and easy to find by Camion's signing algorithm [10], see also [50]). So the recognition of signed balanced bipartite graphs reduces to the recognition of balanceable bipartite graphs.

Clearly the class of balanceable bipartite graphs is closed under taking induced subgraphs, but it is not closed under edge pivoting. Consider for example a graph G that consists of a $3PC(x, y)$ where all the paths have length 3 together with an edge xy. This graph is balanceable, but if we pivot on the middle edge of any of the paths, edge xy disappears and we get the original 3-odd-path configuration $3PC(x, y)$, which is not balanceable (since no matter how we sign its edges two of the paths will have weights that are congruent (mod 4) and would hence induce a hole of weight 2 (mod 4)). Observe that it also follows that G is not regular. So balanceable bipartite graphs properly contain regular bipartite graphs.

The following theorem characterizes balanceable bipartite graphs in terms of excluded induced subgraphs, and provides a convenient way to work with this class.

Theorem 4.3 (Truemper [121]) *A bipartite graph is balanceable if and only if it does not contain an odd bipartite wheel nor a 3-odd-path configuration.*

The first known polynomial time recognition algorithm for balanced matrices (or equivalently, balanced bipartite graphs) is given by Conforti, Cornuéjols and Rao [47], and it is based on the following decomposition theorem.

Theorem 4.4 (Conforti, Cornuéjols and Rao [47]) *If a bipartite graph is balanced but not totally unimodular, then it has a double star cutset.*

These results were later extended to balanceable bipartite graphs. The first known polynomial time recognition algorithm for balanceable bipartite graphs is given by Conforti, Cornuéjols, Kapoor and Vušković [44], and it is based on the following decomposition theorem.

Theorem 4.5 (Conforti, Cornuéjols, Kapoor and Vušković [44]) *A connected balanceable bipartite graph is either strongly balanceable or R_{10} (a certain 10-element graph), or has a 2-join, a 6-join or a double star cutset.*

We observe that the 2-joins in the above theorem are in fact of a special type that we call connected non-path 2-joins and describe in Section 9. The major difficulty in using Theorems 4.4 and 4.5 to construct decomposition based recognition algorithms is the double star cutsets. For the 2-join and 6-join it is possible to construct blocks of decomposition that are balancedness-preserving and keep the decomposition tree polynomial in size (see [44]), but it is not clear how to do that for the double star cutset. The double star cutsets in Theorems 4.4 and 4.5 are actually more structured, but that does not help, the problem appears even when trying to use just the star cutsets. Consider for example an odd wheel (H, x) whose every sector is of length 2. This wheel can be decomposed with a star cutset $S = N[x]$. If we construct blocks of decomposition as we did for the clique cutset decomposition in Section 3.1, we get that all the blocks of decomposition are balanced (or balanceable), but (H, x) is not. (We observe that it was precisely for the decomposition of wheels that double star cutsets are needed in the proofs of Theorems 4.4 and 4.5.) One might add some more information to the blocks to make the decomposition balancedness-preserving, but then the decomposition tree blows up in size. To deal with this problem, a technique called *cleaning* was developed by Conforti and Rao [54], which enabled them to recognize linear balanced matrices in polynomial time. This technique was further developed and used in obtaining decomposition based polynomial time recognition algorithms for balanced matrices [47], balanced $0, \pm 1$ matrices [44], and a new level of cleaning had to be developed for recognition of even-hole-free graphs [46, 62], that was also used in the cleaning for recognition of perfect graphs [22].

We now describe the cleaning procedure in the context of its use for recognizing balanced bipartite graphs. A hole of length 2 (mod 4) is called an *unbalanced hole*. Given an input graph G, the cleaning procedure produces, in polynomial time, a clean graph G', such that G is balanced if and only if G' is balanced, and if G contains an unbalanced hole then G' contains a clean unbalanced hole (i.e. an unbalanced hole for which there are no nodes outside the hole that have problematic neighbors on the hole, which can be used as centers of star cutsets to break the hole). This is done by studying the structure of a smallest unbalanced hole in a graph, showing that such a hole contains a fixed number of nodes that see all the problematic neighbors of the hole, and using that information to remove them. Once we have a clean graph G', decomposition by (double) star cutsets can be applied safely, since it will now be balancedness-preserving, as well as lead to a polynomial decomposition tree.

Using Theorem 4.5 Chudnovsky and Seymour prove the following decomposition theorem for balanceable bipartite graphs, resolving a conjecture from [44]. We observe that this decomposition theorem does not help with the recognition algorithm, since the double star cutsets are still used.

Theorem 4.6 (Chudnovsky and Seymour [28]) *A balanceable bipartite graph that is not regular has a double star cutset.*

The following conjecture is the last unresolved conjecture about balanced (balanceable) bipartite graphs in Cornuéjols' book [55].

Conjecture 4.7 (Conforti and Rao [53]) *Every balanced bipartite graph contains an edge that is not the unique chord of a cycle.*

This conjecture was proved recently for linear balanced bipartite graphs and balanced bipartite graphs whose maximum degree is at most 3 in [4] using the idea of extreme decomposition (in fact in this paper the analogous form of this conjecture for balanceable bipartite graphs is proved for 4-hole-free balanceble bipartite graphs and subcubic balanceable bipartite graphs).

5 Classes closed under minor taking

In this section we briefly consider graph classes that are not only closed under deletion of vertices, but also under deletion and contraction of edges, i.e. classes of graphs that are closed under minor taking. Some important examples of such classes are cycle-free graphs (or forests), series-parallel graphs, planar graphs or more generally classes of graphs embeddable in any fixed surface.

A graph H is a *minor* of a graph G, if it is isomorphic to a graph that can be produced from G by a sequence of contracting edges, and deleting vertices and edges. A class of graphs $\mathcal{G}$ is *minor-closed*, if for every $G \in \mathcal{G}$, every minor of G also belongs to $\mathcal{G}$. Trivially, every minor-closed class of graphs can be characterized by a list of excluded minors, by just listing all the graphs that are not in the class. Wagner conjectured that this can always be done by a *finite* list of excluded minors. This famous conjecture was proved by Robertson and Seymour in their monumental work on revealing the structure of minor-closed families of graphs, with far reaching algorithmic consequences.

Theorem 5.1 (Robertson and Seymour [111]) *Every minor-closed class of graphs can be characterized by a finite family of excluded minors.*

The proof of this theorem is based on the following structural characterization: if a minor-closed class of graphs does not contain all graphs, then every graph in it is "glued" together in a tree-like fashion from graphs that can almost be embedded in a fixed surface. To be more specific we need to introduce the concept of tree-decomposition [108]. A *tree-decomposition* of a graph G is a pair (T, W), where T is a tree and $W = (W_t : t \in V(T))$ is such that:

(i) $\cup_{t \in V(T)} W_t = V(G)$, and every edge of G has both endnodes in some W_t, and

(ii) if $t, t', t'' \in V(T)$ and t' lies on the path from t to t'' in T, then $W_t \cap W_{t''} \subseteq W_{t'}$.

The *width* of (T, W) is the $\max\{|W_t| - 1 : t \in V(T)\}$, and the *tree-width* of G is the least integer k such that G has a tree-decomposition of width k.

Theorem 5.2 (Robertson and Seymour [109]) *For every planar graph H there is an integer $k > 0$ such that if a graph is H-minor-free, then its tree-width is at most k.*

In other words, if a graph does not contain some planar graph as a minor, then it has bounded tree-width, and hence it can be constructed from bounded sized graphs by "gluing" them together in a tree-like structure. In [110] an analogous construction is given for H-minor-free graphs in general, starting with graphs embedded in a connected closed surface with genus at most k, adding more nodes in a specified way, and "gluing" such pieces together in a tree-like fashion. This time the pieces that are "glued on" are not necessarily of bounded size, but the parts that are being glued over are.

This structural characterization leads to an $\mathcal{O}(n^3)$ algorithm to test whether a graph G is H-minor-free (although there is a constant factor that depends super-polynomially on the size of G). Together with Theorem 5.1 we get the following algorithmic consequence.

Theorem 5.3 *Every minor-closed class of graphs can be recognized in polynomial time.*

This theoretically beautiful result, has its practical shortcomings. Unless a minor-closed class of graphs is given by its finite list of excluded minors, from Theorem 5.1 we only get an existence of a polynomial time algorithm.

There are further algorithmic consequences for graph classes that have tree-decompositions of bounded tree-width, as is the case for example with any minor-closed family that does not include all planar graphs (by Theorem 5.2). Many problems that are NP-hard in general, such as the independent set problem or the coloring problem, can be solved by dynamic programming in linear time when the input graph has bounded tree-width. In fact, each problem that can be formulated in Monadic Second Order Logic can be solved in linear time on graphs of bounded tree-width [57].

In the terminology used in this survey, a tree-decomposition of width k corresponds to decomposing a graph into blocks of size at most $k + 1$ by a sequence of "non-crossing" node cutset decompositions, where the cutsets are all of size at most k. Indeed, let (T, W) be a tree-decomposition of a graph G, let t_1t_2 be an edge of T, and for $i = 1, 2$, let T_i be the subgraph of $G \setminus t_1t_2$ that contains t_i. Then $W_{t_1} \cap W_{t_2}$ is a cutset of G that separates $\cup_{t\in V(T_1)} \setminus (W_{t_1} \cap W_{t_2})$ from $\cup_{t\in V(T_2)} \setminus (W_{t_1} \cap W_{t_2})$. Clearly the size of all such cutsets is at most the width of (T, W). Let us now say that for $A, B \subseteq V(G)$, (A, B) is a *separation* of G if $A \cup B = V(G)$ and there are no edges between $A \setminus B$ and $B \setminus A$. Two separations (A, B) and (C, D) *do not cross* if one of the following holds: $A \subseteq C$ and $B \supseteq D$, or $A \subseteq D$ and $B \supseteq C$, or $A \supseteq C$ and $B \subseteq D$, or $A \supseteq D$ and $B \subseteq C$. So a tree-decomposition corresponds to a family of cross-free separations of a graph G.

As we shall see in this survey, we cannot hope for such strong structure results, with sweeping algorithmic consequences, for graph classes that are closed just under vertex deletion. Their structure is a lot more general, so that much stronger cutsets are needed for their decomposition which makes it a lot more difficult to make use of in algorithms. On the other hand Geelen, Gerards and Whittle have worked on generalizing results and techniques of Robertson and Seymour's Graph Minor Theory to matroids representable over finite fields, see [74]. They have shown that binary matroids are well-quasi-ordered by minors, and that any minor-closed property can be tested in polynomial time for binary matroids.

6 $(3PC(\cdot,\cdot), 3PC(\Delta,\cdot), 3PC(\Delta,\Delta), \text{wheel})$-free graphs

Cycle-free graphs are an example of a graph class that does not contain any of the Truemper configurations. This class of graphs is closed under minor taking and is in fact the class of K_3-minor-free graphs. The graphs in this class have tree-width at most 1. Outerplanar graphs (or $(K_4, K_{2,3})$-minor-free graphs), generalize cycle-free graphs and also do not contain any of the Truemper configurations. Their tree-width is at most 2, meaning that they can be decomposed by a sequence of non-crossing node cutsets of size at most 2 into cliques of size at most 3. Triangulated, or hole-free graphs, are another generalization of cycle-free graphs, that are not closed under minor taking, but still have the property of not containing any of the Truemper configurations. They can be decomposed by clique cutsets into cliques (as we have discussed in Section 3.1). The following class introduced in [42] generalizes all these classes of graphs.

Let γ be a $\{0,1\}$ vector whose entries are in one-to-one correspondence with the holes of a graph G. G is *universally signable* if for all choices of vector γ, there exists a subset F of the edge set of G such that $|F \cap H| \equiv \gamma_H \pmod 2$, for all holes H of G. From Theorem 2.1 it is easy to obtain the following characterization of universally signable graphs in terms of forbidden induced subgraphs.

Theorem 6.1 ([42]) *A graph is universally signable if and only if it is* $(3PC(\cdot,\cdot), 3PC(\Delta,\cdot), 3PC(\Delta,\Delta), wheel)$*-free.*

This characterization of universally signable graphs is then used to obtain the following decomposition theorem.

Theorem 6.2 (Conforti, Cornuéjols, Kapoor and Vušković [42]) *A connected* $(3PC(\cdot,\cdot), 3PC(\Delta,\cdot), 3PC(\Delta,\Delta), wheel)$*-free graph is either a clique or a hole or has a clique cutset.*

By the discussion in Section 3.1 it is clear how Theorem 6.2 can be used to construct efficient decomposition based algorithms for the recognition of the class, for finding the size of a largest clique, or independent set, and coloring the class. From Theorem 6.2 it is easy to deduce that every universally signable graph has a vertex that is simplicial or of degree 2. Recently it was shown in [2] that using LexBFS one can find in linear time an ordering of vertices $v_1, \ldots, v_n$ of a universally signable graph G such that for $i = 1, \ldots, n$, v_i is simplicial or of degree 2 in $G[\{v_1, \ldots, v_i\}]$. This implies a linear time robust algorithm for the maximum weight clique and coloring problems.

As we shall see in the following sections, once Truemper configurations are allowed to appear in a graph class, one will need more complex cutsets to decompose the class.

7 $(3PC(\Delta,\cdot), 3PC(\Delta,\Delta), \text{wheel})$-free graphs

A multigraph is called *series-parallel* if it arises from a forest by applying the following operations: adding a parallel edge or subdividing an edge. A *series-parallel graph* is a series-parallel multigraph with no parallel edges. Series-parallel graphs are an example of a graph class that is $(3PC(\Delta,\cdot), 3PC(\Delta,\Delta), \text{wheel})$-free. This class of graphs is closed under minor taking and is in fact the class of K_4-minor-free graphs. Their tree-width is at most 2. In this section we describe three more classes that are $(3PC(\Delta,\cdot), 3PC(\Delta,\Delta), \text{wheel})$-free, but are also closed under taking induced subgraphs.

7.1 (ISK4, wheel)-free graphs

A *subdivision* of a graph G is obtained by subdividing edges of G into paths of arbitrary length (at least 1). An *ISK4* is a subdivision of a K_4. Note that graphs that have no ISK4 as a subgraph are precisely series-parallel graphs. ISK4-free graphs are studied by Lévêque, Maffray and Trotignon in [87]. They prove a decomposition theorem for this class that uses double star cutsets. Unfortunately this does not lead to a recognition algorithm for ISK4-free graphs, which remains an open problem. In [87] a complete structural characterization of (ISK4, wheel)-free graphs is given, which we now describe.

A node cutset $S = \{a, b\}$ of a connected graph G is a *proper 2-cutset* if a and b are nonadjacent and both of degree at least 3, and such that $V(G) \setminus S$ can be partitioned into X and Y so that: $|X| \geq 2, |Y| \geq 2$; there are no edges between X and Y; and both $G[X \cup S]$ and $G[Y \cup S]$ contain an ab-path, but neither is a chordless ab-path.

Given a graph G, an induced subgraph K of G, and a set C of vertices of $G \setminus K$, the *attachment* of C over K is $N(C) \cap V(K)$. When a set $S = \{u_1, u_2, u_3, u_4\}$ induces a square (i.e. a 4-hole) in a graph G, with u_1, u_2, u_3, u_4 in this order along the square, a *link* of S is an induced pp'-path P of G such that either $p = p'$ and $N_S(p) = S$, or $N_S(p) = \{u_1, u_2\}$ and $N_S(p') = \{u_3, u_4\}$, or $N_S(p) = \{u_1, u_4\}$ and $N_S(p') = \{u_2, u_3\}$, and no interior vertex of P has a neighbor in S. A link with ends p, p' is said to be *short* if $p = p'$, and *long* otherwise. A *rich square* (resp. *long rich square*) is a graph K that contains a square S such that $K \setminus S$ has at least two components and every component of $K \setminus S$ is a link (resp. long link) of S.

A graph is *chordless* if all its cycles are chordless. It is easy to check that a line graph $G = L(R)$ is wheel-free if and only if R is chordless.

Theorem 7.1 (Lévêque, Maffray and Trotignon [87]) *An (ISK4, wheel)-free graph either has a clique cutset or a proper 2-cutset, or is one of the following types:*

- *a series-parallel graph,*
- *a complete bipartite graph,*
- *line graph of a chordless graph with maximum degree at most 3, or*
- *a long rich square.*

The structure of chordless graphs is given by the following theorem (that was implicitly proved in [119] and explicitly stated and proved in [87]). A graph G is *sparse* if every edge of G has an endnode that is of degree at most 2. Note that chordless graphs were first studied in the 1960s by Dirac [66] and Plummer [104]. A description of their work can also be found in [3].

Theorem 7.2 ([119, 87]) *A connected chordless graph is either sparse or has a 1-cutset or proper 2-cutset.*

Theorems 7.1 and 7.2 are used in [87] to recognize (ISK4, wheel)-free graphs in $\mathcal{O}(n^2m)$ time, as well as to show that (ISK4, wheel)-free graphs are 3-colorable and to give an $\mathcal{O}(n^2m)$ time coloring algorithm.

7.2 Unichord-free graphs

The class of graphs that do not contain a cycle with a unique chord, also known as *unichord-free graphs*, is studied by Trotignon and Vušković in [119], where they obtain the following structure theorem for this class.

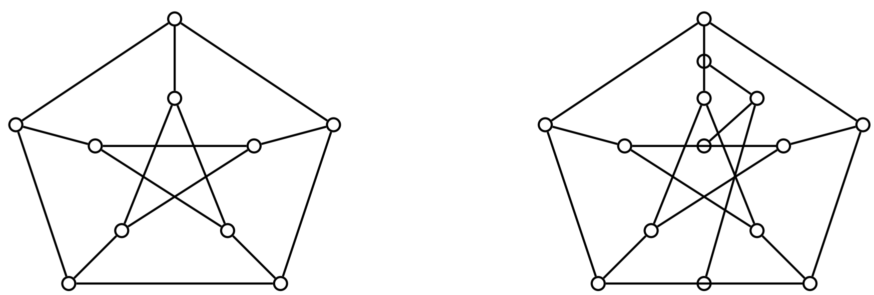

Figure 3: Petersen and Heawood graph

The Petersen and Heawood graphs are the two famous graphs, depicted in Figure 3, that were discovered at the end of the XIXth century in the research on the four color conjecture ([103], [79]), and they also appear as basic classes of unichord-free graphs. A graph is *strongly 2-bipartite* if it is 4-hole-free and bipartite with bipartition (X, Y) where X is the set of all degree 2 vertices of G and Y is the set of all nodes of G of degree at least 3. A node cutset $S = \{a, b\}$ of a connected graph G is a *special 2-cutset* if a and b are nonadjacent and both of degree at least 3, and such that $V(G) \setminus S$ can be partitioned into X and Y so that: $|X| \geq 2, |Y| \geq 2$; there are no edges between X and Y; and both $G[X \cup S]$ and $G[Y \cup S]$ contain an ab-path. A 1-join with split $(X_1, X_2, A_1, \emptyset, \emptyset, A_2, \emptyset, \emptyset)$ is *proper* if both A_1 and A_2 are stable sets of size at least 2.

Theorem 7.3 (Trotignon and Vušković [119]) *A connected unichord-free graph either has a 1-cutset, a special 2-cutset, or a proper 1-join, or is one of the following types:*

- *a clique,*
- *a hole of length at least 7,*

- *a strongly 2-bipartite graph, or*
- *an induced subgraph of the Petersen or the Heawood graph.*

We note that the decomposition theorem above in fact implies a complete structure theorem for unichord-free graphs. The decompositions in Theorem 7.3 can be reversed into compositions in such a way that every unichord-free graph can be built starting from basic graphs, that can be explicitly constructed, and gluing them together with prescribed composition operations, and all graphs built this way are unichord-free. This also implies a straightforward decomposition based recognition algorithm for this class that runs in $\mathcal{O}(nm)$ time.

Since unichord-free graphs are diamond-free, any edge of a unichord-free graph is contained in a unique maximal clique. Hence to find a maximum clique it is enough to look for common neighbors of every edge. This leads to an $\mathcal{O}(nm)$ time algorithm. In [119] an $\mathcal{O}(n+m)$ algorithm is given for the maximum clique problem for unichord-free graphs. It is based on the fact that a connected unichord-free graph that contains a triangle is either a clique or has a 1-cutset. Also in [119] it is shown how Theorem 7.3 can be used to obtain an $\mathcal{O}(nm)$ coloring algorithm for unichord-free graphs. It turns out that every unichord-free graph G is either 3-colorable or has an $\omega(G)$-coloring, and in particular $\chi(G) \leq \omega(G) + 1$. The problem of finding a maximum stable set of a unichord-free graph is NP-hard (follows from 2-subdivisions [105]).

Another characterization of unichord-free graphs is given by McKee in [97]: unichord-free graphs are precisely the graphs whose all minimal separators are stable sets (where a *separator* in a graph G is a set $S \subseteq V(G)$ such that $G \setminus S$ has more connected components than G).

7.3 Propeller-free graphs

Motivated by trying to understand the structure of wheel-free graphs, whose recognition remains an open problem, Aboulker, Radovanović, Trotignon and Vušković studied in [3] a subclass of wheel-free graphs known as *propeller-free graphs*. A *propeller* is a a graph that consists of a cycle C and a node x that has at least two neighbors on C. Let $\mathcal{C}_0$ be the class of graphs that have no node that has at least two neighbors of degree at least 3, $\mathcal{C}_1$ the class of graphs that have no propeller as a subgraph, and $\mathcal{C}_2$ the class of propeller-free graphs. Clearly $\mathcal{C}_0 \subsetneq \mathcal{C}_1 \subsetneq \mathcal{C}_2$.

First let us point out that by considering a longest path it is easy to show that graphs in $\mathcal{C}_2$ must always have a node of degree at most 2, and hence they are 3-colorable, see [3]. Observe that since a clique on 4 nodes is a propeller finding the size of a largest clique in a propeller-free graph can easily be done in polynomial time. On the other hand, finding a maximum independent set of a propeller-free graph is NP-hard (follows easily from [105], see also [119]).

The following decomposition theorems are given in [3], and used to obtain an $\mathcal{O}(nm)$ recognition algorithm for class $\mathcal{C}_1$, and an $\mathcal{O}(n^2m^2)$ recognition algorithm for class $\mathcal{C}_2$.

A 2-cutset $\{a, b\}$ of a graph G is an S_2-*cutset* (resp. K_2-*cutset*) if ab is not an edge (resp. is an edge). An S_2-cutset is *proper* if nodes of $G \setminus \{a, b\}$ can be partitioned into sets X and Y so that no node of X is adjacent to a node of Y, and neither $G[X \cup \{a,b\}]$ nor $G[Y \cup \{a,b\}]$ is a chordless ab-path. A K_2-cutset is *proper* if $G \setminus \{a,b\}$ contains no node adjacent to both a and b. A 3-cutset $\{u,v,w\}$ of a graph G is an *I-cutset* if $G[\{u,v,w\}]$ contains exactly one edge.

Theorem 7.4 (Aboulker, Radovanović, Trotignon and Vušković [3]) *A connected graph in $\mathcal{C}_1$ is either in $\mathcal{C}_0$ or it has a 1-cutset, a proper K_2-cutset or a proper S_2-cutset.*

Theorem 7.5 (Aboulker, Radovanović, Trotignon and Vušković [3]) *A graph in $\mathcal{C}_2$ is either in $\mathcal{C}_1$ or it has an I-cutset.*

Furthermore, it is shown in [3] that propeller-free graphs admit an extreme decomposition, which is used to prove that 2-connected propeller-free graphs must always have an edge whose endnodes are of degree 2. This implies that propeller-free graphs can also be edge-colored in polynomial time.

8 $(3PC(\Delta,\cdot), \text{proper wheel})$-free

Let (H,x) be a wheel. A *sector* of (H,x) is a minimal subpath of H, of length at least one, whose endnodes are neighbors of x on H. A sector is *short* if it is of length one, and *long* otherwise. (H,x) is a *triangle-free wheel* if it has no short sectors. (H,x) is a *universal wheel* if x is adjacent to all nodes of H, i.e. it has no long sectors. (H,x) is a *line wheel* if it has four sectors, exactly two of which are short, and the short sectors have no common node. (H,x) is a *fanned wheel* if it has exactly one long sector. A *proper wheel* is a wheel that is not a triangle-free wheel, a universal wheel, a line wheel, or a fanned wheel with 2 or 3 short sectors. We now consider three different subclasses of the class of $(3PC(\Delta,\cdot), \text{proper wheel})$-free graphs.

8.1 Cap-free graphs

A *cap* is a hole together with a node that is adjacent to exactly two adjacent nodes on the hole. Cap-free graphs were studied in [43], where the focus was on obtaining polynomial time algorithms for recognizing whether a cap-free graph contains an odd (respectively even) hole. Note that the only Truemper configurations that cap-free graphs can contain are $3PC(\cdot,\cdot)$'s and wheels that are either triangle-free, universal or fanned with exactly two short sectors. The following decomposition theorem is obtained in [43] for this class, generalizing the decomposition theorem for Meyniel graphs obtained by Burlet and Fonlupt in [7].

A graph G contains a *1-amalgam* (or *amalgam*) (X_1, X_2, K, A_1, A_2) if $V(G) = X_1 \cup X_2 \cup K$, where X_1, X_2 and K are disjoin sets, $|X_1| \geq 2$, $|X_1| \geq 2$ and the nodes of K induce a clique in G (possibly K is empty). Furthermore, for $i = 1, 2$, $\emptyset \neq A_i \subseteq X_i$; every node of A_1 is adjacent to every node of A_2, and these are the only edges between X_1 and X_2; and every node of K is adjacent to every node of $A_1 \cup A_2$. Amalgams were first introduced in [7], and they generalize 1-joins, as a 1-amalgam with $K = \emptyset$ corresponds to a 1-join.

A *basic cap-free graph* G is either a triangulated graph or a biconnected triangle-free graph with at most one additional node, that is adjacent to all other nodes of G

Theorem 8.1 (Conforti, Cornuéjols, Kapoor and Vušković [43]) *A cap-free graph is either basic or it has a 1-amalgam.*

Cap-free graphs can easily be recognized in polynomial time directly, but in [43] Theorem 8.1 is used to obtain decomposition based recognition algorithms for cap-free even-signable and cap-free odd-signable graphs (even-signable graphs are a generalization of odd-hole-free graphs, and odd-signable graphs are a generalization of even-hole-free graphs; they are formally defined in Section 9).

Since triangle-free graphs are cap-free (basic), it follows that the problems of coloring and finding the size of a largest independent set are both NP-hard for cap-free graphs. On the other hand, it is easy to see how to use Theorem 8.1 to obtain a polynomial time algorithm to solve the maximum weight clique problem for cap-free graphs, see [51].

Theorem 8.1 is a generalization of analogous result obtained by Burlet and Fonlupt [7] for *Meyniel* graphs, which are exactly (cap, odd-hole)-free graphs. The decomposition of Meyniel graphs by 1-amalgams is used in [7] to obtained the first known polynomial time recognition algorithm for this class. Subsequently, Roussel and Rusu [113] obtained a faster algorithm for recognizing Meyniel graphs (of complexity $\mathcal{O}(m^2)$), that is not decomposition based.

Hertz [80] gives an $\mathcal{O}(nm)$ algorithm for coloring and obtaining a largest clique of a Meyniel graphs. This algorithm is based on contractions of even pairs. Roussel and Rusu [114] give an $\mathcal{O}(n^2)$ algorithm that colors a Meyniel graph without using even pairs. This algorithm "simulates" even pair contractions and it is based on lexicographic breadth-first search and greedy sequential coloring. In [9] Cameron, Lévêque and Maffray give another $\mathcal{O}(n^2)$ algorithm for coloring Meyniel graphs, which takes as input any graph and finds either a clique and a coloring of the same size or a Meyniel obstruction (i.e. an odd cycle of length at least 5 with at most one chord).

Conforti and Gerards [51] show how to obtain a polynomial time algorithm for solving maximum weight independent set problem, on any class of graphs that is decomposable by amalgams into basic graphs for which one can solve the maximum weight independent set problem in polynomial time. In particular, using Theorem 8.1, they obtain a polynomial time algorithm for solving the maximum weight independent set problem for (cap, odd-hole)-free graphs (i.e. Meyniel graphs) and (cap, even-hole)-free graphs (and more generally, cap-free odd-signable graphs). Whether (cap, even-hole)-free graphs can be colored in polynomial time remains an open problem.

8.2 Claw-free graphs

A *claw* is a complete bipartite graph $K_{1,3}$. Observe that claw-free graphs are $3PC(\cdot,\cdot)$-free and the only wheels they may contain are line wheels, universal wheels whose rim is of length 4 or 5, and fanned wheels with 2 or 3 short sectors.

Structural study of claw-free graphs started in the context of perfect graphs. Claw-free Berge graphs have been shown to be perfect by Parthasarathy and Ravindra [101] by exploiting properties of minimally imperfect graphs. Another proof, based on properties of minimally imperfect graphs, is given by Giles, Trotter and Tucker in [75]. The first insight into the structure of claw-free Berge graphs is given by Chvátal and Sbihi [15]. They use the following characterization of claw-free Berge graphs to obtain a polynomial time recognition algorithm for this class (based on decomposition by clique cutsets). A graph is *elementary* if its edges can be colored by two colors in such a way that edges xy and yz have distinct colors whenever x and z are nonadjacent. It is easy to see that every elementary graph is claw-free Berge.

Theorem 8.2 (Chvátal and Sbihi [15]) *A claw-free graph G with no clique cutset is Berge if and only if it has at least one of the following properties:*

(i) G is elementary,

(ii) $\alpha(G) \geq 3$ and G contains no hole of length at least 5.

The following strengthening of Theorem 8.2 describes graphs that satisfy (ii) more precisely. A *cobipartite* graph G is the complement of bipartite graph, and *cobipartition* of G is its vertex partition (X, Y) such that X and Y are cliques. A graph is *peculiar* if it can be obtained as follows: take three, pairwise vertex-disjoint, cobipartite graphs $(A_1, B_1), (A_2, B_2), (A_3, B_3)$ such that each of them has at least one pair of nonadjacent vertices; add all edges between every two of them; then take three cliques K_1, K_2, K_3 that are pairwise disjoint and disjoint from the A_i's and B_i's; add all the edges between K_i and $A_j \cup B_j$ for $j \neq i$; there is no other edge in the graph.

Theorem 8.3 (Chvátal and Sbihi [15]) *A claw-free Berge graph either has a clique cutset or it is elementary or peculiar.*

Maffray and Reed [93] further strengthen Theorem 8.3 by giving complete description of the structure of elementary graphs.

An edge is *flat* if it does not lie in a triangle. Let xy be a flat edge of a graph G and let B be a cobipartite graph, disjoin from G, with cobipartition (X, Y) such that there is at least one edge between X and Y in B. Let G' be the graph obtained from $G \setminus \{x, y\}$ and B by adding all possible edges between X and $N(x) \setminus \{y\}$ and between Y and $N(y) \setminus \{x\}$. We say that G' is obtained from G by *augmenting along xy with augment B*.

Now let $x_1y_1, \ldots, x_hy_h$ be pairwise non-incident flat edges of G, and let $B_1, \ldots, B_h$ be pairwise disjoint cobipartite graphs, that are also disjoint from G, with cobipartitions $(X_1, Y_1), \ldots, (X_h, Y_h)$. Let G' be the graph obtained from G by augmenting respectively each edge x_iy_i with augment B_i. Graph G' is called an *augmentation* of G.

A *line graph* of a graph G, denoted by $L(G)$, is a graph whose vertices are edges of G, and two vertices of $L(G)$ are adjacent if and only if the corresponding edges of G have a common vertex.

Theorem 8.4 (Maffray and Reed [93]) *G is elementary if and only if G is an augmentation of a line graph of bipartite multigraph.*

Theorem 8.3 and Theorem 8.4 yield a new proof of the perfection of claw-free Berge graphs: by directly showing that peculiar graphs are perfect, using the above structural characterization of elementary graphs to show directly that they are perfect, and the fact that composing along clique cutsets preserves perfection. We now show that Theorems 8.3 and 8.4 in fact imply the following decomposition theorem for claw-free Berge graphs.

Corollary 8.5 *If G is a connected claw-free Berge graph, then either G has a clique cutset, a 1-join (whose one side is a homogeneous set that is a cobipartite graph) or a 2-join (whose one side is a homogeneous pair of cliques), or G is cobipartite or a line graph of a bipartite multigraph.*

It is easy to see that peculiar graphs have a 1-join whose one side is in fact a homogeneous set that induces a cobipartite graph. Let G be a connected claw-free graph with flat edge xy, and let G' be an augmentation of G along xy with augment B with cobipartition (X, Y). Let $N_x = N(x) \setminus \{y\}$ and $N_y = N(y) \setminus \{x\}$. We first observe that since G is claw-free, it follows that both N_x and N_y are cliques. If $V(G) = N[x]$ then G' is a cobipartite graph. In particular, a cobipartite graph itself can be viewed as an augmentation of the line graph of a bipartite graph consisting of just two adjacent vertices. If $N_y = \emptyset$ and $V(G) \setminus N[x] \neq \emptyset$ then N_x is a clique cutset. Let us now assume that G' is not cobipartite, $G' \neq G$, and that G' does not have a clique cutset. Then it follows that $(X \cup Y, V(G) \setminus \{x, y\}, X, Y, \emptyset, N_x, N_y, \emptyset)$ is a split of a 2-join. Given a 2-join of a graph H with split $(X_1, X_2, A_1, B_1, \emptyset, A_2, B_2, \emptyset)$, let us construct the blocks of decomposition by this 2-join as follows: block H_1 (resp H_2) is the graph obtained from $H[X_1]$ (resp. $H[X_2]$) by adding an edge a_2b_2 (resp. a_1b_1), all edges between a_2 and A_1 (resp. a_1 and A_2), and all edges between b_2 and B_1 (resp. b_1 and B_2). We observe that augmenting along a flat edge is the reverse of decomposing along a 2-join with such construction of the blocks of decomposition.

Given a graph G we say that the two 2-joins (X_1, X_2) and (Y_1, Y_2) of G are *non-crossing* if $X_1 \subseteq Y_1$ or $Y_1 \subseteq X_1$. Theorem 8.4 in fact shows that connected elementary graphs can be decomposed by a sequence of non-crossing 2-joins into cobipartite graphs and a line graph of a bipartite multigraph. Furthermore, if the graph G contains a line graph of a bipartite multigraph H (and H is a maximal such graph in G), then at each step of the decomposition of G a cobipartite graph is split off from the skeleton of H. Moreover, it is shown in [93] how to efficiently find such a sequence of 2-joins.

Minty [98] (corrected by Nakamura and Tamura [99]) showed that there is a polynomial algorithm to find a stable set of maximum weight in a claw-free graph by generalizing the algorithm of Edmonds [67, 68] for finding a maximum weighted matching in a graph. The problems of finding a largest clique and a minimum coloring are both NP-hard for claw-free graphs [83] (finding α in a triangle-free graph is NP-hard, and hence so is ω in a claw-free graph; edge-coloring on general graphs can be reduced to vertex coloring on claw-free graphs and hence the HP-hardness of that problem).

Hsu and Nemhauser [84] gave a combinatorial polynomial time algorithm which finds a maximum weighted clique and a minimum coloring of claw-free perfect graphs. Note that finding a maximum weighted clique in a claw-free perfect graph follows easily from the fact that the neighborhood of any vertex in such a graph is cobipartite. Li and Zang [88] gave a combinatorial polynomial time algorithm which finds a minimum weighted coloring of claw-free perfect graphs and is based on Theorems 8.3 and 8.4.

Chudnovsky and Seymour, in a series of papers [30, 31, 32, 33, 34, 35, 36], extend the above structural characterization of claw-free Berge graphs, to claw-free graphs in general. They show how all claw-free graphs can be obtained through explicit constructions starting from a few basic classes that can all be described explicitly. The full structural characterization they obtain is too complicated to explain. Here we state the decomposition theorem they obtain in [33] and then use in [34] for describing the construction. *Basic claw-free graphs* consist of seven subclasses, some of which are line graphs (of multigraphs), induced subgraphs of icosahedron, circular interval graphs and antiprismatic graphs (claw-free graphs in which every four vertices induce a subgraph with at least two edges). Antiprismatic graphs are further studied in [30] and [31]. We state the decomposition theorem in a weakened form and then explain the strong form actually obtained in [33].

Theorem 8.6 (Chudnovsky and Seymour [33]) *A connected claw-free graph is either basic claw-free or it has a 1-join, 2-join, M-join or 6-join.*

The decomposition theorem in [33] is stronger than the statement we gave above in two ways of key importance for being able to reverse the decomposition into a construction. First, the general joins used in the decomposition theorem in [33] have a particular structure, some of which is directly implied by the fact that the graph is assumed to be connected claw-free, but some is not. For example the 1-joins used in [33] have the following property: if neither side of the split of a 1-join is a homogeneous set then both special sets are cliques (this follows from being connected claw-free), and otherwise at least one of the sides is a homogeneous set and in this case at least one of the two possible homogeneous sets is a clique (note that this is not directly implied by being claw-free when both sides are homogeneous sets). For the 2-join the requirement is that at least one side has special sets that are cliques, and if neither side is a homogeneous pair then all special sets must be cliques. For the M-join, either one side is a homogeneous pair of cliques (that are neither complete nor anticomplete to each other) or all special sets are cliques. For the 6-join, all special sets are cliques and the two sides consist only of special sets.

The second important strengthening of Theorem 8.6 is that in [33] the decomposition theorem is actually proved for claw-free trigraphs. A *trigraph* is an object that generalizes a graph: in a graph every pair of vertices is either adjacent or nonadjacent, and in a trigraph every pair of vertices is either adjacent, or nonadajacent or semi-adjacent. A general join with split (X_1, X_2) in a trigraph has exactly the same requirements for adjacent and nonadjacent pairs of vertices from different sides of the split as in the graph version. In other words, if a pair of vertices are semi-adjacent they must both belong to the same side of the split. The decomposition theorem for claw-free trigraphs is used to strengthen the structure of needed general 2-joins to the point that decompositions can be reversed into compositions.

It is interesting to observe that out of all types of general 2-joins, the only one that does not appear in this decomposition theorem is the N-join. Here is why. Suppose that G is a claw-free graph that has an N-join but does not have any of the cutsets described in Theorem 8.6. Then it is easy to see that G must have a clique cutset S such that $V(G)$ can be partitioned into sets S, V_1, V_2 with $|V_i| \geq 2$, for $i = 1, 2$. If G has such a clique cutset, then (as shown in [33]) it follows that G must be a linear interval graph, i.e. one of the basic graphs (since every linear interval graph is also a circular interval graph).

Let us point out that the NP-hardness of the coloring problem and the maximum clique problem on claw-free graphs stems from the NP-hardness of these problems on the basic subclasses. For example, coloring line graphs is NP-hard, and finding a maximum clique in the class of graphs with no stable set of size 3 (a subclass of antiprismatic graphs) is already NP-hard. On the other hand, the chromatic number of a claw-free graph is bounded by the function of the size of its largest clique: it is easy to see that for a claw-free graph G, $\chi(G) \leq \omega(G)^2$, and that this is not far from being best possible since every graph with no stable set of size 3 is claw-free. One consequence of the structure theory for claw-free graphs is the following boundedness of the chromatic number for claw-free graphs that do contain a stable set of size 3.

Theorem 8.7 (Chudnovsky and Seymour [36]) *If G is a connected claw-free graph with $\alpha(G) \geq 3$, then $\chi(G) \leq 2\omega(G)$ (and this is asymptotically best possible).*

8.3 Bull-free graphs

A *bull* is a graph with five vertices a, b, c, d, e and five edges ab, bc, cd, be, ce. Bull-free graphs cannot contain $3PC(\Delta, \cdot)$'s, the only $3PC(\Delta, \Delta)$'s they can have are $\bar{C}_6$'s (i.e. the complements of holes of length 6), and the only wheels they can have are triangle-free wheels, universal wheels, fanned wheels with 2 short sectors, and wheels whose rim is a 5-hole and whose centre has 4 neighbors on the rim.

The study of bull-free graphs also started in the context of perfect graphs. First Chvátal and Sbihi [39] proved that bull-free Berge graphs are perfect by obtaining the following decomposition theorem.

Theorem 8.8 (Chvátal and Sbihi [39]) *A connected bull-free Berge graph is either bipartite or cobipartite, or it has a homogeneous pair or a star cutset in the graph or its complement.*

Since bipartite and cobipartite graphs are perfect and minimal imperfect graphs cannot have homogeneous pairs [39], nor star cutsets [38], nor star cutsets in the complement (which follows from the Perfect Graph Theorem: a graph is perfect if and only if its complement is perfect [89]), it follows that bull-free Berge graphs must be perfect.

Reed and Sbihi [107] showed how bull-free perfect graphs can be recognized in polynomial time by decomposing them with homogeneous sets and finding vertices whose removal does not change whether the graph is Berge or not, and hence avoiding decomposition by star cutsets.

De Figueiredo and Maffray [63] give a combinatorial strongly polynomial time algorithm for solving the maximum weighted clique problem on bull-free perfect graphs. Since this class is self-complimentary, this algorithm implies combinatorial polynomial time algorithms for maximum weighted stable set problem, minimum weighted coloring problem and minimum weighted clique covering problem. Their algorithm is based on the following decomposition theorem. A graph is *weakly triangulated* if it is (hole, antihole)-free. A graph is *transitively orientable* if it admits a *transitive orientation*, i.e. an orientation of its edges with no circuit and with no P_3 abc with the orientation $\vec{ab}$ and $\vec{bc}$. Such graphs are also called *comparability graphs*.

Theorem 8.9 (De Figueiredo and Maffray [63]) *A connected bull-free Berge graph is either weakly triangulated, transitively orientable, complement of a transitively orientable graph, or it has a homogeneous set or a homogeneous pair.*

A maximum weighted clique of a weakly triangulated graph can be found in strongly polynomial time by the algorithm in [81], of a transitively orientable graph by the algorithm in [82], and of the complement of a transitively orientable graph by the algorithm in [8].

The complete structural characterization of bull-free graphs in general is done by Chudnovsky in a series of papers [18, 19, 20, 21]. This characterization is too difficult to explain precisely here, but we give some flavor of it. First let us consider some examples of bull-free graphs. Triangle-free graphs are clearly bull-free, and since a bull is a self-complementary structure, so are their complements. Note that from these two classes of graphs it follows that the maximum clique problem, maximum stable set problem and the vertex coloring problem are all NP-hard for bull-free graphs. Another example of a bull-free graph is an *ordered split graph*: a graph G whose vertex set is a union of a clique $\{k_1, \dots, k_n\}$ and a stable set $\{s_1, \dots, s_n\}$, and s_i is adjacent to k_j if and only if $i + j \leq n + 1$. A larger bull-free graph can be created from smaller ones using the operation of *substitution*: input are two bull-free graphs G_1 and G_2 with disjoint vertex sets, and vertex $v \in V(G_1)$; output is a new graph G whose vertex set is $V(G_1) \cup V(G_2) \setminus \{v\}$ and whose edge set is $E(G_1 \setminus \{v\}) \cup E(G_2) \cup \{xy : x \in V(G_1) \setminus \{v\}, y \in V(G_2), \text{ and } xv \in E(G_1)\}$. We observe that this composition operation is the reverse of the homogeneous set decomposition.

Chudnovsky's construction of all bull-free graphs starts from three explicitly constructed classes of *basic bull-free graphs*: $\mathcal{T}_0, \mathcal{T}_1$ and $\mathcal{T}_2$. $\mathcal{T}_0$ is a class of graphs with few nodes, the graphs in $\mathcal{T}_1$ are built from a triangle-free graph F and a collection of disjoint cliques with prescribed attachments in F (so triangle-free graphs are in this class, and also ordered split graphs), and $\mathcal{T}_2$ generalizes graphs G that have a pair uv of vertices, so that uv is dominating both in G and $\bar{G}$. Furthermore, each graph G in $\mathcal{T}_1 \cup \mathcal{T}_2$ comes with a list $\mathcal{L}_G$ of "expandable edges". Chudnovsky shows that every bull-free graph that is not obtained by substitution from smaller ones, can be constructed from a basic bull-free graph by expanding the edges in $\mathcal{L}_G$ (where edge expansion is an operation corresponding to "reversing the homogeneous pair decomposition"). To prove this result, again it was convenient to work on trigraphs, and the first step is to obtain the following decomposition theorem for bull-free trigraphs.

Theorem 8.10 (Chudnovsky [18, 19, 20, 21]) *If G is a bull-free trigraph, then either G or $\bar{G}$ is basic bull-free, or G has a homogeneous set or a homogeneous pair.*

The fact that the theorem is proved on trigraphs makes it possible to put enough structure on the homogeneous pairs actually needed in the decomposition to allow for the reversal of the decomposition into a composition.

Recall that the way homogeneous sets and homogeneous pairs are defined for trigraphs, is the same as for graphs when it comes to adjacent and nonadjacent pairs that go across the split, the semi-adjacent pairs are only allowed to be fully contained in a side of a split. So in some sense the above decomposition theorem is saying that there is a sequence of non-crossing decompositions by homogeneous sets and homogeneous pairs that can break the graph down to a basic graph. We can see this by thinking of semi-adjacent edges in the trigraph as marker edges used in the construction of blocks of decomposition by homogeneous pairs.

One consequence of Chudnovsky's characterization of bull-free graphs is that the Erdős-Hajnal conjecture holds for them.

Conjecture 8.11 (Erdős and Hajnal [69]) *For every graph H, there exists $f(H) > 0$, such that if G is H-free, then G contains either a clique or a stable set of size at least $|V(G)|^{f(H)}$*

Theorem 8.12 (Chudnovsky and Safra [27]) *If G is a bull-free graph, then G contains a stable set or a clique of size at least $|V(G)|^{\frac{1}{4}}$.*

The proof of Theorem 8.12 is actually based on the following decomposition theorem.

Theorem 8.13 (Chudnovsky [18, 27]) *If G is a bull-free graph that contains a hole H of length at least 5, and vertices $c, a \in V(G) \setminus V(H)$ such that c is complete to $V(H)$ and a is anticomplete to $V(H)$, then G has a homogeneous set.*

The structure theorem for bull-free graphs is also used to derive a structure theorem for bull-free perfect graphs [25], which is then used in [102] to derive combinatorial polynomial time algorithm for maximum weighted clique problem on bull-free perfect graphs that is a bit faster than the algorithm in [63].

9 Excluding some wheels and some 3-path-configurations

The class of regular bipartite graphs and the class of balanceable bipartite graphs, that generalizes it, were discussed in Section 4. As we have seen, the only Truemper configurations that balanceable bipartite graphs can have are bipartite wheels whose center has an even number of neighbors on the rim, and $3PC(u, v)$'s where u and v are on the same side of the bipartition. In this section we discuss three more well studied classes of graphs where some 3-path-configurations and some wheels are excluded, but enough is left in to make them structurally quite complex, namely the classes of even-hole-free graphs, odd-hole-free graphs and perfect graphs.

9.1 Even-hole-free graphs

The class of even-hole-free graphs is structurally quite similar to the class of perfect graphs, which was the key initial motivation for their study. The first major structural study of even-hole-free graphs was done by Conforti, Cornuéjols, Kapoor and Vušković in [45] and [46]. They were focused on showing that even-hole-free graphs can be recognized in polynomial time (a problem that at that time was not even known to be in NP), and their primary motivation was to develop techniques which can then be used in the study of perfect graphs. In [45] a decomposition theorem is obtained for even-hole-free graphs, based on which the first known polynomial time recognition algorithm for even-hole-free graphs is constructed in [46]. This research kick-started a number of other studies of even-hole-free graphs which we survey in this section. A more detailed survey of even-hole-free graphs is given in [125].

The class of even-hole-free graphs is also of independent interest due to its relationship to β-perfect graphs introduced by Markossian, Gasparian and Reed [96]. For a graph G, let $\delta(G)$ be the minimum degree of a vertex in G. Consider the following total order on $V(G)$: 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 all the vertices already removed. Coloring greedily on this order gives the upper bound $\chi(G) \leq \beta(G)$, where $\beta(G) = \max\{\delta(G') + 1 : G'$ is an induced subgraph of $G\}$. A graph is *β-perfect* if for each induced subgraph H of G, $\chi(H) = \beta(H)$. It is easy to see that β-perfect graphs belong to the class of even-hole-free graphs, and that this containment is proper.

The essence of even-hole-free graphs is actually captured by their generalization to signed graphs. A graph is *odd-signable* if there exists an assignment of $0, 1$ weights to its edges that makes every chordless cycle of odd weight. We say that a wheel (H, x) is *even* if x has an even number of neighbors on H. The following characterization of odd-signable graphs can be easily derived from Theorem 2.1.

Theorem 9.1 ([43]) *A graph is odd-signable if and only if it does not contain an even wheel, a $3PC(\cdot,\cdot)$ nor a $3PC(\Delta,\Delta)$.*

All decomposition theorems for even-hole-free graphs which we now describe are in fact proved for 4-hole-free odd-signable graphs, and the above characterization of odd-signable graphs is repeatedly used in the proofs.

A 2-join with split $(X_1, X_2, A_1, B_1, \emptyset, A_2, B_2, \emptyset)$ is *connected* if for $i = 1, 2$, $G[X_i]$ contains a path whose one endnode is in A_i and the other in B_i. It is a *path 2-join* if for some $i \in \{1, 2\}$, $G[X_i]$ is a chordless path whose one endnode is in A_i and the other in B_i. A *non-path 2-join* is a 2-join that is not a path 2-join. 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 vertices.

Theorem 9.2 (Conforti, Cornuéjols, Kapoor and Vušković [45]) *A connected even-hole-free graph is either an extended clique tree, or it has a k-star cutset for $k \leq 3$ or a connected non-path 2-join.*

This theorem was strong enough to be used in the construction of a polynomial time recognition algorithm for even-hole-free graphs in [46], but even at that time it was suspected that a stronger decomposition theorem was possible. The strengthening of Theorem 9.2 was eventually given in [62].

Theorem 9.3 (da Silva and Vušković [62]) *A connected even-hole-free graph is either an extended clique tree, or it has a star cutset or a connected non-path 2-join.*

We observe that in the decomposition theorems in [45] and [62], the basic graphs are defined in a more specific way, but for the purposes of the algorithms the statements of Theorems 9.2 and 9.3 suffice. As in the case of the decomposition based recognition algorithm for balanced bipartite graphs, described in Section 4.2, the problem in using the above theorems for constructing a recognition algorithm for even-hole-free graphs are the star cutsets. For 2-joins it is possible to construct the blocks of decomposition that are class-preserving for the class of even-hole-free graphs (by replacing a side of a 2-join by a path of appropriate length, which clarifies the usefulness of connected non-path 2-joins in the above decomposition theorems), and lead to a polynomial decomposition tree. To use the decomposition by star cutsets, one first needs to *clean* the graph (as described in Section 3). The decomposition based recognition algorithm for even-hole-free graphs in [46] is of complexity of about $\mathcal{O}(n^{40})$. In [24] an $\mathcal{O}(n^{31})$ recognition algorithm for even-hole-free graphs is given, that first cleans the graph and then directly looks for an even hole (using the shortest-paths detector technique described in Section 2.1). In [62] an $\mathcal{O}(n^{19})$ decomposition based algorithm is obtained. Finally, by using Theorem 9.3 Chang and Lu [11] obtain an $\mathcal{O}(n^{11})$ recognition algorithm for even-hole-free graphs. They improve the complexity by introducing a new idea of a "tracker" that allows for fewer graphs that need to be recursively decomposed by star cutsets, and they improve the complexity of the cleaning procedure by first looking for certain structures, using the three-in-a-tree algorithms from [29], before applying the cleaning. We observe that detecting whether a graph contains a $3PC(\cdot,\cdot)$ or a $3PC(\Delta,\Delta)$ can be done in $\mathcal{O}(n^{35})$ time [23]. The high complexity of all these algorithms is due to the cleaning procedure.

The following intermediate result is used as one of the steps in the proof of Theorem 9.3, and as we shall see later, it is of an independent interest. A *diamond* is the graph obtained from a clique on 4 nodes by removing an edge. A *bisimplicial cutset* is a node cutset that either induces a clique or two cliques with exactly one common node. Note that a bisimplicial cutset is a very special type of a star cutset.

Theorem 9.4 (Kloks, Müller and Vušković [85]) *A connected (even-hole, diamond)-free graph is either an extended clique tree, or it has a bisimplicial cutset or a connected non-path 2-join.*

We now survey known results related to combinatorial optimization on even-hole-free graphs. In Section 10 we shall see that 2-joins can be used in a decomposition based optimization algorithm, but it is not clear how to use star cutsets. The decomposition by star cutsets can, on the other hand, be used to obtain local structural properties that can then be used in algorithms.

The complexities of finding a maximum independent set and an optimal coloring are not known for even-hole-free graphs. One can find a maximum clique of an even-hole-free graph in polynomial time, since as observed by Farber [70], 4-hole-free graphs have $\mathcal{O}(n^2)$ maximal cliques and hence one can list them all in polynomial time. The following structural characterization of even-hole-free graphs leads to a faster algorithm for computing a maximum clique in an even-hole-free graph.

Theorem 9.5 (da Silva and Vušković [61]) *Every even-hole-free graph has a node whose neighborhood is triangulated.*

This result follows from the fact that universal wheels in even-hole-free graphs can be decomposed in a particular way by star cutsets. For any node x in a graph G, a maximal clique belongs to $G[N[x]]$ or $G \setminus \{x\}$. Therefore Theorem 9.5 reduces the problem of finding a maximum clique in an even-hole-free graph to the problem of finding a maximum clique in a triangulated graph, which as we have seen in Section 3.1 can be done efficiently. Observe that in order to find a maximum clique, it is not necessary that we know that the input graph is even-hole-free. The algorithm proceeds by attempting to construct an ordering of vertices $x_1, \dots, x_n$ of the input graph G such that, for every $i = 1, \dots, n$, the neighborhood of x_i in $G_i = G[\{x_i, \dots x_n\}]$ is triangulated. If it cannot complete the sequence, then it follows from Theorem 9.5 that the input graph is not even-hole-free. Otherwise, we get a sequence of triangulated graphs $G_1, \dots, G_n$ such that every maximal clique of G belongs to exactly one of them. It follows that there are at most $n + 2m$ maximal cliques in an even-hole-free graph and all of them can be generated in $\mathcal{O}(n^2m)$ time (and hence in the same time a maximum weighted clique in a weighted even-hole-free graph can be found). In [2] it is shown how LexBFS can be used to find the above ordering of vertices, reducing the complexity of finding a maximum weighted clique to $\mathcal{O}(nm)$. Again, the algorithms discussed are robust in the sense that they either correctly compute the desired clique or they correctly identify the input graph as not being even-hole-free.

Here is another property of even-hole-free graphs that shows that this class is χ-bounded (i.e. the chromatic number is bounded by a function of the size of a largest clique). A *bisimplicial vertex* is a vertex whose set of neighbors induces a graph that is a union of two cliques.

Theorem 9.6 (Addario-Berry, Chudnovsky, Havet, Reed and Seymour [1]) *Every even-hole-free graph has a bisimplicial vertex.*

It is interesting to observe that Theorem 9.6 is also obtained using decomposition, although in [1] not all even-hole-free graphs are decomposed, but enough structures are decomposed using special double star cutsets (star cutsets and cutsets that become double star cutsets after some edges are added) to obtain the desired result. It clearly implies the following corollary.

Corollary 9.7 ([1]) *If G is even-hole-free then $\chi(G) \leq 2\omega(G) - 1$.*

Recall that β-perfect graphs are a subclass of even-hole-free graphs that can be efficiently colored, by coloring greedily on a particular easily constructable ordering of vertices. Unfortunately it is not known whether β-perfect graphs can be recognized in polynomial time. In [96] it is shown that (even-hole, diamond, cap)-free graphs are β-perfect, and in [64] it is shown that (even-hole, diamond, cap-on-6-vertices)-free graphs are β-perfect. These results are further generalized in [85] where it is shown that (even-hole, diamond)-free graphs are β-perfect, and hence can be both recognized and colored in polynomial time. This result follows from the following property of (even-hole, diamond)-free graphs, that is obtained by using

Theorem 9.4. A vertex is *simplicial* if its neighborhood set induces a clique, and it is a *simplicial extreme* if it is either simplicial or of degree 2.

Theorem 9.8 (Kloks, Müller and Vušković [85]) *Every (even-hole, diamond)-free graph has a simplicial extreme.*

Theorem 9.8 and the following property of minimal β-imperfect graphs, imply that (even-hole, diamond)-free graphs are β-perfect.

Lemma 9.9 (Markossian, Gasparian and Reed [96]) *A minimal β-imperfect graph that is not an even hole, contains no simplicial extreme.*

Corollary 9.10 ([85]) *Every (even-hole, diamond)-free graph is β-perfect.*

Note that the fact that (even-hole, diamond)-free graphs have simplicial extremes implies that for such graphs G, $\chi(G) \leq \omega(G) + 1$.

9.2 Perfect graphs and odd-hole-free graphs

A graph G is *perfect* if for every induced subgraph H of G, $\chi(H) = \omega(H)$. In 1961 Berge [5] made a conjecture that characterizes perfect graphs in terms of excluded induced subgraphs in the following way: *a graph is perfect if and only if it does not contain an odd hole nor an odd antihole* (where an *antihole* is a complement of a hole). The graphs that do not contain an odd hole nor an odd antihole are known as *Berge* graphs. It is easy to see that perfect graphs must be Berge, so the essence of the conjecture is to show that Berge graphs must be perfect. This famous conjecture, known as the Strong Perfect Graph Conjecture (SPGC), sparked an enormous amount of diverse research until it was finally proved in 2002 by Chudnovsky, Robertson, Seymour and Thomas [26], and is now known as the Strong Perfect Graph Theorem.

The approach that eventually worked for proving the SPGC is the decomposition method. This approach entails proving a decomposition theorem for Berge graphs, in such a way that the undecomposable (basic) graphs are simple enough so that the SPGC can be proved directly for them, and the cutsets used have the property that no minimum counter-example to the conjecture can have them. This approach was used to prove the SPGC for a number of subclasses of graphs (i-triangulated graphs using clique cutsets [72], weakly triangulated graphs using star cutsets [78], bull-free graphs using homogeneous pairs and star cutsets [14]), but the one subclass that came closest to revealing the structure of Berge graphs in general is the class of 4-hole-free Berge graphs. In [48] Conforti, Cornuéjols and Vušković prove the SPGC for 4-hole-free graphs by the following decomposition theorem, and the fact that it was already known that no minimal imperfect graph can have a star cutset [38], and that if a a minimal imperfect graph has a 2-join then it must be an odd hole [56].

Theorem 9.11 (Conforti, Cornuéjols and Vušković [48]) *A 4-hole-free Berge graph is either bipartite or line graph of a bipartite graph, or it has a star cutset or a connected non-path 2-join.*

Before we describe the decomposition theorem for Berge graphs in general, we state the decomposition theorem for odd-hole-free graphs (a superclass of Berge graphs), that also preceded the work in [26].

Theorem 9.12 (Conforti, Cornuéjols and Vušković [49]) *An odd-hole-free graph is either bipartite, line graph of a bipartite graph or complement of a line graph of a bipartite graph, or it has a double star cutset or a connected non-path 2-join.*

We observe that, as in the study of even-hole-free graphs, a convenient setting for the study of odd-hole-free graphs is their generalization to signed graphs. A graph is *even-signable* if there exists an assignment of $0, 1$ weights to its edges that makes every triangle of odd weight and every hole of even weight. An *odd wheel* is a wheel that induces an odd number of triangles. The following characterization of even-signable graphs can easily be derived from Theorem 2.1.

Theorem 9.13 ([43]) *A graph is even-signable if and only if it does not contain an odd wheel nor a* $3PC(\Delta, \cdot)$.

We now describe the decomposition theorems for Berge graphs, by first introducing the specific cutsets used and the basic graphs.

A 2-join with split $(X_1, X_2, A_1, B_1, \emptyset, A_2, B_2, \emptyset)$ is a *P_3-path 2-join* if for some $i \in \{1, 2\}$, $G[X_i]$ induces a path on three nodes whose one endnode is in A_i and the other in B_i. A *non-P_3-path 2-join* is a 2-join that is not a P_3-path 2-join.

The homogeneous pair as defined in Section 3.2 was first introduced by Chvátal and Sbihi in [39], where it was also shown that no minimal imperfect graph has a homogeneous pair. The definition that we give here is a slight variation that is used in [26]. A *homogeneous pair* is a partition of $V(G)$ into six non-empty sets (A, B, C, D, E, F) such that:

- every vertex in A has a neighbor in B and a non-neighbor in B, and vice versa;
- the pairs (C, A), (A, F), (F, B), (B, D) are complete;
- the pairs (D, A), (A, E), (E, B), (B, C) are anticomplete.

If S is a skew cutset in a graph G, then $(S, V(G) \setminus S)$ is also called a *skew partition* of G. A *balanced skew partition* is a skew partition (S, T) with the additional property that every induced path of length at least 2 in G with ends in S and interior in T has even length, and every induced path of length at least 2 in $\overline{G}$ with ends in T and interior in S has even length. If (S, T) is a balanced skew partition we say that the skew cutset S is *balanced*. Balanced skew partitions were first defined in [26] where it was also shown that no minimum counter-example to the strong perfect graph conjecture admits a balanced skew partition.

A *double split graph* is a graph G constructed as follows. Let $m, n \geq 2$ be integers. Let $A = \{a_1, \ldots, a_m\}$, $B = \{b_1, \ldots, b_m\}$, $C = \{c_1, \ldots, c_n\}$, $D = \{d_1, \ldots, d_n\}$ be four disjoint sets. Let G have vertex set $A \cup B \cup C \cup D$ and edges in such a way that:

- a_i is adjacent to b_i for $1 \leq i \leq m$. There are no edges between $\{a_i, b_i\}$ and $\{a_{i'}, b_{i'}\}$ for $1 \leq i < i' \leq m$;

- c_j is non-adjacent to d_j for $1 \leq j \leq n$. There are all four edges between $\{c_j, d_j\}$ and $\{c_{j'}, b_{j'}\}$ for $1 \leq j < j' \leq n$;
- there are exactly two edges between $\{a_i, b_i\}$ and $\{c_j, d_j\}$ for $1 \leq i \leq m$, $1 \leq j \leq n$ and these two edges are disjoint.

Note that $C \cup D$ is a non-balanced skew cutset of G and that $\overline{G}$ is a double split graph. Note that in a double split graph, vertices in $A \cup B$ all have degree $n+1$ and vertices in $C \cup D$ all have degree $2n + m - 2$. Since $n \geq 2, m \geq 2$ implies $2n - 2 + m > 1 + n$, it is clear that given a double split graph the partition $(A \cup B, C \cup D)$ is unique. Hence, we call *matching edges* the edges that have an end in A and an end in B.

A graph is said to be *basic* if one of $G, \overline{G}$ is either a bipartite graph, the line-graph of a bipartite graph or a double split graph.

The following theorem was first conjectured in a slightly different form in [48]. A corollary of it is the Strong Perfect Graph Theorem.

Theorem 9.14 (Chudnovsky, Robertson, Seymour and Thomas, [26]) *Let G be a Berge graph. Then either G is basic or G has a homogeneous pair or a balanced skew partition, or one of $G, \overline{G}$ has a connected non-P_3-path 2-join.*

The theorem that we state now is due to Chudnovsky who proved it from scratch, that is without assuming Theorem 9.14. Her proof uses the notion of a *trigraph*. The theorem shows that homogeneous pairs are not necessary to decompose Berge graphs. Thus it is a result stronger than Theorem 9.14.

Theorem 9.15 (Chudnovsky, [17, 16]) *Let G be a Berge graph. Then either G is basic, or one of $G, \overline{G}$ has a connected non-P_3-path 2-join or a balanced skew partition.*

In [22] Chudnovsky, Cornuéjols, Liu, Seymour and Vušković show that Berge graphs (and hence perfect graphs) can be recognized in polynomial time. As expected, cleaning (that, recall from Section 3, was developed in order to be able to use cutsets such as star cutsets in decomposition based recognition algorithms) was the key to the work in [22]. What was surprising, as Chudnovsky and Seymour observed, was that once the cleaning is performed, one does not need the decomposition based algorithm, one can simply look for the odd hole directly (using the shortest-paths detector technique described in Section 2.1). In [22] two recognition algorithms for Berge graphs are given: an $\mathcal{O}(n^9)$ Chudnovsky/Seymour style algorithm that uses the direct method, and an $\mathcal{O}(n^{18})$ decomposition based algorithm (that uses Theorem 9.12). Whether odd-hole-free graphs can be recognized in polynomial time remains an open problem.

Finding a maximum clique, a maximum independent set and an optimal coloring can all be done in polynomial time for perfect graphs. This result of Grötschel, Lovász and Schrijver uses the ellipsoid method and a polynomial time separation algorithm for a certain class of positive semidefinite matrices related to Lovász's upper bound on the Shannon capacity of a graph [90]. The question remains whether these optimization problems can be solved for perfect graphs by purely combinatorial polynomial time algorithms, avoiding the numerical instability of the ellipsoid method. Some partial results in this direction are described in Section 10. We observe that for a number of classes we have seen so far, the decomposition theorem is used to prove the existence of a vertex with a particular neighborhood, which is then used to obtain some optimization algorithms. Such a result is not yet known for perfect graphs.

The complexities of finding a maximum independent set and an optimal coloring for odd-hole-free graphs are not known. Finding a maximum clique is NP-hard for odd-hole-free graphs (follows from 2-subdivisions [105]: if G' is the graph obtained from G by subdividing every edge twice then $\alpha(G') = \alpha(G) + |E(G)|$; also all holes of G' are of length at least 9, and hence $\overline{G'}$ does not contain a hole of length at least 5).

10 Combinatorial optimization with 1-joins and 2-joins

In this section we consider how decompositions with 1-joins and 2-joins can be used for construction of different optimization algorithms.

10.1 1-Joins

1-Join decompositions (also known as split decompositions) were used for circle graph recognition [71, 116] and parity graph recognition [40, 60]. Cunningham [58] showed how 1-join decompositions can be used for the independent set problem, and Cicerone and Di Stefano [40] showed how this algorithm can be applied to parity graphs. Rao [106] shows how to use 1-join decompositions for the clique and coloring problems. To describe these results we first need to describe the blocks of decomposition by a 1-join.

Let $(X_1, X_2, A_1, \emptyset, \emptyset, A_2, \emptyset, \emptyset)$ be a 1-join of a graph G. The *block of decomposition by a 1-join* are graphs $G_1 = G[X_1 \cup \{m_2\}]$ (where m_2 is any vertex of A_2) and $G_2 = G[X_2 \cup \{m_1\}]$ (where m_1 is any vertex of A_1). It turns out that if a graph has a 1-join then it has an *extreme 1-join*, i.e. a 1-join where one of the blocks of decomposition does not have a 1-join. So one can construct an extreme decomposition tree by 1-joins, similarly to the extreme decomposition tree by clique cutsets described in Section 3.1. Dahlhaus shows that this decomposition tree can in fact be constructed in linear time ([60], see also [13]). (The first algorithm for decomposing a graph by 1-joins, of complexity $\mathcal{O}(n^3)$, was given in [58], this was later improved to an $\mathcal{O}(nm)$ algorithm in [71], and to an $\mathcal{O}(n^2)$ algorithm in [91]). The fact that one can compute an extreme decomposition tree by 1-joins is quite useful when constructing optimization algorithms.

To solve the independent set, clique and coloring problems using 1-join decomposition we need to move to the weighted versions of these problems. Let $w : V(G) \longrightarrow \mathcal{N}$ be a weight function for a graph G. When H is an induced subgraph of G, $w(H)$ denotes the sum of the weights of vertices in H. By $\alpha_w(G)$ we denote the weight of a maximum weighted independent set of G, and by $\omega_w(G)$ we denote the weight of a maximum weighted clique of G. From the discussion above and the following lemmas it is easy to see how to obtain polynomial time algorithms to solve the weighted independent set and the weighted clique problems for graphs that are decomposable by 1-joins into basic graphs for which these problems can be solved in polynomial time. Similarly, one can also solve the weighted chromatic number problem [106]. For $a \in \mathcal{N}$, denote by $w|_{v \to a}$ the function on domain $V(G) \cup \{v\}$ such that $w|_{v \to a}(v) = a$ and $w|_{v \to a}(u) = w(u)$ for all $u \in V(G) \setminus \{v\}$.

Lemma 10.1 ([58]) *Let* $a = \alpha_w(G_2 \setminus N_{G_2}[m_1])$ *and* $a' = \alpha_w(G_2 \setminus m_1)$. *Then* $\alpha_w(G) = \alpha_{w|_{m_2 \to a' - a}}(G_1) + a$.

Lemma 10.2 ([106]) *Let* $a = \omega_w(G_2[N_{G_2}[m_1])$. *Then* $\omega_w(G) = max\{\omega_w(G_2 \setminus m_1), \omega_{w|_{m_2 \to a}}(G_1)\}$.

10.2 2-Joins

To use 2-joins in a decomposition based optimization algorithm is a lot more difficult. In [120] Trotignon and Vušković focused on developing techniques for combinatorial optimization with 2-joins, by considering two classes of graphs decomposable by 2-joins into basic graphs for which we know how to solve the respective optimization problems in polynomial time. They give combinatorial polynomial time algorithms for finding the size of a largest independent set in even-hole-free graphs with no star cutset; as well as finding the size of a largest independent set, the size of a largest clique and an optimal coloring for Berge graphs with no skew cutset, no 2-join in the complement and no homogeneous pair. The coloring algorithm can be implemented to run in $\mathcal{O}(n^7)$ time, and all the other ones in $\mathcal{O}(n^6)$ time. Coloring of Berge graphs actually follows from being able to compute the size of a largest independent set and largest clique ([76, 77]), so these two problems are the focus of the work in [120].

Using 2-joins in combinatorial optimization algorithms requires building blocks of decomposition and asking at least two questions for at least one block, while for recognition algorithms one question suffices. Applying this process recursively can lead to an exponential blow-up even when the decomposition tree is linear in size of the input graph. In [120] this problem is bypassed by using *extreme 2-joins*, i.e. 2-joins whose one block of decomposition is basic. Graphs in general do not have extreme 2-joins, this is a special property of 2-joins in graphs with no star cutset.

Consider the following way of constructing blocks of decomposition by 2-joins. Let $(X_1, X_2, A_1, B_1, \emptyset, A_2, B_2, \emptyset)$ be a 2-join of the graph G. The *blocks of decomposition by 2-join* are graphs $G_1^{k_1}$ and $G_2^{k_2}$ defined as follows. $G_1^{k_1}$ is obtained by replacing X_2 by a *marker path* P_2, of length k_1, from a vertex a_2 complete to A_1, to a vertex b_2 complete to B_1 (the interior of P_2 has no neighbor in X_1). The block $G_2^{k_2}$ is obtained similarly by replacing X_1 by a marker path P_1 of length k_2. It is easy to see that in an even-hole-free graph or an odd-hole-free graph, all paths from a node in A_i to a node in B_i with interior in $X_i \setminus (A_i \cup B_i)$ have the same parity. So if we are careful about the parity of the marker paths, the blocks of decomposition will be class-preserving for the classes of even-hole-free graphs and odd-hole-free graphs.

The graph G in Figure 4 has exactly two 2-joins, one is represented with bold lines, and the other is equivalent to it. Both of the blocks of decomposition are isomorphic to graph H (where dotted lines represent paths of arbitrary length, possibly of length 0), and H has a 2-join whose edges are represented with bold lines. So G does not have an extreme 2-join.

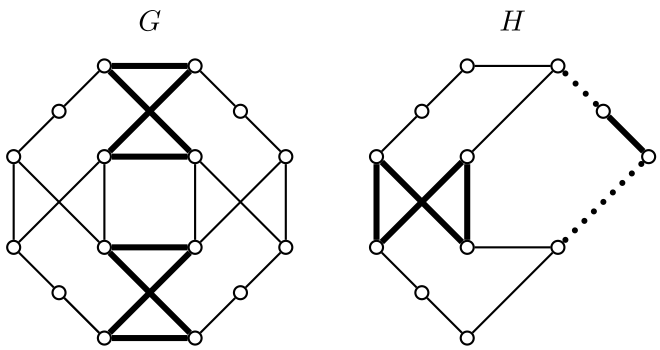

Figure 4: A graph G with no extreme 2-join

With the above construction of blocks of decomposition by 2-joins, clearly one needs to use non-path 2-joins in algorithms. For optimization algorithms it is essential that the 2-joins used are extreme. Can these 2-joins be found efficiently in a graph? The first algorithm for detecting 2-joins, of complexity $\mathcal{O}(n^3m)$, was given by Cornuéjols and Cunningham in [56]. In [12] this complexity is improved to $\mathcal{O}(n^2m)$, and an algorithm of the same complexity is given for detecting non-path 2-joins, as well as an $\mathcal{O}(n^3m)$ algorithm for detecting extreme non-path 2-joins. Note that finding an extreme non-path 2-join reduces to finding a minimally-sided non-path 2-join.

We now give a method from [120] that can be used to solve the maximum weighted clique problem for any class of graphs that can be decomposed with extreme (non-path) 2-joins into basic graphs for which the problem can be solved efficiently. To be able to apply arguments inductively one needs to switch to the weighted version of the problem. Let G be a weighted graph with a weight function $w : \mathcal{N}^+ \longrightarrow V(G)$. Let $\omega(G)$ denote the weight of a maximum weighted clique of G.

Let G_1 and G_2 be the blocks of decomposition by a 2-join of G. Let us also assume that the lengths of the marker paths are at least 3 (this is important in [120] because there it is not just important that the parity of holes is preserved in the blocks, but also the property of not having a star cutset). Let $P_1 = a_1, x_1, \ldots, x_k, b_1$ be the marker path of G_2, where a_1 is adjacent to all of A_2 and b_1 is adjacent to all of B_2. The weights of vertices of G_2 are modified as follows:

- for every $u \in X_2$, $w_{G_2}(u) = w_G(u)$;
- $w_{G_2}(a_1) = \omega(G[A_1])$;
- $w_{G_2}(b_1) = \omega(G[B_1])$;
- $w_{G_2}(x_1) = \omega(G[X_1]) - \omega(G[A_1])$;
- $w_{G_2}(x_i) = 0$, for $i = 2, \ldots, k$.

With such modification of weights it can be shown that $\omega(G) = \omega(G_2)$ [120]. Now if our 2-join is an extreme 2-join, we may assume that block G_1 is undecomposable and hence basic in the sense that the maximum weighted clique problem can be solved on that block efficiently. In particular, all of the weights needed to be computed for modifying the weights of G_2 as above can be computed efficiently. We note that this method of computing a maximum clique in the case of even-hole-free graphs (with no star cutset) is not so interesting since the algorithm described in Section 9.1 is more efficient.

Using 2-joins to compute a maximum stable set is more difficult since stable sets can completely overlap both sides of the 2-join. In [120] a simple class of graphs $\mathcal{C}$ decomposable along extreme 2-joins into bipartite graphs and line graphs of cycles with one chord is given for which computing a maximum stable set is NP-hard. Here is how $\mathcal{C}$ is constructed. A *gem-wheel* is a graph made of an induced cycle of length at least 5 together with a vertex adjacent to exactly four consecutive vertices of the cycle. Note that a gem-wheel is a line-graph of a cycle with one chord. A *flat path* of a graph G is a path of length at least 2, whose interior vertices all have degree 2 in G, and whose ends have no common neighbors outside the path. *Extending a flat path* $P = p_1, \dots, p_k$ of a graph means deleting the interior vertices of P and adding three vertices x, y, z and the following edges: p_1x, xy, yp_k, zp_1, zx, zy, zp_k. *Extending a graph* G means extending all paths of $\mathcal{M}$, where $\mathcal{M}$ is a set of flat paths of length at least 3 of G. Class $\mathcal{C}$ is the class of all graphs obtained by extending 2-connected bipartite graphs. From the definition, it is clear that all graphs of $\mathcal{C}$ are decomposable along extreme 2-joins. One leaf of the decomposition tree is the underlying bipartite graph, and all the others leaves are gem-wheels. The following is shown by Naves [100], and the proof of it can be found in [120].

Theorem 10.3 (Naves [100, 120]) *The problem whose instance is a graph G from $\mathcal{C}$ and an integer k, and whose question is "Does G contain a stable set of size at least k" is NP-complete.*

Let $\mathcal{C}^{PARITY}$ be the class of graphs in which all holes have the same parity. In [120] it is shown how to use 2-joins to compute a maximum stable set in $\mathcal{C}^{PARITY}$.

Let G be a graph with a weight function w on the vertices and $(X_1, X_2, A_1, B_1, \emptyset, A_2, B_2, \emptyset)$ a 2-join of G. For $i = 1, 2$, $D_i = X_i \setminus (A_i \cup B_i)$. For any graph H, $\alpha(H)$ denotes the weight of a maximum weighted stable set of H. Let $a = \alpha(G[A_1 \cup D_1])$, $b = \alpha(G[B_1 \cup D_1])$, $c = \alpha(G[D_1])$ and $d = \alpha(G[X_1])$. The blocks of decomposition by a 2-join that would be useful for computing a largest stable set can be done as follows.

A *flat claw* of a weighted graph G is any set $\{q_1, q_2, q_3, q_4\}$ of vertices such that:

- the only edges between the q_i's are q_1q_2, q_2q_3 and q_4q_2;
- q_1 and q_3 have no common neighbor in $V(G) \setminus \{q_2\}$;
- q_4 has degree 1 in G and q_2 has degree 3 in G.

Define the *even block* G_2 with respect to a 2-join $X_1|X_2$ in the following way. Keep X_2 and replace X_1 by a flat claw on $q_1, \dots, q_4$ where q_1 is complete to A_2 and q_3 is complete to B_2. Give the following weights: $w(q_1) = d - b$, $w(q_2) = c$, $w(q_3) = d - a$, $w(q_4) = a + b - d$. It can be shown that all weights are in fact non-negative.

A *flat vault* of graph G is any set $\{r_1, r_2, r_3, r_4, r_5, r_6\}$ of vertices such that:

- the only edges between the r_i's are such that r_3, r_4, r_5, r_6, r_3 is a 4-hole;
- $N(r_1) = N(r_5) \setminus \{r_4, r_6\}$;
- $N(r_2) = N(r_6) \setminus \{r_3, r_5\}$;
- r_1 and r_2 have no common neighbors;
- r_3 and r_4 have degree 2 in G.

Define the *odd block* G_2 with respect to a 2-join in the following way. Replace X_1 by a flat vault on $r_1, \dots, r_6$. Moreover r_1, r_5 are complete to A_2 and r_2, r_6 are complete to B_2. Give the following weights: $w(r_1) = d - b$, $w(r_2) = d - a$, $w(r_3) = w(r_4) = c$, $w(r_5) = w(r_6) = a + b - c - d$. It can be shown that all weights are non-negative, if $c + d \leq a + b$ holds.

By adequately choosing when to use even or odd blocks, it can be shown that for a 2-join in a graph G in $\mathcal{C}^{PARITY}$, $\alpha(G_2) = \alpha(G)$.

We observe that such construction of blocks is not class-preserving, so it would not allow for inductive use of the decomposition theorems. This problem is avoided in [120] by building the decomposition tree in two stages. First using blocks of decomposition constructed as we discussed at the beginning of this section (that are class-preserving). In the second stage the decomposition tree is reprocessed to replace marker paths by gadgets designed for even and odd blocks. This results in the leaves of the decomposition tree that are not basic as in the decomposition theorems used, but some extensions of these basic classes, for which it is shown that the weighted stable set problem can be computed efficiently.

Recently this work was extended by Chudnovsky, Trotignon, Trunck and Vušković [37] to obtain an $\mathcal{O}(n^7)$ coloring algorithm for perfect graphs with no balanced skew-partition, by focusing on decompositions by 2-joins, their complements and homogeneous pairs. Here the notion of trigraphs was quite helpful in obtaining the desired extreme decompositions.

Acknowledgements

We thank Maria Chudnovsky and Nicolas Trotignon for helpful discussions about the presentation of some of the material. This work was partially supported by EPSRC grant EP/H021426/1 and Serbian Ministry of Education and Science projects 174033 and III44006.

References

[1] L. Addario-Berry, M. Chudnovsky, F. Havet, B. Reed, and P. Seymour, Bisimplicial vertices in even-hole-free graphs, *Journal of Combinatorial Theory B* **98** (2008), 1119-1164.

[2] P. Aboulker, P. Charbit, M. Chudnovsky, N. Trotignon, and K. Vušković, LexBFS, structure and algorithms, preprint (2012), submitted.

[3] P. Aboulker, M. Radovanović, N. Trotignon, and K. Vušković, Graphs that do not contain a cycle with a node that has at least two neighbors on it, to appear in *SIAM Journal on Discrete Mathematics*.

[4] P. Aboulker, M. Radovanović, N. Trotignon, T. Trunck, and K. Vušković, Linear balanceable and subcubic balanceable graphs, preprint (2012), submitted.

[5] C. Berge, Färbung von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind (Zusammenfassung), *Technical report, Wiss. Z. Martin Luther Univ. Math.-Natur. Reihe (Halle-Wittenberg)* (1961).

[6] C. Berge, Sur Certains Hypergraphes Généralisant les Graphes Bipartis, in *Combinatorial Theory and its Applications* I (eds. P. Erdös, A. Rényi and V. Sós), *Colloquia Mathematica Societatis János Bolyai* **4**, North-holland, Amsterdam (1970), 119-133.

[7] M. Burlet and J. Fonlupt, Polynomial algorithm to recognize a Meyniel graph, *Discrete Math* **21** (1984), 225-252.

[8] K. Cameron, Antichain sequences, *Order* **2** (1985), 249-255.

[9] K. Cameron, B. Lévêque, and F. Maffray, Coloring vertices of a graph or finding a Meyniel Obstruction, *Theoretical Computer Science* **428** (2012), 10-17.

[10] P. Camion, Characterization of totally unimodular matrices, *Proceedings of the American Mathematical Society* **16** (1965), 1068-1073.

[11] H.-C. Chang and H.-I. Lu, A faster algorithm to recognize even-hole-free graphs, *Proceedings of 23rd Annual ACM-SIAM Symposium on Discrete Algorithms* (2012), 1286-1297.

[12] P. Charbit, M. Habib, N. Trotignon, and K. Vušković, Detecting 2-joins faster, to appear in *Journal of Discrete Algorithms*.

[13] P. Charbit, F. de Montgolfier, and M. Raffinot, Linear time split decomposition revisited, *SIAM Journal on Discrete Mathematics* **26 (2)** (2012), 499-514.

[14] V. Chvátal and N. Sbihi, Bull-free Berge graphs are perfect, *Graphs and Combinatorics* **3** (1987), 127-139.

[15] V. Chvátal and N. Sbihi, Recognizing claw-free Berge graphs, *Journal of Combinatorial Theory B* **44** (1988), 154-176.

[16] M. Chudnovsky, Berge trigraphs and their applications, PhD thesis, Princeton University (2003).

[17] M. Chudnovsky, Berge trigraphs, *Journal of Graph Theory* **53 (1)** (2006), 1-55.

[18] M. Chudnovsky, The structure of bull-free graphs I: Three-edge paths with centers and anticenters, *Journal of Combinatorial Theory B* **102 (1)** (2012), 233-251.

[19] M. Chudnovsky, The structure of bull-free graphs II and III: A summary, *Journal of Combinatorial Theory B* **102 (1)** (2012), 252-282.

[20] M. Chudnovsky, The structure of bull-free graphs II: elementary trigraphs, manuscript.

[21] M. Chudnovsky, The structure of bull-free graphs III: global structure, manuscript.

[22] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, and K. Vušković, Recognizing Berge graphs, *Combinatorica* **25 (2)** (2005), 143-186.

[23] M. Chudnovsky and R. Kapadia, Detecting a theta or a prism, *SIAM Journal on Discrete Math* **22** (2008), 1164-1186.

[24] M. Chudnovsky, K. Kawarabayashi, and P. Seymour, Detecting even holes, *Journal of Graph Theory* **48** (2005), 85-111.

[25] M. Chudnovsky and I. Penev, The structure of bull-free perfect graphs, to appear in *Journal of Graph Theory*.

[26] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, The strong perfect graph theorem, *Annals of Mathematics* **164 (1)** (2006), 51-229.

[27] M. Chudnovsky and S. Safra, The Erdős-Hajnal conjecture for bull-free graphs, *Journal of Combinatorial Theory B* **98** (2008), 1301-1310.

[28] M. Chudnovsky and P. Seymour, Solution of three problems of Cornuéjols, *Journal of Combinatorial Theory B* **98** (2008), 116-135.

[29] M. Chudnovsky and P. Seymour, The three-in-a-tree problem, *Combinatorica* **30 (4)** (2010), 387-417.

[30] M. Chudnovsky and P. Seymour, Claw-free Graphs I: Orientable prismatic graphs, *Journal of Combinatorial Theory B* **97** (2007), 867-901.

[31] M. Chudnovsky and P. Seymour, Claw-free Graphs II: Non-orientable prismatic graphs, *Journal of Combinatorial Theory B* **98** (2008), 249-290.

[32] M. Chudnovsky and P. Seymour, Claw-free Graphs III: Circular interval graphs, *Journal of Combinatorial Theory B* **98** (2008), 839-938.

[33] M. Chudnovsky and P. Seymour, Claw-free Graphs IV: Decomposition theorem, *Journal of Combinatorial Theory B* **98** (2008), 839-938.

[34] M. Chudnovsky and P. Seymour, Claw-free Graphs V: Global structure, *Journal of Combinatorial Theory B* **98** (2008), 1373-1410.

[35] M. Chudnovsky and P. Seymour, Claw-free Graphs VI: Coloring claw-free graphs, *Journal of Combinatorial Theory B* **100** (2010), 560-572.

[36] M. Chudnovsky and P. Seymour, Claw-free Graphs VII: Quasi-line graphs, submitted.

[37] M. Chudnovsky, N. Trotignon, T. Trunck, and K. Vušković, Coloring perfect graphs with no balanced skew-partitions, preprint (2012), submitted.

[38] V. Chvátal, Star cutsets and perfect graphs, *Journal of Combinatorial Theory B* **39** (1985), 189-199.

[39] V. Chvátal and N. Sbihi, Bull-free Berge graphs are perfect, *Graphs and Combinatorics* **3** (1987), 127-139.

[40] S. Cicerone and D. Di Stefano, On the extension of bipartite graphs to parity graphs, *Discrete Applied Mathematics* **95** (1999), 181-195.

[41] M. Conforti, G. Cornuéjols, G. Gasparyan, and K. Vušković, Perfect graphs, partitionable graphs and cutsets, *Combinatorica* **22 (1)** (2002), 19-33.

[42] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković, Universally signable graphs, *Combinatorica* **17 (1)** (1997), 67-77.

[43] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković, Even and odd holes in cap-free graphs, *Journal of Graph Theory* **30** (1999), 289-308.

[44] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković, Balanced $0, \pm 1$ matrices, Part I: Decomposition theorem, and Part II: Recognition algorithm, *Journal of Combinatorial Theory B* **81** (2001), 243-306.

[45] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković, Even-hole-free graphs, Part I: Decomposition theorem, *Journal of Graph Theory* **39** (2002), 6-49.

[46] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković, Even-hole-free graphs, Part II: Recognition algorithm, *Journal of Graph Theory* **40** (2002), 238-266.

[47] M. Conforti, G. Cornuéjols, and M.R. Rao, Decomposition of balanced matrices, *Journal of Combinatorial Theory B* **77** (1999), 292-406.

[48] M. Conforti, G. Cornuéjols, and K. Vušković, Square-free perfect graphs, *Journal of Combinatorial Theory Series B* **90** (2004), 257-307.

[49] M. Conforti, G. Cornuéjols, and K. Vušković, Decomposition of odd-hole-free graphs by double star cutsets and 2-joins, *Discrete Applied Mathematics* **141** (2004), 41-91.

[50] M. Conforti, G. Cornuéjols, and K. Vušković, Balanced matrices, *Discrete Mathematics* **306** (2006), 2411-2437.

[51] M. Conforti and A.M.H. Gerards, *Stable sets and graphs with no even holes*, preprint (2003), unpublished.

[52] M. Conforti, B. Gerards, and A. Kapoor, A theorem of Truemper, *Combinatorica* **20 (1)** (2000), 15-26.

[53] M. Conforti and M.R. Rao, Structural properties and decomposition of linear balanced matrices, *Mathematical Programming* **55** (1992), 129-168.

[54] M. Conforti and M.R. Rao, Testing balancedness and perfection of linear matrices, *Mathematical Programming* **61** (1993), 1-18.

[55] G. Cornuéjols, Combinatorial Optimization: Packing and Covering, *SIAM*, Philadelphia, PA (2001). CBMS-NSF Regional Conference Series in Applied Mathematics 74.

[56] G. Cornuéjols and W. H. Cunningham, Composition for perfect graphs, *Discrete Mathematics* **55** (1985), 245-254.

[57] B. Courcelle, The monadic second order logic of graphs. I. Recognizable sets of finite graphs, *Information and Computation* **85 (1)** (1990), 12-75.

[58] W.H. Cunningham, Decomposition of directed graphs, *SIAM Journal on Algebraic and Discrete Methods* **3 (2)** (1982), 214-228.

[59] W. H. Cunningham and J. Edmonds, A combinatorial decomposition theory, *Canadian Journal of Mathematics* **32 (3)** (1980), 734-765.

[60] E. Dahlhaus, Parallel algorithms for hierarchical clustering and applications to split decomposition and parity graph recognition, *Journal of Algorithms* **36 (2)** (2000), 205-240.

[61] M.V.G. da Silva and K. Vušković, Triangulated neighborhoods in even-hole-free graphs, *Discrete Mathematics* **307** (2007), 1065-1073.

[62] M.V.G da Silva and K. Vušković, Decomposition of even-hole-free graphs with star cutsets and 2-joins, to appear in *Journal of Combinatorial Theory B.*

[63] C.M.H. de Figueiredo and F. Maffray, Optimizing bull-free perfect graphs, *SIAM Journal on Discrete Mathematics* **18** (2004), 226-240.

[64] C.M.H. de Figueiredo and K. Vušković, A class of β-perfect graphs, *Discrete Mathematics* **216** (2000), 169-193.

[65] G.A. Dirac, On rigid circuit graphs, *Abh. Math. Sem. Univ. Hamburg* **25** (1961), 71-76.

[66] G.A. Dirac, Minimally 2-connested graphs, *Journal für die Reine und Angewandte Mathematik* **228** (1967), 204-216.

[67] J. Edmonds, Maximum matching and a polytope with 0,1-vertices, *J. Res. Nat. Bur. Standards* **69B** (1965), 125-130.

[68] J. Edmonds, Paths, trees and flowers, *Canad. J. Math.* **17** (1965), 449-467.

[69] P. Erdős and A. Hajnal, Ramsey-type theorems, *Discrete Applied Mathematics* **25** (1989), 37-52.

[70] M. Farber, On diameters and radii of bridged graphs, *Discrete Mathematics* **73** (1989), 249–260.

[71] C.P. Gabor, W.L. Hsu, and K.J. Supowit, Recognizing circle graphs in polynomial time, *Journal of the ACM* **36 (3)** (1989), 435-473.

[72] T. Gallai, Graphen mit triangulierbaren ungeraden Vielecken, *Magyar Tud. Akad. Mat. Kutado Int. Közl.* **7** (1962), 3-36.
English translation given by F. Maffray and M. Preissmann in *Perfect Graphs*, J.L. Ramírez-Alfonsín and B.A. Reed (editors), Willey (2001), 25-66.

[73] M. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W.H. Freeman, San Francisco (1979).

[74] J. Geelen, B. Gerards, and G. Whittle, Towards a matroid-minor structure theory, *Combinatorics, Complexity and Chance*, A tribute to Dominic Welsh, (eds. G. Grimmett and C. McDiarmid), Oxford University Press (2007), 72-82.

[75] R. Giles, L.E. Trotter, and A. Tucker, The strong perfect graph theorem for a class of partition able graphs, in *Topics on Perfect Graphs* (C. Berge and V. Chvátal, eds.), North Holland, Amsterdam (1984), 161-167.

[76] M. Grötschel, L. Lovász, and A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica* **1** (1981), 169-197.

[77] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, Springer Verlag (1988).

[78] R.B. Hayward, Weakly triangulated graphs, *Journal of Combinatorial Theory B* **39** (1985), 200-209.

[79] P.J. Heawood, Map colour theorem, *Quart J Pure Appl Math* **24** (1890), 332-338.

[80] A. Hertz, A fast algorithm for colouring Meyniel graphs, *Journal of Combinatorial Theory B* **50** (1990), 231-240.

[81] R. Hayward, C.T. Hoàng and F. Maffray, Optimizing weakly triangulated graphs, *Graphs and Combinatorics* **5** (1989), 339-349. See erratum in vol. 6 (1990) 33-35.

[82] C.T. Hoàng, Efficient algorithms for minimum weighted colouring of some classes of perfect graphs, *Discrete Applied Mathematics* **55** (1994), 133-143.

[83] W.L. Hsu, Efficient algorithms for some packing and covering problems on graphs, Ph.D. Dissertation, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York, (1980).

[84] W.L. Hsu and G.L. Nemhauser, Algorithms for maximum weighted cliques, minimum weighted clique covers, and minimum colorings of claw-free perfect graphs, in *Topics on Perfect Graphs* (eds. C. Berge and V. Chvátal), North-Holland, Amsterdam (1984), 357-369.

[85] T. Kloks, H. Müller, and K. Vušković, Even-hole-free graphs that do not contain diamonds: a structure theorem and its consequences, *Journal of Combinatorial Theory B* **99** (2009), 733-800.

[86] B. Lévêque, D.Y. Lin, F. Maffray, and N. Trotignon, Detecting induced subgraphs, *Discrete Applied Mathematics* **157** (2009), 3540-3551.

[87] B. Lévêque, F. Maffray, and N. Trotignon, On graphs with no induced subdivision of K_4, *Journal of Combionatorial Theory B* **102** (2012), 924-947.

[88] X. Li and W. Zang, A combinatorial algorithm for minimum weighted coloring of claw-free perfect graphs, *Journal of Combinatorial Optimization* **9 (4)** (2005), 331-347.

[89] L. Lovász, Normal hypergraphs and the perfect graph conjecture, *Discrete Mathematics* **2** (1972), 253-267.

[90] L. Lovász, On the Shannon capacity of a graph, *IEEE Trans. Inform. Theory* **25** (1979), 1-7.

[91] T.-H. Ma and J. Spinrad, An $\mathcal{O}(n^2)$ algorithm for undirected split decomposition, *Journal of Algorithms* **16 (1)** (1994), 154-160.

[92] F. Maffray and M. Preissmann, On the NP-completeness of the k-colorability problem for triangle-free graphs, *Discrete Math.* **162** (1996), 313-317.

[93] F. Maffray and B. Reed, A description of claw-free perfect graphs, *Journal of Combinatorial Theory B* **75** (1999), 134-156.

[94] F. Maffray and N. Trotignon, Algorithms for perfectly contractile graphs, *SIAM Journal on Discrete Mathematics* **19 (3)** (2005), 553-574.

[95] F. Maffray, N. Trotignon, and K. Vušković, Algorithms for square-$3PC(\cdot,\cdot)$-free graphs, *SIAM Journal on Discrete Math* **22 (1)** (2008), 51-71.

[96] S.E. Markossian, G.S. Gasparian, and B.A. Reed, β-perfect graphs, *Journal of Combinatorial Theory B* **67** (1996), 1-11.

[97] T. McKee, Independent separator graphs, *Utilitas Mathematica* **73** (2007), 217-224.

[98] G.J. Minty, On maximal independent sets of vertices in claw-free graphs, *Journal of Combinatorial Theory B* **28** (1980), 284-304.

[99] D. Nakamura and A. Tamura, A revision of Minty's algorithm for finding a maximum weighted stable set of a claw-free graph, *Journal of Operations Research Society of Japan* **44 (2)** (2001), 194-204.

[100] G. Naves, personal communication (2009).

[101] K.R. Parthasarathy and G. Ravindra, The strong perefect graph conjecture is true for $K_{1,3}$-free graphs, *Journal of Combinatorial Theory B* **21** (1976), 212-223.

[102] I. Penev, Forbidden substructures in graphs and trigraphs, and related coloring problems, PhD thesis, Columbia University (2012).

[103] J. Petersen, Sur le théorèm de Tait, *L'Intermédiaire Math* **5** (1898), 225-227.

[104] M.D. Plummer, On minimal blocks, *Transactions of the American Mathematical Society* **134** (1968), 85-94.

[105] S. Poljak, A note on the stable sets and coloring of graphs, *Comment. Math. Univ. Carolin.* **15** (1974), 307-309.

[106] M. Rao, Solving some NP-complete problems using split decomposition, *Discrete Applied Mathematics* **156 (14)** (2008), 2768-2780.

[107] B. Reed and N. Sbihi, Recognizing bull-free perfect graphs, *Graphs and Combinatorics* **11** (1995), 171-178.

[108] N. Robertson and P.D. Seymour, Graph minors III. Planar tree-width, *Journal of Combinatorial Theory B* **36** (1984), 49-64.

[109] N. Robertson and P.D. Seymour, Graph minors V. Excluding a planar graph, *Journal of Combinatorial Theory B* **41** (1986), 92-114.

[110] N. Robertson and P.D. Seymour, Graph minors XVI. Excluding a non-planar graph, *Journal of Combinatorial Theory B* **89** (2003), 43-76.

[111] N. Robertson and P.D. Seymour, Graph minors XX. Wagner's Conjecture, *Journal of Combinatorial Theory B* **92 (2)** (2004), 325-357.

[112] D.J. Rose, R.E. Tarjan and G.S. Leuker, Algorithmic aspects of vertex elimination on graphs, *SIAM J. Comput.* **5** (1976), 266-283.

[113] F. Roussel and I. Rusu, Holes and dominoes in Meyniel graphs, *Internat. J. Found. Comput. Sci.* **10** (1999), 127-146.

[114] F. Roussel and I. Rusu, An $\mathcal{O}(n^2)$ algorithm to color Meyniel graphs, *Discrete Mathematics* **235 (1-3)** (2001), 107-123.

[115] P. Seymour, Decomposition of regular matroids, *Journal of Combinatorial Theory B* **28** (1980), 305-359.

[116] J. Spinrad, Recognition of circle graphs, *Journal of Algorithms* **16 (2)** (1994), 264-282.

[117] R.E. Tarjan, Decomposition by clique separators, *Discrete Mathematics* **55** (1985), 221-232.

[118] N. Trotignon, Decomposing Berge graphs and detecting balanced skew partitions, *Journal of Combinatorial Theory Series B* **98** (2008), 173-225.

[119] N. Trotignon and K. Vušković, A structure theorem for graphs with no cycle with a unique chord and its consequences, *Journal of Graph Theory* **63 (1)** (2010), 31-67.

[120] N. Trotignon and K. Vušković, Combinatorial optimization with 2-joins, *Journal of Combinatorial Theory B* **102** (2012), 153-185.

[121] K. Truemper, Alpha-balanced graphs and matrices and GF(3)-representability of matroids, *Journal of Combinatorial Theory Series B* **32** (1982), 112-139.

[122] K. Truemper, A decomposition theory for matroids. V. Testing of matrix total unimodularity, *Journal of Combinatorial Theory Series B* **49** (1990), 241-281.

[123] K. Truemper, *Matroid Decomposition*, Academic Press, Boston (1992).

[124] W.T. Tutte, A homotopy theorem for matroids I,II, *Transactions of the American Mathematical Society* **88** (1958), 144-174.

[125] K. Vušković, Even-hole-free graphs: a survey, *Applicable Analysis and Discrete Mathematics* **4 (2)** (2010), 219-240.

[126] S.H. Whitesides, An algorithm for finding clique cut-sets, *Information Processing Letters* **12** (1981), 31-32.

[127] G. Zambelli, On perfect graphs and balanced matrices, PhD thesis, Carnegie Mellon University (2004).

School of Computing
University of Leeds
Leeds LS2 9JT, UK
and
Faculty of Computer Science (RAF)
Union University
Knez Mihajlova 6/VI
11000 Belgrade, Serbia
k.vuskovic@leeds.ac.uk