Monopolar Graphs: Complexity of Computing Classical Graph Parameters

Michele Barbato*, Dario Bezzi

Università degli Studi di Milano, Dipartimento di Informatica, OptLab, Via Bramante 65, 26013 Crema (CR), Italy

Abstract

A graph $G = (V, E)$ is monopolar if V can be partitioned into a stable set and a set inducing the union of vertex-disjoint cliques. Motivated by an application of the clique partitioning problem on monopolar graphs to the cosmetic manufacturing, we study the complexity of computing classical graph parameters on the class of monopolar graphs. We show that computing the clique partitioning, stability and chromatic numbers of monopolar graphs is **NP**-hard. Conversely, we prove that every monopolar graph has a polynomial number of maximal cliques thus obtaining that a maximum-weight clique can be found in polynomial time on monopolar graphs.

Keywords: Computational complexity, Monopolar graph, Maximum-weight clique, Clique partitioning, Stable set, Graph coloring

1. Introduction

We consider simple undirected graphs whose terminology can be found in [2]. Given a graph $G = (V, E)$, a partition (A, B) of V is *monopolar* if A is a stable set and $G[B]$, the graph induced by B in G, is a *cluster*, that is, the union of vertex-disjoint cliques. The graph G is *monopolar* if its vertex set admits a monopolar partition.

Recently, monopolar graphs have been used to detect core-periphery structure of protein interaction networks [3]. ILP formulations and heuristic methods are given in [3] to extract a monopolar subgraph from a general graph by removing as few edges as possible. Here, the input graph represents a protein interaction network measurement affected by independent stochastic errors and the extracted monopolar subgraph corresponds to the real structure of the observed network.

Our interest in monopolar graphs stems from their relation to another real-world problem, which arises in cosmetic manufacturing and is described at the end of this introduction.

From a theoretical perspective, monopolar graphs have been mainly studied in connection with other graph classes, such as polar graphs first defined in [25] and unipolar graphs treated, *e.g.,* in [7, 10, 24]. All these classes can be concisely described by means of the following definition used in [16]. Given Π_A and Π_B two graph properties, $G = (V, E)$ is a (Π_A, Π_B)-*graph* if V is partitionable into A and B such that $G[A]$ has property Π_A and $G[B]$ has property Π_B. Monopolar graphs are easily seen to be the (K_2-free, P_3-free)-graphs, see *e.g.,* [3]. Similarly, *polar* graphs can be defined as the ($\overline{P_3}$-free, P_3-free)-graphs and the *unipolar* graphs as the ($\overline{K_2}$-free, P_3-free)-graphs. Note that polar graphs generalize both unipolar and monopolar graphs.

Most of works concerned with monopolar graphs are focused on the *monopolarity recognition problem*, consisting in deciding whether a given input graph is monopolar. Monopolar recognition is relevant for solving the analogous problem of recognizing polar graphs. Indeed, for several special classes of input graphs, the monopolarity recognition problem admits polynomial-time algorithms which are also used as subroutines to efficiently recognize polar graphs in those classes, see *e.g.,* [5, 8, 9]. Other efficient

*Corresponding author.
Email addresses: michele.barbato@unimi.it (Michele Barbato), dario.bezzi@unimi.it (Dario Bezzi)

algorithms for monopolarity recognition are given if the number of maximal cliques in the cluster induced by a monopolar partition is treated as a fixed parameter [16], for superclasses of chair-free and hole-free input graphs, and for classes of input graphs with bounded clique- or tree-width, see [18] and the references therein. On the other hand, the results in [11] imply that it is **NP**-complete to recognize (mono)polar graphs in general and the same holds for (mono)polar recognition of K_3-free input graphs [6, 17] and K_3-free planar input graphs of maximum degree three [18].

The **NP**-completeness of recognizing (mono)polar graphs contrasts with the fact that unipolar graphs can be recognized in polynomial time, as shown in [7, 10, 24]. In fact, [10] also shows that unipolar graphs are perfect (see *e.g.,* [15, Sect. 9.2] for the definition of perfect graphs). Hence it is well-known [15, Chapt. 9] that the stability, chromatic, clique and clique partitioning numbers of unipolar graphs can be computed in polynomial time and, to this end, specific combinatorial algorithms exploiting the unipolar structure are provided in [10].

Conversely, little seems to be known about the complexity of determining the same four parameters on monopolar graphs. In particular, a polynomial-time algorithm for the stability number is guaranteed to exist in monopolar $2P_3$-free graphs, see [19], while [22] provides efficient combinatorial algorithms for computing the clique and stability numbers of (mono)polar graphs which are trivially perfect, as defined in [14].

Contribution. We contribute to the investigation on the complexity of computing classical graph parameters on monopolar graphs. We prove that determining the clique partitioning, stability and chromatic numbers on monopolar graphs is **NP**-hard. The **NP**-hardness of the chromatic number computation is derived from the **NP**-completeness of the 3-Colorability problem on monopolar graphs. The **NP**-hardness of computing the clique partitioning number is proven along with the **NP**-hardness of recognizing a positive clique partitioning-stability number gap on monopolar graphs. All these complexity results are obtained by reductions of classical **NP**-complete problems and involve graphs whose vertex set is explicitly partitioned in a monopolar fashion. Hence they hold even if a monopolar partition is known. Clearly, they also extend to the more general class of polar graphs and to the weighted versions of the considered problems. Subsequently, we prove that the Max-Weight Clique problem can be solved in polynomial time on monopolar graphs. We derive this latter result from a more general one, namely, that (K_m-free, P_3-free)-graphs have a polynomial number of maximal cliques for every fixed $m \geq 1$.

A Monopolar Graph Model for Manufacturing. We conclude the introduction by describing the aforementioned real-world problem that can be modelled by means of monopolar graphs. The problem, which we call Partitioning-Covering, is as follows. We are given a set of *ingredients* N, a set of *containers* $\mathcal{C} = \{C_1, \ldots, C_k\}$ with $C_i \subseteq N$ for every $i \in \{1, \ldots, k\}$ and a set of d cosmetic products, each obtained by combining ingredients in the containers. Let $P_j \subseteq N$ be the set of ingredients needed for making product $j \in \{1, \ldots, d\}$ and $\mathcal{P} = \{P_1, \ldots, P_d\}$. The goal is to decide whether there exists a partition of $\mathcal{C}$ into d subsets $S_1, \ldots, S_d$ such that $P_j \subseteq \bigcup_{C \in S_j} C$ for every $j = 1, \ldots, d$. The sets N, $\mathcal{C}$ and $\mathcal{P}$ define a *feasible* instance of the Partitioning-Covering problem whenever such a partition exists.

By definition, a feasible instance of the Partitioning-Covering problem admits an assignment of each container to exactly one product. The covering condition $P_j \subseteq \bigcup_{C \in S_j} C$ for every $j = 1, \ldots, d$ guarantees that every product can be obtained by using the ingredients in its assigned containers. The partitioning condition is imposed because every product requires a series of time-consuming tasks to be performed on its assigned containers. Hence we assume that an ingredient in a container assigned to product $j \in \{1, \ldots, d\}$ cannot be used to make product $\ell \neq j$, even if it is not present in P_j.

Let I be an instance of the Partitioning-Covering problem defined by N, $\mathcal{C}$ and $\mathcal{P}$ as above. We model I as a monopolar graph $\mathcal{G}_I = (\mathcal{V}_I, \mathcal{E}_I)$ as follows. The vertex set $\mathcal{V}_I$ is given by the union of two sets A and B such that A contains a vertex v_C for each element of $C \in \mathcal{C}$ and B contains a vertex $v_{i,P}$ for every pair $\{i, P\}$ with $i \in N$ and $P \in \mathcal{P}$ such that $i \in P$. The edge set $\mathcal{E}_I$ is obtained by linking v_C and $v_{i,P}$ whenever $i \in C \cap P$ for some $C \in \mathcal{C}$ and $P \in \mathcal{P}$ and by linking $v_{i,P}$ and $v_{j,P}$ whenever $i, j \in P$ for some $P \in \mathcal{P}$. Then (A, B) is a monopolar partition of $\mathcal{V}_I$, since A is a stable set and $\mathcal{G}_I[B]$ is a cluster whose maximal cliques are in one-to-one correspondence with the elements of $\mathcal{P}$.

Proposition 1.1 below reveals a relation between the Partitioning-Covering problem and the clique partitioning number of monopolar graphs.

Proposition 1.1. *Let N, $\mathcal{C}$ and $\mathcal{P}$ define an instance I of the* Partitioning-Covering *problem and let $\mathcal{G}_I$ be its corresponding graph. Then I is feasible if and only if the clique partitioning number of $\mathcal{G}_I$ is $|\mathcal{C}|$.*

Proof. Throughout the proof we use the notation adopted in the description of $\mathcal{G}_I = (\mathcal{V}_I, \mathcal{E}_I)$. We observed that $A = \{v_C : C \in \mathcal{C}\}$ is a stable set of $\mathcal{G}_I$ and (A, B) with $B = \mathcal{V}_I \setminus A$ is a monopolar partition. For $j = 1, \dots, d$, let H_j be the maximal clique of $\mathcal{G}_I[B]$ corresponding to $P_j \in \mathcal{P}$.

If I is feasible there exists a partition $S_1, \dots, S_d$ of $\mathcal{C}$ such that $P_j \subseteq \bigcup_{C \in S_j} C$ for $j = 1, \dots, d$. For every $C \in \mathcal{C}$ let $j(C) \in \{1, \dots, d\}$ be the unique index such that $C \in S_{j(C)}$. We define K_C as the subgraph of $\mathcal{G}_I$ induced by v_C and its neighborhood in $H_{j(C)}$. The subgraph K_C is a clique for every $C \in \mathcal{C}$. Let now $j \in \{1, \dots, d\}$ and $i \in P_j$. Then $i \in C^\star$ for some $C^\star \in S_j$. Note that $j(C^\star) = j$ and $v_{i,P_j} \in H_j$. Then $v_{i,P_j} \in K_{C^\star}$. Hence the set $\{K_C : C \in \mathcal{C}\}$ is a clique cover of $\mathcal{G}_I$. It follows that $\mathcal{G}_I$ can be partitioned into at most $|\mathcal{C}|$ cliques. Since $|\mathcal{C}| = |A|$ and A is a stable set, the clique partitioning number of $\mathcal{G}_I$ is exactly $|\mathcal{C}|$.

Let now $\mathcal{K}$ be a clique partition of $\mathcal{G}_I$ consisting of $|\mathcal{C}|$ cliques. For every $j = 1, \dots, d$, let $\mathcal{K}_j \subseteq \mathcal{K}$ be such that every vertex of H_j belongs to a clique of $\mathcal{K}_j$. The maximal cliques of $\mathcal{G}_I[B]$ are vertex-disjoint, so $\mathcal{K}_j \cap \mathcal{K}_\ell = \emptyset$ for all distinct $j, \ell \in \{1, \dots, d\}$. Since every vertex of B belongs to some clique of $\mathcal{K}$, we extend $\mathcal{K}_1, \dots, \mathcal{K}_d$ to a partition of $\mathcal{K}$ by including in $\mathcal{K}_1$ every clique $K \in \mathcal{K}$ with $K \cap B = \emptyset$. From $|\mathcal{C}| = |A|$ and A being stable, every clique of $\mathcal{K}$ contains exactly one vertex of A. It follows that the sets $S_j = \{C \in \mathcal{C} : v_C \in K \text{ for some } K \in \mathcal{K}_j\}$ for $j = 1, \dots, d$ partition $\mathcal{C}$. Let us take $j \in \{1, \dots, d\}$ and $i \in P_j$. Vertex v_{i,P_j} belongs to some $K^\star \in \mathcal{K}_j$. Hence there exists $C^\star \in \mathcal{C}$ such that $v_{C^\star} \in K^\star$ and, as a consequence, $\{v_{C^\star}, v_{i,P_j}\} \in \mathcal{E}_I$. Then $i \in C^\star \cap P_j$. This ensures that $P_j \subseteq \bigcup_{C \in S_j} C$ for $j = 1, \dots, d$. These properties of $S_1, \dots, S_d$ prove that I is feasible. □

2. Complexity Results

Throughout this section, the symbols $\alpha(G)$, $\chi(G)$ and $\omega(G)$ respectively denote the stability, chromatic and clique number of a graph G. The clique partitioning number is indicated by $\overline{\chi}(G)$ to emphasize that it equals the chromatic number of the complement $\overline{G}$, see *e.g.,* [15, Sect. 9.4].

2.1. NP-Hardness Results

Clique Partitioning Monopolar Graphs and Related Problems. The Partitioning-Covering problem of the introduction is easily seen to be in **NP**. We now prove that it is **NP**-complete. This, together with Proposition 1.1 and the monopolarity of $\mathcal{G}_I$ for every Partitioning-Covering instance I, implies that it is **NP**-hard to compute the clique partitioning number of generic monopolar graphs.

Our construction relies on a reduction from the well-known Set Covering problem. An instance of the Set Covering problem is a triple $(\mathcal{U}, \mathcal{T}, \ell)$ where $\mathcal{U}$ is a set, $\mathcal{T}$ is a collection of k subsets of $\mathcal{U}$ such that $\bigcup_{T \in \mathcal{T}} T = \mathcal{U}$ and $\ell \leq k$ is a positive integer. A subset $\mathcal{T}' \subseteq \mathcal{T}$ such that $\bigcup_{T \in \mathcal{T}'} T = \mathcal{U}$ is said to *cover* $\mathcal{U}$, and it is called a *feasible cover* if it additionally satisfies $|\mathcal{T}'| \leq \ell$. Deciding whether a generic instance of the Set Covering problem has a feasible cover is **NP**-complete [12, p. 222].

Given a Set Covering instance $J = (\mathcal{U}, \mathcal{T}, \ell)$ as above, we construct an instance of the Partitioning-Covering problem described in the introduction as follows. First, let $E = \{e_1, \dots, e_{k-\ell}\}$ be a set of dummy elements such that $e_i \notin \mathcal{U}$ for $i = 1, \dots, k-\ell$. We define ingredients $N = \mathcal{U} \cup E$, containers $\mathcal{C} = \{T \cup E : T \in \mathcal{T}\}$ and $\mathcal{P} = \{\mathcal{U}, \{e_i\} : i = 1, \dots k - \ell\}$. Let I_J be the Partitioning-Covering instance defined by N, $\mathcal{C}$ and $\mathcal{P}$. We observe that the size of I_J is polynomial in the size of J.

Lemma 2.1. *Instance J has a feasible cover if and only if I_J is feasible. Thus, the* Partitioning-Covering *problem is* **NP***-complete.*

Proof. It is not restrictive to assume that a feasible cover $\mathcal{T}'$ of J consists of ℓ elements of $\mathcal{T}$. Let $\mathcal{T} \setminus \mathcal{T}' = \{T_1, \dots T_{k-\ell}\}$. We consider the partition of $\mathcal{C}$ given by the sets $S_i = \{T_i \cup E\}$ for every $i = 1, \dots, k - \ell$ and $S_{k-\ell+1} = \{T \cup E : T \in \mathcal{T}'\}$. We assign S_i to $\{e_i\}$ for every $i = 1, \dots, k - \ell$ and $S_{k-\ell+1}$ to $\mathcal{U}$. Then I_J is feasible since $|\mathcal{P}| = k - \ell + 1$ and $\mathcal{T}'$ covers $\mathcal{U}$.

Conversely, if I_J is feasible, $\mathcal{C}$ is partitioned so that every part is assigned to exactly one element of $\mathcal{P}$. Let $S \subseteq \mathcal{C}$ be the part assigned to $\mathcal{U}$ in such a partition. Since $|\mathcal{C}| = k$ and $|E| = k - \ell$ then S contains at most ℓ sets $C_1, \dots, C_h$ of $\mathcal{C}$ with $h \leq \ell$. Finally, $\mathcal{T}' = \{T_1, \dots, T_h\}$ defined by $T_i = C_i \setminus E$ for every $i \in \{1, \dots, h\}$ is a feasible cover of J, since $T_i \subseteq \mathcal{T}$ and $e_j \notin \mathcal{U}$ for $j = 1, \dots, k - \ell$, so $\mathcal{T}'$ covers $\mathcal{U}$. Hence the Partitioning-Covering problem is **NP**-complete. □

Proposition 2.2. *Computing the clique partitioning number on the class of monopolar graphs is* **NP***-hard.*

Proof. Immediate from Proposition 1.1 and Lemma 2.1, the graph $\mathcal{G}_I$ being monopolar for every Partitioning-Covering instance I, as proven in the introduction. □

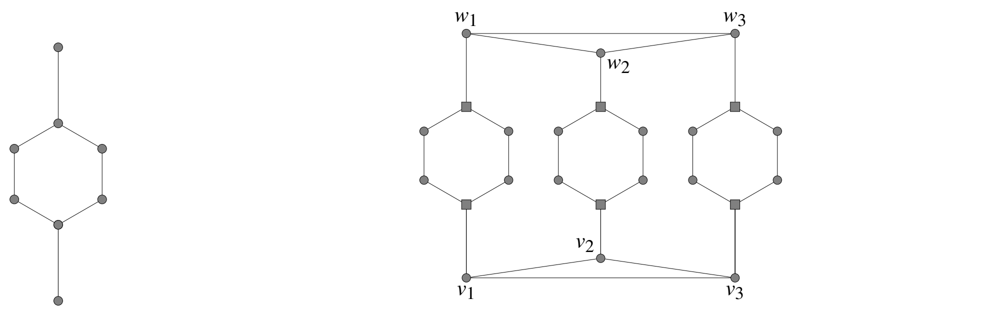
The specific structure of instance I_J constructed for the proof of Lemma 2.1 also allows us to prove that it is **NP**-hard to determine whether $\overline{\chi}(G) = \alpha(G)$ for a monopolar graph G. This latter problem has been shown to be **NP**-hard on generic graphs in [4]. For next proposition, we adapt the proof of [4].

Proposition 2.3. *Deciding whether $\overline{\chi}(G) = \alpha(G)$ for a generic monopolar graph G is* **NP***-hard even if some minimum stable set of G is known.*

Proof. Given an instance $J = (\mathcal{U}, \mathcal{T}, \ell)$ of the Set Covering problem, let I_J be the Partitioning-Covering instance constructed as above, with C its set of containers. Let also $\mathcal{G}_{I_J} = (\mathcal{V}_{I_J}, \mathcal{E}_{I_J})$ be the graph corresponding to I_J as described in the introduction. Clearly, $\mathcal{G}_{I_J}$ has size polynomial in the size of J. Moreover, $\mathcal{V}_{I_J}$ has a monopolar partition (A, B) with every vertex in A corresponding to an element of C and every maximal clique of $\mathcal{G}_{I_J}[B]$ corresponding to an element of $\mathcal{P}$. In particular, $\mathcal{G}_{I_J}$ has a vertex in B for every set $\{e_i\}$ with $i = 1, \ldots, k - \ell$. We call F the set of these vertices. Then $A \cup F$ induces a complete bipartite subgraph of $\mathcal{G}_{I_J}$. From $\ell \geq 1$ we get $|F| \leq |A| - 1$. By construction, every vertex in the maximal clique of $\mathcal{G}_{I_J}[B]$ corresponding to $\mathcal{U}$ is adjacent to at least one vertex in A, since $\mathcal{T}$ covers $\mathcal{U}$. Finally, we observe that $|C| = |\mathcal{T}|$, so $\alpha(\mathcal{G}_{I_J}) = |A| = |C| = |\mathcal{T}|$. By Proposition 1.1 and Lemma 2.1, a polynomial-time algorithm for deciding whether $\overline{\chi}(G) = \alpha(G)$ for every monopolar graph G allows one to determine whether $\overline{\chi}(\mathcal{G}_{I_J}) = |\mathcal{T}|$ and, as a consequence, whether J has a feasible cover. This proves the result. □

Stability Number of Monopolar Graphs. We give a reduction of the 3-Colorability problem on general graphs to the stable set problem on monopolar graphs. In the 3-Colorability problem we have to decide whether a given input graph admits a proper coloring with at most three colors. The 3-Colorability problem is **NP**-complete, see [12, p. 191].

For our purposes, we consider the gadget shown in Figure 2.1a. Its vertices of degree one will be called *extreme*. Let $G = (V, E)$ be a graph. We construct a graph $H_G = (V_G, E_G)$ from G by replacing each vertex $v \in V$ by three vertices v_1, v_2 and v_3 linked to form a K_3 and by joining the two cliques corresponding to v and w as in Figure 2.1b whenever $\{v, w\} \in E$. More precisely, for every pair $\{v_i, w_i\}$ where $i = 1, 2, 3$ and $\{v, w\}$ is an edge of G, we add a gadget having v_i and w_i as extreme vertices.



(a) Gadget used in H_G.

(b) Transformation of an edge $\{v, w\}$ of G into a monopolar subgraph of H_G. Square vertices are a stable set, round vertices induce a cluster.

Figure 2.1

Computing $\alpha(H_G)$ is enough to solve the 3-Colorability problem on G, as we prove in next lemma.

Lemma 2.4. *Let $G = (V, E)$ be a graph and $H_G = (V_G, E_G)$ be the associated monopolar graph defined above. Then G is 3-colorable if and only if $\alpha(H_G) = |V| + 9|E|$.*

Proof. Let $\mathcal{I} = \{1, 2, 3\}$, $C = \{v_i \in V_G : v \in V, i \in \mathcal{I}\}$ and $D = V_G \setminus C$. The graph $H_G[C]$ is a cluster consisting of $|V|$ vertex-disjoint K_3 graphs, hence $\alpha(H_G[C]) = |V|$. The graph $H_G[D]$ is the union of

$3|E|$ vertex-disjoint cycles of length six, thus $\alpha(H_G[D]) = 9|E|$. Since C and D partition V_G, we get that $\alpha(H_G) \leq |V| + 9|E|$. The same argument also proves that the right-hand-side value is reached only by the cardinality of stable sets including exactly one vertex for each K_3 corresponding to a vertex of V and exactly three vertices per cycle being part of the gadgets corresponding to the edges of G.

So, if S is a maximum stable set of H_G of cardinality $|V| + 9|E|$, we get that $v_i \in S$ implies $w_i \notin S$ whenever $\{v, w\} \in E$. Otherwise, S would contain at most two vertices in the cycle of the gadget having v_i and w_i as extreme vertices. It follows that, whenever $\{v, w\} \in E$, if $v_i \in S$ for some $i \in \mathcal{I}$ then $w_j \in S$ for some $j \in \mathcal{I} \setminus \{i\}$, as S contains one vertex for each K_3 corresponding to a vertex of V. As a consequence, assigning color $i \in \mathcal{I}$ to vertex v such that $v_i \in S$ yields a proper coloring of G using at most three colors.

Conversely, let G be 3-colorable with colors in $\mathcal{I}$. We define the stable set $S_1 = \{v_i \in V_G : v \in V \text{ has color } i \in \mathcal{I}\}$. Let us consider the graph H'_G obtained from H_G by removing all vertices in S_1 and their neighbors. Since every vertex $v \in V$ is assigned a color this implies that all K_3 graphs corresponding to the vertices of G are removed. Moreover, at most one vertex per gadget is removed since for every edge $\{v, w\} \in E$ vertices v and w are assigned distinct colors. It follows that H'_G has $3|E|$ connected components each being either a path on five vertices or a cycle on six vertices. All these connected components admit a stable set of size three, hence a maximum stable set S_2 of H'_G has size $9|E|$. Now, $S = S_1 \cup S_2$ is a stable set of H_G of cardinality $|V| + 9|E|$, hence it is a maximum stable set of H_G. □

Proposition 2.5. *Computing the stability number on the class of monopolar graphs is* **NP**-*hard.*

Proof. The size of $H_G = (V_G, E_G)$ is polynomial in the size of G. By Lemma 2.4 it is enough to prove that H_G is monopolar for every graph G. Let us consider the partition (A, B) of V_G where A contains all vertices of degree three of the gadgets corresponding to the edges of G, while B contains all other vertices of V_G. (Figure 2.1b illustrates this partition on the graph H_{K_2}.) By construction of H_G the vertices of distinct gadgets corresponding to the edges of G are not adjacent except for their extreme vertices, thus A is a stable set. The same argument shows that a P_3 in $H_G[B]$ can only be induced by three extreme vertices of the gadgets used in the construction of H_G. However, every connected subgraph containing three extreme vertices is a K_3. Hence $H_G[B]$ is a cluster and (A, B) a monopolar partition of V_G. □

Chromatic Number of Monopolar Graphs. We conclude the section by proving that the 3-Colorability problem on monopolar graphs is **NP**-complete. This immediately proves that computing the chromatic number of monopolar graphs is **NP**-hard in general. We adapt a well-known reduction of the 3-SAT problem to 3-Colorability problem on general graphs [13, Thm. 2.1].

An instance of the 3-SAT problem is a set of disjunctive clauses each consisting of three literals from a given set of positive and negated variables. The goal is to determine the existence of a *truth assignment* for the instance, *i.e.,* an assignment of boolean values to the variables making all clauses true. The 3-SAT problem is **NP**-complete [12, p. 259].

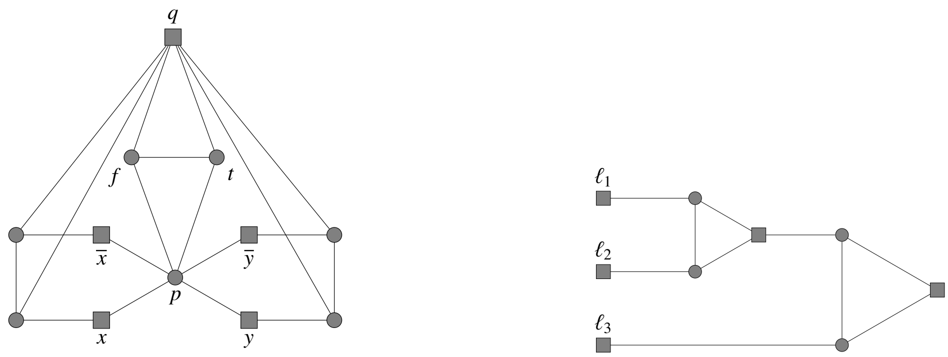
Our reduction relies on two gadgets. The first gadget is constructed by first taking a *diamond* obtained from K_4 by removing an edge. We call p and q the two vertices of the diamond of degree two and f and t the other two vertices. For every variable x of the given instance, we add a cycle of length five having p as a vertex. The neighbors of p in the cycle of variable x will be referred to as x and $\bar{x}$. Finally, we link the remaining two vertices of the cycle to vertex q. In Figure 2.2a we illustrate this gadget for two variables x and y.

The second gadget is depicted in Figure 2.2b and it is the same that is used in [13]. We call it *clause gadget*. A vertex of a clause gadget is a *literal vertex* if it has degree one and *truth vertex* if it has degree two.

Given an instance I of the 3-SAT problem, we construct a graph G_I as follows. We start with a gadget of the first type as above. Subsequently, for every clause $C = (\ell_1, \ell_2, \ell_3)$ of I we create a clause gadget whose literal vertices are identified with the vertices of the first gadget corresponding to the same literals. Finally, we link the truth vertex of each clause gadget to vertices p and f.

Lemma 2.6. *The graph G_I is monopolar for every instance I of the* 3-SAT *problem.*

Proof. The vertex set of the gadget of first type admits a monopolar partition (A, B) with $f, p \in B$ and $x, \bar{x} \in A$ for every variable x, see Figure 2.2a. The vertex set of a clause gadget has a monopolar partition



(a) First monopolar gadget. Square vertices are in A, round vertices in B.

(b) Clause gadget. It is monopolar: Square vertices are in A, round vertices in B.

Figure 2.2

(A, B) in which all literal vertices and the truth vertex belong to A, see Figure 2.2b. Hence identifying the literal vertices across gadgets of different type and linking all truth vertices to p and f does not break the monopolarity. □

We just sketch the proof of next lemma, as the argument is the same as in classical reductions of the 3-SAT problem to the 3-Colorability problem given in [13, Thm. 2.1].

Lemma 2.7. *Given an instance I of the* 3-SAT *problem, the graph G_I is* 3*-colorable if and only if there is a truth assignment for I.*

Proof. The gadget of first type is 3-colorable. Let F and T be the colors respectively assigned to f and t and let N be the third color in such a 3-coloring (note that p and q are colored N). A literal is assigned boolean value `true` if the corresponding vertex in the first gadget is colored T, otherwise it is assigned `false`. It is easy to see that, for every variable x of I, vertices x and $\bar{x}$ cannot be colored N and must have distinct colors, so the above is a consistent assignment of boolean values to the variables. As observed in [13], under the above 3-coloring, the truth vertex of a clause gadget can be colored T if and only if at least one literal vertex in the same clause is. Since the truth vertices are all linked to f and p, the graph G_I is 3-colorable if and only if I admits a truth assignment. □

The proof of the following proposition is now immediate.

Proposition 2.8. *The* 3-Colorability *problem on monopolar graphs is* **NP***-complete. Computing the chromatic number on monopolar graphs is* **NP***-hard.*

In Section 2.2 we show that the largest clique of a monopolar graph can be found in polynomial time. In view of this fact, the result of Proposition 2.8 is quite surprising after observing that $\omega(G) \leq \chi(G) \leq \omega(G) + 1$ for every monopolar graph $G = (V, E)$. It is well-known that the lower bound holds for every graph. For the upper bound note that if (A, B) is a monopolar partition of V, then the maximal cliques of $G[B]$ can be colored with at most $\omega(G)$ colors. Then we can assign an additional color to all vertices in A to obtain a proper coloring of G.

2.2. Polynomial-Time Algorithms for Clique Problems on Monopolar Graphs

We follow to a large extent the definitions given in [21]. A clique is *maximal* if it is not contained in another clique. A graph class is *hereditary* if it is closed under taking induced subgraphs. A hereditary graph class has *few cliques* if there exists a polynomial $p(n)$ such that every $G = (V, E)$ in the class has no more than $p(|V|)$ maximal cliques. The *octahedral graph* O_m is obtained from K_{2m} by removing a perfect matching.

For every $m \in \mathbb{Z}_+$ the octahedral graph O_m is a complete m-partite graph with every part of the partition containing exactly two vertices. Moreover, taking one vertex in each part induces a maximal clique of size m in O_m and this easily shows that the class of graphs containing all octahedral graphs has not few cliques. However, octahedral graphs are the only forbidden graphs in hereditary classes having few cliques, as stated in the following result.[1]

Theorem 2.9 (see [20])**.** *A hereditary graph class* $\mathbf{G}$ *has few cliques if and only if* $O_m \notin \mathbf{G}$ *for some constant* $m \in \mathbb{Z}_+$.

We now prove a corollary of Theorem 2.9.

Corollary 2.10. *Let* $m \geq 1$ *be a fixed integer. The class of* $(K_m$*-free*$, P_3$*-free*$)$*-graphs has few cliques.*

Proof. The class of $(K_m$-free, P_3-free$)$-graphs is hereditary. Thus by Theorem 2.9 it suffices to show that O_{m+1} is not a $(K_m$-free, P_3-free$)$-graph. Assume it is. Let A and B denote a vertex partition of O_{m+1} such that $O_{m+1}[A]$ is K_m-free and $O_{m+1}[B]$ is P_3-free. We recall that O_{m+1} is a complete $(m+1)$-partite graph. If there are m parts in the $(m+1)$-partition of O_{m+1} each having at least one vertex in A, then $O_{m+1}[A]$ contains a K_m. As a consequence, there are at least two parts contained in B but this contradicts the hypothesis that $O_{m+1}[B]$ is P_3-free. □

Corollary 2.11. *Let* $G = (V, E)$ *be a monopolar graph and* $c\colon V \to \mathbb{R}$ *be a weight function on* V*. Then the* Max-Weight Clique *problem* $\max\{c(K)\colon K \text{ is a clique of } G\}$ *can be solved in polynomial time. In particular,* $\omega(G)$ *can be determined in polynomial time.*

Proof. The result holds for all graph classes having few cliques, as shown in [21]. By Corollary 2.10 this is the case for monopolar graphs which are exactly the $(K_2$-free, P_3-free$)$-graphs. □

We conclude with a few observations.

Remark 2.12. Let $G = (V, E)$ be a monopolar graph on n vertices and m edges and let us assume to know a monopolar partition (A, B) of V. Then, representing G by an adjacency list, the maximal cliques of $G[B]$ can be constructed in $O(|B|)$ time. There are $O(|B|)$ such cliques. Every $v \in A$ together with its neighborhood in a maximal clique H of $G[B]$ induces a maximal clique of G, whenever the neighborhood of v in H is nonempty. Moreover, an edge incident to $v \in A$ belongs to exactly one such a clique. It follows that there are $O(m)$ maximal cliques of G with a vertex in A and a vertex in B and all of them can be constructed in $O(m)$ when the maximal cliques of $G[B]$ are known. Every other maximal clique of G is either maximal in $G[B]$ or an isolated vertex of A. Since $|A| + |B| = n$ constructing all maximal cliques of G takes $O(n + m)$ time if (A, B) is known. The above discussion shows that G has $O(n + m)$ maximal cliques. Consequently, the Max-Weight Clique problem can be solved in $O(n + m)$ time on G if (A, B) is known. In general, the h maximal cliques of a graph with n vertices and m edges can be listed in $O(hnm)$ time [23]. Thus the maximal cliques of a monopolar graph can be listed in $O(n^2m + nm^2)$ time if no monopolar partition is explicitly known and this gives a more direct proof of Corollary 2.11.

Remark 2.13. Corollary 2.11 and the results shown in [21] imply that the problem of partitioning the vertices of a monopolar graph into at most k cliques is polynomially solvable whenever k is constant (*i.e.,* it is not part of the input). We sketch the overall idea of a polynomial algorithm, referring the reader to [21, p. 133] for a more detailed treatment. One first shows that only maximal cliques are needed in a k-covering of the vertices into cliques; next, since k is constant and the number of maximal cliques is polynomially bounded, enumerating all subsets of maximal cliques of cardinality k can be done in polynomial time; finally, evaluating whether a set of cliques covers all vertices can be done in quadratic time and this yields the result.

Acknowledgements

This research is partially funded by Regione Lombardia, POR FESR 2014-2020, project AD-COM – Advanced Cosmetic Manufacturing [1].

[1] We point out that we were not able to find the resource [20]. However, Theorem 2.9 is cited in [21] as well as in the following webpage maintained by the author of [20]: `http://www.eprisner.de/Journey/Cliques.html`.

References

[1] AD-COM – Advanced Cosmetic Manufacturing. https://ad-com.net/?lang=en. Accessed: 2019-01-28.

[2] J. A. Bondy and U. S. R. Murty. *Graph theory, volume 244 of Graduate Texts in Mathematics*. Springer, New York, 2008.

[3] S. Bruckner, F. Hüffner, and C. Komusiewicz. A graph modification approach for finding core–periphery structures in protein interaction networks. *Algorithms for Molecular Biology*, 10(1):16, 2015.

[4] S. Busygin and D. V. Pasechnik. On NP-hardness of the clique partition-independence number gap recognition and related problems. *Discrete Mathematics*, 306(4):460–463, 2006.

[5] R. Churchley and J. Huang. Line-polar graphs: characterization and recognition. *SIAM Journal on Discrete Mathematics*, 25(3):1269–1284, 2011.

[6] R. Churchley and J. Huang. On the polarity and monopolarity of graphs. *Journal of Graph Theory*, 76(2):138–148, 2014.

[7] R. Churchley and J. Huang. Solving partition problems with colour-bipartitions. *Graphs and Combinatorics*, 30(2):353–364, 2014.

[8] T. Ekim, P. Heggernes, and D. Meister. Polar permutation graphs. In *International Workshop on Combinatorial Algorithms*, pages 218–229. Springer, 2009.

[9] T. Ekim, P. Hell, J. Stacho, and D. de Werra. Polarity of chordal graphs. *Discrete Applied Mathematics*, 156(13):2469–2479, 2008.

[10] E. M. Eschen and X. Wang. Algorithms for unipolar and generalized split graphs. *Discrete Applied Mathematics*, 162:195–201, 2014.

[11] A. Farrugia. Vertex-partitioning into fixed additive induced-hereditary properties is NP-hard. *The Electronic Journal of Combinatorics.*, 11(1):R46, 2004.
URL: http://www.combinatorics.org/Volume_11/PDF/v11i1r46.pdf.

[12] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W.H. Freeman & Co., New York, 1999.

[13] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical computer science*, 1(3):237–267, 1976.

[14] M. C. Golumbic. Trivially perfect graphs. *Discrete Mathematics*, 24(1):105–107, 1978.

[15] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.

[16] I. Kanj, C. Komusiewicz, M. Sorge, and E. J. van Leeuwen. Parameterized algorithms for recognizing monopolar and 2-subcolorable graphs. *Journal of Computer and System Sciences*, 92:22–47, 2018.

[17] H.-O. Le and V. B. Le. The NP-completeness of $(1,r)$-subcolorability of cubic graphs. *Information Processing Letters*, 81(3):157–162, 2002.

[18] V. B. Le and R. Nevries. Complexity and algorithms for recognizing polar and monopolar graphs. *Theoretical Computer Science*, 528:1–11, 2014.

[19] V. V. Lozin and R. Mosca. Polar graphs and maximal independent sets. *Discrete Mathematics*, 306(22):2901–2908, 2006.

[20] E. Prisner. Graphs with few cliques. *Graph theory, Combinatorics, and Algorithms*, 1:2, 1995.

[21] B. Rosgen and L. Stewart. Complexity results on graphs with few cliques. *Discrete Mathematics and Theoretical Computer Science*, 9(1):127–136, 2007.

[22] M. Talmaciu and E. Nechita. On polar, trivially perfect graphs. *International Journal of Computers Communications & Control*, 5(5):939–945, 2010.

[23] S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing*, 6(3):505–517, 1977.

[24] R. I. Tyshkevich and A. A. Chernyak. Algorithms for the canonical decomposition of a graph and recognizing polarity. *Izvestia Akad. Nauk BSSR, ser. Fiz.-Mat. Nauk*, 6:16–23, 1985.

[25] R. I. Tyshkevich and A. A. Chernyak. Decomposition of graphs. *Cybernetics*, 21(2):231–242, 1985.