Discrete Mathematics and Theoretical Computer Science DMTCS vol. **15:3**, 2013, 13–20

Clique cycle transversals in graphs with few P_4's

Raquel S. F. Bravo[1†] Sulamita Klein[2‡] Loana Tito Nogueira[1§] Fábio Protti[1¶]

[1] *Fluminense Federal University, Niterói, RJ, Brazil*
[2] *Federal University of Rio de Janeiro, RJ, Brazil*

received 26th Sep. 2012, *revised 27th June 2013, 20th Aug. 2013*, *accepted 16th Sep. 2013*.

A graph is extended P_4-laden if each of its induced subgraphs with at most six vertices that contains more than two induced P_4's is $\{2K_2, C_4\}$-free. A cycle transversal (or feedback vertex set) of a graph G is a subset $T \subseteq V(G)$ such that $T \cap V(C) \neq \emptyset$ for every cycle C of G; if, in addition, T is a clique, then T is a clique cycle transversal (cct). Finding a cct in a graph G is equivalent to partitioning $V(G)$ into subsets $\mathcal{C}$ and $\mathcal{F}$ such that $\mathcal{C}$ induces a complete subgraph and $\mathcal{F}$ an acyclic subgraph. This work considers the problem of characterizing extended P_4-laden graphs admitting a cct. We characterize such graphs by means of a finite family of forbidden induced subgraphs, and present a linear-time algorithm to recognize them.

Keywords: Clique, Cycle Transversal, Extended P_4-laden graph, Feedback Vertex Set

1 Introduction

A *cycle transversal* or *feedback vertex set* of a graph G is a subset $T \subseteq V(G)$ such that $T \cap V(C) \neq \emptyset$ for every cycle C of G. In other words, the removal of a cycle transversal leaves a graph without cycles. The problem of finding a minimum cycle transversal in an arbitrary graph G is a classical NP-hard problem [6, 9, 12] which has been extensively studied in several fields related to algorithms and complexity. It has first appeared within the context of combinatorial circuit design, and has applications in deadlock prevention in operating systems, database systems, genome assembly, constraint satisfaction, and Bayesian inference in artificial intelligence. We refer to the survey by Festa, Pardalos, and Resende [4] for further details on the algorithmic study of cycle transversal problems in a variety of areas, including approximation algorithms, linear programming and polyhedral combinatorics. In [1], the authors extend the NP-hardness of finding minimum cycle transversals in general graphs to bipartite graphs with maximum degree four. Algorithmic issues involving C_k transversals (for fixed k) in graphs with bounded degree, where C_k denotes a chordless cycle with k vertices, are discussed in [8].

If a cycle transversal T is also a clique, we say that T is a *clique cycle transversal*, or simply *cct*. A graph admits a cct if and only if it can be partitioned into a complete subgraph and a forest; by this reason such a graph is called a $(\mathcal{C}, \mathcal{F})$-*graph*. Clique cycle transversals are studied in the more general context of graph partitions, especially *sparse-dense partitions* [3]: if T is a cct in a graph G then $G[T]$ is the 'dense' part, and $G - T$ the 'sparse' part. In [1], the authors describe a polynomial-time algorithm for recognizing graphs with cct, and a linear-time algorithm for cographs with cct based on a characterization of such cographs in terms of forbidden induced subgraphs. Distance-hereditary graphs with cct have been investigated in [2].

Denote by P_4 a graph with four vertices a, b, c, d and edges ab, bc, cd. This work considers a class of graphs with few P_4's, the extended P_4-laden graphs. We say that a graph is extended P_4-laden if each of its induced subgraphs with at most six vertices that contains more than two induced P_4's is $\{2K_2, C_4\}$-free. This class strictly contains the class of P_4-sparse graphs e, consequently, cographs. Our main result is a characterization of extended P_4-laden graphs with cct in terms of a family of twelve forbidden induced subgraphs. We also describe a linear-time algorithm for recognizing this class of graphs as a consequence of the given characterization. These results extend the previous ones for the class of cographs.

†Email: raquelbr@cos.ufrj.br. Supported by CAPES and FAPERJ.
‡Email: sula@cos.ufrj.br. Partially supported by CNPq and FAPERJ.
§Email: loana@ic.uff.br. Partially supported by CNPq and FAPERJ.
¶Email: fabio@ic.uff.br. Partially supported by FAPERJ.

2 Background

For standard definitions and notation in graph theory, see [11].

Extended P_4-laden graphs were characterized by [7] in terms of a decomposition technique and special graphs, called *quasi-spiders* and *pseudo-split graphs*. Extended P_4-laden graphs can be represented by a tree decomposition; the resulting tree is called the *primeval tree*.

We say that a graph G is *p-connected* if for every partition of $V(G)$ into nonempty disjoint sets A and B there exists a crossing P_4, that is, a P_4 containing vertices from both A and B. The *p-connected components*, or simply *p-components*, of a graph are the maximal induced subgraphs which are p-connected. It is worth mentioning that a p-component has either one vertex or at least four vertices [7]. A p-connected component H of a graph G is said to be *trivial* if $|H| = 1$; otherwise, H is *nontrivial*.

A p-connected graph G is said to be *separable* if there exists a partition of $V(G)$ into nonempty disjoint sets V_1 and V_2 such that each P_4 which contains vertices from both sets has its endpoints in V_2 and its midpoints in V_1. In this case, we say that G has a *separation* (V_1, V_2).

The following theorem gives us information on the structure of extended P_4-laden graphs.

Theorem 1 [10] *Let $G = (V, E)$ be a graph. Then exactly one of the following statements holds:*

1. *G is disconnected;*
2. *$\overline{G}$ is disconnected;*
3. *There exists a unique separable p-component H with separation (H_1, H_2) such that every vertex outside H is adjacent to all vertices in H_1 and to no vertex in H_2;*
4. *G is p-connected.*

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be disjoint graphs. The *union* and the *join* of G_1 and G_2 are graphs resulting, respectively, from the following operations:

- $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$
- $G_1 + G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup \{xy \mid x \in V_1, y \in V_2\})$

Operations $\cup$ and $+$ are related to the first two cases of Theorem 1.

Let $G_1 = (V_1, E_1)$ be a separable p-connected graph with separation (V_1^1, V_1^2) and $G_2 = (V_2, E_2)$ be an arbitrary graph disjoint from G_1. The third case of Theorem 1 is represented by the following operation:

- $G_1 \, \mathcal{P} \, G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup \{xy \mid x \in V_1^1, y \in V_2\})$

As described by Jamison *et al.* [10], every graph G is either a p-connected graph or can be uniquely obtained from its p-components and its weak vertices (i.e, those vertices that are not contained in any nontrivial p-component of G) by a finite sequence of operations $\cup$, $+$, and $\mathcal{P}$. Moreover, Theorem 1 suggests a way to represent an arbitrary graph G by the primeval tree T_G, which is unique up to isomorphism. Each internal node q of T_G receives a label $i \in \{\cup, +, \mathcal{P}\}$. Such a label indicates that the subgraph associated with the subtree rooted at q is obtained by performing an i-operation on the children of q. The leaves of the tree are the p-components of G.

Figure 1 represents a graph G and its primeval tree T_G.

We say that G is a *spider* if $V(G)$ can be partitioned into three subsets $\mathcal{S}$, $\mathcal{K}$, and $\mathcal{R}$ such that $\mathcal{S}$ is an independent set, $\mathcal{K}$ is a clique, $|\mathcal{S}| = |\mathcal{K}| \geq 2$, and there exists a bijective function $f : \mathcal{S} \to \mathcal{K}$ such that either $N_G(v) = \{f(v)\}$, for every $v \in \mathcal{S}$ (*thin spider*), or $N_G(v) = \mathcal{K} - \{f(v)\}$, for every $v \in \mathcal{S}$ (*thick spider*); in addition, every vertex in $\mathcal{R}$ is adjacent to every vertex in $\mathcal{K}$ and non-adjacent to every vertex in $\mathcal{S}$. We say that $\mathcal{S}$ and $\mathcal{K}$ are, respectively, the *legs* and *body* of the spider, and $\mathcal{R}$ is the *head* of the spider. Figure 2 depicts a thick and a thin spider.

A graph $G = (V, E)$ is *split* if V can be partitioned into a clique $\mathcal{K}$ and an independent set $\mathcal{S}$ (with no restrictions on the edges between $\mathcal{S}$ and $\mathcal{K}$), i.e., $G = (\mathcal{S} \cup \mathcal{K}, E)$. For simplicity, we write $G = (\mathcal{S}, \mathcal{K})$ when the edge set E is irrelevant for our purposes. It is worth observing that such a partition is not necessarily unique. It has been proved [5] that a graph G is split if and only it does not contain C_5, C_4, or $\overline{C_4} = 2K_2$ as an induced subgraph.

Given a split graph $G = (\mathcal{S}, \mathcal{K})$, we say that G is *original* if every vertex of $\mathcal{S}$ has a non-neighbor in $\mathcal{K}$ and every vertex of $\mathcal{K}$ has a neighbor in $\mathcal{S}$.

A graph G is *pseudo-split* if its vertex set has a partition $(\mathcal{S}, \mathcal{K}, \mathcal{R})$ such that $\mathcal{S}$ is an independent set, $\mathcal{K}$ is a clique, and every vertex of $\mathcal{R}$ is adjacent to every vertex of $\mathcal{K}$ and non-adjacent to every vertex of $\mathcal{S}$, and $\mathcal{S} \cup \mathcal{K}$ induces a split graph. As for the spider, $\mathcal{S}$, $\mathcal{K}$, and $\mathcal{R}$ can be seen, respectively, as the legs, body and head of G; if $\mathcal{R}$ is empty, then G is said to be a *headless pseudo-split graph*.

Observe that the complement of a pseudo-split graph is also a pseudo-split graph. Moreover, every spider is a pseudo-split graph. A *quasi-spider* is either a spider or a graph obtained from a spider $\mathscr{S} = (\mathcal{S}, \mathcal{K}, \mathcal{R})$ by

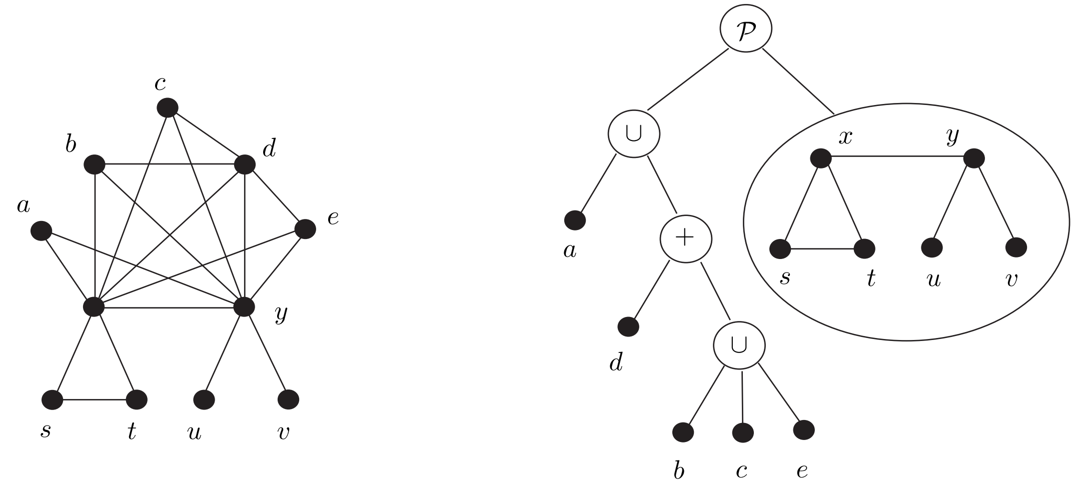

Fig. 1: A graph G and its primeval tree T_G.

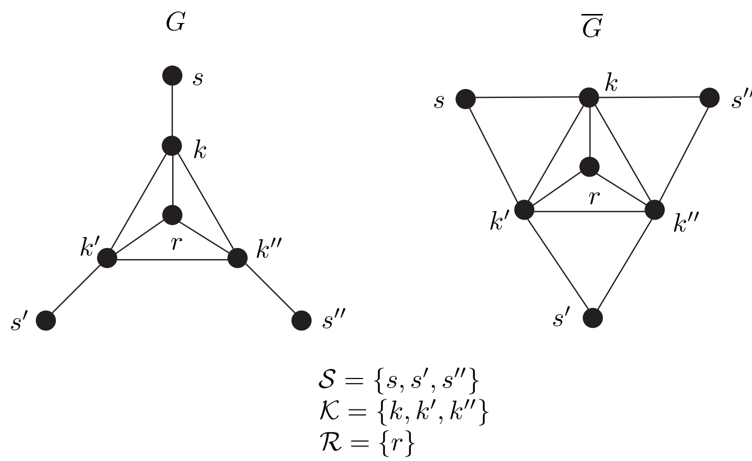

Fig. 2: G is a thin spider and $\overline{G}$ is a thick spider.

replacing exactly one vertex $v \in \mathcal{S} \cup \mathcal{K}$ by a graph H isomorphic to K_2 or $\overline{K_2}$, where vertices of H have the same neighborhood as $v \in \mathcal{S} \cup \mathcal{K}$. The vertex v that has been replaced by a K_2 or I_2 is called a *q-vertex*. A quasi-spider is a *thin quasi-spider* if it is originated from a thin spider, otherwise it is a *thick quasi-spider*. Figure 3 shows two examples of quasi-spiders: a thin one, with a vertex from set $\mathcal{S}$ replaced by K_2, and a thick one, with a vertex from set $\mathcal{K}$ replaced by $\overline{K_2}$.

The following theorem characterizes extended P_4-laden graphs.

Theorem 2 [7] *A graph G is an extended P_4-laden graph if and only if for every induced subgraph H of G exactly one of the following statements holds:*

1. *H is disconnected;*
2. *$\overline{H}$ is disconnected;*
3. *H is a quasi-spider whose head induces an extended P_4-laden graph;*
4. *H is a pseudo-split graph whose head induces an extended P_4-laden graph;*
5. *H is isomorphic to C_5, P_5 or $\overline{P_5}$;*
6. *H is a trivial graph.*

3 Characterization

Now we are ready to present the main result of this work.

Theorem 3 *An extended P_4-laden graph G admits a clique cycle transversal if and only if G does not contain any of the graphs depicted in Figure 4 as an induced subgraph.*

PROOF: By testing all possible cliques in each graph H of Figure 4, we can see that none of them meets all the cycles in H. Below we apply this argument to each graph in Figure 4:

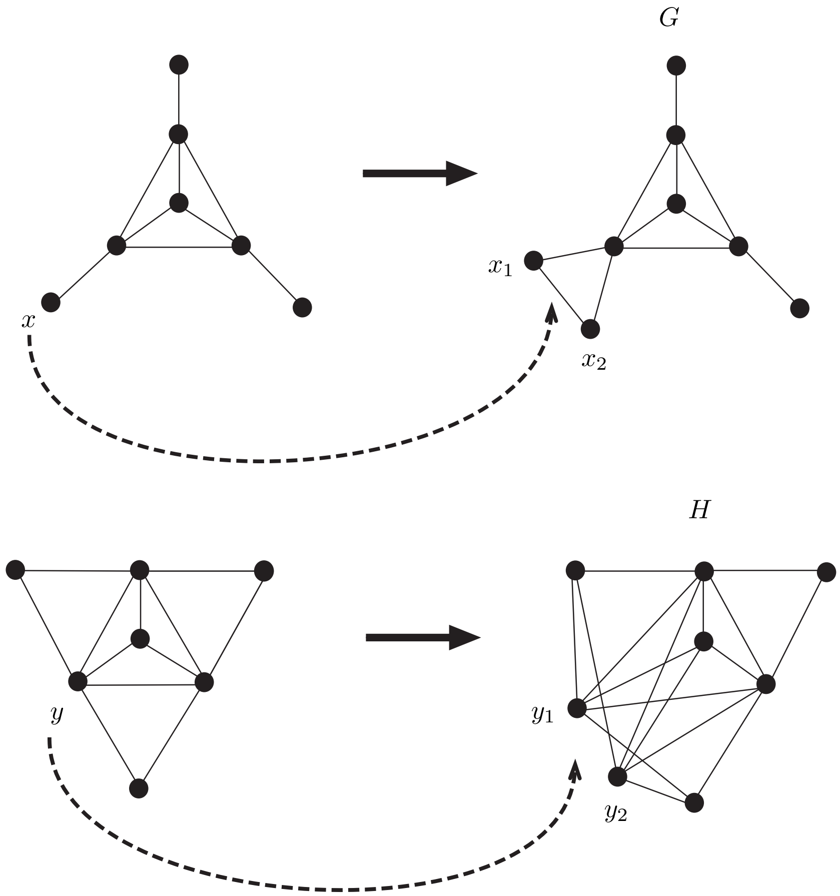

Fig. 3: G is a thin quasi-spider and H is a thick quasi-spider.

- $2C_3$: by removing any clique from this graph, there still remains a connected component which is a cycle; this argument also applies to the graphs $2C_4$, $2C_5$, $C_3 \cup C_4$, $C_3 \cup C_5$, and $C_4 \cup C_5$.
- $I_2 + I_2 + I_2$: in this graph, any clique C is formed by choosing at most one vertex of each independent set of size 2; thus, the vertices not chosen induce a graph that contains a C_3 as a subgraph.
- $2K_2 + I_2$: any clique in this graph is formed by taking one or two vertices from a copy of K_2, plus at most one vertex lying outside that copy; thus, the remaining graph still contains a C_3.
- $I_3 + I_3$: since this graph is bipartite, any clique contains at most two vertices; by removing any vertex or edge, it is easy to see that there still remains an $I_2 + I_2$ (i.e., a C_4) as a subgraph.
- $C_5 + I_2$: any clique in this graph is formed by choosing a vertex or an edge from the C_5, plus at most one vertex outside the C_5; after removing such a clique, the remaining graph contains a C_3 formed by an edge of the C_5 plus a vertex outside the C_5.
- $I_3 + P_4$: the argument for this graph is similar as above: cliques are formed by choosing a vertex or an edge from the P_4, plus at most one vertex outside the P_4; vertices not chosen induce a graph containing $I_2 + K_2$ or $I_2 + I_2$ as a subgraph.
- $P_4 + P_4$: in this graph, any clique C is formed by taking a vertex or an edge of each P_4; thus, the removal of C still a graph containing $K_2 + K_2$, $K_2 + I_2$, or $I_2 + I_2$ as a subgraph, and all of them contain at least one cycle.

Hence, if G contains some subgraph H listed in Figure 4 as an induced subgraph then G admits no cct. Conversely, suppose that G contains no induced subgraph H listed in Figure 4 as an induced subgraph, and suppose that G admits no cct. In this case, G contains a minimal induced subgraph G' with no cct, i.e., for every vertex $v \in V(G')$, $G' - v$ admits a cct. Without loss of generality, assume $G' = G$. Consider the following three cases, according to Theorem 2:

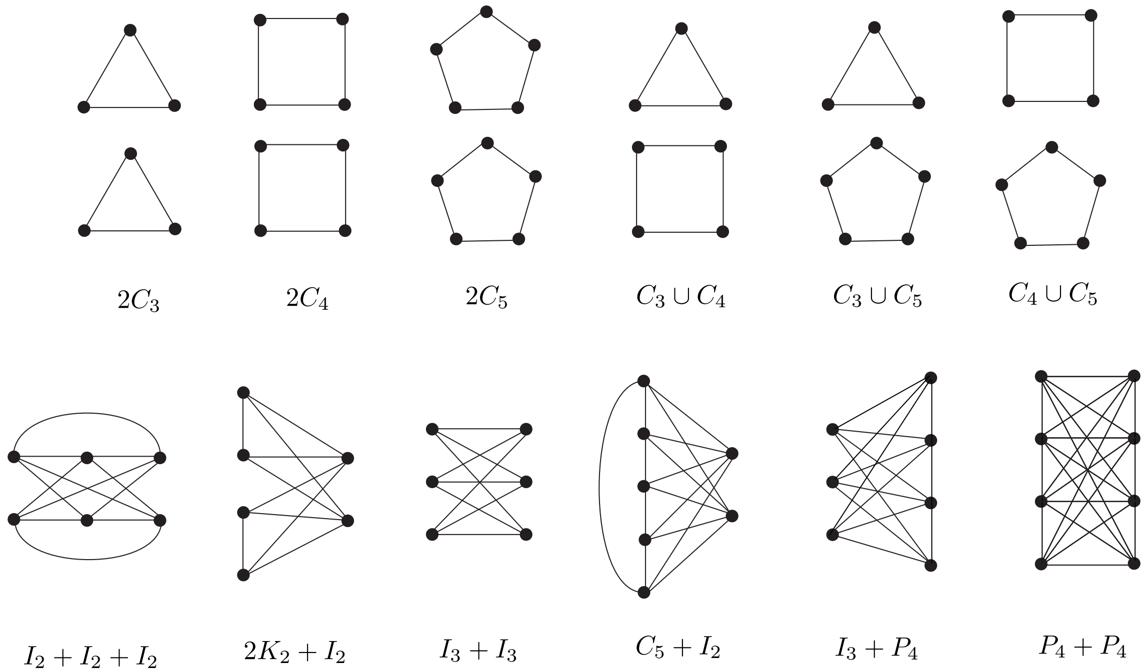

Fig. 4: Minimal forbidden subgraphs for extended P_4-laden graphs with cct.

CASE 1: G is disconnected.

Write $G = G_1 \cup G_2 \cup \dots G_k$, with $k \geq 2$ and each G_i connected. By minimality of G, each G_i admits a cct. Moreover, each G_i is not a tree, otherwise $G' = G - G_i$ would admit a cct, as well as $G = G' \cup G_i$. Then, each G_i must contain a cycle (C_3, C_4, or C_5, since extended P_4-laden graphs contain no induced C_h for all $h \geq 6$). It follows that G contains one of the following graphs as an induced subgraph: $C_3 \cup C_3$, $C_3 \cup C_4$, $C_3 \cup C_5$, $C_4 \cup C_4$, $C_4 \cup C_5$, or $C_5 \cup C_5$. This is a contradiction.

CASE 2: $\overline{G}$ is disconnected.

Write $G = G_1 + G_2 + \cdots + G_k$, where $k \geq 2$ and $\overline{G_i}$ is connected for $1 \leq i \leq k$. If some G_i is a complete subgraph, by minimality of G the graph $G - G_i$ admits a cct, as well as G. Contradiction. Hence, assume that every G_i contains a pair of non-adjacent vertices. If $k \geq 3$ then G contains $I_2 + I_2 + I_2$ as an induced subgraph, a contradiction. If $k = 2$, we consider three possibilities:

- *CASE 2.1:* Both G_1 and G_2 contain I_3. Then G contains $I_3 + I_3$ as an induced subgraph, a contradiction.

- *CASE 2.2:* Exactly one of G_1, G_2 contains I_3.

  Without loss of generality, suppose that G_1 contains I_3. Subgraph G_2 cannot contain C_5, otherwise G contains $I_2 + C_5$ as an induced subgraph. In addition, G_2 cannot contain P_4, otherwise G contains $I_3 + P_4$ as an induced subgraph. Therefore, $\overline{G_2}$ does not contain C_3, C_5, and P_4, i.e., $\overline{G_2}$ is a triangle-free connected cograph, which implies that $\overline{G_2}$ is complete bipartite. We conclude that G_2 is a union of two disjoint, nonempty cliques C and C'. Assume that $|C| \geq |C'|$. If $|C'| \geq 2$ then G contains $I_2 + 2K_2$ as an induced subgraph. Thus $|C| \geq |C'| = 1$.

  If G_1 is a split graph then $V(G_1)$ can be partitioned into an independent set S_1 and a clique C_1; hence G admits cct $C \cup C_1$, a contradiction. Therefore, G contains $I_2 + C_5$ (if G_1 contains C_5), or $I_2 + 2K_2$ (if G_1 contains $2K_2$), or $I_2 + I_2 + I_2$ (if G_1 contains C_4) as an induced subgraph; hence, all these cases are contradictory.

- *CASE 2.3:* Both G_1 and G_2 do not contain I_3.

  If G_1 or G_2 contains C_5 then G contains $I_2 + C_5$ as an induced subgraph. Therefore both G_1 and G_2 do not contain C_5. If both G_1 and G_2 contain P_4 then G contains $P_4 + P_4$ as an induced subgraph, a contradiction. Without loss of generality, suppose that G_2 does not contain P_4. Since C_5 and P_4 are self-complementary graphs, note that $\overline{G_2}$ does not contain C_3, C_5, and P_4; hence, $\overline{G_2}$ is a triangle-free connected cograph, as in Case 2.2. From this fact, we derive exactly the same contradictions as in that case.

CASE 3: G and $\overline{G}$ are connected. By Theorem 2, the analysis of this case is as follows.

If G is trivial or $G = P_5$ or $G = \overline{P_5}$ or $G = C_5$ then G admits a cct, which is a contradiction.

If G is a pseudo-split graph whose vertex set has a partition $(\mathcal{S}, \mathcal{K}, \mathcal{R})$, where $\mathcal{R}$ is the head and induces an extended P_4-laden graph, then, by minimality of G, $G[\mathcal{R}]$ admits a cct, namely C_1. In this case, G also admits a cct, namely $\mathcal{K} \cup C_1$, which is again a contradiction.

If G is a quasi-spider whose head $\mathcal{R}$ induces an extended P_4-laden graph, we analyze three subcases:

- *CASE 3.1:* G is a quasi-spider with a q-vertex $s \in \mathcal{S}$ replaced by either an I_2 or a K_2 with vertices u, v. In this case, the subset $(\mathcal{S} - \{s\}) \cup \{u, v\} \cup \mathcal{K}$ induces a subgraph with cct $\mathcal{K}$. Notice that, by minimality, $G[\mathcal{R}]$ admits a cct, say C_1. Consequently, G admits a cct $\mathcal{K} \cup C_1$, since $(\mathcal{S} - \{s\}) \cup \{u, v\} \cup (\mathcal{R} - C_1)$ induces a forest. Contradiction.

- *CASE 3.2:* G is a quasi-spider with a q-vertex $k \in \mathcal{K}$ replaced by a K_2 with vertices u, v. In this case, the subset $\mathcal{S} \cup (\mathcal{K} - \{k\} \cup \{u, v\}$ induces a subgraph with cct $(\mathcal{K} - \{k\}) \cup \{u, v\}$. By minimality, $G[\mathcal{R}]$ admits a cct C_1. Consequently, G admits cct $(\mathcal{K} - \{k\}) \cup \{u, v\} \cup \mathcal{C}_1$, which is a contradiction.

- *CASE 3.3:* G is a quasi-spider with a q-vertex $k \in \mathcal{K}$ replaced by an I_2 with vertices u, v. We analyze two additional possibilities:

  - $G[\mathcal{R}] = (C, I)$ is a split graph, where I is an independent set and C is a clique. Then, G admits a cct $(\mathcal{K} - \{u\}) \cup C$, because $\mathcal{S} \cup I \cup \{u\}$ induces a forest. This is a contradiction.
  - $G[\mathcal{R}]$ is not a split graph, i.e., contains $2K_2$, C_4, or C_5 as an induced subgraph. Then G contains one of the following graphs as an induced subgraph: $I_2 + 2K_2$, $I_2 + I_2 + I_2$, or $I_2 + C_5$. Contradiction. This concludes the proof of the theorem. $\square$

4 Recognizing extended P_4-laden graphs with cct in linear time

The proof of the previous theorem directly gives a linear-time recognition algorithm for extended P_4-laden graphs with cct (Algorithm 1, described below).

The linear-time complexity is justified by the following facts:

(i) the primeval tree can be computed in linear time [10];

(ii) all the **if**-conditions in the algorithm can be checked in linear time.

In order to prove (ii), it suffices to show that the **if**-conditions based on recursive calls (lines 7, 13, 38, and 42) can be checked in linear time. (The remaining if-conditions are all easily checkable in linear time.) We use an inductive argument. First, it is clear that for a trivial graph one can check in linear time whether it admits a cct. Now we consider how recursive calls are made as Algorithm 1 traverses the primeval tree of the input graph G. In line 7 of the algorithm, a recursive call for each subtree associated with child subgraph G_i of G is made. In line 13, a recursive call is made only for the subtree associated with child subgraph G_1. Finally, in lines 38 and 42, a recursive call is made only for the subtree associated with child subgraph $G[\mathcal{R}]$. In any case, the recursive calls are made on subtrees corresponding to vertex-disjoint subgraphs of G. Assuming by induction that for graphs H with fewer than n vertices (where $n = |V(G)|$) one can check whether H admits a cct in time linear in the size of H, we conclude that all the **if**-conditions in lines 7, 13, 38, and 42 can be checked in time linear in the size of G.

5 Conclusions

In this work, we use the structure of the extended P_4-laden graphs and its primeval tree decomposition to provide a characterization by forbidden subgraphs of extended P_4-laden graphs which admit clique cycle transversal, or, alternatively, which can be partitioned into a complete graph and a forest (acyclic graph). As a consequence, we develop a linear time recognition algorithm for this class of graphs.

Algorithm 1: Checking a cct in an extended P_4-laden graph G

Input: Primeval tree T_G of G, where $G_1, G_2, \ldots, G_k$ are the children of root node G
Output: YES and a cct of G, or NO and a forbidden induced subgraph

1 **if** *G is disconnected* **then**
2 **if** *G_i is acyclic for all $i = 1, \ldots, k$* **then** output YES and cct $\emptyset$;
3 **else if** *there exist $G_i, G_j (i \neq j)$ containing cycles B_i, B_j, respectively* **then**
4 output NO and forbidden subgraph $B_i \cup B_j$
5 **else if** *there is exactly one G_i containing a cycle* **then**
7 **if** *G_i recursively admits a cct C_i* **then** output YES and cct C_i;
8 **else** output NO and forbidden subgraph H_i of G_i
9 **else if** *$\overline{G}$ is disconnected* **then**
10 **if** *G_i is trivial for all $i = 1, \ldots, k$* **then** output YES and cct $V(G)$;
11 **else if** *there is exactly one nontrivial G_i, say G_1* **then**
13 **if** *G_1 recursively admits a cct C_1* **then** output YES and cct $C_1 \cup V(G_2) \cup \cdots \cup V(G_k)$;
14 **else** output NO and forbidden subgraph H_1 of G_1
15 **else if** *there are nontrivial G_i, G_j, G_k (for distinct i, j, k)* **then** output NO and forbidden subgraph $I_2 + I_2 + I_2$
16 **else if** *there are exactly two nontrivial subgraphs G_i, G_j (say G_1, G_2)* **then**
17 **if** *both G_1 and G_2 contain I_3* **then** output NO and forbidden subgraph $I_3 + I_3$;
18 **else if** *exactly one of the subgraphs G_1, G_2 contains I_3, say G_1* **then**
19 **if** *G_2 contains C_5* **then** output NO and forbidden subgraph $I_2 + C_5$;
20 **else if** *G_2 contains P_4* **then** output NO and forbidden subgraph $I_3 + P_4$;
21 **else**
23 Let C, C' be cliques s.t. $V(G_2) = C \cup C'$ with $|C| \geq |C'|$;
24 **if** $|C'| \geq 2$ **then** output NO and forbidden subgraph $I_2 + 2K_2$;
25 **else if** *G_1 is split with $V(G) = S_1 \cup C_1$* **then** output YES and cct $C \cup C_1$;
27 **else** output NO and forbidden subgraph $I_2 + H$, where $H \in \{2K_2, C_4, C_5\}$
28 **else if** *both G_1 and G_2 do not contain I_3* **then**
29 **if** *G_1 or G_2 contains C_5* **then** output NO and forbidden subgraph $I_2 + C_5$;
30 **else if** *G_1 and G_2 contain P_4* **then** output NO and forbidden subgraph $P_4 + P_4$;
31 **else**
32 suppose that G_2 does not contain P_4;
33 perform the same code in lines 23–27
34 **else if** *G and $\overline{G}$ are connected* **then**
35 **if** *G is trivial or $G = P_5$ or $G = \overline{P_5}$ or $G = C_5$* **then** output YES and a corresponding cct: $\emptyset$, $\emptyset$, C_3, or K_2;
36 **else if** *G is a pseudo-split graph with $V(G) = (\mathcal{S}, \mathcal{K}, \mathcal{R})$* **then**
38 **if** *$G[\mathcal{R}]$ recursively admits a cct C_1* **then** output YES and cct $\mathcal{K} \cup C_1$;
39 **else** output NO and forbidden subgraph H of $G[\mathcal{R}]$
40 **else if** *G is a quasi-spider with a q-vertex $s \in \mathcal{S}$ replaced by I_2,* **or** *a q-vertex $s \in \mathcal{S}$ replaced by K_2,* **or** *a q-vertex $k \in \mathcal{K}$ replaced by a K_2 with vertices u, v* **then**
42 **if** *$G[\mathcal{R}]$ recursively admits a cct C_1* **then**
43 output YES and the corresponding cct: $\mathcal{K} \cup C_1$, $\mathcal{K} \cup C_1$, or $(\mathcal{K} - \{k\}) \cup \{u, v\} \cup C_1$
44 **else** output NO and corresponding forbidden subgraph H of $G[\mathcal{R}]$
45 **else if** *G is a quasi-spider with a q-vertex $k \in \mathcal{K}$ replaced by a I_2 with vertices u, v* **then**
46 **if** *$G[\mathcal{R}]$ is split with $G[\mathcal{R}] = (I, C)$* **then** output YES and cct $(\mathcal{K} - \{u\}) \cup C$;
47 **else** output NO and forbidden subgraph $I_2 + H$, where $H \in \{2K_2, C_4, C_5\}$

References

[1] Bransdstädt, A., Brito, S. P., Klein, S., Nogueira, L. T., and Protti, F. Cycle transversals in perfect graphs and cographs. *Theoretical Computer Science* 469 (2013) 15-23. See also Corrigendum, vol. 487 (2013) 103-105.

[2] Bransdstädt, A., Esposito, S., Nogueira, L. T., and Protti, F. Clique cycle transversals in distance hereditary graphs. *VII Latin-American Algorithms,Graphs, and Optimization Symposium*, Playa del Carmen, Mexico, april 2013. *Electronic Notes in Discrete Mathematics*, to appear.

[3] Feder, T., Hell, P., Klein, S., and Motwani, R. Complexity of graph partition problems. *31st Annual ACM Symposium on Theory of Computing*, 464–472. Plenum Press, New York, 1999.

[4] Festa, P. and Pardalos, P. M., Resende, M. G. C. Feedback Set Problems. *Encyclopedia of Optimization* (2009), 1005–1016.

[5] Foldes, S., and Hammer, P. L. Split graphs. *Proceedings of the Eighth Southeastern Conference on Combinatorics, Graph Theory and Computing* (Louisiana State Univ., Baton Rouge, La., 1977), pp. 311–315. *Congressus Numerantium*, No. XIX, Utilitas Math., Winnipeg, Man., 1977.

[6] Garey, M. R., and Johnson, D. S. Computers and Intractability–A Guide to the Theory of NP-completeness, *Freeman, San Francisco*, 1979.

[7] Giakoumakis, V. P_4-laden graphs: a new class of brittle graphs. *Information Processing Letters* 60 (1996) 29–36.

[8] Groshaus, M., Hell, P., Klein, S., Nogueira, L. T., and Protti, F. Cycle Transversals in Bounded Degree Graphs. *Discrete Mathematics and Theoretical Computer Science* 13:1 (2011) 45–66.

[9] Karp, R. M. Reducibility among combinatorial problems, *Complexity of Computer Computations*, R.E. MILLER, J.W. THATCHER (eds.), Plenum Press (1972) 85-103

[10] Jamison, B., and Olariu, S. p-components and the homogeneous decomposition of graphs *SIAM Journal Discrete Mathematics* 8 (1995) 448–463.

[11] West, D. B. Introduction to Graph Theory. Prentice-Hall, New Jersey, 1996.

[12] Yannakakis, M. Node- and edge-deletion NP-complete problems. *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing – STOC'78*, pp. 253–264, 1978, ACM Press.