Discrete Applied Mathematics 261 (2019) 186–192

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Equitable coloring of hypergraphs

Hanna Furmańczyk [a], Paweł Obszarski [b,*]

[a] Institute of Informatics, University of Gdańsk, Wita Stwosza 57, 80-309 Gdańsk, Poland
[b] Department of Algorithms and System Modeling, Gdańsk University of Technology, Narutowicza 11/12, 80-233 Gdańsk, Poland

ARTICLE INFO

Article history:
Received 31 March 2017
Received in revised form 9 January 2019
Accepted 15 January 2019
Available online 11 February 2019

Keywords:
Equitable coloring
Hypergraph
Linear hypertree
Feasible sequence

ABSTRACT

A hypergraph is equitably k-colorable if its vertices can be partitioned into k sets/color classes in such a way that monochromatic edges are avoided and the number of vertices in any two color classes differs by at most one. We prove that the problem of equitable 2-coloring of hypergraphs is NP-complete even for 3-uniform hyperstars. Finally, we apply the method of dynamic programming for designing a polynomial-time algorithm to equitably k-color linear hypertrees, where $k \geq 2$ is fixed.

1. Introduction

A *hypergraph* is a pair $H = (V, \mathbb{E})$, where V (or $V(H)$) is the n element set of vertices of H and $\mathbb{E}$ (or $\mathbb{E}(H)$) is a family of m non-empty subsets of V called edges or *hyperedges*. Let $[k]$ denote the set of integers $\{1, \dots, k\}$. A *k-coloring* of $H = (V, \mathbb{E})$ is a mapping $c : V \to [k]$ such that for each $e \in \mathbb{E}$ with $|e| \geq 2$ there exist $u, v \in e$ with $c(u) \neq c(v)$. That is, there is no monochromatic edge of size at least two. If every edge e is of cardinality r, then a hypergraph H is called *r-uniform*. Simple graphs are 2-uniform hypergraphs. The chromatic number $\chi(H)$ of H is the smallest k such that H has a k-coloring. Note that the set $\mathbb{E}$ is often defined as a multiset, thus allowing multiple hyperedges spanned on the same set of vertices. However, such hyperedges do not have influence on vertex coloring, so we omit them. For a similar reason, we may assume that each edge contains at least two vertices.

In this paper, we consider the problem of equitable coloring of hypergraphs. An *equitable vertex k-coloring* of a hypergraph $H = (V, \mathbb{E})$ is a partition of the vertex set V into subsets $V_1, \dots, V_k$ such that no V_j contains an edge e_i with $|e_i| \geq 2$, and $\| V_x| - |V_y \| \leq 1$, for each $x, y \in [k]$. The smallest k such that H admits an equitable vertex k-coloring is called the *equitable chromatic number* and is denoted by $\chi_=(H)$. In other words, an equitable k-coloring of an n-vertex graph is an assignment of colors to the vertices in such a way that monochromatic edges are avoided and each color class is of size $\lceil n/k \rceil$ or $\lfloor n/k \rfloor$. Examples of such colorings are given in Fig. 1(b) and Fig. 3. The above definition can be viewed as an extension of equitable vertex coloring for simple graphs.

The notion of equitable colorability of simple graphs was introduced by Meyer in 1973 [17]. This definition was generalized to hypergraphs by Berge and Sterboul [2] in 1977. To the best of our knowledge, only a few papers concerning equitable coloring of hypergraphs have been published since then (see [20–22]).

It is worth pointing out that equitable k-colorability of a graph or hypergraph G does not imply that G is equitably $(k+1)$-colorable. A counterexample is the complete bipartite graph $K_{3,3}$ which can be equitably colored with two colors, but not with three. Examples of non-equitably k-colorable hypergraphs are given in [2].

* Corresponding author.
E-mail addresses: hanna@inf.ug.edu.pl (H. Furmańczyk), pawel.obszarski@pg.gda.pl (P. Obszarski).

https://doi.org/10.1016/j.dam.2019.01.016
0166-218X/

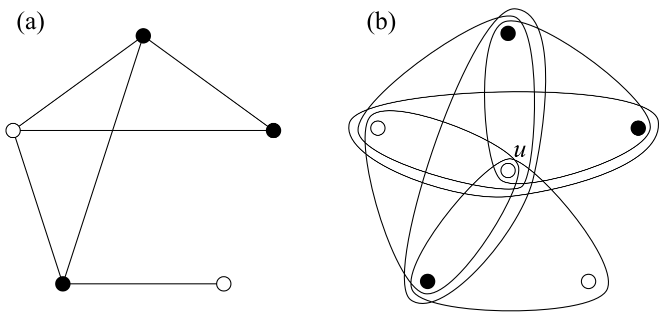

Fig. 1. (a) Simple graph G on 5 vertices with independent set of size 2 (white vertices). (b) 3-uniform hypergraph H constructed from G and its equitable 2-coloring (white and black vertices).

The model of equitable coloring of simple graphs has received considerable attention due to its multiple practical applications. Every time when we have to divide a system with binary conflict relations into equal or almost equal conflict-free subsystems we can model this situation by means of equitable graph coloring. In particular, one motivation for equitable coloring suggested by Meyer [17] concerns scheduling problems. In this application, the vertices of a graph represent a collection of (unit execution times) tasks to be performed and an edge connects two tasks that should not be executed at the same time. A coloring of this graph represents a partition of tasks into subsets that may be performed simultaneously. Hyperedges of size greater than two could be understood as a constraint that forbids scheduling all the vertices of e at the same time. Some applications of equitable coloring in scheduling are also discussed in [5,9,10].

Hypergraphs in general are very useful in real-life problems modeling, for example in chemistry, telecommunications, and many other fields of science and engineering [3]. They have also applications in image representation [14]. Thus, generalization of equitable coloring of simple graphs to hypergraphs seems to be justified.

Let us recall some basic properties concerning hypergraphs.

Definition 1.1. We say that a hypergraph H has an underlying (host) *graph G (spanned on the same set of vertices) if each hyperedge of H induces a connected subgraph in G. Furthermore, it is assumed that for each edge e_G in G there exists a hyperedge e_H in H such that $e_G \subseteq e_H$.*

Note that for each hypergraph its host graph can be found by replacing each hyperedge e with a clique of size $|e|$. However, such graph will be a maximal host, while we often search for a sparse one. That is why other techniques of host searching need to be applied.

Definition 1.2. A hypergraph that has an underlying tree is a hypertree.

By analogy *hyperstars*, *hypercycles*, and *hyperpaths* are defined.

Definition 1.3. A hypergraph is linear *if each pair of edges has at most one vertex in common.*

Definition 1.4 (*[1]*)**.** *A* subhypergraph H_A *of a hypergraph* $H = (V, \mathbb{E})$ *induced by a vertex subset* $A \subseteq V$ *is defined as* $H_A = (A, \{E_i \cap A : E_i \in \mathbb{E}, E_i \cap A \neq \emptyset\}$.

The remainder of the paper is organized as follows. In the next section we establish the complexity status of equitable coloring of hypergraphs, while in Section 3 we give a polynomial-time algorithm for equitable k-coloring of linear hypertrees, where $k \geq 2$ is fixed.

2. Complexity status

In general, the problem of equitable coloring of simple graphs with the minimum number of colors is NP-hard and remains so for corona products of graphs [7]. We are now interested in the computational complexity of the following problem:

Equitable k-Coloring of Hypergraph, ECH(H, k)
Instance: A hypergraph H on n vertices and an integer k.
Question: Does there exist an equitable vertex k-coloring of H?

and its subproblem:

Equitable Coloring of 3-uniform Hypergraph, ECH$_3(H, k)$
Instance: A 3-uniform hypergraph H on n vertices and an integer k.
Question: Does there exist an equitable vertex k-coloring of H?

It is known [16] that 2-coloring of a hypergraph is NP-complete. However, this does not seem to directly imply the hardness of ECH$(H, 2)$. In most cases equitable coloring is not easier than ordinary coloring, but there are also known graphs such that the problem of ordinary coloring is NP-complete while the equitable coloring of these graphs is solvable in polynomial time. Let us define the *join* $G_1 + G_2$ of graphs G_1 and G_2 with disjoint vertex sets V_1 and V_2 and edge sets X_1 and X_2 as the graph union $G_1 \cup G_2$ together with all the edges joining V_1 and V_2. Let N_n be a graph on n vertices without edges, i.e. the graph $(\{v_1, \ldots, v_n\}, \emptyset)$, while let K_n denote a clique on n vertices.

Vertex k-Coloring
Instance: A graph G on n vertices and an integer k.
Question: Does there exist a proper vertex k-coloring of G?

Fact 2.1. Vertex k-Coloring *is NP-complete when restricted to graphs of the form $G + N_1$, while $\chi_=(G + N_1)$ can be found in polynomial time.*

Proof. It is known that Vertex k-Coloring of a general graph is NP-complete [15], for $k \geq 3$, and remains so also for graphs of type $G + N_1$ and $k \geq 4$. Let us consider now equitable vertex coloring of such a graph $G + N_1$. The color assigned to the vertex $v \in V(N_1)$ cannot be used to color any other vertex in G. Since coloring must be equitable, every other color in this coloring can be used at most twice. Thus, we need to partition the vertex set of G into minimum number of color classes of size at most 2. This problem is equivalent to finding maximal matching in the complement of G, which may be done in polynomial time [18,23]. □

In this section we show that ECH$_3(H, 2)$ is NP-complete even for hyperstars. Note that ECH$(H, 2)$ restricted to hyperpaths is decidable in polynomial time. Indeed, every hyperpath on $n \geq 2$ vertices is equitably 2-colorable and such a coloring corresponds to an equitable 2-coloring of the underlying path. A similar result holds for hypercycles and we leave the details to the reader.

We will show that the ECH$_3(H, k)$ is NP-complete. The proof is based on a reduction from a variant of the following well-known NP-complete problem [11]:

Independent Set, IS(G, l)
Instance: A simple graph G on n vertices and an integer l.
Question: Does G have an independent set of size at least l?

Fact 2.2. *IS$(G, \lceil n/2 \rceil - 1)$ is NP-complete.*

Proof. Note that if we have NP-complete problem IS(G, l) with an arbitrary G and l there are three cases $l = \lceil n/2 \rceil - 1$, $l < \lceil n/2 \rceil - 1$, or $l > \lceil n/2 \rceil - 1$. If $l = \lceil n/2 \rceil - 1$, then there is no need of transformation. If $l < \lceil n/2 \rceil - 1$ we can modify G by adding $n - 2l - 2$ isolated vertices. In this case we get the new graph $G' = G \cup N_{n-2l-2}$. Then, finding IS$(G', \lceil n'/2 \rceil - 1)$ is equivalent to finding IS(G, l). Similarly, if $l > \lceil n/2 \rceil - 1$ then graph G is joined with K_{2l+2-n} that leads us to graph $G' = G + K_{2l+2-n}$, and again IS$(G', \lceil n'/2 \rceil - 1)$ is equivalent to IS(G, l). □

Now we are ready to prove

Theorem 2.3. *The problem ECH$_3(H, 2)$ is NP-complete even if H is a hyperstar.*

Proof. Membership in NP is obvious. As already mentioned, we will reduce from IS$(G, \lceil n/2 \rceil - 1)$ (cf. Fact 2.2).

Given a graph G on n_G vertices, we construct an instance of ECH$_3(H, 2)$ as follows. We add a dominating vertex u to G and we let H to be the hypergraph with hyperedges $\{x, y, u\}$, for $\{x, y\} \subseteq E(G)$. Clearly, H is 3-uniform hyperstar and has $n_H = n_G + 1$ vertices. An example of such reduction is given in Fig. 1.

If H can be equitably 2-colorable, then let V_1 and V_2 be the appropriate color classes with $u \in V_1$ and $|V_1| = \lfloor n_H/2 \rfloor$. Clearly, the vertices in $V_1 \backslash \{u\}$ form an independent set in G of size $\lfloor n_H/2 \rfloor - 1 = \lceil n_G/2 \rceil - 1$.

Conversely, if G has an independent set I_G of size $\lceil n_G/2 \rceil - 1$, then vertices of $I_G \cup \{u\}$ will form the first color class and the remaining $\lfloor n_G/2 \rfloor + 1$ vertices the second one. It is easy to see that this 2-coloring of H is proper and equitable. □

Note that ordinary coloring of 3-uniform hyperstar H is solvable in linear time. Furthermore, $\chi(H) = 2$ as long as $|V(H)| > 2$. This means that hyperstars form an exemplary hypergraph class for which proper coloring is polynomially solvable while equitable coloring is hard. A hierarchy of bipartite graphs and hypergraphs with the corresponding complexity status for ordinary coloring and equitable coloring is given in Fig. 2.

3. Equitable coloring of linear hypertrees

Since our 2-coloring problem is NP-complete, we propose a polynomial-time algorithm for more specified classes of hypergraphs, namely linear hypertrees. We apply the method of dynamic programming, used for example in [6,12], for determining the set of counters for every proper coloring of H with colors $\{1, 2\}$.

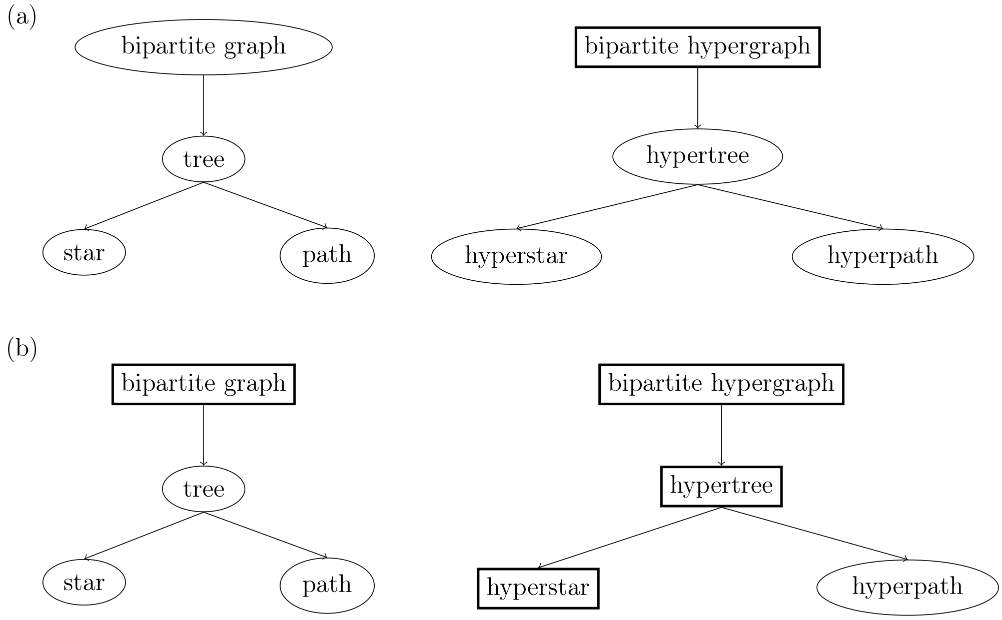

Fig. 2. Hierarchy of bipartite graph and bipartite hypergraphs with the computational complexity status of the problem of (a) ordinary coloring and (b) equitable coloring: polynomial cases in circles, NP-complete cases in rectangles.

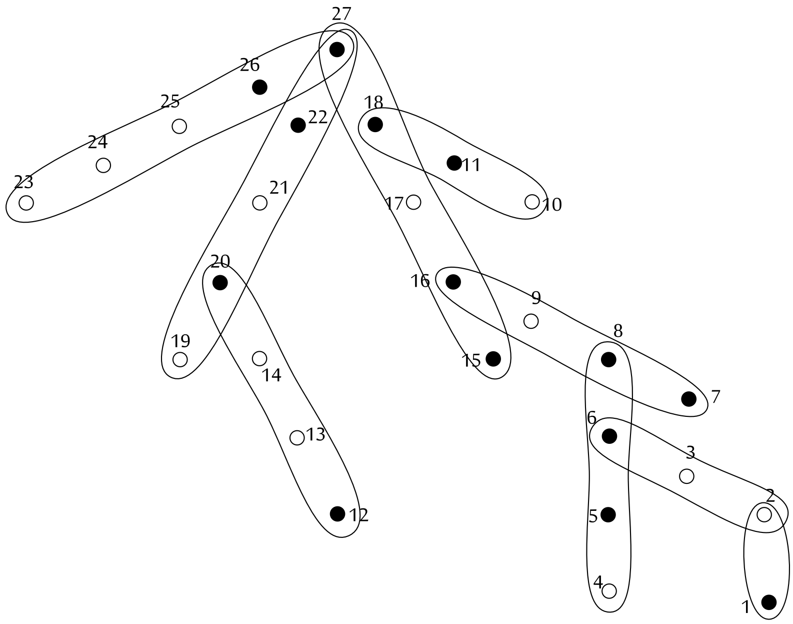

Fig. 3. An example of a linear hypertree H together with its vertex numbering achieved due to aBFS and with its exemplary equitable 2-coloring for a counter (14, 13) (black and white vertices).

Definition 3.1. For a given coloring of an n-vertex hypertree H with colors $\{1, \ldots, k\}$ the counter *of this coloring is a sequence* $(n_1, \ldots, n_k)$, $n_1 + \cdots + n_k = n$, *where* n_i *is the number of vertices colored with color i. A counter is* equitable *if* $\max_{i,j\in[k]} |n_i - n_j| \leq 1$.

For a given coloring, a counter is sometimes called a *feasible sequence*, in the literature. In further consideration we will need a special operation on sets of counters X and Y, defined as follows:

$$X + Y = \{(x_1 + y_1, \ldots, x_k + y_k) : (x_1, \ldots, x_k) \in X \wedge (y_1, \ldots, y_k) \in Y\},$$

where $\emptyset + X = \emptyset$ for any set X. Of course, $X + Y = Y + X$.

Given a linear hypertree $H = (V, \mathbb{E})$ we need to determine an underlying tree $T = (V, \mathbb{E}')$ with a distinguished root $r \in V$ such that each hyperedge $e \in \mathbb{E}$ lies on a path from the root of the underlying tree to its leaf. Furthermore, there is a need for preparing sequence of vertices for dynamic programing. For this purpose we introduce adapted Breadth First Search algorithm, defined as follows.

algorithm aBFS(H);
begin
  $T := (V, \emptyset)$;
  let Q be an empty queue;
  $i := |V(H)|$;
  choose arbitrary vertex that will represent root;
  add root to the queue Q;
  while Q is non-empty **do begin**
    take top vertex v from Q;
    assign index i to v;
    decrement i by 1;
    while there is unmarked edge e incident to v **do begin**
      take unmarked e incident to v;
      mark e;
      add to queue all vertices incident to e apart from vertex v;
      add to T a path starting with v with vertices in the same order as added to Q
    end
  end
end;

Mind that there are some other more complex algorithms finding underlying tree of linear hypertees and general hypertree as well. One based on maximal spanning trees algorithm was proposed in [19], while another may be inspired by chordal graphs recognition.

Theorem 3.2. *There exists a polynomial-time algorithm* ($O(n^2)$) *for checking whether a given linear hypertree H on n vertices can be equitably colored with* 2 *colors.*

Proof. Let us notice that $\chi_=(H) = 1$ if and only if $n = 1$. Thus we can assume that $n \geq 2$.

We will describe a procedure which checks if a given linear hypertree H satisfies $\chi_=(H) = 2$ and produces an appropriate equitable 2-coloring, as long as it is feasible. Because the number of colorings may be exponential, we consider only coloring counters. Thus, we will determine a set of counters for every proper coloring with colors $\{1, 2\}$ for some kind of subhypertrees of H. It is possible to extend this algorithm to one that finds equitable coloring. It is enough to keep with each counter an exemplary coloring.

Let us assume that the vertices $\{v_1, \dots, v_n\}$ are numbered as in the order achieved in aBFS (cf. Fig. 3).

For each $i \in [n]$ we define a hypertree D_i as a connected component of a subhypertree $H_{\{v_1,\dots,v_i\}}$ containing vertex v_i (cf. Definition 1.4).

Let f_i denote an edge in hypertree H containing v_i such that $f_i \notin \mathbb{E}(D_i)$ and there is an edge $y_i \in \mathbb{E}(D_i)$ such that $y_i \subset f_i$. Mind that $|y_i| \geq 1$ or f_i may not exist.

For every $i \in [n]$ we will recursively define sets:

- $Col^*(v_i)^k$ – a set of counters of colorings of D_i with colors $\{1, 2\}$ such that coloring restricted to $D_i - y_i$ is proper, a vertex v_i is colored with k, $k \in \{1, 2\}$, and the edge y_i is monochromatic. If f_i does not exist, $Col^*(v_i)^k = \emptyset$.
- $Col(v_i)^k$ – a set of counters of every proper coloring of hypergraph D_i with $\{1, 2\}$, in which a vertex v_i is colored with k, $k \in \{1, 2\}$.

We will consider some cases, depending on the situation in D_i.

Case 1. D_i contains only one vertex, v_i, and hence only one edge $\mathbb{E}(D_i) = \{\{v_i\}\}$.

Such case appears for example for D_1, D_{10}, or D_{23} for the hypergraph H from Fig. 3.

The way of determining counter sets is as follows:

$$Col^*(v_i)^1 := \{(1, 0)\}$$

$$Col^*(v_i)^2 := \{(0, 1)\}$$

$$Col(v_i)^1 = Col(v_i)^2 := \emptyset.$$

Case 2. D_i contains more than one vertex and there is only one edge in D_i containing vertex v_i.

Such case appears for example for D_5, D_{14}, or D_{21} for the hypergraph H from Fig. 3.

Let $\{e\}$ be the edge containing v_i. Let j be the biggest index among vertices in e, $j < i$. Then

$$Col(v_i)^1 := (Col(v_j)^1 \cup Col(v_j)^2 \cup Col^*(v_j)^2) + \{(1, 0)\},$$

$$Col(v_i)^2 := (Col(v_j)^1 \cup Col(v_j)^2 \cup Col^*(v_j)^1) + \{(0, 1)\}.$$

If f_i exists, then:

$$Col^*(v_i)^1 := Col^*(v_j)^1 + \{(1, 0)\},$$

$$Col^*(v_i)^2 := Col^*(v_j)^2 + \{(0, 1)\},$$

else

$$Col^*(v_i)^1 = Col^*(v_i)^2 := \emptyset.$$

Case 3. D_i contains more than one vertex and there are at least two edges in D_i containing vertex v_i. Moreover, f_i exists and $|y_i| = 1$.

Such case appears for example for D_2 for the hypergraph H from Fig. 3.

We consider graph $D_i - y_i$ as a subhypergraph D_i in the further part. Note that the vertex v_i is the first vertex of f_i and the last vertex in all edges of D_i that is being colored now. Due to the second fact, we will determine $Col(v_i)^k$, $k \in \{1, 2\}$. But finally, due to the definition of counter sets, we will determine only $Col^*(v_i)^k$, $k \in \{1, 2\}$ for the vertex v_i. To achieve this, we number the edges of D_i containing v_i randomly: $\{y^i_1, \ldots, y^i_{\deg_{D_i}(v_i)}\}$, where $\deg_{D_i}(v_i)$ denotes the number of edges in D_i to which vertex v_i belongs to, i.e. its degree. We will count counter sets for v_i by considering consecutive edges of D_i incident to v_i. First, we consider y^i_1 and treat it as the only edge in D_i containing v_i. We count counter sets as in Case 2. Now we consider y^i_l, for $l = 2, \ldots, \deg_{D_i}(v_i)$. Let j be the biggest index among vertices in the edge y^i_l, $j < i$. Then

$$Col(v_i)^1 := (Col(v_j)^1 \cup Col(v_j)^2 \cup Col^*(v_j)^2) + Col(v_i)^1,$$

$$Col(v_i)^2 := (Col(v_j)^1 \cup Col(v_j)^2 \cup Col^*(v_j)^1) + Col(v_i)^2.$$

Finally, after $\deg_{D_i}(v_i)$ iterations, since v_i is the first vertex in f_i thus f_i is monochromatic, do:

$$Col^*(v_i)^1 := Col(v_i)^1,$$

$$Col^*(v_i)^2 := Col(v_i)^2,$$

$$Col(v_i)^1 := \emptyset,$$

$$Col(v_i)^2 := \emptyset.$$

Case 4. D_i contains more than one vertex and there are at least two edges in D_i containing vertex v_i. Moreover, f_i does not exist or $|y_i| \geq 2$.

Such case appears for example for D_6, D_8, or D_{27} for the hypergraph H from Fig. 3.

If f_i exists, then let $y^i_1 := y_i$, otherwise y^i_1 is a random edge of D_i containing v_i. We number the remaining edges of D_i containing v_i in a random way: $\{y^i_1, \ldots, y^i_{\deg_{D_i}(v_i)}\}$. We will count counter sets for v_i by considering consecutive edges from $\{y^i_1, \ldots, y^i_{\deg_{D_i}(v_i)}\}$. First, we consider y^i_1 and treat it as the only edge in D_i containing v_i. We count counter sets as in Case 2. Now we consider y^i_l, for $l = 2, \ldots, \deg_{D_i}(v_i)$. Let j be the biggest index among vertices in the edge y^i_l, $j < i$. Then

$$Col(v_i)^1 := (Col(v_j)^1 \cup Col(v_j)^2 \cup Col^*(v_j)^2) + Col(v_i)^1,$$

$$Col(v_i)^2 := (Col(v_j)^1 \cup Col(v_j)^2 \cup Col^*(v_j)^1) + Col(v_i)^2,$$

$$Col^*(v_i)^1 := (Col(v_j)^1 \cup Col(v_j)^2 \cup Col^*(v_j)^2) + Col^*(v_i)^1,$$

$$Col^*(v_i)^2 := (Col(v_j)^1 \cup Col(v_j)^2 \cup Col^*(v_j)^1) + Col^*(v_i)^2.$$

Finally, after $\deg_{D_i}(v_i)$ iterations, a counter set for v_i is determined.

We are interested in a counter set for v_n. The hypertree H is equitably 2-colorable if and only if there exists a counter $(\lfloor n/2 \rfloor, \lceil n/2 \rceil)$ in $Col(v_n)^1 \cup Col(v_n)^2$.

Note that each time two subhypertrees are joined, new counters for a local root vertex v_i are calculated which requires some operations. In order to count the complexity of the algorithm we estimate the number of such operations. Let us assume that the first subhypertree is spanned on a vertex set W_1 and the second one on a set W_2. Note that the worst case number of operations $4(|W_1|+1) \cdot 4(|W_2|+1)$ is proportional to the number of pairs in $W_1 \times W_2$. As $W_1 \cap W_2 = \emptyset$ and the pairs in various joins are always different, the total number of operations is proportional to the total number of pairs in $V \times V$. Thus, the complexity may be estimated by $O(n^2)$. □

Note that our algorithm can be easily modified to calculate a counter set for colorings of a linear hypertree for bigger k.

Corollary 3.3. *Let $k \geq 2$ be fixed integer. Then there exists a polynomial-time algorithm for checking whether a given linear hypertree H on n vertices can be equitably colored with k colors.* □

4. Conclusion

In the paper the complexity status of equitable coloring of hypergraphs was established and a polynomial-time algorithm for equitable k-coloring of linear hypertrees was proposed. It is worth pointing out that the algorithm from the proof of Theorem 3.2 leads to determining all feasible sequences/counters for vertex k-coloring of such hypergraphs. Thus, our method can be applied in any other model of linear hypertrees coloring where the model definition is connected with cardinality constraints. An example of such graph/hypergraph coloring model is, next to equitable coloring, bounded coloring [13].

The subject has a great potential for development. There is still space for tightening the gap between the known polynomial and NP-complete subproblems. Moreover, it would be interesting to determine other hypergraph classes for which ordinary coloring and equitable coloring have different complexities. Notice also that, to the best of our knowledge, equitable coloring of mixed hypergraphs, as a structure of many potential applications (cf. [8]), has not been studied. Finally, research on generalizing the known conjectures concerning equitable coloring of simple graphs to hypergraphs would be desirable. We think for example about Coloring Conjecture (ECC) [17] or Equitable Δ-Coloring Conjecture (EΔCC) [4].

References

[1] C. Berge, Graphs and hypergraphs, North-Holland, 1989.
[2] C. Berge, F. Sterboul, Equipartite colorings in graphs and hypergraphs, J. Comb. Theory Ser. B 22 (1977) 97–113.
[3] A. Bretto, Applications of hypergraph theory: a brief overview, in: Hypergraph Theory: An Introduction, Springer International Publishing, Heidelberg, 2013, pp. 111–116.
[4] B.L. Chen, K.W. Lih, Equitable coloring of trees, J. Combin. Theory Ser. B 61 (1994) 83–87.
[5] H. Furmańczyk, Equitable coloring of graph products, Opuscula Math. 26 (2006) 31–44.
[6] H. Furmańczyk, K. Giaro, M. Kubale, Equitable coloring of cacti and edge-cacti in polynomial time, Int. J. Pure Appl. Math. 27 (3) (2006) 379–392.
[7] H. Furmańczyk, K. Kaliraj, M. Kubale, J.V. Vivin, Equitable coloring of corona products of graphs, Adv. Appl. Discrete Math. 11 (2013) 103–120.
[8] H. Furmańczyk, A. Kosowski, P. Żyliński, Scheduling with precendence constraints. mixed graph coloring in series-parallel graphs, in: LNCS, vol. 4967, 2007, pp. 1001–1008.
[9] H. Furmańczyk, M. Kubale, Scheduling of unit-length jobs with cubic incompatibility graphs on three uniform machines, Discrete Appl. Math. http://dx.doi.org/10.1016/j.dam.2016.01.036 (in press).
[10] H. Furmańczyk, M. Kubale, Equitable and semi-equitable coloring of cubic graphs and its application in batch scheduling, Arch. Control Sci. 25 (1) (2015) 109–116.
[11] M.R. Garey, D.S. Johnson, L. Stockmeyer, Some simplified NP-complete problems, Theoret. Comput. Sci. 1 (1976) 237–267.
[12] K. Giaro, M. Kubale, P. Obszarski, A graph coloring approach to scheduling of multiprocessor tasks on dedicated machines with availability constraints, Discrete Appl. Math 157 (17) (2009) 3625–3630.
[13] P. Hansen, A. Hertz, J. Kuplinsky, Bounded vertex colorings of graphs, Discrete Math. 111 (1993) 305–312.
[14] D. Ibaroudene, R. Acharya, Linear hypertree for multi-dimensional image representation, Inform. Sci. 68 (1993) 123–154.
[15] R.M. Karp, Reducibility among combinatorial problems, in: Complexity of computer computations, Springer, Boston, MA, 1972.
[16] L. Lovasz, Coverings and colorings of hypergraphs, Util. Math. 8 (1973) 3–12.
[17] W. Meyer, Equitable coloring, Amer. Math. Monthly 80 (1973) 920–922.
[18] S. Micali, V.V. Vazirani, An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs, in: Twenty First Annual IEEE Symp. on the Foundations of Computer Science, 1980, pp. 17–27.
[19] P. Obszarski, J. Dąbrowski, Hipergrafowy model szeregowania w rozrzedzonych systemach zadań wieloprocesorowych, Zesz. Nauk. WETI PG 10 (2006) 499–506.
[20] D.A. Shabanov, Equitable two-colorings of uniform hypergraphs, Eur. J. Comb. 43 (2015) 185–203.
[21] I. Shirgazina, Equitable colorings of non-uniform simple hypergraphs, Electron. Notes Discrete Math. 49 (2015) 699–703.
[22] I. Shirgazina, Equitable colorings of nonuniform hypergraphs, Math. Notes 99 (2016) 444–456.
[23] V.V. Vazirani, A proof of the MV matching algorithm, 2014. http://www.cc.gatech.edu/~vazirani/new-proof.pdf (unpublished manuscript).