Faster Algorithms for Graph Monopolarity

Geevarghese Philip[1] [0000−0003−0717−7303] and Shrinidhi Teganahally Sridhara[2] [0000−0002−0288−8598]

[1] Chennai Mathematical Institute, India and UMI ReLaX
gphilip@cmi.ac.in
https://www.cmi.ac.in/~gphilip
[2] Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, [F-33400 Talence], France shrinidhi.teganahally-sridhara@u-bordeaux.fr

Abstract. A graph $G = (V, E)$ is *monopolar* if its vertex set admits a partition $V = (C \uplus I)$ where $G[C]$ is a *cluster graph* and I is an *independent set* in G; this is a *monopolar partition* of G. The Monopolar Recognition problem—deciding whether an input graph is monopolar—is known to be NP-hard even in very restricted graph classes.

We develop fast exact exponential-time and parameterized algorithms for Monopolar Recognition. Our exact algorithm solves Monopolar Recognition in $\mathcal{O}^\star(1.3734^n)$ time on input graphs with n vertices, where the $\mathcal{O}^\star()$ notation hides polynomial factors. In fact, we develop algorithms that solve the more general problems Monopolar Extension and List Monopolar Partition in the same running time. These are the first improvements over the trivial $\mathcal{O}^\star(2^n)$-time algorithms for all these problems. These problems cannot be solved in $\mathcal{O}^\star(2^{o(n)})$ time, under ETH.

Our FPT algorithms solve Monopolar Recognition in $\mathcal{O}^\star(3.076^{k_v})$ and $\mathcal{O}^\star(2.253^{k_e})$ time where k_v and k_e are, respectively, the sizes of the smallest vertex and edge modulators of the input graph to claw-free graphs. These results are a significant addition to the small number of FPT algorithms currently known for Monopolar Recognition.

Keywords: Graph Monopolarity · Fixed-parameter tractability · Exponential-time algorithms

1 Introduction

In this work we derive fast exponential-time and fixed-parameter tractable (FPT) algorithms for recognizing monopolar graphs. All our graphs are finite, undirected, and simple. Graph H is a *cluster graph* if each connected component of H is a complete graph. A partition of the vertex set of a graph $G = (V, E)$ into two sets C and I is a *monopolar partition* of G if the subgraph $G[C]$ induced by set C is a cluster graph and the set I is an independent set in G. A graph G is *monopolar* if it has a monopolar partition. The primary focus of this work is on the algorithmic problem of recognizing monopolar graphs:

MONOPOLAR RECOGNITION
Input: An undirected graph $G = (V, E)$ on n vertices.
Question: If G is monopolar, then output Yes. Otherwise, output No.

Monopolar graphs generalize both bipartite graphs (C is an independent set) and split graphs (C is a single clique). While both split graphs and bipartite graphs can be recognized in polynomial time, MONOPOLAR RECOGNITION is NP-hard even in very restricted graph classes such as triangle-free planar graphs of maximum degree 3 [12]. We propose fast exponential-time and FPT algorithms that solve MONOPOLAR RECOGNITION in general graphs. Each of our algorithms also outputs a monopolar partition, if the input graph is monopolar. Our first result is a fast exact exponential-time algorithm for MONOPOLAR RECOGNITION:

Theorem 1. MONOPOLAR RECOGNITION *can be solved in* $\mathcal{O}^\star(1.3734^n)$ *time: There is an algorithm which takes a graph* G *on* n *vertices as input, runs in* $\mathcal{O}^\star(1.3734^n)$ *time, and correctly decides if* G *is monopolar. If* G *is monopolar, then this algorithm also outputs one monopolar partition of the vertex set of* G.

Here the $\mathcal{O}^\star()$ notation hides polynomial factors. As far as we know, this is the first improvement over the trivial exact algorithm for this problem. We in fact obtain a similarly fast algorithm for the more general MONOPOLAR EXTENSION problem introduced by Le and Nevries [13]. Let C', I' be two disjoint subsets of the vertex set of a graph $G = (V, E)$. A monopolar partition $V = C \uplus I$ of G with $C' \subseteq C$ and $I' \subseteq I$ is called a *monopolar extension* of the pair (C', I'). Graph G is (C', I')*-monopolar extendable* if it admits a monopolar partition extending (C', I').

MONOPOLAR EXTENSION
Input: An undirected graph $G = (V, E)$ on n vertices, and subsets $C' \subseteq V$ and $I' \subseteq V$.
Question: If G has a monopolar partition which extends (C', I'), then output YES. Otherwise, output NO.

Churchley and Huang [2] defined LIST MONOPOLAR PARTITION as another generalization of MONOPOLAR RECOGNITION. Given a graph $G = (V, E)$ and lists $\{\emptyset \neq L(v) \subseteq \{\hat{C}, \hat{I}\}\ ;\ v \in V\}$, a *monopolar partition of* G *that respects the list function* L is a mapping $f : V \to \{\hat{C}, \hat{I}\}$ such that (i) $f(v) \in L(v)$ holds for all $v \in V$, (ii) $f^{-1}(\hat{I})$ induces an independent set in G, and (iii) $f^{-1}(\hat{C})$ induces a cluster graph in G. The LIST MONOPOLAR PARTITION problem asks if there is a monopolar partition of a graph that respects a given list function:

LIST MONOPOLAR PARTITION
Input: An undirected graph $G = (V, E)$ on n vertices and a list function L defined on the vertices of G such that $\emptyset \neq L(v) \subseteq \{\hat{C}, \hat{I}\}$ for all $v \in V$.
Question: If there exists a monopolar partition of G that respects the list function L, then output YES. Otherwise, output NO.

The problems MONOPOLAR EXTENSION and LIST MONOPOLAR PARTITION are computationally equivalent modulo polynomial time (see Lemma 4), and our exponential speed-up in solving Monopolar Extension directly transfers to such a speed-up for List Monopolar Partition as well:

Theorem 2. *Let G be a graph on n vertices.*

1. *There is an algorithm that takes an instance (G, C', I') of* Monopolar Extension *as input, runs in $\mathcal{O}^\star(1.3734^n)$ time, and correctly decides if G admits a monopolar partition that is an extension of (C', I'). If such a monopolar extension exists then this algorithm also outputs one such monopolar partition of G.*
2. *There is an algorithm that takes an instance (G, L) of* List Monopolar Partition *as input, runs in $\mathcal{O}^\star(1.3734^n)$ time, and correctly decides if G admits a monopolar partition that respects the list function L. If G does admit such a monopolar partition, then this algorithm also outputs one such monopolar partition of G.*

Assuming the Exponential Time Hypothesis, there is no algorithm that can solve Monopolar Recognition—and hence, Monopolar Extension or List Monopolar Partition—in $2^{o(n)}$ time [10, Prop. 9.1]. Any improvement over the running times of Theorem 1 or Theorem 2 can therefore only be in the form of a smaller constant base for the exponential term.

We present two FPT algorithms for Monopolar Recognition parameterized by measures of *distance from triviality.* Churchley and Huang showed that Monopolar Recognition can be solved in polynomial time on claw-free graphs [3]. We show that Monopolar Recognition can be solved in FPT time for the parameter being two natural *deletion distances* to claw-free graphs.

Theorem 3. [$\star$] *There is an algorithm that takes a graph G on n vertices as input, runs in $\mathcal{O}^\star(3.076^{k_v})$ time, and correctly decides if G is monopolar. If G is monopolar, then this algorithm also outputs one monopolar partition of the vertex set of G. Here k_v is the smallest number of vertices that need to be deleted from G to obtain a claw-free graph.*

Theorem 4. [$\star$] *There is an algorithm that takes a graph G on n vertices as input, runs in $\mathcal{O}^\star(2.253^{k_e})$ time, and correctly decides if G is monopolar. If G is monopolar, then this algorithm also outputs one monopolar partition of the vertex set of G. Here k_e is the smallest number of edges that need to be deleted from G to obtain a claw-free graph.*

As far as we know, there are only two previously known results on the parameterized complexity of Monopolar Recognition (summarized below). Theorem 3 and Theorem 4 are thus a significant addition to the set of known FPT results for the problem.

An $\mathcal{O}^\star(2^n)$-time algorithm for Monopolar Recognition follows more or less directly from the definition of the problem. This can easily be improved to $\mathcal{O}^\star(3^{\frac{n}{3}}) \approx \mathcal{O}^\star(1.4423^n)$ time by observing that we need only consider inclusion-maximal independent sets as candidates for the part I in a monopolar partition.

To reduce this further to the $\mathcal{O}^\star(1.3734^n)$ bound of Theorem 1 we exploit structural properties of monopolar graphs, building in particular upon the work of Le and Nevries who showed that MONOPOLAR EXTENSION can be solved in polynomial time on *chair-free* graphs[3] [13, Corollary 3]. We significantly strengthen this result of Le and Nevries:

Theorem 5. [$\star$] *There is a polynomial-time algorithm that solves* MONOPOLAR EXTENSION *for instances $(G, (C', I'))$ where C' is a vertex modulator of G to chair-free graphs. If graph G has a monopolar partition that is an extension of (C', I'), then this algorithm also outputs one such partition.*

We may, in fact, assume C' to be an *inclusion-minimal* vertex modulator:

Lemma 1. [$\star$] *Let $G = (V, E)$ be a monopolar graph. There exists a subset $C' \subseteq V$ of its vertices such that (i) C' is a vertex modulator of G to chair-free graphs; (ii) no proper subset of C' is a vertex modulator of G to chair-free graphs, and (iii) graph G has a monopolar partition $V = (C \uplus I)$ where $C' \subseteq C$.*

It is *not* true that *every* vertex modulator to chair-free graphs can be made to "live inside" the cluster part of some monopolar partition of G. Indeed, consider a *chair-free* monopolar graph $\hat{G}$. It is not the case that arbitrary subsets of its vertices—all of which are, trivially, modulators to chair-free graphs—belong to the cluster part in some monopolar partition of $\hat{G}$. What Lemma 1 says is that there is *at least one such*—inclusion-minimal—modulator in *every* monopolar graph.

MONOPOLAR RECOGNITION thus reduces to the problem of finding a minimal vertex modulator to chair-free graphs that belongs to the "cluster part" of *some* (unknown) monopolar partition of the input graph, or ruling out that such a modulator exists. Guessing the vertices of each induced chair[4] that belongs to this unknown modulator results in a 31-way branching algorithm where the number of undecided vertices drops by 5 in each branch, yielding an $\mathcal{O}^\star(31^{\frac{n}{5}}) \approx \mathcal{O}^\star(1.987^n)$-time algorithm. Instead of looking for a "good" minimal modulator to chair-free graphs, we repeatedly find a "fresh" chair, with all undecided vertices. We then carefully branch on its vertices, assigning each vertex either to the "independent set part" or the "cluster part" of a putative monopolar partition. We stop the branching when *no* induced chair has *all* its vertices undecided; we then apply Theorem 5 to solve the remaining instance. The careful branching and early stopping lead to the considerable speed-up from the trivial $\mathcal{O}^\star(2^n)$ to the $\mathcal{O}^\star(1.3734^n)$ of Theorem 1 and Theorem 2. We get our fast FPT algorithms by "branching towards" Theorem 5, as well.

MONOPOLAR RECOGNITION has received a considerable amount of attention starting around the year 2008. Linear-time algorithms for MONOPOLAR RECOGNITION are known for *cographs* [8], *chordal graphs* [7], and *line graphs* [1]. The problem can be solved in polynomial time on *P_5-free graphs*, *chair-free graphs*, *hole-free graphs*, *maximal planar graphs*, *claw-free graphs*, *permutation graphs*,

[3] See section 2 for a definition.
[4] See Figure 1(f).

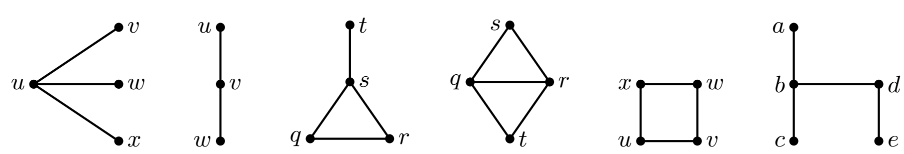
and *co-comparability graphs* [12,13,2,3,4]. In contrast, the problem is known to be NP-hard on *triangle-free planar graphs of maximum degree* 3 [12].

We are aware of only two papers that take up the parameterized complexity of Monopolar Recognition. These are both by the same set of authors, namely, Kanj, Komusiewicz, Sorge and van Leeuwen [10,11]. In the first paper [10], they show—*inter alia*—that Monopolar Recognition is FPT parameterized by the *number of cliques* on the "cluster side" $G[C]$ of a(n unknown) monopolar partition $V(G) = C \uplus I$ of the input graph G. They derive an algorithm that, given a graph G and a positive integer k, decides in $\mathcal{O}^\star(2^k)$ time whether G has a monopolar partition $V(G) = C \uplus I$ where the "cluster side" $G[C]$ is a disjoint union of at most k cliques. They show also that under ETH, any algorithm that solves Monopolar Recognition must take $2^{\Omega(n+m)}$ time on graphs with n vertices and m edges. In the second work, [11], they show that Monopolar Recognition has a kernel with $\mathcal{O}(k^4)$ vertices, for the same parameter k. Note that this parameter k is not comparable with either of our parameters k_v, k_e; it is easy to see that there are graphs in which each of k_v or k_e is a small constant while k is unbounded, and *vice versa*.

Organization of the rest of the paper. In the next section we list our notation and terminology, and prove some preliminary results. We describe our exact exponential algorithms for monopolarity and prove Lemma 1, Theorem 1, Theorem 2 and Theorem 5 in section 3. We derive our FPT algorithms—and prove Theorem 3 and Theorem 4—in section 4. We summarize our results and list some open problems in section 5. *Proofs of results marked with a* [$\star$] *had to be excluded from this Extended Abstract due to page constraints.*

2 Preliminaries

All our graphs are finite, simple, and undirected. We follow the graph notation and terminology of Diestel [5]. Graph G is *H-free* for a graph H if H does not appear as an *induced subgraph* of G. A graph G is a *cluster graph* if each connected component of G is a complete graph; equivalently, if and only if G is P_3-free (see Figure 1(b)).



(a) Claw (b) P_3 (c) Paw $P(s,t)$ (d) Diamond $D(s,t)$ (e) C_4 (f) Chair $Ch(a,b,c,d,e)$

Fig. 1: Some small named graphs.

Graph G is *claw-free* if it does not have a *claw*—Figure 1(a)—as an induced subgraph. A subset X of the vertex set of graph G is said to be a *vertex modulator of* G *to claw-free graphs* if the graph $G - X$ is claw-free. A subset Y of the edge set of graph G is said to be an *edge modulator of* G *to claw-free graphs* if the graph $G - Y$ is claw-free. Graph G is *chair-free* if it does not have a *chair*—Figure 1(f)—as an induced subgraph. A subset X of the vertex set of graph G is said to be a *vertex modulator of* G *to chair-free graphs* if the graph $G - X$ is chair-free. *We use the short forms "vertex modulator" and "edge modulator", when the fuller forms are clear from the context.*

We use P_k to denote a path with k vertices, and C_k for a cycle with k vertices. Some other small graphs that we need to refer by name are listed in Figure 1. We use $Ch(a, b, c, d, e)$ to denote a *chair* with $\deg(b) = 3, \deg(d) = 2, \deg(a) = \deg(c) = \deg(e) = 1$ as in Figure 1(f) (a chair graph is also defined as $S_{2,1,1}$, which is a claw with edges subdivided once, zero, and zero), $P(s, t)$ to denote a *paw* with $\deg(s) = 3, \deg(t) = 1$, and $D(s, t)$ for a *diamond* with $\deg(s) = \deg(t) = 2$.

We need the following simple properties of monopolar extensions:

Lemma 2. [$\star$] *Let C', I' be two (disjoint) subsets of the vertex set V of graph G, and let $x \notin (C' \cup I')$ be a vertex of degree 1 in G. Then G is (C', I')-monopolar extendable if and only if $G - \{x\}$ is (C', I')-monopolar extendable.*

Lemma 3. [$\star$] *A graph G is (C', I')-monopolar extendable if and only if I' is an independent set in G and the induced subgraph $G - I'$ is $((C' \cup N(I')), \emptyset)$-monopolar extendable.*

Lemma 4. [$\star$] Monopolar Extension *and* List Monopolar Partition *are computationally equivalent modulo polynomial time.*

Le and Nevries [13] prove the following result for claw-free graphs:

Theorem 6. Monopolar Extension *can be solved in $O(n^4)$ time on n-vertex claw-free graphs.*

Our algorithms that prove Theorem 3 and Theorem 7 employ algorithms to solve the following parameterized deletion problems, as subroutines:

Claw-Free Vertex Deletion	**Parameter:** k
Input: An undirected graph $G = (V, E)$, and an integer k.	
Question: If there is a set $X \subseteq V$; $\lvert X\rvert \leq k$ such that deleting $G - X$ is a claw-free graph, then output one such set X. Otherwise, output No.	

Claw-Free Edge Deletion	**Parameter:** k
Input: An undirected graph $G = (V, E)$, and an integer k.	
Question: If there is a set $X \subseteq E$; $\lvert X\rvert \leq k$ such that deleting X from G results in a claw-free graph, then output one such set X. Otherwise, output No.	

To the best of our knowledge, the fastest way to solve Claw-Free Vertex Deletion is by phrasing it as an instance of a 4-Hitting Set problem and then applying the best known FPT algorithm for 4-HITTING SET due to Dom et al. [6, Theorem 3.1]. This yields an algorithm that solves CLAW-FREE VERTEX DELETION in $\mathcal{O}^\star(3.076^k)$ time. Similarly, the fastest algorithm for CLAW-FREE EDGE DELETION involves a reduction to 3-HITTING SET and then applying the argument underlying Dom et al.'s algorithm for this problem [6, Proposition 3.1] which uses an algorithm for VERTEX COVER as black box. Using the current best running time of $\mathcal{O}^\star(1.253^k)$ for VERTEX COVER [9] here, we get an algorithm that solves CLAW-FREE EDGE DELETION in $\mathcal{O}^\star(2.253^k)$ time.

These two fast algorithms for modulators to claw-free graphs are thus direct applications of existing results to these problems. We capture these as a theorem for later reference:

Theorem 7. CLAW-FREE VERTEX DELETION can be solved in $\mathcal{O}^\star(3.076^k)$ time, and CLAW-FREE EDGE DELETION can be solved in $\mathcal{O}^\star(2.253^k)$ time.

3 Exact Algorithms for Monopolarity

In this section we outline the proofs of Theorem 1 and Theorem 2. We get these faster running times by exploiting various structural properties of monopolar graphs to lift a theorem of Le and Nevries [13] for chair-free graphs, so that it works for all graphs; see Theorem 5. Le and Nevries [13] showed that MONOPOLAR EXTENSION, and consequently MONOPOLAR RECOGNITION, can be solved in polynomial time on chair-free graphs. They achieved this result by reducing MONOPOLAR EXTENSION on (a superclass of) chair-free graphs to *2*-SAT; our algorithms make extensive use of this reduction.

If the given graph $G = (V, E)$ does not have an induced P_3 then G is a cluster graph. In this case, for any vertex $v \in V$, the partition $V = (V \setminus \{v\}) \uplus \{v\}$ is a monopolar partition of G, and there is nothing more to do. If the graph *does* contain at least one induced P_3, then we make use of the concept of a *C'-good induced* P_3, a notion introduced by Le and Nevries [13].

Definition 1. [13] *Let G be a graph, and let C' be a subset of the vertex set of G. We say that an induced P_3 in G of the form uvw is C'*-good*, if it satisfies* at least one *of the following conditions:*

1. *At least one of the vertices in the set $(N[v] \cup N[u] \cup N[w])$ is in the set C'.*
2. *At least one of the vertices u, v, w is a part of a triangle in G.*
3. *At least one of the edges of the P_3 uvw is part of an induced C_4 in G.*

Otherwise, we say that the P_3 uvw is C'-bad. *Finally, we say that graph G is* C'-good *if every induced P_3 in G is* C'-good.

Le and Nevries [13] show how, given an instance $(G, (C', \emptyset))$ of MONOPOLAR EXTENSION *where G is C'-good*, we can construct in polynomial time a *2*-SAT formula $F(G, C')$ where (i) the variables represent vertices of G, (ii) the clauses are constructed based on the properties of small induced graphs from Figure 1, and (iii) the formula $F(G, C')$ is satisfiable if and only if $(G, (C', \emptyset))$ is a Yes-instance of Monopolar Extension.

Given an instance $(G, (\tilde{C}, \tilde{I}))$ of Monopolar Extension, first, we use Lemma 3 to transform it into an equivalent instance $(G', (C', \emptyset))$ where $C' = \tilde{C} \cup N(\tilde{I})$ and $G' = G - \tilde{I}$. We then construct the corresponding *2*-SAT formula $F(G', C')$ as described below, following Le and Nevries [13]. Intuitively, the formula is set up in such a way that the variable corresponding to a vertex v gets the value 1 in a satisfying assignment of the formula if and only if v belongs to the *independent set* part in some valid monopolar extension.

1. There is a Boolean variable v for each vertex v of G.
2. For each vertex $u \in C'$, add $(\neg u)$ to the formula (a *forced-cluster-clause*).
3. For each edge uv in G, add $(\neg u \vee \neg v)$ to the formula (an *edge-clause*).
4. For each induced paw $P(s, t)$ and each induced diamond $D(s, t)$ in G, add $(s \vee t)$ to the formula (a *paw-diamond-clause*).
5. For each induced 4-cycle $C_4 : uvwx$ in G, add the clauses $(u \vee v)$, $(v \vee w)$, $(w \vee x)$, and $(x \vee u)$ to the formula (C_4*-clauses*).
6. For each vertex $x \in C'$ and each induced P_3 of the form xvw or vxw, add the clause $(v \vee w)$ to the formula (a P_3*-clause*).

The formula $F(G, C')$ is the conjunction of all forced-cluster-clauses, edge-clauses, paw-diamond-clauses, C_4-clauses, and P_3-clauses. Le and Nevries show that *for a C'-good graph G*, the graph G is monopolar extendable with respect to $(C', \emptyset)$ if and only if the formula $F(G, C')$ is satisfiable:

Lemma 5. [13, Lemma 1 and its proof] *Let $G = (V, E)$ be a graph and C' be a subset of vertices such that G is C'-good. Then in polynomial time we can construct an instance $F(G, C')$ of 2*-SAT *with the following properties. First, G is $(C', \emptyset)$-monopolar extendable if and only if $F(G, C')$ is satisfiable. Second, if $F(G, C')$ is satisfiable then, the set I of vertices corresponding to variables set to* True *and the set C of vertices corresponding to variables set to* False *in a satisfying assignment to $F(G, C')$ together form a monopolar partition of G i.e., $V = (C \uplus I)$ is a monopolar partition that extends $(C', \emptyset)$.*

To solve Monopolar Recognition on an input graph G using Lemma 5 we need to find a vertex subset C' of G such that graph G is C'-good. But why should such a vertex set C' exist for *every* monopolar graph G? And even if it did, how could we find one such set? Le and Nevries [13, Corollary 3] show that *if the input graph G is chair-free* then they can use Lemma 5 as a tool to solve an instance $(G, (C', \emptyset))$ of Monopolar Extension in polynomial time *even if* graph G is *not* C'-good. We generalize this further; we show that in fact we can solve an instance $(G, (C', \emptyset))$ of Monopolar Extension in an *arbitrary graph G* in polynomial time if the set C' is a *vertex modulator of G to chair-free graphs*, even if graph G is *not* C'-good. We thus reduce Monopolar Recognition to the problem of finding such a vertex modulator C' of G, and then solving Monopolar Extension on the instance $(G, (C', \emptyset))$.

Lemma 6. [$\star$] *If a vertex subset C' is a vertex modulator of graph $G = (V, E)$ to chair-free graphs, then the instance $(G, (C', \emptyset))$ of* Monopolar Extension *can be solved in polynomial time. Furthermore if graph G is $(C', \emptyset)$-monopolar extendable, then a monopolar partition of G that extends $(C', \emptyset)$ can be obtained in polynomial time as well.*

As List Monopolar Partition and Monopolar Extension are polynomially equivalent as shown in Lemma 4, we immediately get a similar result for List Monopolar Partition when the set of vertices $\{v : L(v) = \{\hat{C}\}\}$ forms a vertex modulator of the input graph:

Corollary 1. *Let (G, L) be an instance of* List Monopolar Partition *where the set $C' = \{v : L(v) = \{\hat{C}\}\}$ of vertices that are assigned to the cluster side, forms a vertex modulator of graph G to chair-free graphs. Such an instance can be solved in polynomial time.*

We now have all the ingredients for proving our generalization of Le and Nevries' polynomial-time algorithm for Monopolar Extension on chair-free graphs, namely, Theorem 5.

Let $(G = (V, E), C', I')$ be an instance of Monopolar Extension. If either the graph G is chair-free or the set C' is a vertex modulator, then we can solve this instance in polynomial time. Such instances form the base cases of Monopolar Extension that are easy to solve. We start with a simple observation:

Lemma 7. [$\star$] *Let $(G = (V, E), C', I')$ be an instance of* Monopolar Extension*. In polynomial time we can either solve the instance or derive an* equivalent *instance $(\tilde{G}, \tilde{C}, \tilde{I})$ of* Monopolar Extension *with the following property: there exists at least one induced chair—say $Ch(a, b, c, d, e)$—in $\tilde{G}$ such that $\{a, b, c, d, e\} \cap (\tilde{C} \cup \tilde{I}) = \emptyset$.*

Let $Ch(a, b, c, d, e)$ be an induced chair in $G = (V, E)$ such that $\{a, b, c, d, e\} \cap (\tilde{C} \cup \tilde{I}) = \emptyset$ holds. Let $V = (C \uplus I)$ be a(n unknown) valid monopolar partition of G. Our algorithm for Monopolar Extension does an exhaustive branching on the vertices b and e—see Figure 1(f) for the notation—being in C or I:

- **$b \in C, e \in C$:** Since $G[C]$ is a cluster graph and the edge be is not present in graph G, vertex d cannot belong to the set C. So we have $d \in I$. We cannot conclude anything about vertices a and c except that they cannot be together in set C.
- **$b \in I, e \in C$:** Since $G[I]$ is edge-less and the edges ab, bc and bd are present in graph G, vertices a, c and d cannot belong to the set I. So we have $a \in C$, $c \in C$ and $d \in C$.
- **$b \in C, e \in I$:** Vertex d cannot belong to the set I as de is an edge and $e \in I$. Thus $d \in C$. Also, vertex a cannot belong to the set C as the edge ad is not present and vertex c cannot belong to the set C as the edge cd is not present. So we have $d \in C$, $a \in I$ and $c \in I$.

$b \in I, e \in I$**:** Here, a, c and d cannot belong to the set I as ab, bc and bd are edges and $b \in I$ where $G[I]$ is needed to be an independent set. So we have $a \in C$, $c \in C$ and $d \in C$.

Let ME-ALGORITHM be a procedure (pseudocode not included in the Extended Abstract due to page constraints) which uses the above branching strategy to solve MONOPOLAR EXTENSION and returns a monopolar partition if the input is a YES-instance. The correctness of procedure ME-ALGORITHM follows from the fact that it does exhaustive branching on valid partitions, and from Corollary 1, Theorem 6, and Lemma 6.

Lemma 8. *The call* ME-ALGORITHM(G, C', I') *terminates in* $\mathcal{O}^\star(1.3734^n)$ *time where* n *is the number of vertices in graph* G*.*

Proof. Let $\mu = \mu(G, C', I') = |V(G) \setminus (C' \cup I')|$ be a measure function that denotes the number of vertices in graph G which are not yet in the sets C', I'. Note that the condition $\mu \leq n$ holds before the procedure ME-ALGORITHM makes any recursive call. The procedure makes recursive calls only when it finds a chair $\mathcal{C}$, all of whose five vertices contribute to the measure μ. Three of these recursive calls cause μ to drop by 5 each, since all five vertices of $\mathcal{C}$ are moved to $C' \cup I'$ in these calls. The remaining recursive call results in μ dropping by 3, since only three vertices—b, e, d—are moved to $C' \cup I'$ by this call.

The procedure stops recursing only when for each chair in graph G, at least one vertex in the chair is in $C' \cup I'$. If a vertex x in a chair $\mathcal{C}$ is moved to the set I', then the procedure moves every neighbour of x to the set C'. It follows that when the recursion stops, the set C' forms a vertex modulator of graph G. The procedure solves this instance in polynomial time using an implementation of Corollary 1.

Let $T(\mu)$ denote the number of nodes in the recursion tree of the procedure, when invoked with an instance with measure μ. The recurrence for $T(\mu)$ is then $T(\mu) = 3T(\mu - 5) + T(\mu - 3)$. This solves to $T(\mu) \leq (1.3734)^\mu$, and the lemma follows. □

We can convert an instance $G = (V, E)$ of MONOPOLAR RECOGNITION to an instance $(G = (V, E), C', I')$ of MONOPOLAR EXTENSION by setting $C' = \emptyset, I' = \emptyset$. This proves Theorem 1.

Combining Lemma 4 and Lemma 8 we get Theorem 2.

4 FPT Algorithms for Monopolarity

In this section we outline the arguments that prove Theorem 3 and Theorem 4; the proofs could not be included in this Extended Abstract due to page constraints.

We have seen in Theorem 7 that we can find a vertex modulator of graph G to claw-free graphs in time $\mathcal{O}^\star(3.076^{k_v})$ where k_v is the size of a smallest such modulator of G. It also says that we can find an *edge* modulator of G to claw-free graphs in time $\mathcal{O}^\star(2.253^{k_e})$ where k_e is the size of a smallest such modulator. Once we have such sets, we show that we can solve monopolarity in arbitrary graphs by making use of Lemma 6[5]. The idea is to find such sets and then guess the partition the elements of these sets are assigned to in any valid monopolar partition and then try to solve monopolar extension to decide monopolarity. Once fleshed out, these arguments prove Theorem 3 and Theorem 4.

5 Conclusion

We derive fast exponential-time and FPT algorithms for Monopolar Recognition. Our exact algorithm for Monopolar Recognition solves the problem in $\mathcal{O}^\star(1.3734^n)$ time on graphs with n vertices, significantly improving the trivial $\mathcal{O}^\star(2^n)$-time algorithm and the more involved $\mathcal{O}^\star(3^{\frac{n}{3}})$-time algorithm. We also show how to solve the more general problems List Monopolar Partition and Monopolar Extension in $\mathcal{O}^\star(1.3734^n)$ time. These are the fastest known exact algorithms for these three problems. We derive two FPT algorithms for Monopolar Recognition with two notions of distance from triviality as the parameters. We show that Monopolar Recognition can be solved in $\mathcal{O}^\star(3.076^{k_v})$ and $\mathcal{O}^\star(2.253^{k_e})$ time, where k_v and k_e are, respectively, the sizes of the smallest vertex and edge modulators of the input graph to claw-free graphs. These results are a significant addition to the small number of FPT algorithms known for Monopolar Recognition.

Le and Nevries [13] showed that if a graph G is chair-free then the instance $(G, (C', \emptyset))$ of Monopolar Extension can be solved in polynomial time for any vertex subset C' of graph G. We significantly generalize this result: we show that the instance $(G, (C', \emptyset))$ of Monopolar Extension can be solved in polynomial time for *any* graph G if C' is a vertex modulator of G to chair-free graphs. This generalization of Le and Nevries' result forms the basis of our fast algorithms.

Open problems. Le and Nevries [13] show that Monopolar Recognition can be solved in polynomial time for a strict superclass of claw-free graphs, namely the class of $\{F_1, F_2, F_3\}$-free graphs for certain small graphs F_1, F_2, F_3. This class includes chair-free graphs. A natural question to ask is whether we can obtain a faster exact algorithm by proving the equivalent of Lemma 6 for $\{F_1, F_2, F_3\}$-free graphs, and then branching on induced subgraphs F_1, F_2, F_3. Recall that algorithms that solve these problem in time which is sub-exponential in n, are ruled out by ETH. It would therefore be interesting to establish a lower bound for the base of the exponential term, perhaps based on SETH.

We can ask analogous questions in the parameterized setting as well: whether we can use the sizes of vertex or edge modulators to $\{F_1, F_2, F_3\}$-free graphs as the parameters, to get faster FPT algorithms. This latter question is made interesting by the fact that each of the graphs $\{F_1, F_2, F_3\}$ has more vertices and edges than a claw. Using the known d-Hitting Set FPT algorithms as a

[5] Observe that a *claw-free* graph is also a *chair-free* graph.

black box to solve Monopolar Recognition, similar to our algorithms for Theorem 3 and Theorem 4, would therefore incur a significantly higher running time in terms of the respective hitting set sizes. We will need clever ideas to get around this slowdown. Since our FPT algorithms depend on algorithms to find claw-free modulators, an improvement in these algorithms would improve our FPT algorithms as well.

Acknowledgements

We extend our heartfelt gratitude to the anonymous reviewers of an earlier version of this manuscript. Their detailed and insightful feedback has helped us greatly improve the presentation.

References

1. Churchley, R., Huang, J.: Line-polar graphs: characterization and recognition. SIAM Journal on Discrete Mathematics **25**(3), 1269–1284 (2011)
2. Churchley, R., Huang, J.: List monopolar partitions of claw-free graphs. Discrete Mathematics **312**(17), 2545–2549 (2012)
3. Churchley, R., Huang, J.: On the polarity and monopolarity of graphs. Journal of Graph Theory **76**(2), 138–148 (2014)
4. Churchley, R., Huang, J.: Solving partition problems with colour-bipartitions. Graphs and Combinatorics **30**, 353–364 (2014)
5. Diestel, R.: Graph Theory: 5th edition. Springer Graduate Texts in Mathematics, Springer (2017), `https://books.google.co.in/books?id=zIxRDwAAQBAJ`
6. Dom, M., Guo, J., Hüffner, F., Niedermeier, R., Truß, A.: Fixed-parameter tractability results for feedback set problems in tournaments. Journal of Discrete Algorithms **8**(1), 76–86 (2010)
7. Ekim, T., Hell, P., Stacho, J., de Werra, D.: Polarity of chordal graphs. Discrete Applied Mathematics **156**(13), 2469–2479 (2008)
8. Ekim, T., Mahadev, N.V., de Werra, D.: Polar cographs. Discrete Applied Mathematics **156**(10), 1652–1660 (2008)
9. Harris, D.G., Narayanaswamy, N.S.: A Faster Algorithm for Vertex Cover Parameterized by Solution Size. In: Beyersdorff, O., Kanté, M.M., Kupferman, O., Lokshtanov, D. (eds.) 41st International Symposium on Theoretical Aspects of Computer Science (STACS 2024). Leibniz International Proceedings in Informatics (LIPIcs), vol. 289, pp. 40:1–40:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2024). `https://doi.org/10.4230/LIPIcs.STACS.2024.40`, `https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.STACS.2024.40`
10. Kanj, I., Komusiewicz, C., Sorge, M., Van Leeuwen, E.J.: Parameterized algorithms for recognizing monopolar and 2-subcolorable graphs. Journal of Computer and System Sciences **92**, 22–47 (2018)
11. Kanj, I., Komusiewicz, C., Sorge, M., Van Leeuwen, E.J.: Solving partition problems almost always requires pushing many vertices around. SIAM Journal on Discrete Mathematics **34**(1), 640–681 (2020)
12. Le, V.B., Nevries, R.: Recognizing polar planar graphs using new results for monopolarity. In: Algorithms and Computation: 22nd International Symposium, ISAAC 2011, Yokohama, Japan, December 5-8, 2011. Proceedings 22. pp. 120–129. Springer (2011)
13. Le, V.B., Nevries, R.: Complexity and algorithms for recognizing polar and monopolar graphs. Theoretical Computer Science **528**, 1–11 (2014). `https://doi.org/https://doi.org/10.1016/j.tcs.2014.01.032`, `https://www.sciencedirect.com/science/article/pii/S0304397514000814`