ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Discrete Applied Mathematics 131 (2003) 39–50

DISCRETE APPLIED MATHEMATICS

www.elsevier.com/locate/dam

Stability number of bull- and chair-free graphs revisited☆

Andreas Brandstädt[a,*], Chính T. Hoàng[b], Van Bang Le[a]

[a] *Fachbereich Informatik, Universität Rostock, A.-Einstein-Str. 21, D-18051 Rostock, Germany*

[b] *Department of Physics and Computing, Wilfrid Laurier University, Waterloo, Ont., Canada*

Received 22 February 2001; received in revised form 27 July 2001; accepted 5 May 2002

Abstract

De Simone showed that prime bull- and chair-free graphs containing a co-diamond are either bipartite or an induced cycle of odd length at least five. Based on this result, we give a complete structural characterization of prime (bull,chair)-free graphs having stability number at least four as well as of (bull,chair,co-chair)-free graphs. This implies constant-bounded clique width for these graph classes which leads to linear time algorithms for some algorithmic problems. Moreover, we obtain a robust $\mathcal{O}(nm)$ time algorithm for the maximum weight stable set problem on bull- and chair-free graphs without testing whether the (arbitrary) input graph is bull- and chair-free. This improves previous results with respect to structural insight, robustness and time bounds.

Keywords: Modular decomposition; Prime graphs; Maximum weight stable set problem on graphs; Bull- and chair-free graphs; Robust algorithms; Clique width of graphs

1. Introduction

A vertex set in a finite undirected graph is *stable* if its elements are pairwise non-adjacent. The *maximum* (*weight*) *stable set* ($M(W)S$) problem asks for a maximum (vertex weight) stable set in the given graph. The M(W)S problem is a basic algorithmic graph problem occurring in many models in computer science and

☆ Part of this work was done while the first author visited Wilfrid Laurier University, Waterloo, Ontario; the first and second author acknowledge support by NSERC.

* Corresponding author. Tel.: +49-381-498-3363; fax: +49-381-498-3366.

E-mail addresses: ab@informatik.uni-rostock.de (A. Brandstädt), choang@wlu.ca (C.T. Hoàng), le@informatik.uni-rostock.de (V.B. Le).

0166-218X/03/$ - see front matter
PII: S0166-218X(02)00415-8

operations research. It is NP-complete in general, which led to the investigation of a variety of graph classes defined by forbidding small graphs such as claw-free graphs for which a polynomial time algorithm for the MWS problem was given by Minty [25] (and independently for the MS problem by Sbihi [27]) (as usual, the *claw* is the graph consisting of four vertices a,b,c,d such that a is adjacent to the pairwise non-adjacent vertices b,c,d).

In [16], De Simone and Sassano solved the MS problem for bull- and chair-free graphs in time $\mathcal{O}(n^3)$. Note that meanwhile, in [1], Alekseev gave a polynomial time algorithm for the MWS problem on the class of chair-free graphs based on the algorithm for claw-free graphs given by Minty [25].

De Simone [15] showed that prime bull- and chair-free graphs containing a co-diamond are either bipartite or an induced cycle of odd length at least five. We will extend this line of research and show the following results, bringing together the concepts of clique width and robust algorithms which recently attracted much attention:

1. The structure of bull- and chair-free graphs is extremely simple if stability number at least four is assumed.
2. The structure of bull-, chair- and co-chair-free graphs (i.e. (bull, chair)-free graphs with (bull, chair)-free complement graph) is extremely simple.
3. Both graph classes have constant-bounded clique width.
4. We give a robust $\mathcal{O}(nm)$ time algorithm for the MWS problem on bull- and chair-free graphs i.e. if the input graph is bull- and chair-free, the algorithm correctly solves the MWS problem, and if not, the problem is solved as well or the algorithm finds out that the input graph is not (bull,chair)-free. (Such an algorithm is called *robust* in [28].)

The bounded clique width of the classes mentioned above allows to solve all algorithmic problems expressible in a certain kind of monadic second-order logic in linear time [11], among them the MWS problem. The notion of clique width has been introduced in [10] and is intimately related to modular decomposition of a graph.

2. Notions and preliminary results

Throughout this paper, let $G=(V,E)$ be a finite undirected graph without self-loops and multiple edges and let $|V|=n$, $|E|=m$. The edges between two disjoint vertex sets X,Y form a *join* (*co-join*) if for all pairs $x\in X$, $y\in Y$, $xy\in E$ ($xy \notin E$) holds. A vertex $z\in V$ *distinguishes* vertices $x,y\in V$ if $zx\in E$ and $zy \notin E$. A vertex set $M\subseteq V$ is a *module* if no vertex from $V\setminus M$ distinguishes two vertices from M i.e. every vertex $v\in V\setminus M$ has either a join or a co-join to M. A module is *trivial* if it is either the empty set, a one-vertex set or the entire vertex set V. Non-trivial modules are called *homogeneous sets*. A graph is *prime* if it contains only trivial modules. The notion of modules plays a crucial role in the *modular* (or *substitution*) *decomposition* of graphs (and other discrete structures) which is of basic importance for the design of efficient algorithms—see e.g. [26] for modular decomposition of discrete structures and its algorithmic use.

Recently, the modular decomposition of graphs attracted much attention. A homogeneous set M is *maximal* if no other homogeneous set properly contains M. It is well known that in a connected graph G with connected complement $\bar{G}$, the maximal homogeneous sets are pairwise disjoint which means that every vertex is contained in at most one maximal homogeneous set. The existence and uniqueness of the *modular decomposition tree* is based on this property, and recently, linear time algorithms were designed to determine this tree—see [24,13,14]. The tree contains the vertices of the graph as its leaves, and the internal nodes are of three types: they represent a join or co-join operation, or a prime subgraph. The graph G^* obtained from G by contracting every maximal homogeneous set to a single vertex is called the *characteristic graph* of G. It is not hard to see that G^* is connected and prime.

Let $N(v) := \{u : u \in V, u \neq v, uv \in E\}$ denote the *neighborhood* of v and $\bar{N}(v) := V \setminus (N(v) \cup \{v\})$ the *non-neighborhood* of v. For $U \subseteq V$ let $G(U)$ denote the subgraph of G induced by U. Throughout this paper, all subgraphs are understood as induced subgraphs. A vertex set $U \subseteq V$ is *stable* (sometimes called *independent*) in G if the vertices in U are pairwise non-adjacent. Let $\bar{G} = (V, \bar{E})$ denote the complement graph of G. A vertex set $U \subseteq V$ is a *clique* in G if U is a stable set in $\bar{G}$.

For a vertex weight function w on V, let $\alpha_w(G)$ denote the maximum weight sum of a stable set in G and let $\omega_w(G) := \alpha_w(\bar{G})$ denote the maximum weight of a clique in G. If $w(v) = 1$ for all vertices v then we omit the index w.

For $k \geqslant 1$, let P_k denote an induced chordless path with k vertices and $k-1$ edges, and for $k \geqslant 3$, let C_k denote an induced chordless cycle with k vertices and k edges. For a P_4 with vertices a, b, c, d and edges ab, bc, cd, the vertices a and d (b and c) are called the *endpoints* (*midpoints*) of the P_4. Note that the P_4 is the smallest non-trivial prime graph and the complement of a P_4 is a P_4 itself (where midpoints and endpoints change their roles).

The *bull* is a 5-vertex graph consisting of a P_4 and a vertex adjacent to its midpoints and non-adjacent to its endpoints, and the *chair* is a 5-vertex graph consisting of a P_4 and a vertex adjacent to one of its midpoints and non-adjacent to its endpoints and the other midpoint. Note that the complement of a bull is a bull itself. The *gem* is a 5-vertex graph consisting of a P_4 and a vertex adjacent to its midpoints and endpoints, and the *co-gem* is its complement graph consisting of a P_4 plus an isolated vertex. See Fig. 1 for all these graphs. The *diamond* is the $K_4 - e$ i.e. a 4-vertex clique minus one edge.

Let $\mathscr{F}$ denote a set of graphs. A graph G is *$\mathscr{F}$-free* if none of its induced subgraphs is in $\mathscr{F}$. There are many papers on the structure and algorithmic use of prime $\mathscr{F}$-free graphs for $\mathscr{F}$ being a set of P_4 extensions; see e.g. [21,22,17–19,23,3–5].

A graph is *matched co-bipartite* if its vertex set is partitionable into two cliques C_1, C_2 with $|C_1| = |C_2|$ or $|C_1| = |C_2| - 1$ such that the edges between C_1 and C_2 form a matching and at most one vertex in C_1 and C_2 is not covered by the matching. Complements of matched co-bipartite graphs will be called *co-matched bipartite* graphs. Examples are the C_6 as a co-matched bipartite graph and its complement, the $\overline{C_6}$ as a matched co-bipartite graph.

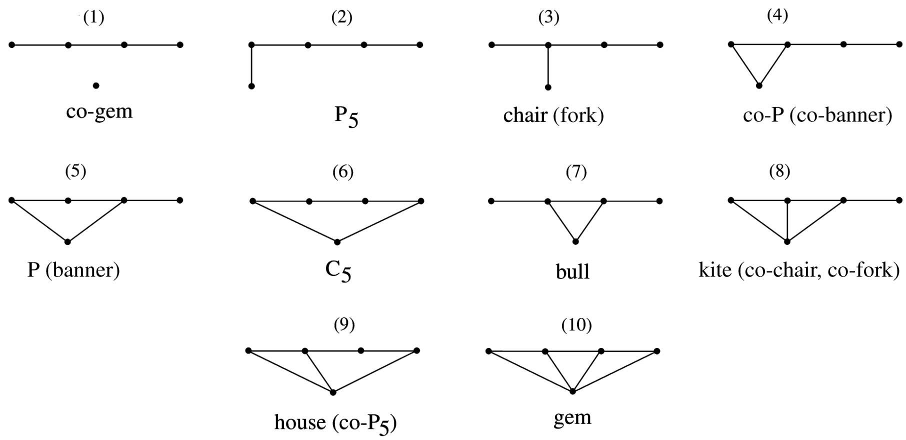

Fig. 1. All one-vertex extensions of a P_4.

If the graph is partitioned into a clique and a stable set instead of two cliques (two stable sets), we get the following notion: A graph is a *thin spider* if its vertex set is partitionable into a clique C and a stable set S such that the edges between C and S form a matching covering all vertices from S, and at most one vertex in C (the *head of the spider*) is not covered by the matching. G is a *thick spider* if it is the complement of a thin spider

Subsequently, we use the following property which follows from results in [15] (see also [16]):

Lemma 1 (Simone [15]). *If a prime bull- and chair-free graph contains an induced co-diamond then it is bipartite or an induced odd cycle* C_{2k+1}, $k \geqslant 3$.

3. Structure of prime (bull,chair)-free graphs G

We first consider the case of prime chair-free bipartite graphs which can be described as follows:

Lemma 2. *Every prime chair-free bipartite graph is co-matched bipartite, a path, or a cycle.*

Proof. Let $G = (A \cup B, E)$ be a prime, chair-free bipartite graph. Let x be a vertex of maximum degree, say $x \in A$. If $d(x) \leqslant 2$ we are done. So, assume that $|N(x)| \geqslant 3$. Note that no two vertices in G have the same neighborhood (two such vertices would form a homogeneous set), so A has the following partition:

$$A = \{x\} \cup Y \cup Z,$$

where $Y := \{v \in A : v$ is adjacent to a vertex but not to all vertices in $N(x)\}$ and $Z := A - (Y \cup \{x\})$. Y is non-empty otherwise $N(x)$ would be a homogeneous set.

Now,

$$\text{Every vertex } y \in Y \text{ is non-adjacent to exactly one vertex in } N(x). \tag{1}$$

If y is non-adjacent to distinct vertices $b, b' \in N(x)$ then y, b, b', x and any vertex in $N(x)$ adjacent to y form a chair. Thus, y is non-adjacent to at most one and hence exactly one vertex in $N(x)$.

For every $y \in Y$ let $b_y \in N(x)$ be the vertex non-adjacent to y.

$$\forall y, y' \in Y, \quad y \neq y' \Rightarrow b_y \neq b_{y'}. \tag{2}$$

Otherwise, $y, y', x, b_y = b_{y'}, b$ form a chair for any vertex $b \in N(x) - \{b_y\}$.

It follows from (1) and (2) that

$$|Y| \leqslant |N(x)|. \tag{3}$$

Moreover,

$$|Y| \geqslant |N(x)| - 1. \tag{4}$$

Otherwise, by (1) and (2), $N(x) - \{b_y : y \in Y\}$ would consist of $|N(x)| - |Y| \geqslant 2$ vertices and it would be a homogeneous set in G.

$$\forall b \in B - N(x): \text{ If } b \text{ is adjacent to a vertex in } Y,\ b \text{ is adjacent to all of them.} \tag{5}$$

Assume that b is adjacent to $y \in Y$ and non-adjacent to $y' \in Y$. As $|N(x)| \geqslant 3$, there exists a vertex $c \in N(x) - \{b_y, b_{y'}\}$. But then x, c, y, y' and b form a chair.

$$Z = \emptyset. \tag{6}$$

Otherwise, by the connectedness of G, there exist vertices $z \in Z$, $b \in B - N(x)$ such that z and b are adjacent and b is adjacent to a vertex in Y. By (5) b is adjacent to all vertices in Y. By (4) there are two distinct vertices $y, y' \in Y$. Thus, y, y', b, z and b_y form a chair.

$$|B - N(x)| \leqslant 1. \tag{7}$$

Otherwise, by (5) and (6), and the connectedness of G, $B - N(x)$ would be a homogeneous set in G.

Now, by (1)–(7), G is a co-matched bipartite graph. □

Lemmas 1 and 2 imply:

Corollary 1. *If G is a prime $(bull, chair)$-free graph containing a co-diamond then G is co-matched bipartite or a path or cycle.*

Theorem 1. *If G is a prime $(bull, chair)$-free graph with $\alpha(G) \geqslant 4$ then G is co-matched bipartite or a path or cycle.*

Proof. First, if G contains an induced co-diamond then, by Lemma 1, G is bipartite or a cycle of odd length and we are done by Lemma 2.

Now assume that G is co-diamond-free. We are going to show that G is bipartite and then we are done by Lemma 2 again.

Let S be a maximal stable set with at least 4 vertices in G, let U be the set of all vertices in G adjacent to all vertices in S, and let $T := V(G) - (S \cup U)$. By definition of S and U, every vertex in T is adjacent to a vertex in S but not to all vertices in S.

$$\text{Every vertex } x \in T \text{ is non-adjacent to exactly one vertex } S. \tag{8}$$

If $x \in T$ is non-adjacent to different vertices $a, b \in S$ then x, a, b together with a neighbor of x in S induce a co-diamond.

For all vertices $x \in T$ let s_x be the vertex in S non-adjacent to x.

$$\forall x \in T \ \exists y \in T \quad \text{such that } s_y \neq s_x. \tag{9}$$

If for all $x, y \in T$ $s_x = s_y$ then $S - s_x$ is a homogeneous set in G.

$$\forall x, y \in T : s_x \neq s_y \Rightarrow xy \in E(G). \tag{10}$$

If $s_x \neq s_y$ but x and y are adjacent then the vertices x, y, s_x, s_y and s induce a bull, where s is any vertex in $S - \{s_x, s_y\}$.

$$\forall x, y \in T, x \neq y : s_x = s_y \Rightarrow xy \in E(G). \tag{11}$$

By (9), there exists a vertex $z \in T$ such that $s_z \neq s_x = s_y$. Hence, $zs_x \in E(G)$. By (10), z is non-adjacent to x, y. But then x, y, z, s_x induce a co-diamond if $xy \in E$.

$$\forall x, y \in T, x \neq y : s_x \neq s_y. \tag{12}$$

Assume that $s_x = s_y$. By (9), there exists a vertex $z \in T$ such that $s_z \neq s_x$. Since $S' = S - \{s_x, s_z\}$ has at least two vertices, there exists a vertex $t \in T$ with $s_t \in S'$ otherwise S' would be a homogeneous set in G. By (10), $\{x, z, t\}$ and $\{y, z, t\}$ are stable sets. By (11), x and y are adjacent. Thus, x, y, z, t induce a co-diamond.

$$T \text{ is a stable set.} \tag{13}$$

This is a corollary from (12) and (10).

$$|T| \geqslant |S| - 1; \text{ in particular, } |T| \geqslant 3. \tag{14}$$

If $|T| \leqslant |S| - 2$, then $S' := S - \{s_x : x \in T\}$ has at least $|S| - (|S| - 2) = 2$ vertices, and S' would be a homogeneous set.

Now, consider the partition $U = U_1 \cup U_2$, where U_1 consists of all vertices in U adjacent to all vertices in T and $U_2 := U - U_1$.

$$\text{No vertex in } U_2 \text{ has a neighbor in } T. \tag{15}$$

Assume that there are adjacent vertices $u \in U_2$ and $x \in T$. By definition of U_2, there is a vertex $y \in T$ non-adjacent to u. By (14), there is a vertex $z \in T - \{x, y\}$. Now, u must be adjacent to z otherwise, by (13), u, x, y, z would induce a co-diamond. Thus, by (13), x, y, z, u, s_x induce a bull.

$$\text{Every vertex in } U_1 \text{ is adjacent to every vertex in } U_2. \tag{16}$$

If $u_1 \in U_1$ is non-adjacent to $u_2 \in U_2$ then, by (13) and (15), the vertices x, y, s_y, u_1, u_2 induce a bull, where x and y are arbitrary different vertices in T.

$$U_1 = \emptyset. \tag{17}$$

Otherwise, by (16) and definition of U_1, $V(G) - U_1$ would be a homogeneous set.

$$|U_2| \leqslant 1. \tag{18}$$

Otherwise, by (15), U_2 would be a homogeneous set.

Now, (13), (15) and (18) imply that G is a bipartite graph. By Lemma 2, the proof of Theorem 1 is completed. □

Theorem 2. *If G is a prime (bull, chair, co-chair)-free graph then G or $\bar{G}$ is co-matched bipartite, a path, or a cycle.*

Proof. By Theorem 1, we may assume that $\alpha(G) \leqslant 3$ as well as $\alpha(\bar{G}) \leqslant 3$. If $\alpha(G) \leqslant 2$ and $\alpha(\bar{G}) \leqslant 2$ then it is well known that G has at most five vertices. Since G is prime, G or $\bar{G}$ must be a path P_4 or the cycle C_5 and we are done.

Thus, we may assume that $\alpha(G) = 3$ or $\alpha(\bar{G}) = 3$. By symmetry, $\alpha(G) = 3$, say, and assume that G is not a path or a cycle.

We are going to show that G or $\bar{G}$ is bipartite. If G is not bipartite, consider three pairwise non-adjacent vertices a, b, c in G. As G is not bipartite, there is an edge $xy \in G - \{a, b, c\}$. Since $\alpha(G) = 3$ and G has no co-diamond (by Lemma 1), each of x, y is adjacent to at least two vertices in a, b, c. Without loss of generality, let x be adjacent to a, b and let y be adjacent to a, say. Now, if y is adjacent to b then G has a diamond (i.e. $\bar{G}$ has a co-diamond), which contradicts Lemma 1. If y is adjacent to c (and non-adjacent to b) then G has a bull. This final contradiction shows that G or $\bar{G}$ must be bipartite, as claimed. By Lemma 2, G or $\bar{G}$ is a co-matched bipartite graph. □

It is not clear whether Theorem 2 implies a linear time recognition for the class of (bull,chair,co-chair)-free graphs; however, the class is contained in the following larger class $\mathscr{C}$ having linear time recognition:

For this purpose, we need the notion of p-connectedness; a graph $G = (V, E)$ is *p-connected* if for every partition $V = V_1 \cup V_2$ with non-empty V_1 and V_2, there is a P_4 with vertices in V_1 and in V_2 (see [2] for a survey on p-connectedness). In the usual way, *p-connected components* of a graph are defined.

Let $\mathscr{C}$ denote the class of graphs G for which every p-connected component of G has the property that its homogeneous sets are P_4-free and its characteristic graph G^* or its complement $\overline{G^*}$ is a

(i) matched co-bipartite graph or
(ii) induced path P_k, $k \geqslant 4$ or
(iii) induced cycle C_k, $k \geqslant 5$.

Due to Theorem 2, the characteristic graphs of (bull,chair,co-chair)-free graphs have one of these types or its complement. Moreover, it is easy to see that in p-connected (chair,co-chair)-free graphs, the homogeneous sets are P_3-free and thus P_4-free.

Note that $\mathscr{C}$ has a linear time recognition algorithm: In [24,13,14], linear time algorithms for constructing the modular decomposition tree, finding the maximal homogeneous sets (and thus, constructing the characteristic graph) are given. This applies also to finding the p-connected components of a graph (see [2]).

4. Cographs, clique width and algorithmic problems

The P_4-free graphs (also called *cographs*) play a fundamental role for graph decomposition; see [6] for a survey on this graph class and related ones. For a cograph G, either G or its complement is disconnected, and the *cotree* of G expresses how the graph can be recursively generated from single vertices by repeatedly applying join and co-join operations.

The cotree representation allows to solve various NP-hard problems in linear time when restricted to cographs, among them the problems maximum weight stable set and maximum weight clique. See [9] for linear time recognition of cographs and [7–9,6] for more informations on P_4-free graphs. Note that the cographs are those graphs whose modular decomposition tree contains only join and co-join nodes as internal nodes.

Based on three operations on vertex-labeled graphs, namely,

- disjoint union (i.e. co-join),
- join between all vertices with label i and all vertices with label j for $i \neq j$, and
- relabeling vertices of label i by label j,

Courcelle et al. [10] introduced the notion of *clique width* $cwd(G)$ of a graph G as the minimum number of labels which are necessary to generate a given graph by using the three operations. Obviously, the clique width of cographs is at most two. A k-*expression* for a graph G of clique width k describes the recursive generation of G by repeatedly applying the three operations using only a set of at most k different labels.

Proposition 1 (Courcelle et al. [11], Courcelle and Olariu [12]). *The clique width of a graph is the maximum of the clique width of its prime graphs, and the clique width of the complement graph $\bar{G}$ of a graph G is at most twice the clique width of G.*

Recently, the concept of clique width of a graph attracted much attention since it gives a unified approach to the efficient solution of many algorithmic graph problems on graph classes of bounded clique width via the expressibility of the problems in terms of logical expressions; in [11], it is shown that every algorithmic problem expressible in a certain kind of monadic second-order logic called $LinEMSOL(\tau_{1,L})$ in [11], is linear time solvable on any graph class with bounded clique width for which a k-expression can be constructed in linear time.

Hereby, in [11] it is mentioned that, roughly speaking, $MSOL(\tau_1)$ is monadic second-order logic with quantification over subsets of vertices but not of edges; $MSOL(\tau_{1,L})$ is the extension of $MSOL(\tau_1)$ with the addition of labels added to the vertices.

LinEMSOL($\tau_{1,L}$) is the extension of *MSOL*($\tau_{1,L}$) which allows to search for sets of vertices which are optimal with respect to some linear evaluation functions. The maximum weight stable set problem is an example of a *LinEMSOL*($\tau_{1,L}$) problem.

Theorem 3 (Courcelle et al. [11]). *Let $\mathscr{C}$ be a class of graphs of clique width at most k such that there is a* (*known*) $\mathcal{O}(f(|E|),|V|)$ *algorithm, which for each graph G in $\mathscr{C}$, constructs a k-expression defining it. Then every LinEMSOL*($\tau_{1,L}$) *problem on $\mathscr{C}$ can be solved* (*constructively*) *in time* $\mathcal{O}(f(|E|),|V|)$.

As an application, in [11] it was shown that P_4-sparse graphs and some variants of their variants have bounded clique width. Hereby, a graph is P_4-*sparse* if no set of five vertices in G induces at least two distinct P_4's [21,22]. From the definition, it is obvious that a graph is P_4-sparse if and only if it contains no C_5, P_5, $\overline{P_5}$, P, $\bar{P}$, chair, co-chair (see Fig. 1).

In [21], it was shown that the prime P_4-sparse graphs are the spiders (which were called *turtles* in [21]), and according to Proposition 1 and the fact that the clique width of spiders is bounded by 4 (which is easy to see), it follows that P_4-sparse graphs have bounded clique width.

In a similar way, other examples lead to bounded clique width. Recently, variants of P_4-sparse graphs attracted much attention because of their applications in areas such as scheduling, clustering and computational semantics. Moreover, all these classes are natural generalizations of cographs.

To relate this approach to our graph classes, note that the clique width of matched co-bipartite graphs and their complements as well as the clique width of induced paths and cycles is at most 4, which is straightforward to see. According to Proposition 1, Theorem 2 implies:

Corollary 2. *The clique width of a* (*bull, chair, co-chair*)*-free graph is bounded by at most* 8, *and an* 8-*expression of such a graph can be constructed in linear time*.

Thus, every *LinEMSOL*($\tau_{1,L}$) problem can be solved in linear time on this graph class if the input graph is known to be (bull,chair,co-chair)-free.

To get linear time *robust* algorithms for *LinEMSOL*($\tau_{1,L}$) expressible problems on (bull,chair,co-chair)-free graphs, we turn over to the larger class $\mathscr{C}$ described in Section 3 since we do not have linear time recognition for (bull,chair,co-chair)-free graphs. The problem solving algorithm first checks whether the input graph is in $\mathscr{C}$, and if not, it is in particular not (bull,chair,co-chair)-free. Otherwise, the clique width of G is bounded by 8, and an 8-expression can be constructed in linear time. Thus, the total time bound for every input graph is linear.

Note that the clique width of (bull,chair)-free graphs is unbounded since every co-bipartite graph is (bull,chair)-free, and in [20] it was shown that an $n \times n$ square grid has clique width $n+1$ which means that bipartite graphs and thus also co-bipartite graphs have unbounded clique width.

5. A robust algorithm for the maximum weight stable set problem on (bull,chair)-free graphs

Let G be an arbitrary input graph. We want to solve the MWS problem for G in such a way that in the case that G is (bull,chair)-free, the algorithm determines $\alpha_w(G)$, and in the other case, the algorithm either determines $\alpha_w(G)$ as well or detects that G is not (bull,chair)-free.

Note that for every graph

$$\alpha_w(G) = \max\{w(v) + \alpha_w(G(\bar{N}(v))) : v \in V\}$$

holds, and for co-gem-free graphs, there is an obvious way to determine $\alpha_w(G)$ in $\mathcal{O}(nm)$ time using the fact that for every vertex v, $G(\bar{N}(v))$ is P_4-free, and for P_4-free graphs, the problem is solvable in linear time.

Algorithm 1 assumes that the input graph G is prime and consists of the following steps:

Algorithm 1. *Input*: A prime graph G with vertex weight function w.
Output: $\alpha_w(G)$ or the answer that G is not bull- or chair-free.

(1) Check whether G is co-gem-free i.e. check for all $v \in V$ whether $G(\bar{N}(v))$ is a cograph.
(2) If yes then for every $v \in V$, determine $\alpha_w(G(\bar{N}(v)))$; now $\alpha_w(G) = \max\{w(v) + \alpha_w(G(\bar{N}(v))) : v \in V\}$
(3) If not (i.e. G contains a co-gem and thus a co-diamond) then check whether G is co-matched bipartite or an induced path or cycle.
 (3.1) If not then G is not (bull,chair)-free.
 (3.2) If yes (i.e. G is co-matched bipartite or an induced path or cycle) then $\alpha_w(G)$ can be computed in an obvious way.

Lemma 3. *Algorithm* 1 *is correct and has time bound* $\mathcal{O}(nm)$.

Proof. The correctness of this algorithm follows from Corollary 1. Step (1) takes $\mathcal{O}(nm)$ time since for every vertex $v \in V$, it can be tested in $\mathcal{O}(m)$ time whether $\bar{N}(v)$ is a cograph, and if yes, $\alpha_w(G(\bar{N}(v)))$ can be determined in $\mathcal{O}(m)$ time using the cotree representation of $G(\bar{N}(v))$. Thus, step (2) can be carried out altogether in time $\mathcal{O}(nm)$. It can be checked in linear time whether a graph is co-matched bipartite or an induced path or cycle. If yes then $\alpha_w(G)$ can be computed directly in an obvious way in $\mathcal{O}(m)$ steps. □

If an arbitrary graph G with vertex weight function w is given, we first construct the modular decomposition tree of G and apply Algorithm 1 repeatedly in a bottom-up way to the prime nodes of the tree. For join and co-join nodes, there is an obvious formula for computing their weighted stability number.

In this well-known way, we obtain a robust algorithm which either determines $\alpha_w(G)$ or answers that G is not bull- or chair-free.

References

[1] V.E. Alekseev, A polynomial algorithm for finding maximum independent sets in fork-free graphs, Discrete Anal. Oper. Res. 6 (4) (1999) 3–19.

[2] L. Babel, S. Olariu, On the p-connectedness of graphs: a survey, Discrete Appl. Math. 95 (1999) 11–33.

[3] A. Brandstädt, (P_5, diamond)-Free graphs revisited: structure and linear time optimization, Discrete Applied Mathematics, in press.

[4] A. Brandstädt, V. Giakoumakis, J.-M. Vanherpe, On prime (P_5,claw)-free, (P_5,bull)-free, and (bull,claw)-free graphs and the maximum stable set problem, Manuscript, 2000.

[5] A. Brandstädt, D. Kratsch, On (P_5,gem)-free and (P_5,gem,co-gem)-free graphs: structure and algorithmic applications, Manuscript, 2000.

[6] A. Brandstädt, V.B. Le, J. Spinrad, Graph classes: a survey, SIAM Monographs on Discrete Mathematics Applications, Vol. 3, SIAM, Philadelphia, PA, 1999.

[7] D.G. Corneil, H. Lerchs, L. Stewart-Burlingham, Complement reducible graphs, Discrete Appl. Math. 3 (1981) 163–174.

[8] D.G. Corneil, Y. Perl, L.K. Stewart, Cographs: recognition, applications, and algorithms, Congr. Numer. 43 (1984) 249–258.

[9] D.G. Corneil, Y. Perl, L.K. Stewart, A linear recognition algorithm for cographs, SIAM J. Comput. 14 (1985) 926–934.

[10] B. Courcelle, J. Engelfriet, G. Rozenberg, Handle-rewriting hypergraph grammars, J. Comput. System Sci. 46 (1993) 218–270.

[11] B. Courcelle, J.A. Makowsky, U. Rotics, Linear time solvable optimization problems on graphs of bounded clique width, Theory Comput. Systems 33 (2000) 125–150.

[12] B. Courcelle, S. Olariu, Upper bounds to the clique width of graphs, Discrete Appl. Math. 101 (2000) 77–114.

[13] A. Cournier, M. Habib, A new linear algorithm for modular decomposition, LIRMM, University Montpellier, 1995 (Preliminary version in: Trees in Algebra and Programming—CAAP '94, Lecture Notes in Computer Science, Vol. 787, Springer, Berlin, 1994, pp. 68–84).

[14] E. Dahlhaus, J. Gustedt, R.M. McConnell, Efficient and practical modular decomposition, Technical Report, TU Berlin FB Mathematik, 524/1996 (1996), in: Conference Proceedings of the Eighth SODA, 1997, pp. 26–35.

[15] C. De Simone, On the vertex packing problem, Graphs Combin. 9 (1993) 19–30.

[16] C. De Simone, A. Sassano, Stability number of bull- and chair-free graphs, Discrete Appl. Math. 41 (1993) 121–129.

[17] J.-L. Fouquet, A decomposition for a class of (P_5, $\overline{P_5}$)-free graphs, Discrete Math. 121 (1993) 75–83.

[18] J.-L. Fouquet, V. Giakoumakis, H. Thuillier, F. Maire, On graphs without P_5 and $\overline{P_5}$, Discrete Math. 146 (1995) 33–44.

[19] V. Giakoumakis, I. Rusu, Weighted parameters in (P_5, $\overline{P_5}$)-free graphs, Discrete Appl. Math. 80 (1997) 255–261.

[20] M.C. Golumbic, U. Rotics, On the clique-width of perfect graph classes, Internat. J. Found. Comput. Sci. 11 (2000) 423–443.

[21] C.T. Hoàng, A class of perfect graphs, M.Sc. Thesis, School of Computer Science, McGill University, Montreal, 1983.

[22] C.T. Hoàng, Perfect graphs, Ph.D. Thesis, School of Computer Science, McGill University, Montreal, 1985.

[23] B. Jamison, S. Olariu, A unique tree representation for P_4-sparse graphs, Discrete Appl. Math. 35 (1992) 115–129.

[24] R.M. McConnell, J. Spinrad, Linear-time modular decomposition and efficient transitive orientation of comparability graphs, Proceedings of the Fifth Annual ACM–SIAM Symposium on Discrete Algorithms, Arlington, Virginia, 1994, pp. 536–543.

[25] G.J. Minty, On maximal independent sets of vertices in claw-free graphs, J. Combin. Theory Ser. B 28 (1980) 284–304.

[26] R.H. Möhring, F.J. Radermacher, Substitution decomposition for discrete structures and connections with combinatorial optimization, Ann. Discrete Math. 19 (1984) 257–356.
[27] N. Sbihi, Algorithme de recherche d'un stable de cardinalité maximum dans un graphe sans étoile, Discrete Math. 29 (1980) 53–76.
[28] J.P. Spinrad, Representations of graphs, Book Manuscript, Vanderbilt University, Nashville, TN, 1998.