Graph Subcolorings: Complexity and Algorithms*

Jiří Fiala[1,2], Klaus Jansen[1], Van Bang Le[3]†, and Eike Seidel[1]

September 14, 2001

[1] Institut für Informatik und Praktische Mathematik,
Christian-Albrechts-Universität zu Kiel, 24098 Kiel, Germany.
{jfi,kj,ese}@informatik.uni-kiel.de

[2] Institute of Theoretical Computer Science,
Charles University, Prague, Czech Republic.[1]

[3] Institut für Theoretische Informatik, Fachbereich Informatik,
Universität Rostock, 18051 Rostock, Germany.
le@informatik.uni-rostock.de

Abstract

In a graph coloring, each color class induces a disjoint union of isolated vertices. A graph subcoloring generalizes this concept, since here each color class induces a disjoint union of complete graphs. Erdős and independently Albertson et al. proved that every graph of maximum degree at most 3 has a 2-subcoloring. We point out in this paper that this fact is best possible with respect to degree-constraints by showing that the problem of recognizing 2-subcolorable graphs with maximum degree 4 is *NP*-complete, even when restricted to triangle-free planar graphs. Moreover, in general, for fixed k, recognizing k-subcolorable graphs is *NP*-complete on graphs with maximum degree at most k^2. In contrast, we show that, for arbitrary k, k-SUBCOLORABILITY can be computed efficiently on graphs of bounded treewidth, on cographs and on graphs with bounded cliquewidth.

*Research supported in part by EU ARACNE project HPRN-CT-1999-00112 and EU APPOL project IST-1999-14084. An extended abstract of this paper was presented at the 27th Int. Workshop on Graph-Theoretic Concepts in Computer Science (WG 2001)

†Financial support by Deutsche Forschungsgemeinschaft is gratefully acknowledged.

[1]Supported by the Ministry of Education of the Czech Republic as project LN00A056

1 Introduction and results

A *k-coloring* of a graph G is a partition of the vertices into k pairwise disjoint sets $V_1, \ldots, V_k$ such that for every $i = 1, 2, \ldots, k$, each color class V_i consists of isolated vertices, i.e. it forms a stable set. The smallest k for which the graph G has a k-coloring is called the *chromatic number* of G, denoted by $\chi(G)$. Graph coloring is a well studied topic, both for its theoretic and algorithmic aspects. It is well-known that testing 3-COLORABILITY is *NP*-complete for triangle-free graphs with maximum degree 4 ([22]) and for planar graphs with maximum degree 4 ([18]). Testing 3-COLORABILITY is easy for graphs with maximum degree 3 (by Brooks theorem) and for triangle-free planar graphs (by Grötzsch theorem). 2-colorable graphs can be recognized in linear time.

Graph coloring has been generalized in several ways and by a number of authors; see [2] for a comprehensive survey. In this paper we address one of these generalized colorings. A partition $V_1, \ldots, V_k$ of vertex set of a graph G is called a *k-subcoloring* of G if each color class V_i induces in G a disjoint union of complete subgraphs (of various sizes). The *subchromatic number* $\chi_s(G)$ of G is the smallest integer k for which G has a k-subcoloring. Subcolorings have been discussed in [2, 7, 8, 23]. It turns out that subcolorings have many interesting properties similar to colorings, and every k-coloring is also a k-subcoloring, hence $\chi(G) \geq \chi_s(G)$. Among other results we would like to mention the following properties of graph subcolorings:

(1) For every $k \geq 1$, there is a triangle-free graph G_k with $\chi_s(G_k) = k$ [2, 23].

(2) For every planar graph G, $\chi_s(G) \leq 4$. In addition, if G is outerplanar, $\chi_s(G) \leq 3$. These bounds are tight ([7]).

(3) For every graph G with maximum degree Δ, $\chi_s(G) \leq \lfloor \frac{\Delta}{2} \rfloor + 1$ ([2]).

By (3), every graph with maximum degree at most 3 is 2-subcolorable. Actually, this fact follows also from a theorem due to Erdős [15] saying that every graph G has a bipartite spanning subgraph H such that the degree in H of every vertex is at least one half of its degree in G. Thus, if the maximum degree in G is at most 3, every bipartition of H defines a sub-bipartition of G. Moreover, such a bipartite spanning subgraph H of G can be found in polynomial time, easily by a "local improvement" technique.

Albertson et al. [2] then pointed out the difficulties involved in characterizing 2-subcolorable graphs by giving a number of examples. In this paper we prove the following theorems (Sect. 2.1 and 2.2):

Theorem 1 *Recognizing* 2*-subcolorable graphs of maximum degree* 4 *is NP-complete, even on triangle-free planar graphs.*

Albertson informed us that, independently, Gimbel also proved Theorem 1 with a completely different reduction.

Formally we define k-SUBCOLORABILITY as a decision problem whose input is a graph G and the question is: Is $\chi_s(G) \leq k$? Notice that the *NP*-completeness of k-SUBCOLORABILITY for the class of *all* graphs follows from a theorem by Achlioptas [1]. We prove in Sect. 2.4 that:

Theorem 2 *For every fixed* $k \geq 2$*,* k-SUBCOLORABILITY *is NP-complete for graphs of maximum degree at most* k^2*.*

For constant k, k-SUBCOLORABILITY can be expressed as a monadic second order logic formula, and hence can be tested in linear time for graphs with bounded treewidth. Due to the fact that for these graphs $\chi_s(G) \leq c$ for a constant c, we get that there exists an algorithm that in linear time determine $\chi_s(G)$ for graphs with bounded treewidth. On the other side, the general algorithm is unnecessarily complicated for our purpose and in Sect. 3.1 we present a simpler algorithm solving this problem.

On the other hand $\chi_s(G)$ can be arbitrary large for graphs with constant cliquewidth (or cographs that have cliquewidth ≤ 2).

Theorem 3 *For every* k*,* k-SUBCOLORABILITY *can be efficiently decided on graphs of bounded treewidth, on cographs and on graphs with bounded cliquewidth (given a construction tree).*

Note that our result on graphs with bounded cliquewidth is new because, for arbitrary k, k-SUBCOLORABILITY cannot be expressed in the so-called monadic second order logic.

2 The *NP*-completeness of the subcoloring problem

We prove Theorem 1 by reducing the Not-all-equal 3-satisfiability problem, which was proven *NP*-complete by Schaefer [25] (see also [17, Problem LO3]).
Problem: *Let* $\mathcal{C}$ *be a Boolean formula consisting of of* m *clauses such that every clause has exactly three distinct literals. The the decision problem whether there exists a satisfying assignment for* $\mathcal{C}$ *such that each clause in* $\mathcal{C}$ *has at least one false (and at least one true) literal is NP-complete.*

We denote the class of all formulas that allow such not all equal assignment by *NAE3SAT*.

2.1 Triangle-free graphs of maximum degree 4

We first prove in this section the *NP*-completeness of 2-SUBCOLORABILITY for non-planar triangle-free graphs of maximum degree 4 (the problem is clearly in *NP*). Let $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ be a Boolean formula consisting of of m clauses over variables $x_1, x_2, \ldots, x_n$ such that every clause C_j of $\mathcal{C}$ contains exactly three literals, $C_j = (l'_j \vee l''_j \vee l'''_j)$. We will construct a triangle-free graph $G = G(\mathcal{C})$ of maximum degree 4 such that G has a 2-subcoloring if and only if $\mathcal{C} \in NAE3SAT$.

Before we describe the construction of G, observe to connector graphs depicted in Fig. 1.

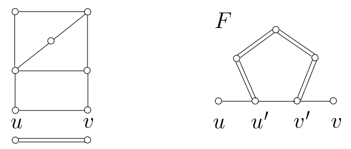


Figure 1: Two connector graphs

The first graph has the property, that under any 2-subcoloring, vertices u and v have distinct colors. Its symbolic representation is shown below, and we will use this simplified drawing when a larger graph contains this connector as a subgraph. Such example is depicted in the connector graph F (in the right part of Fig. 1). Observe that under any 2-subcoloring of F the pair u, v is always colored by the same color, distinct from the color used on the pair u', v'.

The graph G consists of three parts: clause gadgets, variable gadgets and connectors. The clause gadget is very simple: For each clause C_j, we insert into G a unique path of length two P_3 , with vertices labeled l'_j, l''_j and l'''_j. Observe that every clause gadget allows all possible 2-subcolorings such that both colors are used and (by the definition of 2-subcoloring) it is impossible to color the P_3 by one color.

For each variable x_i we insert in G a copy of graph H_i depicted in Figure 2.

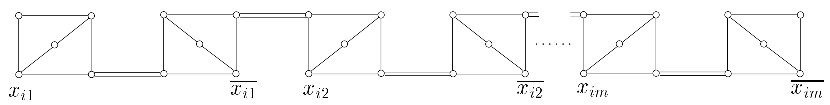


Figure 2: The variable gadget H_i

Lemma 4 *The graph H_i is 2-subcolorable. Any its 2-subcoloring contains vertices $x_{i1}, x_{i2}, \ldots, x_{im}$ in the same color class and vertices $\overline{x_{i1}}, \overline{x_{i2}}, \ldots, \overline{x_{im}}$ are colored by the other color.*

We complete the construction of the graph G by connecting clause and variable gadgets by inserting a copy of the connector graph F for each literal l_j^α of C, where the vertex u of F is merged with the corresponding vertex l_j^α in a clause gadget, and the vertex v with the vertex x_{ij} if the literal l_j^α equals to x_i, or with the vertex $\overline{x_{ij}}$ if $l_j^\alpha = \neg x_i$. (We make the construction for all possible $\alpha = \prime, \prime\prime, \prime\prime\prime$.)

Observe that all graphs involved in the construction of the graph G are triangle free and of maximum degree 4, and even the final composition does not violate this property.

We now are going to show that $\mathcal{C} \in \mathit{NAE3SAT}$ if and only if G has a 2-subcoloring.

Let ϕ be a truth assignment for $\mathcal{C}$ in which every clause has at least one true and at least one false valued literal. We define a 2-subcoloring of G as follows: For every variable x_i, color all x_{ij} red if and only if $\phi(x_i) = \mathsf{true}$, and use blue color otherwise. Then extend this subcoloring to a unique 2-subcoloring of H_i. This is possible as we have seen by Lemma 4. Next extend this 2-subcolorings for all connectors F. Since ϕ was a feasible *NAE3SAT* assignment, every clause gadget (path of length 3) is also properly 2-subcolored. Observe also, that no conflict happened due to vertex merging in the construction of G, since in every F vertices u and u' and also v and v' have different colors.

In the opposite direction, assume any 2-subcoloring of G in red and blue. We define the assignment ϕ for $\mathcal{C}$ as follows: $\phi(x_i) = \mathsf{true}$ if x_{ij} is red for some j; otherwise $\phi(x_i) = \mathsf{false}$. By Lemma 4, this assignment is well-defined. Due to the properties of connectors F it holds that in every clause gadget two of l_j', l_j'', l_j'' have different colors. Therefore, each clause C_j has at least one true and at least on false literal by the truth assignment ϕ.

2.2 Planar graphs of maximum degree 4

In this section, we construct a triangle-free planar graph G' from the graph G obtained in the previous section, such that G is 2-subcolorable if and only if G' is 2-subcolorable.

Note, that G can be embedded in the plane, in polynomial time, such that every edge is a straight line and all edge crossing occurs only on (v, v')

edges of the connector graphs F, and at each crossing point meets exactly two edges.

This makes the use of the "crossover" technique in proving NP-completeness of Planar Graph 3-Colorability, described among others in [17], possible.

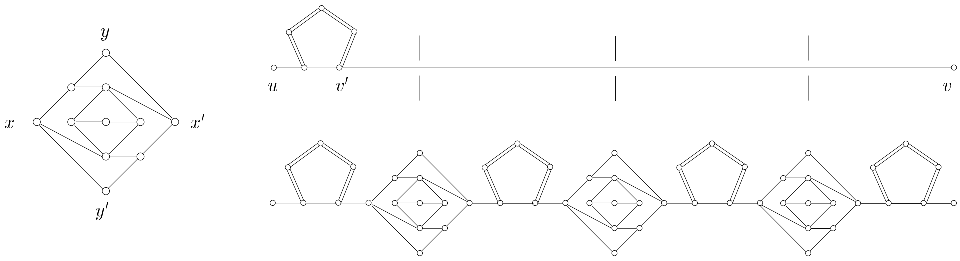


Figure 3: The "crossover" graph and the crossing replacement

The "crossover" in our construction is the graph depicted in Figure 3 on the left side and has the following properties:

- In any 2-subcoloring, vertices x and x' belong to the same color class, and also vertices y and y' have the same color (not necessarily the same as x, x').
- There exists a 2-subcoloring such that x and y belong to the same color class, and also another 2-subcoloring such that x and y are colored by different colors.

The construction of the planar graph G' from G is very similar to the construction for Planar Graph 3-Colorability. We replace each crossing point by the "crossover" graph and join these crossovers by connectors F (see Fig. 3, right).

Observe, that the graph G' has a 2-subcoloring if and only if G does.

Assume G has a 2-subcoloring. Such a subcoloring of G can be extended to a 2-subcoloring for G' as follows: For every edge (v', v) in G use the color of v on vertices u and v of all connectors F added during removed crossovers. Such coloring can be extended to the coloring of G'.

In the opposite direction, assume that G' has a 2-subcoloring. Then, due to properties of connector F (u and v are colored the same) and "crossover" graph (opposite vertices maintain the same color), we get that the color restriction on the original vertices is a proper 2-subcoloring of G.

2.3 3-Subcolorability of planar graphs

The 3-SUBCOLORABILITY of planar graphs is also *NP*-complete. We show a simple reduction from PLANAR GRAPH 3-COLORABILITY. Assume that G is a planar graph whose proper 3-coloring is questioned. We replace each edge (u, v) by a graph depicted in Fig. 4 composed of four copies of path P_{11} and six additional vertices. (see [7]).

By a case study, it is easy to check, that under any 3-subcoloring of this replacement graph, vertices u, u' and u'' have the same color, distinct from the color used on v'', v' and v. And the result for the reduction is straightforward.

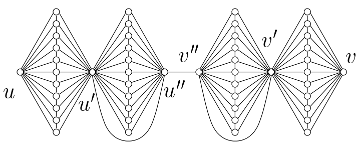


Figure 4: The edge replacement graph

Corollary 5 *The planar* 3-SUBCOLORABILITY *is NP-complete on planar graphs.*

Recall that every planar graph is 4-subcolorable and that every outerplanar graph is 3-subcolorable.

2.4 The hardness of k-subcoloring for $k \geq 3$

In this section we show that, for each fixed $k \geq 2$ the k-SUBCOLORABILITY is a *NP*-complete problem on graphs with maximum degree at most k^2.

Lemma 6 *If φ is a k-subcoloring of a graph G and H is an induced subgraph of G then the restriction of φ on H is a k-subcoloring of H.*

Note that Lemma 6 does not hold for subgraphs in general.

Lemma 7 *For every $k \geq 2$, the complete k-partite graph $K_{k,\ldots,k,k+1}$ consisting of $k - 1$ (small) partitions with k vertices and one (big) partition of $k + 1$ vertices has exactly one (up to permutation) k-subcoloring; this k-subcoloring is also its unique k-coloring. The graph $K_{k,\ldots,k,k+1}$ cannot be subcolored with less than k colors.*

The reduction from $(k-1)$-SUBCOLORABILITY to k-SUBCOLORABILITY goes as follows: Let G be the graph for which the existence of a k-subcoloring is questioned and let $V(G) = \{v_1, \ldots, v_n\}$. Then the graph G', which is assumed as an instance for k-SUBCOLORABILITY is constructed by:

- Take n copies $H_1, \ldots, H_n$ of the $K_{k,\ldots,k,k+1}$;
- in each H_i, label four distinct vertices of the big class with $x_1^i, x_2^i, x_3^i, x_4^i$, and label one vertex in each of the $k-1$ small classes with y_j^i, $1 \le j \le k-1$;
- add edges $(x_1^i, y_j^{i-1}), (x_2^i, y_j^{i-1})$ for all $1 \le j \le k-1, 2 \le i \le n$;
- add edges $(x_3^i, v_i), (x_4^i, v_i)$ for all $1 \le i \le n$.

We claim that G is $k-1$-subcolorable if and only if G' is k-subcolorable:

(1) Assume G can be subcolored with $k-1$ colors. Then subcolor, in each H_i, the $k-1$ small classes with these $k-1$ colors (each class gets one color), and take one new color for the big class. This is a k-subcoloring for G'.

(2) Consider a k-subcoloring of G'. Then, by Lemma 6, the restriction of this subcoloring on each H_i is a k-subcoloring. By Lemma 7, each H_i gets all k colors and each class in H_i is monochromatic. Moreover, the big classes of all H_is have the same color.

We show that, for $1 \le i < n$, x_1^i and x_1^{i+1} have the same color. Assume not, then the color of x_1^{i+1} must occur in a small class of H_i, say y_1^i has this color. But then $x_1^{i+1}, x_2^{i+1}, y_1^i$ induce a monochromatic path P_3 in G', contradicting the definition of k-subcoloring.

No vertex in G can have the color occurring in the big classes of the H_is. Therefore, the restriction of the k-subcoloring of G' on G is a $k-1$-subcoloring of G.

Lemma 8 *For $k \ge 3$, $\Delta(G') = \max\{\Delta(G) + 2, k^2\}$:*

Proof: Observe that $\Delta(G') = \max\{\Delta(G) + 2, d_{G'}(x_1^2), d_{G'}(y_1^2)\}$. By the construction, $d_{G'}(x_1^2) = (k-1)k + (k-1) = k^2 - 1$ and also $d_{G'}(y_1^2) = (k-2)k + (k+1) + 2 = k^2 - k + 3$, hence, for $k \ge 3$, the Lemma follows. □

Proof of Theorem 2 The case $k = 2$ is proven by Theorem 1. The statement for $k \ge 3$, follows from the construction and Lemma 8 and by noting that if G has maximum degree at most $(k-1)^2$ then the graph G' constructed from G as above has maximum degree at most k^2.

3 Polynomially solvable cases — algorithms

In this section we show, that the k-SUBCOLORABILITY problem allows an polynomial time algorithm on restricted classes of input graphs, in particular on graphs with bounded treewidth, cographs and graphs with bounded cliquewidth.

All these classes uses an underlying tree structure for a given graph and with use of dynamic programming perform the feasibility test for all vertices of the tree. Here we would like to introduce notions that will be common for all three forthcoming subsections.

The tree is denoted by T and its *nodes* (to distinguish them from *vertices* of G) are denoted by $X_1, \ldots, X_m$. The tree is rooted, hence the parent-child relation $\succ$ is well defined. Moreover each node has at most two descendants and the size of T is always polynomial in the size of G. Each node X_i is of certain type (this type is sometimes specified by its *label*) and corresponds to a subgraph of G, denoted by G_i.

For each node X_i we build a table $\mathtt{Tab}_i$ of constant or polynomial size, whose entries describe necessary properties of a feasible k-subcoloring ϕ_i of the graph G_i. The situation when the table $\mathtt{Tab}_m$ is nonempty for the root node X_m corresponds to the fact that a proper k-subcoloring of the entire graph G exists.

3.1 Graphs with bounded treewidth

The notion of treewidth was introduced by Robertson and Seymour in [24] via tree decompositions.

Let $G = (V, E)$ be a graph. The *tree decomposition* of G is a tree T, whose nodes X_i are subsets of V and the following is satisfied

1. For each edge $(u, v) \in E$ exists $X_i \in V(T)$ such that $u, v \in X_i$.
2. For any $v \in V(G)$ the sets X_i containing v induce a nonempty connected subtree of T.

The width of a tree decomposition T is $\max_{X_i \in V(T)}\{|X_i|\} - 1$ and the treewidth of G is the minimum width among all possible tree decompositions. We denote treewidth by $tw(G)$. If the treewidth of G is bounded by a constant c, a tree decomposition of width at most c of G can be constructed in linear time ([6]).

For fixed k, k-SUBCOLORABILITY can be expressed in monadic second order logic, which is a language to describe graph properties, using the following constructions: quantifications over vertices, edges, sets of edges, sets of vertices, membership tests, adjacency tests and logic operations. By the results of Courcelle [12] it is known that each problem that can be stated in monadic second order logic can be solved in linear time on graphs with bounded treewidth. Unfortunately, the multiplicative constant grows very fast, essentially it is a tower of 2's whose height is the number of quantifier-alternations of the monadic second order logic formula. In our case the height is at least linear in k.

We present a decision algorithm that for fixed k tests whether the subchromatic number $\chi_s(G) \leq k$ for graphs G with bounded treewidth. For simplicity we restrict ourself to nice tree decomposition.

The *nice tree decomposition* of G ([20]) is a tree decomposition such that: T is a rooted binary tree and for each $i \in V(T)$ at least one of the following cases applies:

- X_i is a leaf and $|X_i| = 1$, then X_i is a *leaf node*.
- X_i has one son j and $X_i = X_j \cup \{v\}$ for some $v \in V(G) \setminus X_j$, then we call X_i an *introduce node*.
- X_i has one son j and $X_i = X_j \setminus \{v\}$ where $v \in X_j$, then X_i is a *forget node*.
- X_i has two sons j, j', and $X_i = X_j = X_{j'}$ then we call X_i a *join node*.

Any tree decomposition can be transformed in linear time into a nice tree decomposition of the same width ([20]).

Denote by G_i the subgraph of G induced by vertices of X_i and G'_i the subgraph of G induced by vertices of $\bigcup_{j \succ i} X_j$, where $j \succ i$ means that j is a descendant of i i.e. i lies on the path from j to the root of T.

Let ϕ_i be a k-subcoloring of G_i, then the *color clique* of ϕ_i is any inclusion-maximal set $K \subseteq V(G_i) = X_i$, such that all vertices of K have the same color under ϕ_i and are mutually adjacent in G_i. In other words, a color clique is any clique that belongs to a color class of ϕ_i.

The entries of $\texttt{Tab}_i$ consists of several pairs (ϕ_i, g_{ϕ_i}), where ϕ_i is a feasible k-subcoloring of G_i, that might be extended to a subcoloring of G'_i and g_{ϕ_i} is a function assigning to each color clique K of ϕ_i a boolean variable which helps us to properly define a new k-subcoloring in the inductive step. Note that a single ϕ_i may occur in some entry of $\texttt{Tab}_i$ several times with different functions g_{ϕ_i}. However, as G has bounded tree width, the number of all pairs (ϕ_i, g_{ϕ_i}) is bounded by a constant. The evaluation of $\texttt{Tab}_i$ goes as follows:

1. If $X_i = \{v\}$ is a *leaf node* then $\texttt{Tab}_i$ contains all k possible k-subcolorings ϕ_i of $G_i = (\{v\}, \emptyset)$ and for the only color clique $K = \{v\}$ and all ϕ_i set $g_{\phi_i}(K) = \mathsf{true}$.

2. Let X_i be a *forget node* with the son X_j, and $\texttt{Tab}_j$ has been already computed. Then let $\texttt{Tab}_i$ contains all entries from $\texttt{Tab}_j$, restricted to set X_i. Take $(\phi_j, g_{\phi_j}) \in \texttt{Tab}_j$, and let ϕ_i be the restricted k-subcoloring. For the color clique K of ϕ_j containing the vertex $v = X_j \setminus X_i$ set $g_{\phi_i}(K \setminus \{v\}) = \mathsf{false}$ (if $K \setminus \{v\}$ is nonempty). For all other color cliques L of ϕ_i let $g\phi_i(L) = g\phi_j(L)$. Remove duplicated entries in $\texttt{Tab}_i$, if some exists.

3. Let X_i be an *introduce node* with the son X_j, $v \in V(G)$ is the added vertex and $\texttt{Tab}_j$ is already known. Then for every pair $(\phi_j, g_{\phi_j}) \in \texttt{Tab}_j$ and every k-subcoloring ϕ_i of G_i, such that ϕ_i restricted onto X_j is equal to ϕ_j, find a color clique K of ϕ_i containing v. If $K = \{v\}$ or $g_{\phi_j}(K \setminus \{v\}) = \mathsf{true}$, then add into $\texttt{Tab}_i$ entry (ϕ_i, g_{ϕ_i}) where $g_{\phi_i}(K) = \mathsf{true}$ and set $g_{\phi_i}(L) = g_{\phi_j}(L)$ for all other color cliques $L \neq K$ of ϕ_i.

4. Let X_i be a *join node* with sons X_j and $X_{j'}$ and ϕ_i a k-subcoloring of G_i. Then for all possible combinations of $(\phi_j, g_{\phi_j}) \in \texttt{Tab}_j$ and $(\phi_{j'}, g_{\phi_{j'}}) \in \texttt{Tab}_{j'}$ add the entry $(\phi_i, g_{\phi_j} \wedge g_{\phi_{j'}})$ into $\texttt{Tab}_i$ if and only if $\phi_i = \phi_j = \phi_{j'}$ and for each color clique K of ϕ_i the value of $g_{\phi_j}(K) \vee g_{\phi_{j'}}(K)$ is true. Again, if some entries are present more times, store only one.

5. Compute the values of $\texttt{Tab}_i$ for all nodes X_i in the tree, as described in steps 1–4. The graph G allows a k-subcoloring if and only if the table entry $\texttt{Tab}_m$ is nonempty for the root X_m.

To see that the algorithm is correct we think that steps 2, 3 and 4 deserve further explanation.

In step 2 we remember in the function g the fact, that a certain color clique K has already lost a vertex v, and future extension of K by v' would cause that the color class will contain induced P_3, since the edge (v, v') does not belong to G. Therefore in step 3 we try extend only those color cliques which might be extended. The same argument is used in step 4, since it is impossible to identify two color cliques when both of them already forget a vertex. Note that various functions g_{ϕ_i} for a single k-subcoloring ϕ_i may appear during executing steps 2 and 4.

This discussion concludes the proof of the first part of Theorem 3. For a graph G with tree-decomposition of width bounded by a constant c the decision of k-SUBCOLORABILITY can be performed as fast as the evaluation of the table $(\mathtt{Tab}_i)_{i \in V(T)}$, that is in time $O(n2^c k^{c+1})$. This expression is linear in n.

Note that finding the minimum k such that G allows a k-subcoloring can be done in time $O(n^{c+2})$ by running at most n tests for all $k < n$.

3.2 Cographs

Cographs belong to the class of graphs with bounded cliquewidth. Due to the results of Courcelle [13] problems that are expressible in monadic second order logic are linear time solvable in graph with bounded cliquewidth, but the constants involved are equally bad as in the case of graphs with bounded treewidth, as discussed in beginning of Section 3.1. as in the case for graphs with bounded treewidth. Hence there is a linear time algorithm to decide k-SUBCOLORABILITY, k fixed, for graphs of bounded cliquewidth.

In this section we present a $O(n^4)$ algorithm to compute the subchromatic number of cographs (graphs of bounded cliquewidth at most 2). In particular, k-SUBCOLORABILITY (k arbitrary) is efficiently solvable for cographs.

Cographs ([9]) are inductively defined as follows.

- Every single vertex graph is a cograph.
- If G_j and $G_{j'}$ are two cographs, then the disjoint union $G_j \cup G_{j'}$ is cograph.
- Similarly the join $G_j + G_{j'}$ of two cographs is a cograph. The join graph $G_j + G_{j'}$ is obtained from disjoint union $G_j \cup G_{j'}$ by adding all edges between vertices of G_j and $G_{j'}$.

With each cograph $G = (V, E)$ we associate cotree T. Each leaf node X_i of the cotree T represents a vertex $v \in V$, and in this case $G_i = (v_i, \emptyset)$. Note that each vertex $v \in V$ is represented exactly once by a leaf node in T.

Internal nodes of T have either label $\cup$ or $+$. If a parent X_i of nodes X_j and X_j' carries label $\cup$, then $G_i = G_j \cup G_{j'}$, and similarly, if it is labeled by $+$, then $G_i = G_j + G_{j'}$.

There is a linear time algorithm for recognizing whether a given graph G is a cograph and, if so, for constructing a cotree T of G (see [10]).

Our algorithm for determining the subchromatic number of a cograph relies on the following notion. A subcoloring ϕ_i of a graph G_i is of *type* (α, β) if ϕ_i has α color classes each of which induces a clique (called *small* *classes*) and β remaining classes (called *big classes*). If ϕ_i is of type (α, β), we also call ϕ_i an (α, β)-*subcoloring.*

We write $(\alpha, \beta) \preceq (\gamma, \delta)$ and say that (α, β) *minorizes* (γ, δ) if it simultaneously holds that $\beta \leq \delta$ and $\alpha + \beta \leq \gamma + \delta$. It is clear that from a subcoloring of type (α, β) any subcoloring of type (γ, δ) with $(\alpha, \beta) \preceq (\gamma, \delta)$ can be derived by adding extra colors or claiming some small color classes as big.

Consider an (α_i, β_i)-subcoloring ϕ_i of a graph G_i which arose by disjoint union $\cup$ or by join $+$ of two graphs G_j and $G_{j'}$. In the following, for each $t = j, j'$ we denote by ϕ_t be the restriction of ϕ_i on G_t and assume that ϕ_t is of type (α_t, β_t).

Lemma 9 *If $G_i = G_j + G_{j'}$ then for any (α_i, β_i)-subcoloring ϕ_i of G_i it holds: $\alpha_i \geq \max\{\alpha_j, \alpha_{j'}\}$ and $\beta_i = \beta_j + \beta_{j'}$.*

Proof: The first equality follows from the fact, that any small class of ϕ_i may consist of at most two small color classes, one in ϕ_j and one in $\phi_{j'}$. The second equality express the fact that big color class of ϕ_i is big in exactly one of ϕ_j or $\phi_{j'}$. $\square$

Lemma 10 *If ϕ_j and $\phi_{j'}$ are subcolorings of type (α_j, β_j) and $(\alpha_{j'}, \beta_{j'})$, then a $(\max\{\alpha_j, \alpha_{j'}\}, \beta_j + \beta_{j'})$-subcoloring of $G_i = G_j + G_j'$ can be obtained from ϕ_j and $\phi_{j'}$.*

Proof: A $(\max\{\alpha_j, \alpha_{j'}\}, \beta_j + \beta_{j'})$-subcoloring of G_i can be obtained by a combination $\min\{\alpha_j, \alpha_{j'}\}$ small classes of G_j and $G_{j'}$ into the same color class of G_i and leaving the other color classes disjoint. $\square$

Lemma 11 *If $G_i = G_j \cup G_{j'}$ then $\beta_i \geq \max\{\beta_j, \beta_{j'}\}$, $\alpha_i + \beta_i \geq \alpha_t + \beta_t$ $(t = j, j')$, and $\alpha_i + 2\beta_i \geq \alpha_j + \beta_j + \alpha_{j'} + \beta_{j'}$.*

Proof: Let

- r denote the number of the big color classes C of ϕ_i such that, for each $t = j, j'$, $C \cap G_t$ is a big class in ϕ_t,
- r_t denote the number of big classes C of ϕ_i such that, for $t \neq t'$, $C \cap G_t$ is a small class in ϕ_t but $C \cap G_{t'}$ is a big class in $\phi_{t'}$,
- q_t denote the number of big classes of ϕ_i belonging only to ϕ_t,

- s denote the remaining big classes of ϕ_i, which are small both in G_j and $G_{j'}$.
- l_t denote the number of the small classes of ϕ_i belonging to ϕ_t.

The first statement of the Lemma follows directly from:

$$\begin{aligned} \alpha_i &= l_1 + l_2, & \beta_i &= r + r_1 + r_2 + q_1 + q_2 + s, \\ \alpha_j &= r_1 + l_1 + s, & \beta_j &= r + r_2 + q_1, \\ \alpha_{j'} &= r_2 + l_2 + s, & \beta_{j'} &= r + r_1 + q_2. \end{aligned}$$

Moreover, $\alpha_j + \beta_j = \alpha_i + \beta_i - l_2 - q_2$ and $\alpha_{j'} + \beta_{j'} = \alpha_i + \beta_i - l_1 - q_1$, hence the second statement holds. The third statement then follows from $\alpha_j + \beta_j + \alpha_{j'} + \beta_{j'} = 2(\alpha_i + \beta_i) - l_1 - l_2 - q_1 - q_2 = \alpha_i + 2\beta_i - q_1 - q_2$. $\square$

In view of Lemmas 9 and 10, we are interested in (α_i, β_i)-subcolorings with small number β_i of big classes. A way to obtain such a subcoloring of $G_i = G_j \cup G_{j'}$ from ϕ_j and $\phi_{j'}$ is as follows: We first merge the $\min\{\beta_j, \beta_{j'}\}$ pairs of big classes of ϕ_j and $\phi_{j'}$, and then combine as many as possible of the $|\beta_j - \beta_{j'}|$ remaining big classes together with some small classes into a new color class of ϕ_i. The number of remaining small classes of ϕ_j is then $\kappa_j := \alpha_j - \min\{\alpha_j, \max\{0, \beta_{j'} - \beta_j\}\}$. Similarly ϕ_2 contains $\kappa_{j'} := \alpha_{j'} - \min\{\alpha_{j'}, \max\{0, \beta_j - \beta_{j'}\}\}$ remaining small classes.

Note that $\kappa_j = \alpha_j$ (if $\beta_j \geq \beta_{j'}$) or $\kappa_{j'} = \alpha_{j'}$ otherwise. Finally we combine κ, $0 \leq \kappa \leq \min\{\kappa_j, \kappa_{j'}\}$, small classes of ϕ_j with k small classes of $\phi_{j'}$ and get a $(\kappa_j + \kappa_{j'} - 2\kappa, \max\{\beta_j, \beta_{j'}\} + \kappa)$-subcoloring ϕ_i of $G_i = G_j \cup G_{j'}$.

This and Lemma 10 suggest the following algorithm for determining $\chi_s(G)$, assuming that the cotree T of a cograph G is given.

For each node X_i of T the algorithm stores in $\texttt{Tab}_i$ the type (α_i, β_i) of all possible (α_i, β_i)-subcolorings ϕ_i of the graph G_i that are relevant for computing $\chi_s(G)$ as follows:

1. For each leaf node X_i of T, put $(1, 0)$ into $\texttt{Tab}_i$.

2. If X_i has label $+$ and sons $X_j, X_{j'}$, then for all combinations of entries $(\alpha_j, \beta_j) \in \texttt{Tab}_j$ and $(\alpha_{j'}, \beta_{j'}) \in \texttt{Tab}_{j'}$ put into $\texttt{Tab}_i$ the entry (α_i, β_i), where $\alpha_i = \max\{\alpha_j, \alpha_{j'}\}$ and $\beta_i = \beta_j + \beta_{j'}$. Remove all minorized entries, if some exists.

3. If X_i has label $\cup$ and sons $X_j, X_{j'}$, then for all combinations of entries $(\alpha_j, \beta_j) \in \texttt{Tab}_j$ and $(\alpha_{j'}, \beta_{j'}) \in \texttt{Tab}_{j'}$ perform the following computation:

3.1. Set $\kappa_j := \alpha_j - \min\{\alpha_j, \max\{0, \beta_{j'} - \beta_j\}\}$ and $\kappa_{j'} := \alpha_{j'} - \min\{\alpha_{j'}, \max\{0, \beta_j - \beta_{j'}\}\}$.

3.2. For each κ varying from 0 to $\min\{\kappa_j, \kappa_{j'}\}$ put into $\texttt{Tab}_i$ the entry (α_i, β_i), where $\alpha_i = \kappa_j + \kappa_{j'} - 2\kappa$ and $\beta_i = \max\{\beta_j, \beta_{j'}\} + \kappa$.

Remove all minorized entries, if some exists.

4. Return $\chi_s(G) = \min\{\alpha_m + \beta_m : (\alpha_m, \beta_m) \in \texttt{Tab}_m\}$ for the root X_m of T.

Note that the number of all entries (α_i, β_i) stored in each $\texttt{Tab}_i$ is bounded by k. Moreover, as discussed by Lemmas 9 and 10 and after Lemma 11, if $(\alpha_i, \beta_i) \in \texttt{Tab}_i$ then there exists a subcoloring of G_i of type (α_i, β_i).

The following lemma shows the correctness of the algorithm:

Lemma 12 *For every node X_i of T and every subcoloring ϕ_i of G_i of type (γ_i, δ_i), there exists a pair $(\alpha_i, \beta_i) \in \texttt{Tab}_i$ such that $(\alpha_i, \beta_i) \preceq (\gamma_i, \delta_i)$.*

Proof: By induction on the level of X_i. The statement of the lemma is correct for leaves of the cotree. So, let X_i be an internal node of T, and let $X_j, X_{j'}$ be the two sons of X_i. For $t = j, j'$ let ϕ_t denote the restriction of ϕ_i on G_t and suppose that ϕ_t is of type (γ_t, δ_t). By induction there exists $(\alpha_t, \beta_t) \in \texttt{Tab}_t$ such that

$$(\alpha_t, \beta_t) \preceq (\gamma_t, \delta_t). \tag{I}$$

We distinguish two cases.

Case 1: X_i is $+$ node.
Set $\alpha_i := \max\{\alpha_j, \alpha_{j'}\}$ and $\beta_i := \beta_j + \beta_{j'}$. Then, according to step 2 of the algorithm, some entry in $\texttt{Tab}_i$ minorizes (α_i, β_i). We claim that $(\alpha_i, \beta_i) \preceq (\gamma_i, \delta_i)$: By the induction hypothesis (I) and Lemma 9, $\beta_i = \beta_j + \beta_{j'} \leq \delta_j + \delta_{j'} = \delta_i$. To see the second condition in the definition of $\preceq$ we may assume without loss of generality that $\alpha_j \leq \alpha_{j'}$. Then

$$\begin{aligned}
\alpha_i + \beta_i &= \alpha_{j'} + \beta_j + \beta_{j'} \leq \beta_j + \gamma_{j'} + \delta_{j'} \\
&\leq \delta_j + \gamma_{j'} + \delta_{j'} = \gamma_{j'} + \delta_i \\
&= \max\{\gamma_j, \gamma_{j'}\} + \delta_i \leq \gamma_i + \delta_i.
\end{aligned}$$

Case 2: X_i has label $\cup$.
Let $\kappa_j, \kappa_{j'}$ be the integers computed from (α_j, β_j) and $(\alpha_{j'}, \beta_{j'})$ in step 3 of the algorithm. Note that by (I) and Lemma 11,

$$\max\{\beta_j, \beta_{j'}\} \leq \max\{\delta_j, \delta_{j'}\} \leq \delta_i,$$

hence there exists some integer $\kappa \geq 0$ such that

$$\max\{\beta_j, \beta_{j'}\} + \kappa \leq \delta_i \text{ and } \kappa \leq \min\{\kappa_j, \kappa_{j'}\}.$$

Let κ be the maximum integer satisfying these properties. Note that by the maximality of κ,

$$\text{either } \kappa = \min\{\kappa_j, \kappa_{j'}\} \text{ or } \kappa = \delta_i - \max\{\beta_j, \beta_{j'}\}. \tag{II}$$

Set $\alpha_i := \kappa_j + \kappa_{j'} - 2\kappa$ and $\beta_i := \max\{\beta_j, \beta_{j'}\} + \kappa$. Then according to step 3 of the algorithm, some entry in $\texttt{Tab}_i$ minorizes (α_i, β_i). We claim that $(\alpha_i, \beta_i) \preceq (\gamma_i, \delta_i)$.

By the choice of κ and of β_i, we have $\beta_i \leq \delta_i$. To see the second condition in the definition of $\preceq$, we may assume without loss of generality that $\beta_j \leq \beta_{j'}$. Then $\kappa_j = \alpha_j - \min\{\alpha_j, \beta_{j'} - \beta_j\}$ and $\kappa_{j'} = \alpha_{j'}$.
If $\kappa_j = 0$ then, $\kappa \leq \min\{\kappa_j, \kappa_{j'}\} = 0$ and

$$\begin{aligned} \alpha_i + \beta_i &= \alpha_{j'} - 2\kappa + \beta_{j'} + \kappa \leq \alpha_{j'} + \beta_{j'} \\ &\leq \gamma_{j'} + \delta_{j'} \leq \gamma_i + \delta_i. \end{aligned}$$

If $\kappa_j = \alpha_j - (\beta_{j'} - \beta_j)$ then

$$\alpha_i + \beta_i = \alpha_j - (\beta_{j'} - \beta_j) + \alpha_{j'} - 2\kappa + \beta_{j'} + \kappa = \alpha_j + \beta_j + \alpha_{j'} - \kappa.$$

In this case in consideration of (II) there are two possibilities for κ. If $\kappa = \delta_i - \max\{\beta_j, \beta_{j'}\} = \delta_i - \beta_{j'}$ we get,

$$\begin{aligned} \alpha_i + \beta_i &= \alpha_j + \beta_j + \alpha_{j'} - (\delta_i - \beta_{j'}) \\ &\leq \gamma_j + \delta_j + \gamma_{j'} + \delta_{j'} - \delta_i \\ &\leq (\gamma_i + 2\delta_i) - \delta_i = \gamma_i + \delta_i, \end{aligned}$$

and if $\kappa = \min\{\kappa_j, \kappa_{j'}\}$, then

$$\begin{aligned} \alpha_i + \beta_i &= \alpha_j + \beta_j + \alpha_{j'} - \min\{\kappa_j, \kappa_{j'}\} \\ &= \alpha_j + \beta_j + \alpha_{j'} - \min\{\alpha_j - \beta_{j'} + \beta_j, \alpha_{j'}\} \\ &= \max\{\alpha_{j'} + \beta_{j'}, \alpha_j + \beta_j\} \\ &\leq \max\{\gamma_{j'} + \delta_{j'}, \gamma_j + \delta_j\} \leq \gamma_i + \delta_i. \end{aligned}$$

Thus, in any case, $\alpha_i + \beta_i \leq \gamma_i + \delta_i$. Hence $(\alpha_i, \beta_i) \preceq (\gamma_i, \delta_i)$, and the lemma is proved. □

The direct application of Lemma 12 concludes the proof of the second part of Theorem 3 in time $O(nk^3)$.

This algorithm can be either used to test k-Subcolorability when we consider only those types of subcolorings such that $\alpha + \beta \leq k$, but also in time $O(n^4)$ computes $\chi_s(G)$, since $\chi_s(G) \leq n$.

3.3 Graphs with bounded cliquewidth

Graphs of bounded cliquewidth generalize both the notion of cographs and graphs with bounded treewidth (cf. [14, 11]).

All graph problems that are expressible in monadic second order logic (MSOL) can be solved in linear time on graphs with bounded cliquewidth ([12, 13]), given an expression defining the input graph. There are many problems not expressible in MSOL (for example HAMILTONICITY, k-COLORABILITY (k arbitrary), ...), but, nevertheless, are solvable in polynomial time on graphs with bounded cliquewidth (cf. [16, 21]).

In this section, we extend our problem k-SUBCOLORABILITY (k arbitrary) to this list. Our approach is different from that of [16] and of [21]. SUBCOLORABILITY is, however, much more complicated than COLORABILITY (as one may see in case of cographs already) and it is not clear whether the schemes suggested in [16, 21] can be modified for our problem.

Let us now recall the notion of cliquewidth. Consider the a construction tree T over a finite label set L, which recursively defines a graph G as follows:

- Every *leaf node* X_i with to operation $t(v)$ which means creation of a one-vertex graph $G_i = (\{v\}, \emptyset)$ where v is labeled by $t \in L$.

- The *join node* X_i with operation $\eta_{s,t}$ and one child X_j, inserts to the graph G_j all edges between vertices labeled by s and t. $(s, t \in L)$. We require that labels s and t are distinct, however, some edges between vertices labeled by s and t may already exist.

- The *relabel node* X_i with operation $\rho_{s \to t}$ $(s, t \in L)$ and one son X_j changes all labels s in the graph G_j to t.

- Finally, the *union node* X_i with two children X_j and $X_{j'}$ corresponds to the graph $G_i = G_j \cup G_{j'}$, where all vertices maintain their labels from subgraph G_j and $G_{j'}$ respectively.

The cliquewidth of a graph G is the smallest cardinality of the label set L, such that there exists a construction tree T using this label set L and the G is isomorphic to the graph G_m which corresponds to the root X_m of the tree T.

Every construction tree can be in polynomial time rearranged (cf. [14]), such that

- For each join operation $\eta_{s,t}$ we may assume that in this moment there are no edges between vertices labelled by s and t.

- For each node X_i it is possible to compute in polynomial time a graph G'_i, the subgraph of G_m induced by $V(G_i)$ (i.e. it contains G_i and all edges that will be added later due to join operations on the path from X_i to the root X_m).

- For each X_i it is possible to compute also in polynomial time an auxiliary graph F_i, defined on the label set used on G_i, where two labels s and t are connected in F_i, if on the path from X_i to the root there is a sequence of operations ρ changing labels s to s' and t to t' (possibly in several iterations) followed by $\eta_{s',t'}$. In other words edge (s,t) in F_i has the meaning that later there should be added an edge between vertices that in G_i have labels s and t.

Now assume that the size of the set L is fixed constant c. Consider arbitrary set of labels $K \subseteq L$ and define a (possibly empty) subgraph G_i^K of G'_i induced by all vertices of G_i whose label belongs to K.

Assume that ϕ_i is a k-subcoloring of a graph G'_i and V_a its color class.

The type of the color class V_a is a vector τ of length 2^c, where entries are indexed by sets $K \subseteq L$, and

$$\tau_K(V_a) = \begin{cases} 0 & \text{if } V_a \text{ induces empty graph in } G_i^K, \\ 1 & \text{if } V_a \text{ induces in } G_i^K \text{ single clique with at least one vertex} \\ 2 & \text{if } V_a \text{ induces in } G_i^K \text{ a disjoint union of nonempty cliques.} \end{cases}$$

Observe that $\tau_K(V_a) \le \tau_{K'}(V_a)$ whenever $K \subseteq K'$. Moreover there exist at most $M = 3^{2^c}$ different types of color classes. Let Γ be the set of all possible color class types.

The following definition will help us to control color class types in the time of relabeling operation $\rho_{s\to t}$.

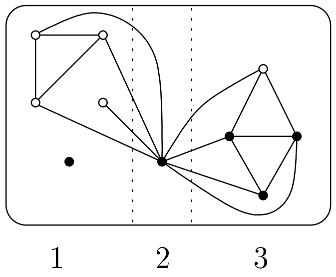
We say that a color class of type ς transmutes into a color class of type τ via relabeling $s \to t$, if

- $\tau_K = \varsigma_K$ if $s, t \notin K$,

- $\tau_K = \varsigma_{K\cup\{s\}}$ if $t \in K, s \notin K$,

- $\tau_K = \tau_{K\setminus\{s\}}$ if $s \in K$.

Observe, that for every type ς, the target type τ via transmutation $s \to t$ is unique. Note that such test can be performed in a constant time, as long as the length of the type is constant.



$$\tau(V_\circ) = (2, 0, 1, 2, 2, 1, 2)$$

$$\tau(V_\bullet) = (1, 1, 1, 2, 2, 1, 2)$$

$$L = \{1, 2, 3\}$$

Figure 5: Example of a 2-subcoloring of a graph of cliquewidth 3

To illustrate these notions, see a 2-subcoloring of a graph depicted in Fig. 5. If we order subsets of L as $(1, 2, 3, 12, 13, 23, 123)$ then the types of the white and black class are

$$\tau(V_\circ) = (2, 0, 1, 2, 2, 1, 2) \qquad \tau(V_\bullet) = (1, 1, 1, 2, 2, 1, 2)$$

Observe also that in this example the relabeling $\rho_{1\to 3}$ transmutes the class $V_\bullet$ into a class of type $(0, 1, 2, 1, 2, 2, 2)$.

For a k-subcoloring ϕ_i we define its characteristic vector $\mathbf{a}$ indexed by color class types $\tau \in \Gamma$ whose entry $\mathbf{a}_\tau$ equals to the number of color classes of ϕ_i that are of type τ.

The following lemma gives us a tool to test whether the characteristic vector $\mathbf{a}$ of a k-subcoloring ϕ_i of G'_i can be composed from the characteristic vectors $\mathbf{b}$ and $\mathbf{c}$ of a k-subcoloring of G'_j and $G'_{j'}$, respectively, during the union operation $G_i = G_j \cup G_{j'}$. In such case we say that $\mathbf{a}$ is *compatible* with a composition of $\mathbf{b}$ and $\mathbf{c}$ in the label graph F_i.

Before stating the lemma we would like to discuss one particular case in the composition of types τ and ω into ς (with respect to F_i) in more detail. Consider $K \subseteq L$, such that $\tau_K = \omega_K = 1$. In this case we have to decide whether $\varsigma_K = 1$ or 2. To get the right answer we first find sets $I, J \subseteq K$ such that $\tau_I = \omega_J = 1$ and $\tau_{K\setminus I} = \omega_{K\setminus J} = 0$. Only two situations make composition of τ and ω possible:

- Either there is no edge in F_i between any $u \in I$ and $v \in J$. In this case we set $\varsigma_K = 2$.
- Or sets I and J are disjoint and F_i contains all edges between vertices from I and from J, and then we set $\varsigma_K = 1$.

Only if one of these two cases applies, we say that τ_K and ω_K *could be merged* into ς_K. Observe also that for a single K we may perform such test in time $O(|K|^2)$.

In Fig. 6 we present two examples for $K = \{1,2,3\}$. In the first one we consider $I = \{1,2\}$ and $J = \{1\}$. Such two types could be merged if and only if F_i does not contain the edge $(1,2)$, and the resulting type is $\varsigma_K = 2$. In the next example assume $I = \{1,2\}$ and $J = \{3\}$. Then these types could be merged if and only if none or both of the two edges $(1,3)$ and $(2,3)$ are in F_i. The existence of dotted edges is not important here.

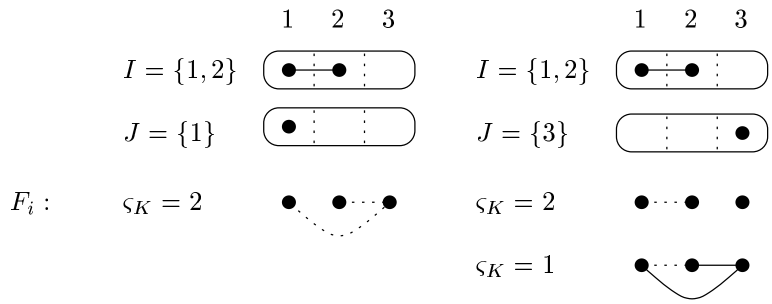


Figure 6: Merging of two color types $\tau_K = \omega_K = 1$.

In any other case of composition of two types τ_K and ω_K, where at least one of them is not 1, we follow the majority principle, i.e. $\varsigma_K = \max\{\tau_K, \omega_K\}$.

Lemma 13 *The type* $\mathbf{a}$ *is compatible with a composition of* $\mathbf{b}$ *and* $\mathbf{c}$ *in the label graph* F_i *if and only if the following system of linear inequalities over variables* $\mathbf{x}_{\varsigma,\tau,\omega}$ *has an integral solution.*

- $\mathbf{x}_{\varsigma,\tau,\omega} \geq 0$,
- $\mathbf{a}_\varsigma = \sum_{\tau,\omega\in\Gamma} \mathbf{x}_{\varsigma,\tau,\omega}$,
- $\mathbf{b}_\tau = \sum_{\varsigma,\omega\in\Gamma} \mathbf{x}_{\varsigma,\tau,\omega}$,
- $\mathbf{c}_\omega = \sum_{\varsigma\tau,\in\Gamma} \mathbf{x}_{\varsigma,\tau,\omega}$,
- $\mathbf{x}_{\varsigma,\tau,\omega} = 0$, *if there exists* $K \subseteq L$ *such that*
  - *either* $(\tau_K \neq 1 \vee \omega_K \neq 1)$ *and* $\varsigma_K \neq \max\{\tau_K, \omega_K\}$,
  - *or* $\tau_K = \omega_K = 1$ *and* τ_K *and* ω_K *could not be merged into* ς_K.

Since the dimension of this instance is bounded by a constant M^3, the corresponding integer linear program could be solved in polynomial time.

Now we are ready to present the decision algorithm. We store at $\texttt{Tab}_i$ all characteristic vectors of all proper k-subcolorings of the graph G'_i. Observe that the number of entries is bounded by n^M where $n = |V(G)|$.

The recursive evaluation of $\texttt{Tab}_i$ goes as follows:

1. For a leaf node X_i with $t(v)$ store in $\texttt{Tab}_i$ the unique characteristic vector $\mathbf{a}$ of the k-subcoloring for which $\{v\} = V_1$ is its the only color class and its type satisfies $\tau_K(V_1) = 1$ if $t \in K \subseteq L$, and $\tau_K(V_1) = 0$ otherwise.

2. All entries of a join node X_i with $\eta_{s,t}$ are taken from its only son X_j. Observe that in this case $G'_j = G_j$ including the vertex labeling, so no new restriction should be applied.

3. For a recolor node X_i with label $\rho_{s \to t}$ and child X_j take every characteristic vector $\mathbf{b} \in \texttt{Tab}_j$ and compute $\mathbf{a}_\tau$ as the sum of all $\mathbf{b}_\varsigma$ where the sum is taken over all types ς which transmutes onto τ via relabeling $s \to t$.

4. For the union node X_i with children $X_j, X_{j'}$ test every possible type $\mathbf{a}$ perform the test whether it is compatible with some type $\mathbf{b} \in \texttt{Tab}_j$ and some type $\mathbf{c} \in \texttt{Tab}_{j'}$. If the test succeeds, put $\mathbf{a}$ into $\texttt{Tab}_i$.

Remember that each step is followed by removing duplicities in table entries, if some appears.

Note that computing $\chi_s(G)$ can be also determined in polynomial time. As, for constant cliquewidth c, the number of types of a color class is 3^{2^c}, the evaluation of the table $\texttt{Tab}_{i \in V(T)}$ can be performed in $O(n^{3^{2^c}})$ time. Thus, after n tests, we can compute $\chi_s(G)$ in $O(n^{3^{2^c}+1})$ time.

4 Conclusion

In this paper we brought both positive and negative results on the computational complexity of k-SUBCOLORING problem.

We showed full complexity classification on planar graphs, in the case of 2-subcolorability we have even refined this classification on the degree condition.

Similarly we have shown that the general k-SUBCOLORABILITY problem is *NP*-complete on graphs of degree at most k^2. Here we would like to point out that in the view of degree constraints the PLANAR GRAPH 3-SUBCOLORABILITY and k-SUBCOLORABILITY is not fully classified and we are convinced that it deserves further research.

To motivate this study we would like to mention that we expect that there is a possibility to construct uniquely k-subcolorable graphs of degree $2k$. Here, we would like to propose a generalization of the complete computational complexity characterization for the case $k = 2$ as stated in the following conjecture:

Conjecture: *For every fixed $k \geq 2$, k-SUBCOLORABILITY is NP-complete for graphs with maximum degree $2k$.*

If true, this conjecture is best possible because every graph with maximum degree at most $2k - 1$ is k-subcolorable cf. Section 1.

As a particular result it would be interesting to see whether the positive answer to 4-SUBCOLORABILITY for planar graphs is as hard as the famous four color theorem, or whether this problem allows a simpler direct proof.

Finally, we would like to remark that in all considered cases (bounded tree width, cographs, bounded cliquewidth), an optimal subcoloring could be derived, in the same time bounds, from the table.

Acknowledgments

The third author (VBL) thanks Hoàng-Oanh Le for her interesting comments. He also thanks the second author (KJ) for inviting him to visit his research group in Kiel.

References

[1] D. Achlioptas, The complexity of G-free graph colourability, *Discrete Math* 165/166 (1997), 31-38

[2] M.O. Albertson, R.E. Jamison, S.T. Hedetniemi, S.C. Locke, The subchromatic number of a graph, *Discrete Math.* 74 (1989) 33-49

[3] S. Arnborg, Efficient algorithms for combinatorial problems on graphs with bounded decomposability – A survey, *BIT*, 25 (1985) 2-23.

[4] H.L. Bodlaender, A tourist guide through treewidth. *Acta Cybernetica*, 11 (1993) 1-23.

[5] H.L. Bodlaender, A partial k-arboretum of graphs with bounded treewidth, *Theoretical Computer Science*, 209 (1998) 1-45

[6] H.L. Bodlaender, A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Computing*, 25 (1996) 1305-1317.

[7] I. Broere, C.M. Mynhardt, Generalized colorings of outerplanar and planar graphs, *Proc. Graph Theory with Applications to Algorithms and Computer Science*, Kalamazoo, Mich., (1984) 151-161

[8] J.L. Brown, D.G. Corneil, On generalized graph colorings, *J. Graph Theory* 11 (1987) 87-99

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

[10] D.G. Corneil, Y. Perl, L.K. Stewart, A linear recognition algorithm for cographs, *SIAM J. Computing* 14 (1985) 926-934

[11] D.G. Corneil, U. Rotics, On the relationship between clique-width and treewidth, In: *Graph-Theoretic Concepts in Computer Science, 27th Int. Workshop (WG 20001), Lecture Notes in Computer Science* 2204 (2001), in print

[12] B. Courcelle, Graph rewriting: an algebraic and logical approach, *Handbook of Theoretical Computer Science*, volume B, (1990) 192-242

[13] B. Courcelle, The monadic second-order logic of graphs I: Recognizable sets of finite graphs, *Information and Computation 85* (1991) 12-75

[14] B. Courcelle, S. Olariu, Uper bounds to the clique-width of graphs, *Discrete Appl. Math.* 101 (2000) 77-114

[15] P. Erdős, Bipartite subgraphs of graphs, *Math. Lapok* 18 (1967) 283-288

[16] W. Espelage, F. Gurski, E. Wanke, How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time. In: *Graph-Theoretic Concepts in Computer Science, 27th Int. Workshop (WG 20001), Lecture Notes in Computer Science* 2204 (2001), in print

[17] M. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W.H. Freeman, San Fransisco, 1979

[18] M.R. Garey, D.S. Johnson, L. Stockmayer, Some simplified *NP*-complete graph problems, *Theoretical Computer Science* 1 (1976) 237-267

[19] H.A. Jung, On a class of posets and the corresponding comparability graphs, *J. Combin. Theory Ser. B* 24 (1978) 125-13

[20] T. Kloks, *Treewidth – Computations and Approximations*, Springer-Verlag, 1994. Lecture Notes in Computer Science 842

[21] D. Kobler, U. Rotics, Polynomial algorithms for partitioning problems on graphs with fixed clique-width. In: *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA 2001)*, pages 468-475

[22] F. Maffray, M. Preissman, On the *NP*-completeness of k-colorability problem of triangle-free graphs, *Discrete Math.* 162 (1996) 313-317

[23] C.M. Mynhardt, I. Broere, Generalized colorings of graphs, *Proc. Graph Theory with Applications to Algorithms and Computer Science*, Kalamazoo, Mich., (1984), 583-594

[24] N. Robertson and P.D. Seymour, Graph minors. II. Algorithmic aspects of tree-width, *Journal of Algorithms* 7 (1986) 309-322.

[25] T.J. Schaefer, The complexity of the satisfability problem, *Proc. 10th Ann. ACM Symp. on theory of computing* (1978) 216-226