Consensus models: Computational complexity aspects in modern approaches to the list coloring problem

Damian Bogdanowicz, Krzysztof Giaro, Robert Janczewski [*]

Department of Algorithms and System Modeling, Faculty of Electronics, Telecommunication and Informatics, Gdańsk University of Technology, Narutowicza 11/12, Gdańsk, Poland

ARTICLE INFO

ABSTRACT

In the paper we study new approaches to the problem of list coloring of graphs. In the problem we are given a simple graph $G = (V, E)$ and, for every $v \in V$, a nonempty set of integers $S(v)$; we ask if there is a coloring c of G such that $c(v) \in S(v)$ for every $v \in V$. Modern approaches, connected with applications, change the question—we now ask if S can be changed, using only some elementary transformations, to ensure that there is such a coloring and, if the answer is yes, what is the minimal number of changes. In the paper for studying the adding, the trading and the exchange models of list coloring, we use the following transformations:

- adding of colors (the adding model): select two vertices u, v and a color $c \in S(u)$; add c to $S(v)$, i.e. set $S(v) := S(v) \cup \{c\}$;
- trading of colors (the trading model): select two vertices u, v and a color $c \in S(u)$; move c from $S(u)$ to $S(v)$, i.e. set $S(u) := S(u) \setminus \{c\}$ and $S(v) := S(v) \cup \{c\}$;
- exchange of colors (the exchange model): select two vertices u, v and two colors $c \in S(u)$, $d \in S(v)$; exchange c with d, i.e. set $S(u) := (S(u) \setminus \{c\}) \cup \{d\}$ and $S(v) := (S(v) \setminus \{d\}) \cup \{c\}$.

Our study focuses on computational complexity of the above models and their edge versions. We consider these problems on complete graphs, graphs with bounded cyclicity and partial k-trees, receiving in all cases polynomial algorithms or proofs of NP-hardness.

1. Introduction

Let $G = (V, E)$ be a simple graph with vertex set[1] V and edge set E. By n and m we mean the number of vertices and the number of edges of G, respectively. By $N(v)$ we denote the *neighborhood* of vertex v, i.e. the set of its neighbors; Δ stands for the maximum degree over all vertices of G. We also use symbol γ to denote the *cyclomatic number* of G, i.e. the minimal number of edges that must be removed from E to make G acyclic.

Let S be a *list assignment* for G, i.e. a function that assigns nonempty finite subsets of $\mathbb{N}$ to vertices of G. By an *S-coloring* of G we mean any function $c \colon V \to \mathbb{N}$ such that

(1) $c(u) \neq c(v)$ for every two adjacent vertices $u, v \in V$;
(2) $c(v) \in S(v)$ for every vertex $v \in V$.

---

* Corresponding author. Tel.: +48 583471064.
  E-mail addresses: Damian.Bogdanowicz@eti.pg.gda.pl (D. Bogdanowicz), giaro@eti.pg.gda.pl (K. Giaro), skalar@eti.pg.gda.pl (R. Janczewski).

[1] All notations introduced here are functions of G. We drop the reference to the graph, if G is clear from the context.

Functions $c: V \to \mathbb{N}$ that satisfy the first of the above conditions only are called *vertex colorings* of G. Vertex colorings $c: V \to \{1, 2, \ldots, k\}$, $k \in \mathbb{N}$ are called *k-colorings*. If G has an S-coloring (k-coloring), we say that the pair (G, S) (graph G) is *list colorable* (*k-colorable*). We also define $l = |\bigcup_{v \in V} S(v)|$ and $L = \sum_{v \in V} |S(v)|$.

The problem of deciding if a pair (G, S) is list colorable (the *list coloring* problem) has been intensively studied. It is NP-complete for many simple classes of graphs and very restricted list assignments. The list of NP-complete cases includes complete bipartite graphs [4,13], line graphs of complete [10] and complete bipartite graphs [2], unions of two intersecting complete graphs [7,13] and even the case when a graph is subcubic, planar and bipartite [8,13]. On the other hand, there is a linear algorithm for trees [9].

In the paper we study consensus models for list colorings. These models were introduced by Mahadev and Roberts in [11] as a tool for modeling some aspects related to error correction in a physical mapping of DNA, where classic coloring seems to be insufficient. In an ideal situation, data resulting from biological experiments (modeled by some graph G with vertices corresponding to DNA fragments of unknown position and list assignment S) is correct and (G, S) is list colorable. In a real situation, both graph and list assignment contain some errors and they must be changed to make (G, S) list colorable. This is the place where consensus models can be applied. These models can be described as follows: given a simple graph G and list assignment S for G; can we transform S into S', using only some elementary transformations, to ensure that (G, S') is list colorable? If it is possible, what is the minimal number of such transformations?

Mahadev and Roberts introduced three types of transformations: *adding*, *trading* and *exchanging* of colors (consensus models related to them are named the *adding*, the *trading* and the *exchange* coloring model, respectively). They can be defined as follows:

- adding of a color: select two vertices u, v and a color $c \in S(u)$; add c to $S(v)$, i.e. set $S(v) := S(v) \cup \{c\}$;
- trading of a color: select two vertices u, v and a color $c \in S(u)$; move c from $S(u)$ to $S(v)$, i.e. set $S(u) := S(u) \setminus \{c\}$ and $S(v) := S(v) \cup \{c\}$;
- exchange of colors: select two vertices u, v and two colors $c \in S(u)$, $d \in S(v)$; exchange c with d, i.e. set $S(u) := (S(u) \setminus \{c\}) \cup \{d\}$ and $S(v) := (S(v) \setminus \{d\}) \cup \{c\}$.

The minimal number of adding (trading, exchanging) operations that is needed to change a pair (G, S) into list colorable one will be called the *added* (*traded*, *exchanged*) *inflexibility* and will be denoted by $I_{ad}(G, S)$ ($I_{tr}(G, S)$, $I_{ex}(G, S)$). If such a change is not possible, we assume $I_{ad}(G, S) = +\infty$ ($I_{tr}(G, S) = +\infty$, $I_{ex}(G, S) = +\infty$).

Fact 1 (*[11]*)**.** *Let G be a simple graph and S be a list assignment for G. Then*

(1) *$I_{ad}(G, S)$ is finite if and only if $l \geq \chi$, where χ is the chromatic number of G;*
(2) *$I_{tr}(G, S)$ is finite if and only if there exists a vertex coloring $c: V \to \bigcup_{v \in V} S(v)$ such that every color i is used at most $|\{v \in V : i \in S(v)\}|$ times;*
(3) *if $I_{ex}(G, S)$ is finite then $I_{tr}(G, S)$ is finite; if $I_{tr}(G, S)$ is finite then $I_{ad}(G, S)$ is finite.* □

Graph-theoretical properties of I_{ad}, I_{tr} and I_{ex} were investigated in [11]. In this paper we focus on complexity of computing I_{ad}, I_{tr} and I_{ex} for different graph classes and identifying computationally hard cases. First observe that the problems under consideration are closely related to the list coloring, because the pair (G, S) is list colorable if and only if $I_{ad}(G, S) = 0$ ($I_{tr}(G, S) = 0$, $I_{ex}(G, S) = 0$).

Fact 2. *Consider any class of graphs $\mathcal{G}$ (with polynomial time verifiable membership), for which the list coloring problem is NP-complete. Then for pairs (G, S) with $G \in \mathcal{G}$:*

(1) *the problems of computing I_{ad}, I_{tr} and I_{ex} are NP-hard;*
(2) *if $\mathcal{G}$ is closed under graph disjoint union operation then for every fixed $t \in \mathbb{N}$ the problems of verifying whether $I_{ad}(G, S) \leq t$, $I_{tr}(G, S) \leq t$ and $I_{ex}(G, S) \leq t$ are NP-complete.*

Proof. (1) Obvious, since verification whether $I_{ad} = 0$ ($I_{tr} = 0$, $I_{ex} = 0$) is assumed to be NP-complete.

(2) It easily follows from the fact that when performing at most t list transformations on the graph made of $2t + 1$ disjoint copies of G, lists of vertices of some copy will not change. □

The above fact leads to the following conclusion: I_{ad}, I_{tr} and I_{ex} can be computed efficiently only for those classes of graphs for which the list coloring problem is polynomially solvable. Moreover, it shows that the problems of verifying inequalities $I_{ad}(G, S) \leq t$, $I_{tr}(G, S) \leq t$ and $I_{ex}(G, S) \leq t$ are NP-complete even for subcubic planar graphs because the list coloring problem for the class is known to be NP-complete [8,13].

Fact 3. *Suppose that the list coloring problem on a certain (polynomially verifiable) class of graphs $\mathcal{G}$ is solvable in polynomial time. Then for every fixed $t \in \mathbb{N}$ the verification whether $I_{ad}(G, S) \leq t$ for $G \in \mathcal{G}$ ($I_{tr}(G, S) \leq t$, $I_{ex}(G, S) \leq t$) is also polynomial.*

Proof. It follows immediately from the fact that there is a polynomial number of ways in which we can add/trade/exchange at most t colors since t is a constant. □

Later we will see that when t is a part of the instance of the problem then it may become NP-hard.

The list coloring problem has its edge version—the so-called *edge list coloring* problem. In the problem, we are given a graph $G = (V, E)$ and *edge list assignment* S, i.e. a function that assigns nonempty finite subsets of $\mathbb{N}$ to edges of G. The goal is to verify whether there is an *edge S-coloring* of G, i.e. a function $c': E \to \mathbb{N}$ so that:

(1) $c'(e) \neq c'(f)$ for every two adjacent edges $e, f \in E$;
(2) $c'(e) \in S(e)$ for every edge $e \in E$.

If G has an edge S-coloring, the pair (G, S) is said to be *edge list colorable*.

Consensus models have their natural edge versions, too. This time we are given a pair (G, S), where G is a simple graph and S is an edge list assignment for G. The goal is to verify if (G, S) can be changed into edge colorable pair (G, S'), using analogous transformations: adding, trading and exchanging of colors. As in vertex consensus models, we ask if it can be done and, if the answer is yes, what is the minimal number of transformations needed to do so. We also define *edge inflexibilities* $I'_{ad}(G, S)$, $I'_{tr}(G, S)$ and $I'_{ex}(G, S)$ as the minimal number of added/traded/exchanged colors that change (G, S) into edge list colorable one (or $+\infty$, if it is not possible). Obviously $I'_{ad}(G, S) = I_{ad}(L(G), S)$, $I'_{tr}(G, S) = I_{tr}(L(G), S)$ and $I'_{ex}(G, S) = I_{ex}(L(G), S)$, where $L(G)$ is the line graph of G.

The remainder of the paper is organized as follows. In Section 2 we show that adding, trading and exchange models are polynomially solvable on complete graphs. Next, in Section 3, we will show that the adding model and its edge version are polynomially solvable on trees and even bounded cyclicity graphs. The trade model is not that simple—NP-completeness occurs even in the case of trees of very simple structure (caterpillars) for both vertex and edge versions. However, bounding the colors set's size by any fixed constant reestablishes polynomiality; procedures for partial k-trees and simple almost trees (edge version) will be presented in Section 4. Similar results for the exchange model will be presented, too.

2. Complete graphs

It is easy to see that the adding and the trading model do not differ much on complete graphs. Indeed, every color can be used only once, so adding is equivalent to trading in this case. Moreover, for any shortest sequence of exchanges that changes (K_n, S) into list colorable (K_n, S') there is an S'-coloring of G such that exactly one of the two colors that appear in any exchange in the sequence will be used in the S'-coloring. This leads to the following conclusion:

Fact 4. *For complete n-vertex graph K_n the inflexibility $I_{ad}(K_n, S)$ is finite if and only if $l \geq n$. Moreover, $I_{ad}(K_n, S) = I_{tr}(K_n, S) = I_{ex}(K_n, S)$.* □

The problem of computing the added (traded/exchanged) inflexibility of complete graphs can be easily reduced to the problem of finding maximum cardinality matching in bipartite graphs. Let us recall that we can find such a matching in a given graph G in $O(m\sqrt{n})$ time [12].

Theorem 5. *Computing of $I_{ad}(K_n, S)$ ($I_{tr}(K_n, S)$, $I_{ex}(K_n, S)$) can be done in $O(L\sqrt{n + l})$ time.*

Proof. Verifying whether $I_{tr}(K_n, S)$ is finite can be done in $O(L)$ time by Fact 4. Then we construct a bipartite graph $G' = (V'_1 \cup V'_2, E')$, where $V'_1 = V(K_n)$, $V'_2 = \bigcup_{v \in V(K_n)} S(v)$ and

$$E' = \{\{v, x\} : v \in V'_1 \wedge x \in V'_2 \wedge x \in S(v)\}.$$

We will show that if M is a maximum cardinality matching in G' then

$$I_{tr}(K_n, S) = n - |M|.$$

($\leq$) We construct coloring c of K_n as follows. In the first step we set $c(v) = x$ for all such vertices v that $\{v, x\} \in M$. In the second step we color with colors from V'_2 all remaining vertices in any way that leads to a proper coloring. This step needs at most $n - |M|$ trades.

($\geq$) Let c be a coloring of K_n that requires $I_{tr}(K_n, S)$ trades. Then $M' = \{\{v, c(v)\} : v \in V'_1 \wedge c(v) \in S(v)\}$ is a matching in G'. Vertices v for which $\{v, c(v)\} \notin M$, are the only that require trades. Thus $I_{tr}(K_n, S) = n - |M'| \geq n - |M|$.

To complete the proof, it suffices to see that $n(G') = n + l$ and $m(G') = L$. □

Notice that even a slight generalization of the problem, to the case when the graph is formed of two intersecting cliques leads to NP-hardness—this follows from already mentioned NP-completeness of the list coloring for such graphs. The problems of computing I'_{ad}, I'_{tr} and I'_{ex} for complete graphs are also NP-hard since the list coloring problem is NP-complete for line graphs of complete graphs [10].

3. Trees and almost trees

The complexity of the three consensus models differs on trees: the adding model is polynomially solvable; the trading and exchange models are NP-complete. The same holds for their edge versions.

The adding model is closely related to the so-called *cost list coloring* problem. In the problem we are given a graph $G' = (V', E')$, list assignment S' for G' and the function $f' : \{(v, x) : v \in V' \wedge x \in S'(v)\} \to \mathbb{N} \cup \{0\}$; we seek for an S'-coloring c of G' that minimizes its *cost*, i.e. the number $\sum_{v \in V'} f'(v, c(v))$.

To see that these problems are related, let us consider an instance of the adding model. It consists of a graph G and list assignment S for G. If we set $G' = G$, $S'(v) = \bigcup_{u \in V} S(u)$ for $v \in V$ and

$$f'(v, x) = \begin{cases} 0 & \text{if } x \in S(v), \\ 1 & \text{otherwise,} \end{cases}$$

then it should become obvious that the cost of any solution of the cost list coloring problem equals $I_{ad}(G, S)$.
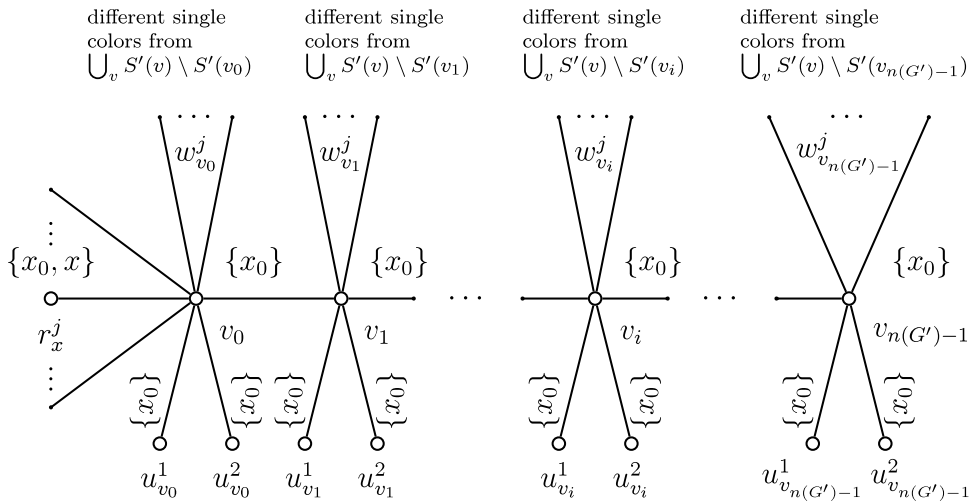
Fig. 1. The scheme of the reduction from the proof of Theorem 8.

The cost list coloring can be solved in linear time for trees by methods analogous to [9]. Moreover, it can be solved for graphs with any fixed γ in $O(n\Delta^{1+2\gamma})$ time (see [5] for even more general case of total coloring). Thus, we obtain the following corollary.

Corollary 6. I_{ad} can be computed in $O(n\Delta^{1+2\gamma})$ time for graphs with fixed γ. I_{ad} can be computed in linear time for trees. $\square$

Similar reasoning can be applied to the edge version of adding model and the edge cost list coloring problem (the edge equivalent of the cost coloring). The edge list coloring problem is solvable for graphs with any fixed γ in $O(n\Delta^{2+\gamma}\log Cn)$ time [5], where $C = \max_{e\in E, x\in S'(e)} f'(e, x)$. Thus, we obtain the following corollary.

Corollary 7. I'_{ad} can be computed in $O(n\Delta^{2+\gamma}\log n)$ time for graphs with fixed γ. I'_{ad} can be computed in $O(n\Delta^2 \log n)$ for trees. $\square$

As we have seen, the adding model and its edge version are polynomially solvable for trees and graphs with bounded cyclomatic number. The trading and exchange models are not that simple.

Theorem 8. *The problem given as follows is NP-complete.*

Instance: *Tree G, list assignment S for G and a positive integer t;*
Question: *Is $I_{tr}(G, S) \leq t$?*

Proof. It is known [4] that the following problem PTHC is NP-complete.

Instance: Path G', list assignment S' for G' and function $p' : \bigcup_{v\in V(G')} S'(v) \to \mathbb{N}$ satisfying $\sum_{v\in V(G')} p'(v) = |V(G')|$;
Question: Is there an S'-coloring of G' such that each color $x \in \bigcup_{v\in V(G')} S'(v)$ is used exactly $p'(x)$ times?

We will show the polynomial reduction from it to our problem.

We construct a pair (G, S) as follows. G will be a supergraph of G' obtained by executing two steps. At first we repeat the following procedure for every vertex $v \in V(G')$:

(1) add two new vertices u_v^1, u_v^2 to G;
(2) add $t(v) = |\bigcup_{u\in V(G')} S'(u) \setminus S'(v)|$ new vertices $w_v^1, w_v^2, \ldots, w_v^{t(v)}$ to G;
(3) make newly added vertices pendant by connecting them with v.

Let v_0 be an arbitrary vertex of G'. In the second step, for each color $x \in \bigcup_{v\in V(G')} S'(v)$ we add to G $p'(x)$ new vertices r_x^1, $r_x^2, \ldots, r_x^{p'(x)}$ and make them pendant by connecting with v_0.

Let $x_0 \notin \bigcup_{v\in V(G')} S'(v)$ be an integer. To receive S, execute the following procedure for every vertex $v \in V(G')$:

(1) set $S(v) = S(u_v^1) = S(u_v^2) = \{x_0\}$;
(2) set $S(w_v^i)$ for $i = 1, 2, \ldots, t(v)$ in such a way that $|S(w_v^i)| = 1$, $S(w_v^i) \neq S(w_v^j)$ for $i \neq j$ and $\bigcup_{i=1}^{t(v)} S(w_v^i) = \bigcup_{u\in V(G')} S'(u) \setminus S'(v)$.

Finally, set $S(r_x^i) = \{x_0, x\}$ for all vertices r_x^i and set $t = n(G')$. The scheme of this reduction is given in Fig. 1.

It is easy to see that G is a tree, S is a list assignment for G and $|V(G)| = \sum_{v \in V(G')}(L(S') - t(v) + 2) + 2n(G') = O(n(G')L(S'))$, so the reduction is polynomial. To complete the proof, it suffices to show that there exists such S'-coloring of G' that each color $x \in \bigcup_{v \in V(G')} S'(v)$ is used exactly $p'(x)$ times if and only if $I_{tr}(G, S) \leq t$.

($\Rightarrow$) Let us suppose that there exists a coloring c' which fulfills restrictions created by a list of allowed colors and a function p' for the path G'. We construct a coloring c of G in the following way:

(1) $c(v) = c'(v)$ for $v \in V(G')$;
(2) $c(u_v^1) = c(u_v^2) = x_0$ for $v \in V(G')$;
(3) $c(w_v^i) \in S(w_v^i)$ is defined unambiguously, because $|S(w_v^i)| = 1$;
(4) $c(r_x^i) = x_0$.

Coloring c is not a legal list coloring, but only vertices $v \in V(G')$ receive colors which are not on their lists. We can obtain a new list assignment for which the coloring is proper using $t = n(G')$ trades by moving the needed colors from lists of vertices r_x^i. Therefore $I_{tr}(G, S) \leq t$.

($\Leftarrow$) Note that, if $I_{tr}(G, S) \leq t$ then $I_{tr}(G, S) = t$, because for each $v \in V(G')$ 3-vertex path G_v induced in G by vertices v, u_v^1, u_v^2 is not list colorable. Thus we need at least one trade for every G_v. The only way to do so is to use one trade to extend $S(v)$ by a color that does not appear on lists of vertices $w_v^1, w_v^1, \ldots, w_v^{t(v)}$ that is a color from the list $S'(v)$. Thus, in each trade a vertex r_x^i must appear exactly once because only vertices r_x^i have lists of size two; the situation fulfills the restrictions created by function p'. Therefore, if $I_{tr}(G, S) \leq t$ then there is a solution to the PTHC problem. $\square$

We can see that the trees used in the proofs of Theorem 8 belongs to a special class of very simple trees called *caterpillars* (paths with some pendant vertices attached to it). Two natural open problems arise: is it possible in polynomial time

(1) to compute $I_{tr}(G, S)$, where G is a path;
(2) to verify if $I_{tr}(G, S)$ is finite, where G is a tree?

By Fact 1, to answer the last question, it suffices to show how to solve the following problem in polynomial time: given a tree G and positive integers $c_1, c_2, \ldots, c_s$; is there an s-coloring of G such that the number of vertices receiving color i is at most c_i? The solution is known if G is a path [3]:

(1) the s-coloring we are looking for exists if and only if $\sum_{i=1}^{s} c_i \geq n$ and $\sum_{i=1}^{s} c_i - \max_{1 \leq i \leq s} c_i \geq \lfloor n/2 \rfloor$;
(2) the s-coloring, if exists, can be obtained by coloring vertices of G (from one endpoint to the other) by color j such that obtained partial coloring is legal and $c_j - r_j$ is as big as possible, where r_j is the number of vertices currently colored by color j (see [3] for details).

Theorem 9. *The problem given as follows is NP-complete.*

Instance: *Tree G, list assignment S for G and a positive integer t;*
Question: *Is $I_{ex}(G, S) \leq t$?*

Proof. We sketch the proof only since it is similar to the previous one. We show that PTHC can be reduced to our problem. We construct (G, S) in a way described in the proof of Theorem 8 and, for every w_v^j, add to G a new vertex $\hat{w}_v^j$ such that w_v^j is its only neighbor and $S(\hat{w}_v^j) = \{x_0\}$. Moreover, $S(r_x^i) = \{x\}$ instead of $\{x_0, x\}$ for all r_x^i. It suffices to show that reduction does not change the answer.

($\Rightarrow$) We use the coloring constructed in the proof of Theorem 8 and extend it by setting $c(\hat{w}_v^j) = x_0$. The rest of this part is analogous to reasoning from the proof of Theorem 8.

($\Leftarrow$) Every G_v needs at least one color (different from x_0) from lists of the vertices from $V(G) \setminus \bigcup_u V(G_u)$, so there are exactly t exchanges. Vertices w_u^j cannot participate in exchanges since otherwise $c(w_u^j) = c(\hat{w}_u^j) = x_0$ in the final coloring c. Therefore t vertices r_x^i must appear in exchanges (each of them in the other exchange). Thus there is a solution to the PTHC problem. $\square$

Theorem 10. *The problem given as follows is NP-complete.*

Instance: *Tree (caterpillar) G, edge list assignment S for G and a positive integer t;*
Question: *Is $I'_{tr}(G, S) \leq t$?*

Proof. Careful analysis of the proof of NP-completeness of the PTHC problem (see the proof of Theorem 8) given in [4] shows that it remains NP-complete even if we assume that

$$\bigcup_{v \in V(G')} (S'(v) \setminus S'(w)) \cap S'(u_1) \cap S'(u_2) = \emptyset \tag{1}$$

for every vertex w of G' that has two neighbors u_1, u_2. We will show the polynomial reduction from the above subproblem of PTHC to our problem.

Without loss of generality we can assume that $V(G') = \{v'_1, v'_2, \ldots, v'_{n(G')}\}$, where v'_1 is one of the endpoints of G', v'_2 is the neighbor of v'_1, v'_3 is the only neighbor of v'_2 different from v'_1 and so on. Let $C = \bigcup_{v \in V(G')} S'(v)$ and $x_0, x_1 \notin C$ be integers. We construct the graph G as follows:
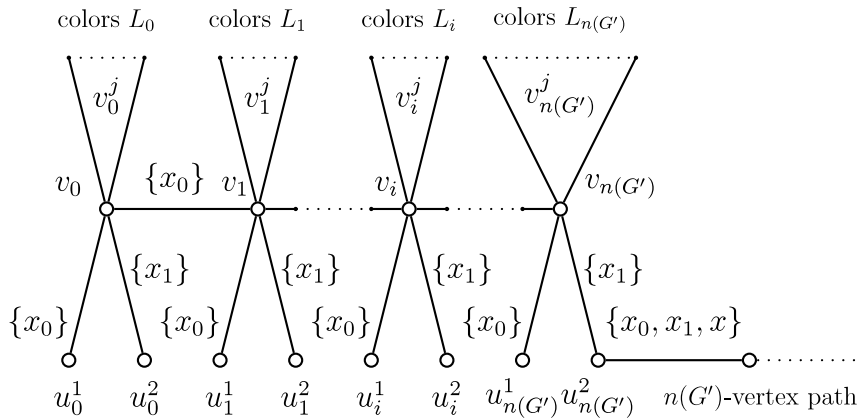
Fig. 2. A scheme of the reduction from the proof of Theorem 10.

(1) $V(G) = \{v_i, u_i^1, u_i^2, v_i^j : 0 \leq i \leq n(G') \wedge j \in L_i\} \cup \{r_x^i : x \in C \wedge 1 \leq i \leq p'(x)\}$, where $L_0 = C \setminus S'(v_1')$, $L_i = C \setminus (S'(v_{i-1}') \cup S'(v_i'))$ for $1 \leq i \leq n(G') - 1$ and $L_{n(G')} = C \setminus S'(v_{n(G')}')$.
(2) v_0 is a neighbor of v_1, v_1 is a neighbor of v_2 and so on;
(3) u_i^1, u_i^2 and v_i^j are pendant vertices connected with v_i;
(4) r_x^i constitutes an $n(G')$-vertex path with one endpoint connected to $u_{n(G')}^2$.

To receive S, we assign sets to edges of G in the following way:

(1) $S(\{v_i, v_{i+1}\}) = \{x_0\}$ for $0 \leq i \leq n(G') - 1$;
(2) $S(\{v_i, u_i^1\}) = \{x_0\}$ and $S(\{v_i, u_i^2\}) = \{x_1\}$ for $0 \leq i \leq n(G')$;
(3) $S(\{v_i, v_i^j\}) = \{j\}$ for $0 \leq i \leq n(G')$ and $j \in L_i$;
(4) for edges incident with r_x^i we assign lists of the form $\{x_0, x_1, x\}$ ($x \in C$) in such a way that every $x \in C$ is an element of exactly $p'(x)$ of those lists.

Finally, we set $t = n(G')$. The scheme of this reduction is given in Fig. 2.

Obviously, G is a tree (it is even a caterpillar) and S is an edge list assignment for G. The reduction is polynomial since $n(G) = 3n(G') + \sum_{i=0}^{n} |L_i| + \sum_{x \in C} p'(x) = O(n(G')l(S'))$. To complete the proof, it suffices to show that there exists an S'-coloring of G' such that each color $x \in \bigcup_{v \in V(G')} S'(v)$ is used exactly $p'(x)$ times if and only if $I'_{tr}(G, S) \leq t$.

($\Leftarrow$) Since $I'_{tr}(G, S) \leq t$ then after at most t trades we should be able to color edges of G with colors from their lists. Let c be such a coloring. We will show that function c' given by $c'(v_i') = c(\{v_{i-1}, v_i\})$ is a solution of PTHC.

Let H be the subgraph of G that consists of edges $\{v_i, v_{i+1}\}$ and $\{v_i, u_i^1\}$ (and all necessary vertices). H contains only one maximal matching that consists of edges $\{v_i, u_i^1\}$. Moreover, H has $2n(G') + 1$ edges and all its edges e satisfy $S(e) = \{x_0\}$. All edges that have color x_0 (as a value of c) must be a matching. This implies that edges $\{v_i, v_{i+1}\}$ are the only ones that enlarged their lists, so there are exactly t trades. There is only one group of edges that could reduce their lists—the edges with lists of cardinality at least 2, i.e. edges incident with vertices r_x^i, so the traded colors must be from C since $S(\{v_i, u_i^2\}) = \{x_1\}$. But there are exactly t such edges and each of them could participate in at most one trade (there is only one color different from x_0 and x_1 on their lists). This means that every such edge participated in one trade and, since there are exactly $p'(x)$ edges with color $x \in C$ on their lists, color x participated in $p'(x)$ trades. To complete the proof, it suffices to show that c' is an S'-coloring of G'. Since edges $\{v_{i-1}, v_i\}$ have neighbors with lists $\{x_0\}$, $\{x_1\}$ and $\{x\}$ for all $x \in L_i \cup L_{i-1}$, then they could receive a color from $C \setminus (L_{i-1} \cup L_i)$ only. But $S'(v_{i-2}') \cap S'(v_i') \subseteq S'(v_{i-1}')$ by (1) for $2 \leq i \leq n(G') - 1$ and $C \setminus (L_{i-1} \cup L_i) = (C \setminus L_{i-1}) \cap (C \setminus L_i) = (S'(v_{i-2}') \cup S'(v_{i-1}')) \cap (S'(v_{i-1}') \cup S'(v_i')) = S'(v_{i-1}')$. The same holds for $i = 1$ and $i = n(G')$.

($\Rightarrow$) Let c' be such S'-coloring of G' that color $x \in C$ is used exactly $p'(x)$ times. We construct edge coloring c of G as follows:

(1) $c(\{v_{i-1}, v_i\}) = c'(w_i)$ for $1 \leq i \leq n(G')$;
(2) $c(\{v_i, u_i^1\}) = x_0$, $c(\{v_i, u_i^2\}) = x_1$;
(3) $c(\{v_i, v_i^j\}) \in S(\{v_i, v_i^j\})$ is defined unambiguously, since $|S(\{v_i, v_i^j\})| = 1$;
(4) the remaining edges are colored alternately with x_0 and x_1.

Coloring c is not a legal edge list coloring, but only edges $\{v_{i-1}, v_i\}$ receive colors which are not on their lists. We can obtain a new list assignment for which the coloring is proper using $t = n(G')$ trades by moving the needed colors from lists of edges incident with vertices r_x^i. Therefore $I'_{tr}(G, S) \leq t$.   □

Theorem 11. *The problem given as follows is NP-complete.*

Instance: *Tree (caterpillar) G, edge list assignment S for G and a positive integer t;*
Question: *Is $I'_{ex}(G, S) \leq t$?*

Proof. We sketch the proof only since it is similar to the proof of Theorem 10. The reduction is almost the same—the only difference is that we use sets $\{x_1, x\}$ instead of $\{x_0, x_1, x\}$ for edges incident to r^i_x.

($\Leftarrow$) Similar reasoning to that from the analogous part of the proof of Theorem 10 leads to the conclusion that there are exactly t exchanges and each of them takes color different from x_0 from lists of edges outside H and adds it to lists of edges $\{v_i, v_{i+1}\}$. Edges that participate in exchanges and do not belong to H must be incident to vertices r^x_j, because all the other edges are incident with some $\{v_j, u^1_j\}$ and $c(\{v_j, u^1_j\}) = x_0$. Moreover, every exchange must add a color different from x_1 to an edge $\{v_i, v_{i+1}\}$, because $c(\{v_j, u^2_j\}) = x_1$. Therefore every edge $\{v_i, v_{i+1}\}$ exchanges color x_0 with a color from C with edge incident one of the vertices r^x_j. The rest of the proof proceeds like in the analogous part of the proof of Theorem 10.

($\Rightarrow$) The reasoning is the same as in the previous proof (the only difference is that we use exchanges instead of trades). $\square$

4. Bounded treewidth graphs

Let k be a positive integer. A graph G is called *k-tree* if and only if it satisfies one of the following conditions:

(1) G is a complete graph on k vertices;
(2) G has such a vertex v that the neighborhood of v induces a clique of size k in G and $G - v$ is a k-tree.

Subgraphs of k-trees are called *partial k-trees*. It is easy to see that trees and connected partial 1-trees are synonyms.

Partial k-trees have very convenient representation for dynamic programming approach, the so-called *tree-decomposition*. A tree-decomposition of a partial k-tree G is a pair (F, Γ), where F is a tree with set of vertices I and $\Gamma = \{X_i : X_i \subseteq V, i \in I\}$ satisfies the following conditions:

(1) $\bigcup_{i \in I} X_i = V$;
(2) for each edge $\{v, w\} \in E$, there is $i \in I$ such that $v, w \in X_i$;
(3) $X_i \cap X_m \subseteq X_j$ for each triple $i, j, m \in I$ such that j is on the path between i and m in F;
(4) for each $i \in I$, $|X_i| \leq k + 1$.

A *nice tree-decomposition* is a tree-decomposition with two new conditions added:

(1) F is a rooted (oriented) binary tree with root r;
(2) each node i of F is of one of the following four types:
  (a) *leaf node* of F (and then $|X_i| = 1$);
  (b) *introduce node* (having one child j and there exist $v \in V$ such that $X_i = X_j \cup \{v\}$);
  (c) *forget node* (having one child j and there exist $v \in V$ such that $X_j = X_i \cup \{v\}$);
  (d) *join node* (with two children j_1, j_2 and $X_i = X_{j_1} = X_{j_2}$).

Every partial k-tree has a nice tree-decomposition [1,7]. A nice tree-decomposition of a given partial k-tree can be found in linear time for any fixed k [1]. The number of nodes in F is also linear in terms of n. For each node $i \in I$ we can define a triple $G_i = (V_i, E_i, X_i)$, where (V_i, E_i) is the subgraph of G induced by

$$V_i = \{v : v \in X_j \text{ and } j \text{ is } i \text{ or a descendant of } i \text{ in } F\}.$$

Now we consider the problem of computing of I_{tr} for partial k-trees in the case when the number of colors is bounded.

Theorem 12. *Let $k, q \in \mathbb{N}$ be constants. We can compute $I_{tr}(G, S)$ for partial k-trees G and list assignments satisfying $l \leq q$ in a polynomial time.*

Proof. Without loss of generality we can assume that $\bigcup_{v \in V} S(v) = \{1, 2, \ldots, l\}$. For every node $i \in I$ we define W_i as the set of all sequences $c' : X_i \to \{1, 2, \ldots, l\}, n_1, n_2, \ldots, n_l, p_1, p_2, \ldots, p_l$ such that c' can be extended to a legal coloring $c : V_i \to \{1, 2, \ldots, l\}$ of G_i satisfying $n_x = |\{v \in c^{-1}(x) : x \in S(v)\}|$ and $p_x = |\{v \in c^{-1}(x) : x \notin S(v)\}|$ for all $1 \leq x \leq l$. Sets W_i will be computed starting from leaves of F and going up to the root. The procedure that computes W_i depends on the type of node i:

(a) i is a leaf node. Then $V_i = \{v\}$ for a certain vertex $v \in V$ and it is easy to see that W_i consists of l sequences and the xth sequence is:

- if $x \in S(v)$: $c', n_1, n_2, \ldots, n_l, 0, \ldots, 0$ where $c'(v) = x$ and $n_j = 1$ for $j = x$ and 0 otherwise;
- if $x \notin S(v)$: $c', 0, \ldots, 0, p_1, p_2, \ldots, p_l$ where $c'(v) = x$ and $p_j = 1$ for $j = x$ and 0 otherwise.

(b) i is an introduce node. Then all neighbors of v in G_i belong to X_i. For each sequence $d, n_1, n_2, \ldots, n_l, p_1, p_2, \ldots, p_l$ from W_j and color $x \in \{1, 2, \ldots, l\} \setminus d(X_j \cap N(G, v))$ we define c' as extension of d to X_i by setting $c'(v) = x$. W_i consists of sequences of the following form:

- if $x \in S(v)$: $c', n'_1, n'_2, \ldots, n'_l, p_1, \ldots, p_l$ where $n'_j = n_j + 1$ for $j = x$ and $n'_j = n_j$ otherwise;
- if $x \notin S(v)$: $c', n_1, \ldots, n_l, p'_1, p'_2, \ldots, p'_l$ where $p'_j = p_j + 1$ for $j = x$ and $p'_j = p_j$ otherwise.

(c) i is a join node. Then all common vertices of G_{j_1} and G_{j_2} belong to X_i and there are no edges that would connect $V_{j_1} \setminus X_i$ with $V_{j_2} \setminus X_i$. For each pair of sequences $d, n_1^1, n_2^1, \ldots, n_l^1, p_1^1, p_2^1, \ldots, p_l^1$ (from W_{j_1}) and $d, n_1^2, n_2^2, \ldots, n_l^2, p_1^2, p_2^2, \ldots, p_l^2$ (from W_{j_2}) we define $dn_x = |\{u \in X_i : d(u) = x \in S(u)\}|$ and $dp_x = |\{u \in X_i : d(u) = x \notin S(u)\}|$; W_i consists of sequences of the form $d, n_1^1 + n_1^2 - dn_1, \ldots, n_l^1 + n_l^2 - dn_l, p_1^1 + p_1^2 - dp_1, \ldots, p_l^1 + p_l^2 - dp_l$.

(d) i is a forget node. Then $W_i = \{(c'|_{V_i}, n_1, \ldots, n_l, p_1, \ldots, p_l) : (c', n_1, \ldots, n_l, , p_1, \ldots, p_l) \in W_j\}$.

Finally, when W_r is known, we can calculate I_{tr}. First, we remove from W_r all sequences $c', n_1, n_2, \ldots, n_l, p_1, p_2, \ldots, p_l$ for which there is x fulfilling $n_x + p_x > |\{v \in V : x \in S(v)\}|$. Second, we note that

$$
I_{tr}(G, S) = \begin{cases} +\infty & \text{if } W_r = \emptyset, \\ \min \left\{ \sum_{x=1}^{l} p_x : (c', n_1, \ldots, n_l, p_1, \ldots, p_l) \in W_r \right\} & \text{otherwise.} \end{cases}
$$

W_i has at most $l^{k+1} n^{2l}$ elements and it can be constructed in $O(n^{4l})$ time even in the most demanding case of a join node. Therefore the algorithm is polynomial and its complexity is $O(n^{4l+1})$. $\square$

As we have shown in the previous section, if we allow unbounded number of colors, the problem becomes NP-complete. The case of computing I_{ad} is different—it is polynomially solvable for partial k-trees without restriction on colors set's size.

Corollary 13. *Let $k \in \mathbb{N}$ be a constant. We can compute $I_{ad}(G, S)$ for partial k-trees G in a polynomial time.*

Proof. Computing of I_{ad} can be reduced to the cost list coloring (see Section 4 for details). The thesis follows now from the fact that this problem is polynomially solvable for partial k-trees [6]. $\square$

Now we prove a result similar to that of Corollary 7, but in the trade model.

Lemma 14. *The line graph of G is a partial $(\Delta + \gamma - 1)$-tree.*

Proof. Graph arising from partial k-tree H by adding a new vertex v (and some edges incident to it) is a partial $(k + 1)$-tree. Indeed, the tree-decomposition of the graph arises from tree-decomposition of H by adding v to all sets X_i.

Since every connected graph can be formed from a spanning tree by adding γ edges (which means that its line graph can be formed from line graph of a tree by adding γ vertices and some edges incident to them), it suffices to prove our claim for trees. But it is easily seen that line graph of a tree G is a partial $(\Delta - 1)$-tree since we can define its tree-decomposition by setting $F = G$, $I = V$ and $X_v = \{e \in E : v \text{ is incident with } e\}$ for all $v \in V$. $\square$

Corollary 15. *Let $t, q \in \mathbb{N}$ be constants. We can compute $I'_{tr}(G, S)$ for graphs G with $\gamma \leq t$ and list assignments satisfying $l \leq q$ in a polynomial time.*

Proof. If $\Delta > q$ then obviously $I'_{tr} = +\infty$. Otherwise the line graph of G is a partial $(q + t - 1)$-tree and the thesis follows by Theorem 12. $\square$

References

[1] H.L. Bodlaender, Treewidth: algorithmic techniques and results, Lecture Notes in Computer Science 1295 (1997) 19–36.
[2] C.J. Colbourn, Complexity of completing partial Latin squares, Discrete Applied Mathematics 8 (1984) 25–30.
[3] D. de Werra, On a multiconstrained model for chromatic scheduling, Discrete Mathematics 94 (1999) 171–180.
[4] M. Dror, G. Finke, S. Gravier, W. Kubiak, On the complexity of a restricted list-coloring problem, Discrete Mathematics 195 (1999) 103–109.
[5] K. Giaro, M. Kubale, Efficient list cost coloring of vertices and/or edges of bounded cyclicity graphs, Discussiones Mathematicae Graph Theory 29 (2009) 361–376.
[6] K. Jansen, The optimum cost chromatic partition problem, Lecture Notes in Computer Science 1203 (1997) 25–36.
[7] K. Jansen, P. Scheffler, Generalized coloring for tree-like graphs, Discrete Applied Mathematics 75 (1997) 135–155.
[8] J. Kratochvil, Z. Tuza, Algorithmic complexity of list colorings, Discrete Applied Mathematics 50 (1994) 297–302.
[9] L.G. Kroon, A. Sen, H. Deng, A. Roy, The optimal cost chromatic partition problem for trees and interval graphs, Lecture Notes in Computer Science 1197 (1997) 279–292.
[10] M. Kubale, Some results concerning the complexity of restricted colorings of graphs, Discrete Applied Mathematics 36 (1992) 35–46.
[11] N.V.R. Mahadev, F.S. Roberts, Consensus list colorings of graphs and physical mapping of DNA, DIMACS, Technical Report, 2002.
[12] S. Micali, V.V. Vazirani, An $O(\sqrt{|V|}\,|E|)$ algorithm for finding maximum matching in general graphs, in: 21st Annual Symposium on Foundations of Computer Science, 1980, pp. 17–27.
[13] Z. Tuza, Graph colorings with local constraints—a survey, Discussiones Mathematicae Graph Theory 17 (1997) 161–228.