Discrete Applied Mathematics xxx (xxxx) xxx

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

# The multicolored graph realization problem

Josep Díaz [a,d], Öznur Yaşar Diner [b,c], Maria Serna [a,d,*], Oriol Serra [c,d]

[a] ALBCOM Research Group, Computer Science Department, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
[b] Computer Engineering Department, Kadir Has University, Istanbul, Turkey
[c] Mathematics Department, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
[d] Institut de Matemàtiques de la UPC-BarcelonaTech (IMTech), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

ARTICLE INFO

*Article history:*
Received 22 April 2021
Received in revised form 10 June 2022
Accepted 15 June 2022
Available online xxxx

*Keywords:*
Multicolored realization problem
Generalized combinatorial problems
Parameterized complexity
Convex bipartite graphs

ABSTRACT

We introduce the *multicolored graph realization* problem (MGR). The input to this problem is a colored graph $(G, \varphi)$, i.e., a graph $G$ together with a coloring $\varphi$ on its vertices. We associate each colored graph $(G, \varphi)$ with a *cluster graph* $(G_\varphi)$ in which, after collapsing all vertices with the same color to a node, we remove multiple edges and self-loops. A set of vertices $S$ is *multicolored* when $S$ has exactly one vertex from each color class. The MGR problem is to decide whether there is a multicolored set $S$ so that, after identifying each vertex in $S$ with its color class, $G[S]$ coincides with $G_\varphi$.

The MGR problem is related to the well-known class of generalized network problems, most of which are NP-hard, like the generalized Minimum Spanning Tree problem. The MGR is a generalization of the multicolored clique problem, which is known to be $W[1]$-hard when parameterized by the number of colors. Thus, MGR remains $W[1]$-hard, when parameterized by the size of the cluster graph. These results imply that the MGR problem is $W[1]$-hard when parameterized by any graph parameter on $G_\varphi$, among which lies treewidth. Consequently, we look at the instances of the problem in which both the number of color classes and the treewidth of $G_\varphi$ are unbounded. We consider three natural such graph classes: chordal graphs, convex bipartite graphs and 2-dimensional grid graphs. We show that MGR is NP-complete when $G_\varphi$ is either chordal, biconvex bipartite, complete bipartite or a 2-dimensional grid. Our reductions show that the problem remains hard even when the maximum number of vertices in a color class is 3. In the case of the grid, the hardness holds even for graphs with bounded degree. We provide a complexity dichotomy with respect to cluster size.



## 1. Introduction

It is well known that graphs are important tools to model many systems in different disciplines. In particular graph partitioning and graph clustering are key techniques in various areas of computer science, engineering, biology, epidemiology, social science, etc. For example, when dealing with the analysis of large social nets, the modelization of infection spreading, route planning, community detection in social networks and high performance computing. In many of these applications large graphs are partitioned as to control the structural connections among the clusters (the elements of the partition). Given a partition of the vertices into clusters, the topological notion of the *graph quotient* provides a way to obtain a cluster graph as a summary of the input graph. The cluster graph provides a simpler and compact form of the

* Corresponding author at: ALBCOM Research Group, Computer Science Department, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain.
*E-mail addresses:* diaz@cs.upc.edu (J. Díaz), oznur.yasar@khas.edu.tr (Ö.Y. Diner), mjserna@cs.upc.edu (M. Serna), oriol.serra@upc.edu (O. Serra).

https://doi.org/10.1016/j.dam.2022.06.031

complex network, extracted from an adequate partition of the data, summarizing the relevant relationships among the clustered data. Graph quotients have many applications in the study of data sets containing complex relationships (see for example [31]) and they have motivated the study of generalized optimization problems.

Classical combinatorial optimization problems can be generalized in a natural way, by considering a related problem relative to a given partition of the vertices of the graph. Those *generalized combinatorial optimization problems* have the following primary features: the graph is given together with a partition of its vertices in clusters and, when considering the feasibility constraints of the graph problem, these are expressed in relation to the clusters, rather than as individual vertices. Some interesting and intensively studied problems belonging to this category are as follows: the generalized traveling sales person problem [14–16], the generalized minimum spanning tree problem [22,25], the generalized shortest path tree problem [1,5], the generalized vehicle routing problem [20,27], the partition graph coloring problem, [3,4], among others. For further references on the category of generalized combinatorial optimization problems, we point to [10,13,26] and references therein.

Further problems involving graphs and a partition are the *multicolored clique* and the *multicolored independent set* problems [12,24]. In this formalism a partition is seen as a coloring (which is not necessarily proper). The goal of the problems is to select a *multicolored set* having a vertex of each color, that induces a clique or an independent set, respectively. The *multicolored clique* problem has been studied from the parameterized complexity point of view [12]. The problem is known to be W[1]-hard, when parameterized by the number of colors, i.e., the number of sets in the partition.

In this paper, we introduce another generalized combinatorial problem, the *multicolored graph realization problem* (MGR): given a graph together with a partition of its vertices (a colored graph), decide whether there is a multicolored set inducing the cluster graph, i.e., the quotient graph with respect to the given partition. In network applications it is quite natural to partition the nodes into non-overlapping clusters. In communication networks such clusters may identify local-area subnetworks and in social networks communities of individuals. Inter cluster connections allow to infer structure of the network in a higher level. We ask whether the high level network structure can be found at the network level. This problem has immediate practical motivations. For instance, in a communication network, such a realization provides a set of nodes and links to reinforce in order to improve the cluster network, without modifying the local-area networks. In a social network, individual representatives provide a proof that the inferred social relationship among communities hold also at the individual level. In data analysis applications, the problem is equivalent to asking whether we can obtain particular data fulfilling all the inferred relations.

The MGR problem is solvable in $O(n^k poly(n))$ time, which is polynomial when the number of colors $k$ is a constant. But, it is W[1]-hard, when parameterized by the number of colors, as it generalizes the multicolored clique problem. Observe that, under this parameterization, the cluster graph has constant size, and therefore, all graph parameters on the cluster graph are constant. We are interested in analyzing the complexity of the MGR problem when both the number of colors and the treewidth of the cluster graph are unbounded. Our first result, based on the complexity of the Multicolored Clique Problem, is stated as follows (see Section 2 for the appropriate definitions and terminology):

**Theorem 1.** *The MGR problem is W[1]-hard when parameterized by the number of colors or parameterized by the treewidth of the cluster graph.*

We next focus on specific classes of graphs for which the complexity of natural problems has been widely studied. In particular in the class of chordal graphs which form an intensively studied graph class both within structural graph theory and within algorithmic graph theory. Recall that several problems that are hard on other classes of graphs such as graph coloring may be solved in polynomial time on chordal graphs [19]. We show that the MGR problem remains NP-complete for colored graphs whose cluster graph is a chordal bipartite graph. In particular, we consider the subclasses of convex bipartite graphs and biconvex bipartite graphs which have been used as a benchmark for complexity of homomorphism problems, see e.g. [6,11,21]. We show that the MGR problem is NP-complete for colored graphs whose cluster graph is biconvex bipartite (see Section 3).

**Theorem 2.** *The MGR problem is NP-complete, for colored graph having a biconvex bipartite cluster graph, even when the cluster size is at most 3.*

We extend this result to cluster graphs that are chordal (see Section 3). The hardness results also hold in case the number of vertices in a color class is constant.

**Theorem 3.** *The MGR problem is NP-complete, for colored graphs having a chordal cluster graph, even when the cluster size is at most 3.*

We complement this result by showing that the MGR problem belongs to FPT, for colored graphs having a convex bipartite cluster graph, when parameterized by the size of the clusters and the maximum degree of the non ordered part.

A third family of bipartite graphs we analyze are the 2–dimensional grid graphs. We are showing the hardness result in this case as well (see Section 4).

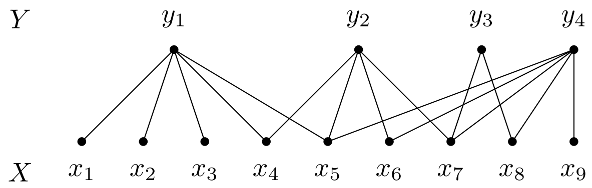



**Fig. 1.** A convex bipartite graph.

**Theorem 4.** *The MGR problem is NP-complete, for colored graphs whose cluster graph is a 2-dimensional grid, even when the cluster size is 6 and the input graph has bounded degree.*

In view of those results, we analyze the computational complexity of the problem with respect to the size of the color classes. We provide a complexity dichotomy with respect to this parameter (see Section 5).

**Theorem 5.** *The MGR problem is NP-complete, for colored graphs with cluster size $s \geq 3$, and polynomial time solvable otherwise.*

We also show that the MGR problem, under the double parameterization by cluster size and treewidth of the cluster graph belongs to FPT.

## 2. Definitions and preliminaries

In this section, we provide the definitions and terminology used in the paper. We follow notation and basic terminology in graph theory from Diestel [7].

We consider finite, simple and undirected graphs $G = (V, E)$, i.e., without multiple edges or loops. For $S \subseteq V$, $G[S]$ represents the graph induced by $S$, defined as $G[S] = (S, E \cap \binom{S}{2})$.

A *chordal graph* is one in which all cycles of four or more vertices have a chord. A bipartite graph is represented by $G = (X \cup Y, E)$, where $X$, $Y$ form a bi-partition of the vertex set and $E \subseteq X \times Y$. Given a bipartite graph $G = (X \cup Y, E)$, an ordering of the vertices $X$ has the *adjacency property* (or the ordering is said to be *convex*) if, for each vertex $v \in Y$, $N(v)$ consists of vertices which are consecutive in the ordering of $X$. *Convex bipartite graphs* are the bipartite graphs $G = (X \cup Y, E)$ that have the adjacency property on one of the partite sets (let us say $X$). *Biconvex bipartite graphs* are the bipartite graphs $G = (X \cup Y, E)$ that have the adjacency property on both partite sets. Fig. 1 shows a convex bipartite graph that is not biconvex. It is known that there are linear time recognition algorithms for these graphs class [23].

*Treewidth.* In our results involving treewidth we will use the particular kind of a decomposition called *nice tree decompositions*. For definitions and notation concerning tree decompositions we refer the reader to [2].

Let $(T, X)$ be a tree decomposition of a graph $G$. We can make any tree $T$ into a rooted tree by choosing a node $r \in V(T)$ as the root, and directing all edges to the root. In this way we can convert a tree decomposition $(T, X)$ into a *rooted tree decomposition*, by fixing one node $r$ as the root in $T$. A rooted tree decomposition $(T, X, r)$ of $G$ allows us to associate to every node in the graph a subgraph of $G$ as follows: For $v \in V(T)$, let $R_T(v)$ denote the set of nodes in the subtree rooted at $v$ (including $v$). For $v \in V(T)$, define $V(v) = \cup_{w \in R_T(v)} X_w$ as the set of vertices included in any bag in the subtree rooted at $v$. Finally, define the associated graph as $G(v) = G[V(v)]$, the subgraph induced by $V(v)$. Observe that $G(r) = G$ and that $X_v$ is a separator in $G$.

A *nice tree decomposition* is a variant in which the structure of the nodes is simpler. A rooted tree decomposition $(T, X)$ is *nice* if each node $u \in V(T)$ can be classified in one of the following four types.

- *start* node: $u$ has no child and $|X_u| = 1$.
- *forget* node: $u$ has one child $v$ and $X_u \subseteq X_v$ and $|X_u| = |X_v| - 1$.
- *introduce* node: $u$ has one child $v$ and $X_v \subseteq X_u$ and $|X_u| = |X_v| + 1$.
- *join* node: $u$ has two children $v$ and $w$ with $X_u = X_v = X_w$.

Given a tree decomposition of width $k$ for a graph $G$, a rooted nice tree decomposition with width at most $k$ for $G$ and a polynomial number of nodes can be obtained in $O(kn)$ time (see for example [2]).

*Complexity classes.* Many NP-complete problems can be associated with one or more parameterizations. A *parameterization* is a function $\kappa$ assigning a non negative integer value to each input $x$ [17]. A *fixed parameter tractable (FPT)* algorithm is an algorithm solving a problem parameterized by $\kappa$ that on input $x$ takes time

$$f(\kappa(x)) \cdot |x|^{\Theta(1)},$$

where $f(\kappa)$ is a (super-polynomial) function that does not depend on $|x|$. The Parameterized Complexity settles the question of whether a parameterized problem is solvable by an FPT algorithm. If such an algorithm exists, we say that the

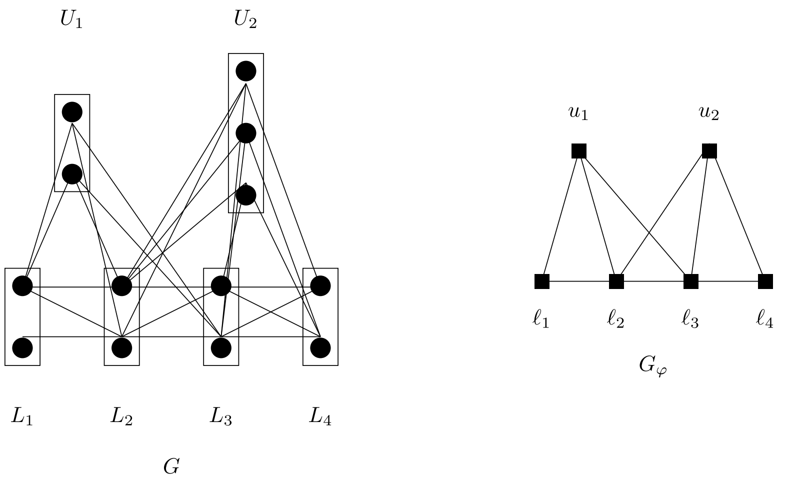



**Fig. 2.** A colored graph and its cluster graph.

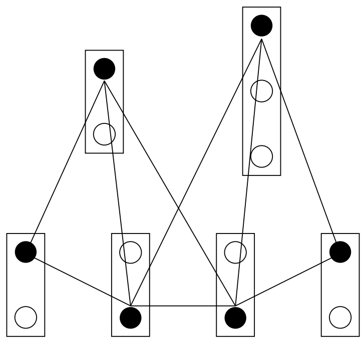


**Fig. 3.** In black a multicolored set $S$ realizing $G_\varphi$, for the colored graph given in Fig. 2.

parameterized problem belongs to the class FPT of fixed parameter tractable problems. In a series of fundamental papers (see [8,9]), Downey and Fellows introduced a series of complexity classes, namely the classes $FPT \subseteq W[1] \subseteq W[2] \subseteq \cdots \subseteq W[SAT] \subseteq W[P]$, and proposed special types of reductions such that hardness for some of the above classes makes it rather impossible that a problem belongs in FPT.

*The multicolored graph realization problem.* A *coloring* of a graph $G = (V, E)$ is a map $\varphi : V \to \mathbb{N}$. Observe that our colorings are not necessarily proper as we do not require that adjacent vertices get different colors. However, all our results will also hold for proper colorings. Given a coloring $\varphi$ on $G$, let $k(G, \varphi)$ be the number of different colors used by $\varphi$. We use the term *colored graph* to refer to a pair $(G, \varphi)$.

Given a colored graph $(G, \varphi)$ with $G = (V, E)$, we say that two vertices $u$ and $v$ are equivalent whenever they get the same color, i.e., $u \sim_\varphi v$ if and only if $\varphi(u) = \varphi(v)$. This is a natural equivalence relation that partitions the vertices of $G$ into non empty color classes. We use $[u]_\varphi$ (or just $[u]$ when $\varphi$ is clear from the context) to denote the color class of $u$.

Given a colored graph $(G, \varphi)$, with color classes $A_1, \ldots, A_k$, the associated *cluster graph* is the graph $G_\varphi = (\{1, \ldots, k\}, E_\varphi)$ where, for $i, j \in \{1, \ldots, k\}$ with $i \neq j$, there is an edge $(i, j) \in E_\varphi$ whenever there is an edge $(u, v) \in E$ with $u \in A_i$ and $v \in A_j$. Fig. 2 gives an example of a colored graph, each rectangle representing a color class, and its associated cluster graph. We say that a set of vertices $S \subseteq V$ is *multicolored* if, for any $1 \leq i \leq k$, we have that $|S \cap A_i| = 1$, i.e., there is exactly one vertex in $S$ from each color class. For a multicolored set $S$, we assume that $S = \{u_1, \ldots, u_k\}$ so that, for $1 \leq i \leq k$, $[u_i] = A_i$. A *multicolored realization* of $G_\varphi$ is a multicolored subset $S \subseteq V$ such that the restriction of $\varphi$ to $S$ is an isomorphism between $G[S]$ and $G_\varphi$, i.e., for $u_i, u_j \in S$, $(u_i, u_j) \in E(G)$ if and only if $(i, j) \in E(G_\varphi)$. Fig. 3 shows a multicolored set realizing $G_\varphi$, for the colored graph given in Fig. 2.

With this notation, we can state the definition of our problem formally.

Multicolored graph realization problem (MGR)

Instance: Undirected graph $G = (V, E)$ and a coloring $\varphi$ of $G$.

Question: Is there a multicolored realization of $G_\varphi$?

Observe that, given a colored graph $(G, \varphi)$ and a multicolored set $S$, we can check, in polynomial time, whether $S$ is a realization of $G_\varphi$. Therefore, the MGR problem belongs to NP.

We are interested in analyzing the computational complexity of the MGR problem under different parameterizations: the *number of used colors* $k(G, \varphi)$, the *cluster size* $s(G, \varphi) = \max_{v \in V} |\varphi^{-1}(\varphi(v))|$, the treewidth of the cluster graph, and other combinations of parameters. Although the problem is defined as usual in its decision form, the algorithms provided in the paper are constructive, they produce a multicolored realization in case that one exists.

Our first results follow from a well known problem called the Multicolored Clique Problem (also known as the Partitioned clique problem), which according to [2], was introduced in [12,24]. We provide here a formal definition of the problem adapted to our notation.

> MULTICOLORED CLIQUE PROBLEM (MC)
>
> INSTANCE: A colored graph $(G, \varphi)$.
>
> QUESTION: Is there a multicolored set $S$ such that $G[S]$ is a clique?

The MC problem is known to be $W[1]$-hard parameterized by the number of colors [12]. This yields the following result in the multicolored problem language.

**Theorem 1.** *The* MGR *problem is* $W[1]$*-hard when parameterized by the number of colors or parameterized by the treewidth of the cluster graph.*

**Proof.** Observe that, in a colored graph, a necessary condition to have a multicolored clique is the cluster graph $G_\varphi$ being itself a clique. Therefore, the MC problem is the particular case of the MGR problem when the cluster graph $G_\varphi$ is a complete graph. On the other hand, when the number of used colors $k$ is bounded, the cluster graph $G_\varphi$ has constant size, and therefore, has bounded treewidth. □

## 3. Chordal and convex bipartite cluster graphs

In this section, we analyze the complexity of the MGR problem on colored graphs having a cluster graph that is chordal or chordal bipartite. We start presenting the NP-hardness for the case of convex bipartite or biconvex bipartite graphs.

**Theorem 2.** *The* MGR *problem is NP-complete, for colored graph having a biconvex bipartite cluster graph, even when the cluster size is at most 3.*

**Proof.** As the MGR problem belongs to NP, we only have to show that the problem is NP-hard. For doing so, we provide a reduction from Monotone 1-in-3 SAT problem, which is known to be NP-complete [29]. The input formula $\Phi$ (in CNF) is the conjunction of $m$ clauses, $C_1, C_2, \ldots, C_m$, over a set of $n$ variables. Furthermore, each clause is the disjunction of exactly three non-negated variables. The problem asks whether it is possible to assign a value in $\{0, 1\}$ to each of the $n$ variables $x_1, x_2 \ldots x_n$, so that, in each clause, exactly one of the three variables is set to 1. As a shorthand, we write each clause as $(x, y, z)$, without the use of the $\vee$ symbol. Furthermore, we assume that the variables in a clause are given in increasing order, i.e. $C_j = (x_{j_1}, x_{j_2}, x_{j_3})$ with $j_1 \leq j_2 \leq j_3$

Given an input formula $\Phi$ to the Monotone 1-in-3 SAT problem, we construct a colored graph $(G, \varphi)$. Instead of describing the coloring $\varphi$, we provide the different color classes and the connection among their vertices. In this way, it is easy to see that the construction provides a colored graph with a biconvex bipartite cluster graph. For each Boolean variable $x_i$, $1 \leq i \leq n$, we create a color class $X_i$ having two vertices $v_i^0$ and $v_i^1$. For each clause $C_j$, $1 \leq j \leq m$, we create a color class $Y_j = \{u_j^{100}, u_j^{010}, u_j^{001}\}$ with one vertex for each possible join assignment with only one 1.

The edge set in $G$ is the following:

1. For a clause $C_j = (x_{j_1}, x_{j_2}, x_{j_3})$, we connect:
   - $u_j^{100}$ with $v_{j_1}^1$, $v_{j_2}^0$ and $v_{j_3}^0$ and with both $v_i^0$ and $v_i^1$, for $1 \leq i \leq n$ with $i \neq j_1, j_2, j_3$;
   - $u_j^{010}$ with $v_{j_1}^0$, $v_{j_2}^1$ and $v_{j_3}^0$ and with both $v_i^0$ and $v_i^1$, for $1 \leq i \leq n$ with $i \neq j_1, j_2, j_3$;
   - $u_j^{001}$ with $v_{j_1}^0$, $v_{j_2}^0$ and $v_{j_3}^1$ and with both $v_i^0$ and $v_i^1$,for $1 \leq i \leq n$ with $i \neq j_1, j_2, j_3$.

Fig. 4 shows the color classes, the vertices, and the connections corresponding to a clause. Notice that $G_\varphi$ is a complete bipartite graph and therefore biconvex bipartite.

Now we show that the construction is indeed a reduction from the Monotone 1-in-3 SAT problem to the MGR problem. As a main argument, we translate the appearance of $v_i^0$ in a multicolored realization (if it exists) as the assignment $x_i = 0$ and the appearance of $v_i^1$ as the assignment $x_i = 1$, and vice versa.

Let us assume that we have an assignment $T$ such that $T(x_i) = t_i \in \{0, 1\}$, for $1 \leq i \leq n$, in which exactly one variable for each clause in $\Phi$ is assigned to 1. Consider the set

$$S = \{v_i^{t_i} \mid 1 \leq i \leq n\} \cup \{u_j^{t_{j_1} t_{j_2} t_{j_3}} \mid 1 \leq j \leq m\}.$$

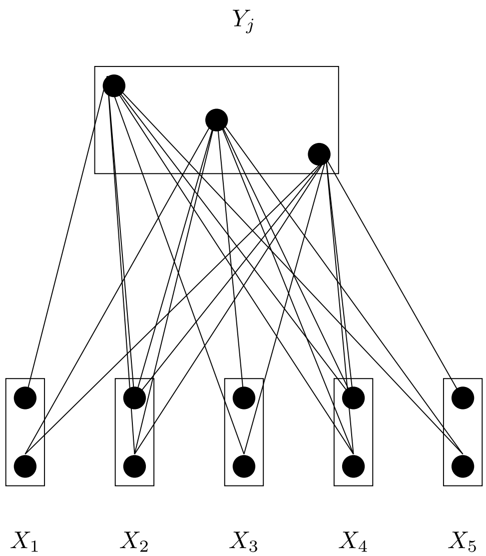



**Fig. 4.** The connections representing clause $C_j = (x_1, x_3, x_5)$, in a formula on 5 variables, in the associated colored graph.

For a clause $C_j = (x_{j_1}, x_{j_2}, x_{j_3})$, by the definition of $G$, the vertex $u_j^{t_{j_1}t_{j_2}t_{j_3}}$ becomes connected with all the vertices $v_i^{t_i}$, for $i \in [j_1, j_3]$. Therefore, $S$ is a multicolored realization of $G_\varphi$.

For the reverse implication, let $X = \cup_{i=1}^n X_i$. Assume that $S$ is a multicolored realization of $G_\varphi$ and that $S \cap X = \{v_1^{t_1}, v_2^{t_2}, \dots, v_n^{t_n}\}$, where $t_i \in \{0, 1\}$, for $1 \le i \le n$. Consider the assignment $T(x_i) = t_i$. By construction, the assignment $T$ is correct, as each variable $x_i$ gets a unique assigned value. Furthermore, observe that, if a clause $C_j = (x_{j_1}, x_{j_2}, x_{j_3})$ contains a variable with assigned value 1, the other variables in $C_j$ are assigned value 0. This is due to the fact that $S$ is a realization of the cluster graph. Without loss of generality, assuming that $t_{j_1} = 1$, $G[S]$ contains $v_{j_1}^1$, then it must contain $v_{j_2}^0$ and $v_{j_3}^0$ as well, as $u_j^{100}$ must belong to $S$; and its neighbors must be contiguous on the interval $[j_1, j_3]$. Thus, for each clause, only one of the three variables has assigned value 1.

Finally, observe that the graph can be constructed in polynomial time from the given formula. □

We can adapt the previous reduction to show that the MGR problem remains hard when the cluster graph is chordal.

**Theorem 3.** *The* MGR *problem is NP-complete, for colored graphs having a chordal cluster graph, even when the cluster size is at most 3.*

**Proof.** We modify slightly the construction in the previous theorem to get a colored graph whose cluster graph is chordal. For doing so, we just add some connections among the lower layers and among the upper layers.

1. For $1 \le i < n$, we connect the vertices in $X_i$ with those in $X_{i+1}$ by a complete bipartite subgraph.
2. For each $1 \le j < k \le m$, we add a complete bipartite graph connecting the vertices in $Y_i$ with the vertices in $Y_k$.

In this way, the cluster graph has a clique connecting the clause vertices, a path connecting the variable vertices, and a complete bipartite graph connecting clauses and variable vertices. Notice that this graph is chordal. As the added connections among clusters were all-to-all, any multicolored subset realizing the cluster graph does it in both graphs. Therefore, the construction is a reduction from the Monotone 1-in-3 SAT problem to the MGR problem on chordal graphs. □

Let $G = (X \cup Y, E)$ be a convex bipartite graph that has the adjacency property with respect to $X$. We define the *spread* of $G$ as the maximum degree of the vertices in $Y$. Notice that Theorem 3 shows the hardness of the MGR problem when the cluster size is bounded but the spread is not necessarily bounded. Our next results give an FPT algorithm solving the MGR problem on colored convex bipartite graphs when parameterized by both the cluster size and the spread.

**Proposition 1.** *The* MGR *problem belongs to FPT, for colored graphs having a convex bipartite cluster graph, when parameterized by the cluster size and the spread.*

**Proof.** Let $(G, \varphi)$ be a colored convex bipartite graph, with cluster size $\ell$ and spread $d$. As there is a linear time algorithm to recognize convex bipartite graphs [23,30], we assume that the color classes are $X_1, \dots, X_\alpha, Y_1, \dots, Y_\beta$ and that $G_\varphi$ is a bipartite cluster graph having the adjacency property with respect to the clusters $X_1, \ldots, X_\alpha$. Therefore, the neighbors of $Y_j$ are consecutive under this ordering. For $1 \le j \le \beta$, let $a_j \le b_j$ be such that $Y_j$ is connected to $X_i$, for $a_j \le i \le b_j$. We devise a dynamic programming algorithm based on the ordering of the $X$ clusters. Let $X = \cup_{i=1}^{\alpha} X_i$ and $Y = \cup_{j=1}^{\beta} Y_j$. For each $i$, $d \le i \le \alpha$, let $P_i = X_{i-d+1} \times \cdots \times X_i$, of tuples formed by $d$ vertices in consecutive layers ending at a vertex in $X_i$. Let $G_i$ be the subgraph induced in $G$ by

$$V_i = (X_1 \cup \cdots \cup X_i) \cup \left(\cup_{j|b_j \le i} Y_j\right).$$

For each $d \le i \le \alpha$, our dynamic programming algorithm keeps a table $M_i$ holding a boolean value for each $p \in P_i$. Thus the table size is $|P_i|$. The entry $M_i(p)$ will be set to 1 whenever there is a multicolored set $S \subseteq V_i$ that is a realization for $G_{i,\varphi}$ such that $S$ contains all the vertices in $p$. Otherwise, the value will be 0.

When $i = d$, for each $p \in P_d$, we have to check whether the set of vertices in $p$ can be extended to a multicolored realization in $G_d$. For this, it is enough to check whether, for each color class $Y_j$ included in $V_d$, there exists a vertex $u_j \in Y_j$ so that it is connected to all the vertices in $p$ belonging to layers in $[a_j, b_j]$. In this case, set $M_d(p) = 1$, and otherwise set $M_d(p) = 0$.

When $d < i \le \alpha$, for $p \in P_i$, we set $M_i(p) = 1$, if

(1) for any $Y_j$ with $b_j = i$, there is a vertex in $Y_j$ connected to all the vertices in $p$ in color classes $X_i$ with $i \in [a_j, b_j]$, and
(2) there exists $p' \in P_{i-1}$, such that the first $d-1$ vertices in $p$ appear as the last vertices in $p'$ and $M_{i-1}(p') = 1$.

Observe that as the spread is $d$ and the considered $Y$ sets are included in $G_i$ but not in $G_{i-1}$, condition (1) guarantees that $p$ can be extended to a multicolored realization with respect to the newly incorporated $Y$ sets. On the other hand, condition (2) guarantees that the vertices in $p$ can be extended to a multicolored realization with respect to $G_{i-1}$, as $p'$ contains all the vertices in $p$ except the one in the last $X$ cluster.

Thus, we can conclude that the proposed algorithm correctly computes $M_i(p)$ for each $i$, $d \le i \le \alpha$, and $p \in P_i$.

Note that $G_\alpha = G$, so if $M_\alpha(p) = 1$, the set of vertices in $p$ can be extended to a multicolored realization for $G$. The last step of our algorithm just checks whether there is a $p \in P_\alpha$ having $M_\alpha(p) = 1$.

For the time complexity, observe that $|P_i| \le \ell^d$. Checking conditions (1) and (2) is the costliest operation. For a given $j$ and $p$, checking condition (1) takes time $O(dn)$. Furthermore, the algorithm performs this checking once for each $j$. So, the overall time is $O(\ell^d dn)$. For given $p \in P_i$ and $p' \in P_{i-1}$, checking condition (2) requires $O(d)$ time. On the other hand, the number of tuples that $p$ can extend is at most $|X_{i-1}|$. This gives an overall time of $O(\ell^2 \ell^d)$. The total cost is $O((nd + \ell^2)\ell^d)$.

Using standard dynamic programming techniques, the algorithm can be adapted to produce a multicolored realization, when one exists, within the same time bounds. □

## 4. Grid cluster graphs

In this section, we consider the MGR problem restricted to colored graphs $(G, \varphi)$ for which the resulting cluster graph $G_\varphi$ is a 2-dimensional grid. Recall that a *two-dimensional grid graph* is a lattice graph, obtained as the Cartesian product of two path graphs, on $n$ and $m$ vertices, respectively. Formally, an $n \times m$ grid graph $L_{n,m}$ has vertex set $\{(i, j) \mid 1 \le i \le n, 1 \le j \le m\}$. Two vertices $(i, j)$ and $(i', j')$ are adjacent if and only if $|i - i'| + |j - j'| = 1$. To show that the problem is hard in this case, we again provide a reduction from the Monotone 1-in-3 SAT problem.

**Theorem 4.** *The* MGR *problem is NP-complete, for colored graphs whose cluster graph is a 2-dimensional grid, even when the cluster size is 6 and the input graph has bounded degree.*

**Proof.** The MGR problem remains in NP when the input graph has a cluster graph which is a 2-dimensional grid. To show that the problem is NP-hard we are going to describe a reduction from the Monotone 1-in-3 SAT problem. Assume that $\Phi$ is a monotone SAT formula on $n$ variables $x_1, \ldots, x_n$ having $m$ clauses $C_1, \ldots, C_m$ each with exactly three variables. For simplicity, as we did before, we assume that clause $j$, $1 \le j \le m$, has the form $C_j = (x_{j_1}, x_{j_2}, x_{j_3})$, with $j_1 < j_2 < j_3$. We construct a colored graph $(G, \varphi)$ in polynomial time from $\Phi$ and will show that $\Phi$ has a valid truth assignment if and only if $G_\varphi$ has a multicolored realization.

Our construction uses several gadgets, each one of which describes a color class of $(G, \varphi)$, see Fig. 5. The graph $G$ will be formed by several copies of those gadgets. We locate them inside a 2-dimensional grid, as shown in Fig. 8. In this way, it will be clear that $G_\varphi$ is indeed a $(n + 2) \times (2m + 1)$ 2-dimensional grid.

*The gadgets.* We use five basic gadgets, each used gadget constitutes a color class in $G$ (see Fig. 5). The first kind, the `Var` gadget, contains two vertices, we refer to them for their positions in the box, left and right. They are used to represent a variable and a selection of one vertex will correspond to an assignment of value to the variable, left vertex with a 1 and the right one with a 0. The second gadget, the `Cl` gadget, contains 3 vertices. We refer to them as the upper, middle and lower vertices. They are used for representing a clause. Those three nodes are used for identifying the three valid assignment values for the variables in the clause, upper with 100, middle with 010, and lower with 001. The third and the four gadgets, the `Var-in-Cl` and the `Var-not-in-Cl` gadgets, contain 3 and 6 vertices, respectively. The first one is used for a variable that appears in a clause and the second when it does not appear. The three node in the `Var-in-Cl`

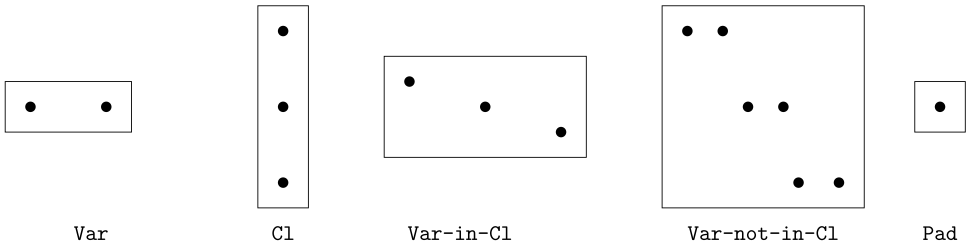



**Fig. 5.** The five basic cluster gadgets.

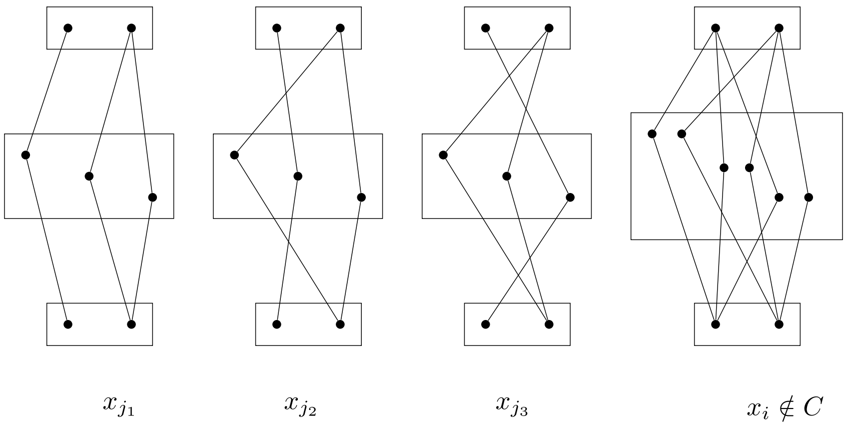



**Fig. 6.** The vertical connections among contiguous `Var`–`Var-in-Cl` and `Var`–`Var-not-in-Cl` gadgets, for a clause $C_j = (x_{j_1}, x_{j_2}, x_{j_3})$.

will be referred as upper, middle and lower. The `Var-not-in-Cl` block has three groups of two vertices (upper, middle and lower groups), inside each group, we use position (left or right) as a reference. Finally, the fifth gadget, the `Pad` gadget contains only one vertex.

*The color classes.* We first describe the color classes of the graph $(G, \varphi)$. Each color class corresponds to one of the basic gadgets. An example of the construction is given in Fig. 8. The cluster graph (described here as a grid of gadgets) has one column for each variable (in the order $x_1 \dots x_n$) and two additional columns, first and last. The upper row starts and ends with a `Pad` gadget and it has one `Var` gadget in each of the columns, i.e., one for each variable.

For each clause, we create two consecutive rows in the grid, following the order of the clauses $C_1, \dots C_m$. The upper row associated with a clause $C_j$, starts and ends with a `Cl` gadget. At column $i$, we place a `Var-in-Cl` gadget, if $x_i \in C_j$, or a `Var-not-in-Cl` gadget, otherwise. The lower row associated with $C_j$ starts and ends with a `Pad` gadget, and contains one `Var` gadget for each variable.

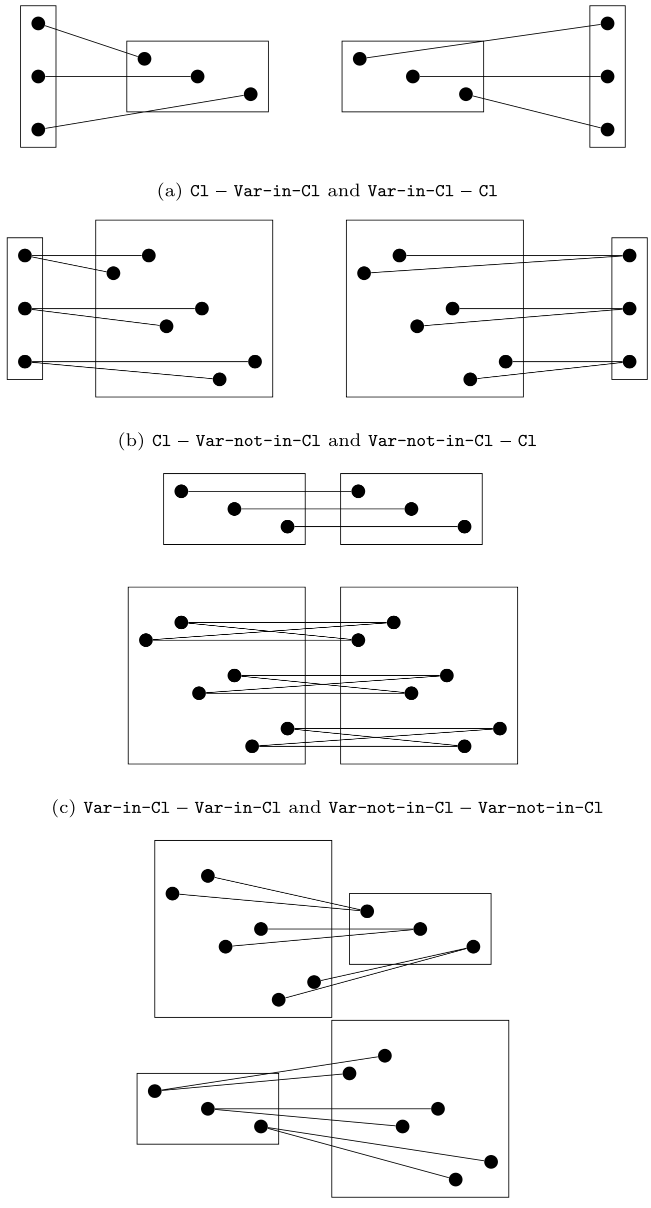
*The connections among vertices.* The connections among vertices in the different color classes depends on the type of gadget and on whether the two color classes are connected in the grid vertically or horizontally. Let us start with the vertical connections. The vertex in a `Pad` gadget is connected to all the vertices in the vertically contiguous `Cl` gadgets. The vertical connections of a `Var-in-Cl` or a `Var-not-in-Cl` gadget and its upper and lower `Var` gadgets are given in Fig. 6.

The horizontal connections are the following: The vertex in a `Pad` gadget is connected to all the vertices in the horizontally contiguous `Var` gadget. The vertices in two horizontally contiguous `Var` gadgets are connected by a complete bipartite graph. The other horizontal connections correspond to contiguous pairs of gadgets from the types `Cl`, `Var-in-Cl` and `Var-not-in-Cl`. The connections among all the possible combinations of such pairs are described in Fig. 7.

Note that the vertical connections described in Fig. 6 guarantee that the left (right) vertex in a `Var` gadget is connected by a path only to a left (right) vertex in another `Var` gadget in the same column. Furthermore, the horizontal connections, as described in 7, always join vertices in the same vertical position (upper, middle or lower).

*Correctness of the reduction.* Observe that the graph together with the coloring can be constructed in polynomial time.

Let us start proving that when $\Phi$ is a yes instance of the Monotone 1-in-3 SAT problem, the constructed colored graph admits a multicolored realization. Let $T$ be a valid assignment to $\Phi$, i.e., letting $T(x_i) = t_i \in \{0, 1\}$, for $1 \le i \le n$, exactly one variable in each clause in $\Phi$ is set to 1.



(a) Cl – Var-in-Cl and Var-in-Cl – Cl

(b) Cl – Var-not-in-Cl and Var-not-in-Cl – Cl

(c) Var-in-Cl – Var-in-Cl and Var-not-in-Cl – Var-not-in-Cl

(d) Var-not-in-Cl – Var-in-Cl and Var-in-Cl – Var-not-in-Cl

**Fig. 7.** The horizontal connections among contiguous gadgets in a clause row.

We define a set $S \subseteq V$ as follows.

- The unique vertex in any Pad gadget belong to $S$.
- For each Var gadget corresponding to variable $x_i$ we add to $S$ the left vertex, if $t_i = 1$, or the right vertex, if $t_i = 0$.
- Consider a clause $C_j = (x_{j_1}, x_{j_2}, x_{j_3})$.
  - From the Cl and the Var-in-Cl gadgets in the column associated with $C_j$, we add to $S$ the upper vertex, if $t_{j_1}t_{j_2}t_{j_3} = 100$, the middle one, if $t_{j_1}t_{j_2}t_{j_3} = 010$, or the lower one, if $t_{j_1}t_{j_2}t_{j_3} = 001$.
  - From a variable $x_i$ that does not appear in $C_j$, we select the upper, middle of lower block, depending on whether $t_{j_1}t_{j_2}t_{j_3}$ is 100, 010 or 001. Inside the selected block, we add to $S$ the left vertex if $x_i = 1$ or the right vertex if $x_i = 0$ to be added to $S$.

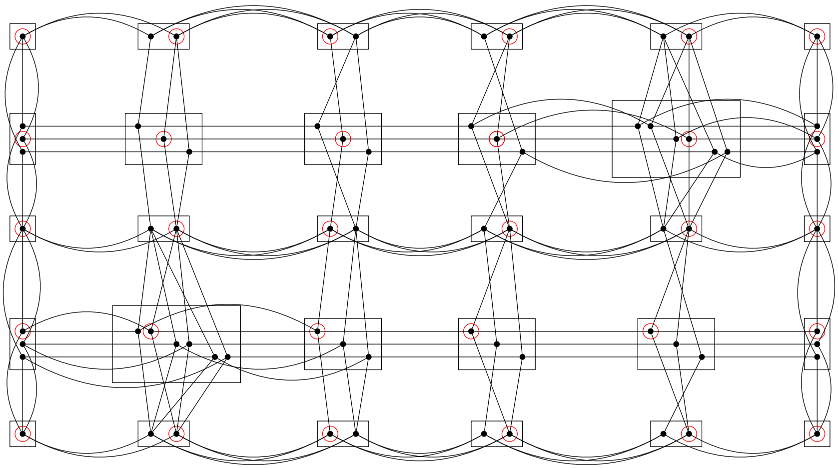


**Fig. 8.** The colored graph obtained from the monotone formula $\Phi = ((x_1, x_2, x_3), (x_2, x_3, x_4))$. The circled vertices form a multicolored realization of the cluster graph.

Note that, $S$ contains exactly one vertex from each color class in $(G, \varphi)$ so, it is multicolored. It remains to show that $S$ is a realization of the cluster graph. Let us first look at the horizontal connections. The circled vertices in the colored graph given in Fig. 8 are the set $S$ associated with the assignment $x_1 = 0$, $x_2 = 1$, $x_3 = 0$ and $x_4 = 0$.

For the rows that alternate `Var` clauses, the connections are all to all; therefore, $S$ realizes all the corresponding connections in the cluster graph. The same happens for the first and the last columns.

Consider a clause $C_j = (x_{j_1}, x_{j_2}, x_{j_3})$ and the corresponding assigned values $t_{j_1}t_{j_2}t_{j_3}$. The vertices in $S$ from the `Cl`, `Var-in-Cl` and `Var-not-in-Cl` in the row are all in the same vertical position (upper, middle, or lower). Therefore, according to the horizontal connections, the horizontal path on this row is realized by $S$ (see Figs. 7 and 8).

Consider a variable $x_i$ with assigned value $t_i$, observe that depending on the value $t_i$, the selected vertices are all on the left ($t_i = 1$) or on the right ($t_i = 0$). Therefore, according to the vertical connections (see Figs. 6 and 8), the vertical path on this column is realized by $S$. We conclude that $S$ is a multicolored realization for $G_\varphi$

To show the opposite direction, we must prove that when $G_\varphi$ has a multicolored realization $S$, $\Phi$ has a truth assignment $T$ in which each clause gets exactly one variable with assigned value 1. We define $T$ as follows: Consider the `Var` gadgets on the top row, we set $T(x_i) = t_i$, being $t_i = 1$ when $S$ contains the left vertex in the gadget and $t_i = 0$ otherwise. As $S$ is multicolored, each $x_i$ is assigned a single truth value. In Fig. 8, the multicolored set defined by the circled vertices translates to the truth assignment $x_1 = 0$, $x_2 = 1$, $x_3 = 0$ and $x_4 = 0$.

Consider a column corresponding to a variable $x_i$ assigned to value $t_i$. When the vertex selected on the top row gadget is the left (respectively right) one, the vertical connections only allow vertical paths that go through left (respectively right) vertices in the gadgets in the column (see Fig. 6). Therefore, for the `Var` gadgets in column $i$, when $t_i = 1$, $S$ contains all the left vertices, and, when $t_i = 0$, $S$ contains all the right vertices.

Consider a clause $C_j = (x_{j_1}, x_{j_2}, x_{j_3})$ and the vertical position (upper, middle or lower) of the vertex in the leftmost `Cl` gadget in the corresponding row. Recall that, according to the definition of the horizontal connections, a horizontal complete path in the cluster graph can only contain vertices in the same vertical position in all the gadgets. Consider the path $p$ induced by the vertices in $S$ from the clusters in the associated row starting from the left. If $p$ starts in the upper vertex, it contains the upper vertices of the `Var-in-Cl` gadget associated to the variables $x_{j_1}, x_{j_2}$ and $x_{j_3}$. The first one is connected only to the left vertex on the vertically contiguous `Var` gadgets, while the other two are connected only to the right vertex on the vertically contiguous `Var` gadgets (see Fig. 6). So, we get that $t_{j_1} = 1$, $t_{j_2} = 0$, and $t_{j_3} = 0$. A similar argument shows that when $p$ starts in the middle (lower) vertex, then $t_{j_1} = 0$, $t_{j_2} = 1$, and $t_{j_3} = 0$ ($t_{j_1} = 0$, $t_{j_2} = 0$, and $t_{j_3} = 1$). Therefore, the constructed formula is a YES instance of the Monotone 1-in-3 SAT problem. □

## 5. Colored graphs with bounded cluster size

We start with presenting a polynomial time algorithm for the particular case of the MGR problem in which the colored graph has cluster size at most 2. Later, we show that the problem becomes NP-complete for graphs with cluster size larger than 2, thus providing a complexity dichotomy with respect to cluster size. In order to get the result, we provide a reduction to the 2-SAT problem: given a boolean formula $\Phi$ in CNF with at most two literals per clause, decide whether $\Phi$ has a satisfying assignment. Recall that the 2-SAT problem can be solved in polynomial time [18].

**Proposition 2.** *The* MGR *problem is polynomial time solvable for colored graphs with cluster size at most 2.*

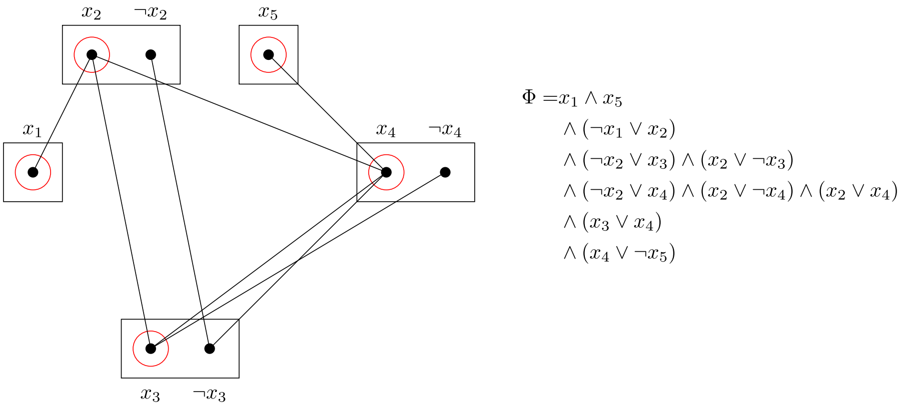



**Fig. 9.** A colored graph with cluster size at most two and the associated 2-SAT formula. The set formed by the circled vertices is a multicolored realization corresponding to the satisfying assignment $x_i = 1$, for $i = 1, \ldots, 5$.

**Proof.** Let $(G, \varphi)$ be a colored graph with cluster size $s(G, \varphi) = 2$. Let $A_1, \ldots, A_k$ be the color classes in $(G, \varphi)$.

From $(G, \varphi)$, we create an instance $\Phi$ of 2-SAT as follows. $\Phi$ has variables $x_1, \ldots x_k$. We associate each vertex in $u \in V$ with a literal $l_u$. If $A_i = \{u\}$, $l_u = x_i$. If $A_i = \{u, v\}$, $l_u = x_i$ and $l_v = \neg x_i$. The clauses in $\Phi$ are the following. For each color class with $|A_i| = 1$, we add clause $x_i$ For each edge $(A_i, A_j) \in E(G_\varphi)$, we add one clause for each missing edge among vertices in the color classes, i.e., for $u \in A_i$, $w \in A_j$ with $(u, w) \notin E(G)$, we add the clause $(\neg l_u \vee \neg l_w)$. Observe that this clause will be satisfied only when at least one of its two literals is assigned the value 0.

An example of the construction is given in Fig. 9. Note that $\Phi$ is a 2-SAT instance, and it can be constructed in polynomial time. Let us show that $(G, \varphi)$ admits a multicolored realization if and only if $\Phi$ is satisfiable.

Assume that $(G, \varphi)$ admits a multicolored realization $S$. Let us consider the assignment $T$ of truth values to the variables of $\Phi$ that makes the literals associated with the vertices in $S$ get value 1, as $S$ is multicolored, $T$ is a valid assignment for $\Phi$. As $S$ is multicolored, it contains all the vertices in color classes with only one vertex. Therefore, $T$ satisfies all the clauses in $\Phi$ with one literal.

Consider a connection $(A_i, A_j) \in E(G_\varphi)$ with some associated clause in $\Phi$. For $u \in A_i$, $w \in A_j$ with $(u, w) \notin E(G)$, at least one of the vertices $u$ or $w$ cannot belong to $S$. Therefore, at least one of $l_u$ or $l_w$ gets value 0 under $t_S$. Thus, the clause $(\neg l_u \vee \neg l_w)$ is satisfied by $T$. We conclude that $T$ satisfies $\Phi$.

For the other direction, assume that $T$ is a satisfying assignment for $\Phi$. Consider the set of vertices $S$ that contains those vertices $u \in V(G)$ such that $T(l_u) = 1$. As $\alpha$ is an assignment, then $S$ is multicolored. Let $S = \{l_1, \ldots l_k\}$. Assume that $S$ is not a multicolored realization of $(G, \varphi)$. In such a case, there must be an edge $(A_i, A_j) \in E(G_\varphi)$ with $(l_i, l_j) \notin E(G)$. Hence, the clause $(\neg l_i \vee \neg l_j)$ will not be satisfied, contradicting the fact that $\alpha$ is a satisfying assignment. Thus we conclude that $S$ is a multicolored realization of $(G, \varphi)$. □

Let us analyze the case of colored graphs with cluster size $s \geq 3$. The reduction provided in the proof of Theorem 3 shows NP-hardness for $s = 3$. To extend the reduction to a value of $s > 3$, we add to the graph constructed in this reduction a large enough set of independent vertices. Those independent vertices are colored in such a way that each color class is completed to have $s$ vertices. After the addition of the independent vertices, the cluster graph remains the same. Furthermore, none of the added vertices can form part of a multicolored realization. Therefore, the problem is NP-complete for $s > 3$. Putting this together with Proposition 2, we get a complexity dichotomy with respect to cluster size.

**Theorem 5.** *The* MGR *problem is NP-complete for colored graphs with cluster size* $s \geq 3$*, and polynomial time solvable otherwise.*

Our last result is an FPT algorithm for the MGR problem parameterized by the treewidth of the cluster graph and the cluster size. Recall that we have already established that the MGR problem parameterized by the treewidth of the cluster graph is $W[1]$-hard (see Theorem 1), and that it is NP-complete, for $s > 2$, when the cluster graph is convex bipartite. Recall that convex bibartite graph can have unbounded treewidth. Our algorithm uses dynamic programming on the tree decomposition.

To describe the algorithm, we assume that as usual, together with the input $(G, \varphi)$, we are given a nice tree decomposition $(T, X, r)$ of the cluster graph $G_\varphi$. To simplify the explanation, we slightly change the notation. For a node $v \in T$, we consider two associated graphs $G_\varphi^u$, the subgraph induced in $G_\varphi$ by the union of all the bags in the subtree rooted at $u$ (a subgraph of $G_\varphi$), and $G_u$, the subgraph induced in $G$ by the union of all the color classes appearing in a bag in the subtree rooted at $u$ (a subgraph of $G$). Observe that by definition $(G_u)_\varphi = G_\varphi^u$.

As each element in a bag corresponds to a color class, we refer directly to the color class. Given a collection of color classes $\mathcal{A} = \{A_1, \ldots, A_k\}$, and a set $S \subseteq V$, we say that $S$ is *multicolored with respect to* $\mathcal{A}$ when $S$ contains exactly one vertex from each color class in $\mathcal{A}$.

**Theorem 6.** *The* MGR *problem when parameterized by the treewidth of the cluster graph and the cluster size belongs to FPT.*

**Proof.** Let $(G, \varphi)$ be a colored graph with cluster size $s$, and let $(T, X, r)$ be a nice tree decomposition of the cluster graph $G_\varphi$ with width $w$. The dynamic programming algorithm will fill, for each node $v \in V(T)$, a boolean table $M_v(S)$ having an entry for each multicolored subset $S$ with respect to the color classes included in $X_v$. At the end of the algorithm, $M_v(S) = 1$ if there is a multicolored set $S'$ in $G_v$ realizing $G_\varphi^v$ such that $S \subseteq S'$, and otherwise $M_v(S) = 0$. Therefore, the value of $M_r(\emptyset)$ will determine whether there is a multicolored realization of $G$ or not. We deal with the table computation for each type of node in the nice tree decomposition separately, as each type of node requires a different kind of recursion and a different correctness guarantee.

*Start node.* Let $u$ be start node of $T$, that is a leaf, with $X_u = \{A\}$ for some color class $A$ in $G_\varphi$. The multicolored subsets are formed by just one vertex in $A$. We set $M_u(\{x\}) = 1$, for $x \in A$, and $M_u(\emptyset) = 0$. As the graph $G_\varphi^u$ is an isolated vertex, the computed values are correct.

*Introduce node.* Let $u$ be an *introduce* node of $T$ and let $v$ be its unique child. Recall by definition $|X_u - X_v| = 1$. Assume that $A$ is the unique color class in $X_u - X_v$.

Then, for each $x \in A$ and each multicolored set $S$ with respect to $X_v$, if $G[\{x\} \cup S]$ is a realization of $G_\varphi^u[X_v]$ and $M_v(S) = 1$, we set $M_v(\{x\} \cup S) = 1$, otherwise we set the value to 0.

Note that all the multicolored sets with respect to $X_u$ are formed by a vertex in $A$ and a multicolored set $S$ with respect to $X_v$. Furthermore, $G_\varphi^v$ does not include the vertex $x$. For a multicolored set $S'$ of $G_\varphi^u$, let $x \in A \cup S'$ and let $S$ be formed by the vertices in $S'$ belonging to the color classes in $X_v$. Then $G[S']$ is a realization of $G_\varphi^u$ if and only if $G[\{x\} \cup S]$ is a realization of $G_\varphi[X_u]$ and $M_v(S) = 1$.

*Forget node.* Let $u$ be a *forget* node of $T$, let $v$ be its unique child and assume that, according to the definition, $A$ is the unique color class in $X_v - X_u$. Then, for each $x \in A$ and each multicolored subsets $S$ with respect to $X_v$, we define $M_u(S) = \wedge_{x \in A} M_v(S \cup \{x\})$. This expression provides the correct value, as we are considering all the possible multicolored supersets of $S$ with respect to $X_v$, if one of them is extendable to a multicolored realization, then $S$ is also extendable.

*Join node.* Let $u$ be a join node of $T$ with children $v$ and $w$. In this case, we have that $X_u = X_v = X_w$

Then, for each multicolored subsets $S$ with respect to $X_u$, we set $M_u(S) = M_u(S) \wedge M_w(S)$. This formula provides the correct value, as for the set $S$ to be extendable to a multicolored realization in $G_u$, $S$ must be extendable to a multicolored realization in both $G_v$ and $G_w$.

*Complexity.* The size of the tables associated with a node is upper-bounded by $s^w$, as we have to select on vertex from each color class with at most $s$ vertices and the number of classes in a bag is at most $w$. To compute the entries, the most complex operation is a forget node in which we have to look at all the elements in a color class. This number is upper-bounded by $n$. As the total number of nodes is polynomial in the number of color classes, the total cost is $O(s^w p(n))$. This function shows that the problem is fixed parameter tractable. □

## 6. Conclusions and further results

We have introduced a new generalized graph problem in order to assess the viability of the associated cluster graph in terms of a possible existing realization. We have studied the complexity of the problem in the parameterized framework. Our results shed light on the hardness of the problem with respect to several parameters.

We can consider also a variant of the MGR problem in which, instead of asking for a multicolored realization of the cluster graph, we are interested in a multicolored realization of a given spanning subgraph:

Multicolored subgraph realization problem (MsGR)

Instance: An undirected graph $G = (V, E)$ and a coloring $\varphi$ of $G$.

Question: Is there a multicolored realization of $H = (V(G_\varphi), E')$, for $E' \subseteq E(G_\varphi)$?

This version of the problem captures another well known problem: the multicolored independent set problem which is also known to be $W[1]$ hard parameterized by the number of colors (see for example [2]). Observe that the MsGR problem includes, as a particular case, the MGR problem. Therefore, all the hardness results provided in this paper hold for the MsGR problem.

Note that in the MGR problem, when an edge is not present in $G_\varphi$, none of the vertices in the corresponding color classes are connected. This is not always the case in the MsGR problem, an edge that is not present in the target graph $H$ might appear in $G_\varphi$. When $(A, B) \in E(G_\varphi)$ but $(A, B) \notin E(H)$, a realization of $H$ must select two not connected vertices, one from $A$ and another from $B$. So, a necessary condition for the existence of a multicolored realization of $H$ is that the bipartite graph $G[A,B]$ connecting the vertices in $A$ with the vertices in $B$ is not a complete bipartite graph. If $G[A,B] \equiv K_{|A|,|B|}$ and $H$ does not contain the edge $(A,B)$, we know that no multicolored realization exists. Otherwise, we can assume that, for each $(A,B) \in E(G_\varphi)$ with $(A,B) \notin E(H)$, we have $E(A,B) = \{(u,v) \in V(G) \mid u \in A, v \in B\} \subsetneq A \times B$. Under this assumption, we can consider the graph $G'$ in which, for each $(A,B) \in E(G_\varphi)$ with $(A,B) \notin E(H)$, we remove from $E(G)$ the edges in $E(A,B)$ and add the edges $A \times B \setminus E(A,B)$. Maintaining the same coloring, we have that $G'_\varphi = G_\varphi$ and that $H$ has a multicolored realization if and only if there is a multicolored realization of $G'_\varphi$. Furthermore, $G'_\varphi$ can be constructed in polynomial time in the size of $G$. In this way we obtain a polynomial time reduction from the MsGR problem to the MGR that preserves all the parameters considered in this paper. Consequently, all the positive results (polynomial time or FPT algorithms) devised for the MGR problem also hold for the MsGR problem.

Recall that a homomorphism from a graph $G = (V,E)$ to a graph $H = (V',F)$ is a function $f$ from $V$ to $V'$ such that, for each $(x,y) \in E$, $(f(x), f(y)) \in F$. If $S \subseteq V$ and $f$ is a homomorphism from $G$ to $G[S]$, $f$ is a *retraction with respect to S* if, for $x \in S$, $f(x) = x$ [28]. Inspired by this notion, we consider the following problem.

Composed retraction problem (CR)

Instance: Undirected graph $G = (V,E)$, a subgraph $H = (V',F)$ of $G$ together with a homomorphism $f$ from $G$ to $H$.

Question: Is there a homomorphism $g$ from $H$ to $G$ such that, for $x \in V'$, $f(g(x)) = x$?

Note that, for colored graphs $(G,\varphi)$ in which the coloring is proper, $\varphi$ is a homomorphism from $G$ to $G_\varphi$. Furthermore, if $S$ is a multicolored realization of $G_\varphi$, then the function $g$ assigning to each vertex in $G_\varphi$ the corresponding colored vertex in $S$ is a homomorphism from $G_\varphi$ to $G$ that verifies $\varphi(g(x)) = x$. On the other hand, if there is a homomorphism $g$ from $G_\varphi$ to $G$ such that, for $x \in V(G_\varphi)$, $\varphi(g(x)) = x$, then $g(V(G_\varphi))$ is a multicolored realization of $G_\varphi$. Therefore, the MGR problem, when the coloring is proper, is a subproblem of the CR problem. Taking into account that the coloring used in the reductions in this paper are proper, all the hardness results provided in this paper hold for the CR problem. It remains open to find other parameterizations under which the CR problem becomes tractable.

## Acknowledgments

We thank the anonymous referees for their careful reading and helpful suggestions. J. Díaz and M. Serna are partially supported by funds from the Spanish Agencia Estatal de Investigación under grant PID2020-112581GB-C21 (MOTION), and from AGAUR under grant 2017-SGR-786 (ALBCOM). Ö. Y. Diner is partially supported by the Scientific and Technological Research Council Tübitak under project BIDEB 2219-1059B191802095 and by Kadir Has University under project 2018-BAP-08. O. Serra is supported by the Spanish Agencia Estatal de Investigación under grant PID2020-113082GB-I00.

## References

[1] O. Cosma, P.C. Pop, I. Zelina, An effective genetic algorithm for solving the clustered shortest-path tree problem, IEEE Access 9 (2021) 15570–15591, http://dx.doi.org/10.1109/ACCESS.2021.3053295.
[2] M. Cygan, F.V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, first ed., Springer Publishing Company, Incorporated, 2015.
[3] M. Demange, T. Ekim, B. Ries, C. Tanasescu, On some applications of the selective graph coloring problem, European J. Oper. Res. 240 (2015) 307–314.
[4] M. Demange, J. Monnot, P. Pop, B. Ries, On the complexity of the selective graph coloring problem in some special classes of graphs, Theor. Comput. Sci. 540–541 (2014) 82–102.
[5] M. D'Emidio, L. Forlizzi, D. Frigioni, S. Leucci, G. Proietti, Hardness, approximability, and fixed-parameter tractability of the clustered shortest-path tree problem, J. Comb. Optim. 38 (1) (2019) 165–184, http://dx.doi.org/10.1007/s10878-018-00374-x.
[6] J. Díaz, O.Y. Diner, M. Serna, O. Serra, On list $k$-coloring convex bipartite graphs, in: C. Gentile, G. Stecca, P. Ventura (Eds.), Graphs and Combinatorial Optimization: From Theory to Applications CTW2020 Proceedings, in: Airo, vol. 5, Sringer, 2021, pp. 15–26.
[7] R. Diestel, Graph Theory, in: Graduate Texts in Mathematics, vol. 173, Springer-Verlag, Heidelberg, 2017.
[8] R.G. Downey, M.R. Fellows, Fixed-parameter tractability and completeness. I. Basic results, SIAM J. Comput. 24 (4) (1995) 873–921.
[9] R.G. Downey, M.R. Fellows, Fixed-parameter tractability and completeness II: On completeness for $W[1]$, Theoret. Comput. Sci. 141 (1–2) (1995) 109–131.
[10] M. Dror, H. Haouari, J. Chaouachi, Generalized spanning trees, European J. Oper. Res. 120 (2000) 583–592.
[11] J. Enright, T. Stewart, G. Tardos, On list coloring and list homomorphism of permutation and interval graphs, SIAM J. Discrete Math. 28 (4) (2014) 1675–1685.
[12] M.R. Fellows, D. Hermelin, F. Rosamond, S. Vialette, On the parameterized complexity of multiple-interval graph problems, Theoret. Comput. Sci. 410 (1) (2009) 53–61, http://dx.doi.org/10.1016/j.tcs.2008.09.065.
[13] C. Feremans, M. Labbe, G. Laporte, Generalized network design problems, European J. Oper. Res. 148 (1) (2003) 1–13.
[14] M. Fischetti, J. González, P. Toth, The Generalized Traveling Salesman and Orienteering Problems, Kluwer, Dordrecht, 2002.
[15] M. Fischetti, J.J. Salaza González, P. Toth, The symmetric generalized traveling salesman polytope, Networks 26 (2) (1995) 113–123.
[16] M. Fischetti, J.J. Salaza González, P. Toth, A branch-and-cut algorithm for the symmetric generalized traveling salesman problem, Oper. Res. 45 (3) (1997) 378–394, http://dx.doi.org/10.1287/opre.45.3.378.
[17] J. Flum, M. Grohe, Parameterized Complexity Theory, in: Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag Berlin Heidelberg, 2006.
[18] M.R. Garey, D.S. Johnson, Computers and Intractability. A Guide to the Theory of NP-Completeness, Freeman and Company, 1979.
[19] F. Gavril, Algorithms for minimum coloring, maximum clique, minimum covering by cliques and maximum independent set of a chordal graph, SIAM J. Comput. 1 (1972) 180–187.
[20] G. Ghiani, G. Improta, An efficient transformation of the generalized vehicle routing problem, European J. Oper. Res. 122 (1) (2000) 11–17.

[21] S. Huang, M. Johnson, D. Paulusma, Narrowing the complexity gap for coloring $(c_s, p_t)$-free graphs, Comput. J. (11) (2015) 3074–3088.
[22] Y. Myung, C. Lee, D. Tcha, On the generalized minimum spanning tree problem, Networks 26 (4) (1995) 231–241, http://dx.doi.org/10.1002/net.3230260407.
[23] D. Nussbaum, S. Pu, J. Sack, T. Uno, H. Zarrabi-Zadeh, Finding maximum edge bicliques in convex bipartite graphs, Algorithmica 64 (2) (2010) 140–149.
[24] K. Pietrzak, On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems, J. Comput. System Sci. 67 (4) (2003) 757–771, http://dx.doi.org/10.1016/S0022-0000(03)00078-3.
[25] P. Pop, The generalized minimum spanning tree problem: An overview of formulations, solution procedures and latest advances, European J. Oper. Res. (283) 1–15.
[26] P. Pop, Generalized Network Design Problems: Modeling and Optimization, De Gruyter, Germany, 2012, http://dx.doi.org/10.1515/9783110267686.
[27] P. Pop, O. Matei, C. Sabo, A. Petrovan, A two-level solution approach for solving the generalized minimum spanning tree problem, European J. Oper. Res. 265 (2) (2018) 478–487.
[28] A. Quilliot, A retraction problem in graph theory, Discrete Math. 54 (1985) 61–72.
[29] T. Schaefer, The complexity of satisfiability problems, in: Proceedings of the 10th Annual ACM Symposium on Theory of Computing, 1978, pp. 216–226.
[30] J.P. Spinrad, A. Brandstädt, L. Stewart, Bipartite permutation graphs, Discrete Appl. Math. 18 (1987) 279–292.
[31] L. Vepštas, Graph quotiens: a topological approach to graphs, Bull. Novosib. Comput. Center Comput. Sci. 41 (2017) 55–89.