PROCEEDINGS OF THE YEREVAN STATE UNIVERSITY

Physical and Mathematical Sciences 2021, **55**(2), p. 113–122

Mathematics

INTERVAL EDGE-COLORINGS OF TREES WITH RESTRICTIONS ON THE EDGES

A. K. SAHAKYAN [1*], R. R. KAMALIAN [2**]

1 *Chair of Discrete Mathematics and Theoretical Informatics, YSU, Armenia*
2 *Chair of Information Technologies and Applied Mathematics, EUA, Armenia*

An edge-coloring of a graph G with consecutive integers $c_1,\ldots,c_t$ is called an interval t-coloring, if all colors are used, and the colors of edges incident to any vertex of G are distinct and form an interval of integers. A graph G is interval colorable, if it has an interval t-coloring for some positive integer t. In this paper, we consider the case, where there are restrictions on the edges of the tree and provide a polynomial algorithm for checking interval colorability that satisfies those restrictions.

https://doi.org/10.46991/PYSU:A/2021.55.2.113

MSC2010: Primary: 05C15; Secondary: 68Q25.

Keywords: trees, interval t-coloring, interval edge-coloring, restrictions on edges, dynamic programming, bipartite matching.

Introduction. All graphs considered in this paper are undirected (unless explicitly said), finite, and have no loops or multiple edges. For an undirected graph G, let $V(G)$ and $E(G)$ denote the sets of vertices and edges of G, respectively. The degree of a vertex $v \in V(G)$ is denoted by $d_G(v)$. Let T be a tree (a connected undirected acyclic graph). Let $d_T(u,v)$ be the length of the path from the vertex u to the vertex v. Since T is a tree, there is exactly one path connecting two vertices.

For a directed graph $\overrightarrow{G}$ if there is an edge from a vertex u to a vertex v we will denote it as $u \to v$. The graph G is called the underlying graph of a directed graph $\overrightarrow{G}$ if $V(G) = V(\overrightarrow{G})$ and $E(G) = \{(u,v) | \text{iff } u \to v \text{ or } v \to u\}$ (between any pair of vertices u and v, if the directed graph has an edge $u \to v$ or an edge $v \to u$, the underlying graph includes the edge (u,v)).

For a tree T and a vertex r, let T_r be the directed graph whose underlying graph is T and in T_r each edge is directed in such a way that for each vertex $v \in T_r$ there is a path in T_r from r to v. We will say that T_r is a rooted tree with a root r. Fig. 1 illustrates the rooted tree T_{v_1} with the root v_1.

[*] E-mail: sahakyan.albert96@gmail.com
[**] E-mail: rrkamalian@yahoo.com

Let T_r be a rooted tree, the depth of a vertex v, denoted by $h(v)$, is the length of the unique path from the root r to the vertex v. A vertex u is said to be the parent of the vertex v, denoted by $p(v)$, if $u \to v$. In that case the vertex v is said to be a child of the vertex u. The children of a vertex $v \in V(T_r)$ are the set $W \subseteq V(T_r)$ of all vertices w of the tree T_r satisfying the condition $v \to w$. A vertex having no children is said to be a leaf vertex. Non-root two vertices $a, b \in V(T_r)$ are said to be sibling vertices if $p(a) = p(b)$. For a vertex v let $S(v)$ be the subtree induced by all the vertices w such that there is a path from v to w in T_r [1]. For a non-root vertex v, let $U(v)$ be the subtree in T_r induced by the subset $V(S(v)) \cup \{p(v)\}$ of its vertices. Fig. 2 illustrates the subtree $U(v_6)$: $V(U(v_6)) = \{v_1, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}\}$ and the edges are all the edges between these vertices.

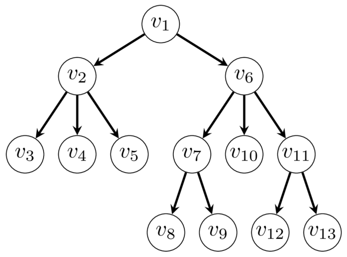

Fig. 1. A rooted tree T_{v_1} with the root v_1.

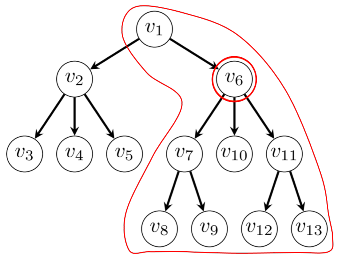

Fig. 2. The subtree $U(v_6)$ in T_{v_1}.

An edge-coloring of a graph G is an assignment of colors to the edges of the graph so that no two adjacent edges have the same color. An edge-coloring of a graph G with colors $1, \ldots, t$ is an interval t-coloring if all colors are used, and the colors of edges incident to each vertex of G form an interval of integers. A graph G is interval colorable, if it has an interval t-coloring for some positive integer t. The set of all interval colorable graphs is denoted by $\mathfrak{N}$. The concept of an interval edge-coloring of a graph was introduced by Asratian and Kamalian [2]. This means that an interval t-coloring is a function $\alpha : E \to \{1, \ldots, t\}$ such that for each edge e the color $\alpha(e)$ of that edge is an integer from 1 to t, for each color from 1 to t there is an edge with that color and for each vertex v all the edges incident to v have different colors forming an interval of integers. For an interval coloring α and a vertex v, the set of all the colors of the incident edges of v is called the spectrum of that vertex in α and is denoted by $S_\alpha(v)$. The smallest and the largest numbers in $S_\alpha(v)$ are denoted by $\underline{S_\alpha}(v)$ and $\overline{S_\alpha}(v)$, respectively.

Interval edge-colorings have been intensively studied in different papers. In [3] it was shown that every tree is from $\mathfrak{N}$. Lower and upper bounds on the number of colors in interval edge-colorings were provided in [4] and the bounds were improved for different graphs: planar graphs [5], r-regular graphs with at least $2r+2$ vertices [6], cycles, trees, complete bipartite graphs [3], n-dimensional cubes [7, 8], complete graphs [9, 10], Harary graphs [11], complete k-parite graphs [12].

In this paper, we consider the case, where there are restrictions on the edges and the problem is to find an interval t-coloring that meets those restrictions. Colorings with restrictions (also known as list colorings and list edge-colorings) were first studied in 1970s in the independent papers [13] and [14]. In [15] a solution for the simplified version of this problem was provided when the restrictions are on the spectrums, the restrictions are strict, and all the spectrums contain the color 1. In [4] and [16] it was shown that for bipartite graphs with maximum degree equal to 3 and with strict restrictions on spectrums the problem of finding an interval t-coloring that meets the restrictions is an NP-complete [17, 18] problem. The problem, where the restrictions are on spectrums can be reduced to the problem, when the restrictions are on the edges. In [19] another problem with restrictions is considered for bipartite graphs, where the restrictions are provided for one "part" of the bipartite graph.

Simplifying the Algorithm. Given a graph G and for each edge $e \in E(G)$ of the graph there is a restriction $R(e) = [l(e), r(e)]$. For a given t we want to answer whether it is possible to find an interval t-coloring α such that for each edge e the color $\alpha(e)$ is inside the restricted range $l(e) \leq \alpha(e) \leq r(e)$. For a given t let $P_t(G,R)$ be a function, which answers that question:

$$P_t(G,R) = \begin{cases} 1, & \text{if there is an interval } t\text{-coloring satisfying the restrictions,} \\ 0, & \text{otherwise.} \end{cases}$$

For an interval t-coloring we need to use all the colors from 1 to t and only those colors. We now want to prove that these restrictions are not important for finding a polynomial algorithm. It means that if for each restriction we can find an interval edge-coloring that uses only the colors that are at least 1 and are at most t, then we can always modify the restrictions and use the same algorithm to find an interval t-coloring.

We will say a coloring α is an interval coloring if it uses positive integers and for each vertex v the spectrum $S_\alpha(v)$ is an interval of integers and all the edges incident to v have different colors.

Lemma 1. *For a connected graph G and an edge-coloring α, if for every vertex the colors of the edges incident to that vertex form an interval of integers, then the set of all colors of G is also an interval of integers.*

Proof. We prove this Lemma by induction. Let $n = |V(G)|$. Let us start with an arbitrary vertex v_1. At each step we are going to add a new vertex that is connected to one of the previously added vertices and show that the newly formed set is also an interval (every color between the minimum and maximum is being used). If at the step k we added vertices $v_1, \ldots, v_k$ so far then let S_k denote the set of colors at that step: $S_k = \bigcup\limits_{i=1}^{k} S_\alpha(v_i)$, and our goal is to show that S_k is an interval of integers for every k. The base case of the induction is S_1, which is equal to $S_\alpha(v_1)$ and is an interval of integers. Suppose we already added vertices $v_1, v_2, \ldots, v_{k-1}$ for some $k \leq n$ and the set of colors so far is S_{k-1}, which is an interval of integers by induction. Since the graph is connected it is possible to find a vertex v_k that is connected with an edge to one of the other vertices in our set. Let $S_k = S_{k-1} \cup S_\alpha(v_k)$. Let us say v_j and v_k

are connected with an edge $e=(v_j,v_k)$, since $j \leq k-1$, $\alpha(e) \in S_{k-1}$ and since e is incident to v_k, $\alpha(e) \in S_\alpha(v_k)$. This means that $S_{k-1} \cap S_\alpha(v_k) \neq \varnothing$ and since S_{k-1} and $S_\alpha(v_k)$ are both intervals of integers then S_k is also an interval of integers, because it is the union of intervals that have a common point. Hence by induction we can get that S_n is an interval of integers, which is the set of all colors of the graph. □

This Lemma is very similar to the Lemma 1.2.1 from [4], where it is proved for interval t-coloring with some additional requirements.

Lemma 2. *If we can detect in polynomial time for any restriction R whether it is possible to find an interval coloring that meets the restrictions R, then:*

a) We can detect in polynomial time whether it is possible to have a coloring with restrictions R that uses the color 1.

b) We can detect in polynomial time whether it is possible to have a coloring with restrictions R that uses the color t and all the colors are less than or equal to t.

Proof. *a)* Suppose that there is a polynomial algorithm $A(G,R)$ answering whether there is an interval coloring of G that satisfies the restrictions R. We want to find an algorithm $A_1(G,R)$ that answers whether it is possible to have an interval coloring of G that satisfies the restrictions R and contains an edge with the color 1. The $A_1(G,R)$ could work this way: for every edge e, if $1 \in [l(e),r(e)]$, then we could make $r(e)=1$ and get a different restriction R_e, where $l(e)=1$ and $r(e)=1$ and run the algorithm A with restrictions R_e. This means that for each such edge e we would run the algorithm $A(G,R_e)$ and since the number of edges is $|E(G)|$ we would run the algorithm A with different restrictions at most $|E(G)|$ times, and if for some e we detect that $A(G,R_e)=1$, then $A_1(G,R)=1$, otherwise, $A_1(G,R)=0$. Hence the algorithm $A_1(G,R)$ is also polynomial.

b) Similar to the item (a) suppose that there is a polynomial algorithm $A(G,R)$ answering whether there is an interval coloring of G that satisfies the restrictions R. We want to find an algorithm $A_t(G,R)$ that answers whether it is possible to have an interval coloring of G that satisfies the restrictions R, there is an edge with the color t, and all the colors are at most t. The $A_t(G,R)$ could work this way: first we would transform all the restrictions in R into restrictions R^t in the following way: for each edge e intersect the restriction $[l(e),r(e)]$ with the interval $[1,t]$ to get a new restriction $[l^t(e),r^t(e)]$. If the intersection is empty for some edge, then $A_t(G,R)$ is 0, and we no longer need to continue. Otherwise, for all new restrictions in R^t , $r^t(e) \leq t$. Now on R^t let's do the following. For every edge e such that $t \in [l^t(e),r^t(e)]$, we could make $l^t(e)=t$ and get a different restriction R^t_e, where $l^t(e)=t$ and $r^t(e)=t$, and run the algorithm A with restrictions R^t_e. This means that for each e we would run the algorithm $A(G,R^t_e)$ and since the number of edges is $|E(G)|$ we would run the algorithm A with different restrictions $|E(G)|$ times and if for some e we detect that $A(G,R^t_e)=1$, then $A_t(G,R)=1$, otherwise, $A_t(G,R)=0$. Hence the algorithm $A_t(G,R)$ is also polynomial. □

Theorem 1. *If there is a polynomial algorithm that detects whether it is possible to have an interval edge-coloring for any restrictions R on the edges of a* *connected graph G, then there is a polynomial algorithm that detects whether there is an interval t-coloring of G for the given restrictions.*

Proof. The difference of interval edge-coloring and interval t-coloring is that the interval t-coloring should use the colors from 1 to t. Since the graph G is connected, if we ensure that all the colors are from 1 to t and the color 1 and the color t are used in the interval coloring, then by Lemma 1 the interval coloring will also be an interval t-coloring since all the colors from 1 to t will be used. Similar to the proofs of the previous Lemmas, let $A(G,R)$ be the algorithm that detects if the graph can have an interval coloring, $A_1(G,R)$ is the algorithm that detects if the graph can have an interval coloring using the color 1, and $A_t(G,R)$ is the algorithm that detects if there is an interval coloring that uses the color t and all the colors are less than or equal to t. We want to find an algorithm $A_{1..t}(G,R)$ that answers whether it is possible to have an interval t-coloring of G that satisfies the restrictions R. The algorithm for $A_{1..t}(G,R)$ could work in this way: first, we would select an edge e and construct R_e restrictions on it such that $r(e) = 1$ (similar to Lemma 2), then we would run the algorithm $A_t(G,R_e)$. If it results to 1 for some e, then $A_{1..t}(G,R)$ is also 1, otherwise, it is 0. Since we run the algorithm $A_t(G,R_e)$ at most $|E(G)|$ times and $A_t(G,R_e)$ is a polynomial algorithm from Lemma 2, the algorithm $A_{1..t}(G,R)$ is also polynomial. □

Theorem 1 means that we do not need to worry about the interval t-coloring for the restrictions. If we can provide a polynomial algorithm that detects whether it is possible to find an interval coloring for any given restrictions R, then we can have a polynomial algorithm that detects if it is possible to find an interval t-coloring. The Theorem is true for any connected graph.

A Polynomial Algorithm for Interval Edge Colorability of a Tree with Restrictions on Its Edges. Here we will provide a polynomial algorithm $A(T,R)$, which for a given tree T will return 1, if it is possible to have an interval edge-coloring with restrictions R and 0 if it is impossible. From Theorem 1 it follows, that if we can find such an algorithm, then we can also find a polynomial algorithm that answers if it is possible to have an interval t-coloring. We assume that for all the restrictions $l(e) \geq 1$, because we are only interested in colorings with positive integers. We are only interested to meet the restrictions for each edge and also to make sure that for every vertex the colors of incident edges are different and form an interval. Without loss of generality, we can assume that $r(e) \leq |V(T)|$, because we normally are interested in interval t-colorings, in which the maximum color can not exceed the number of edges $|E(T)|$ (since having the color 1 is also required) and in the case of trees we have $|E(T)| = |V(T)| - 1$. We can also assume that each edge has a restriction, because otherwise we could assume $l(e) = 1$ and $r(e) = |V(T)|$ for the edges that do not have restrictions. Now the problem is the following.

Problem. Given an arbitrary tree T with $N = |V(G)|$ vertices and given arbitrary restrictions R for every edge e with $1 \leq l(e) \leq r(e) \leq N$. Determine whether it is possible to have an interval coloring $\alpha : E(G) \to \{1,\dots,N\}$ such that for each edge e,

$l(e) \leq \alpha(e) \leq r(e)$ and for every vertex v, $S_\alpha(v)$ is an interval ($\overline{S_\alpha}(v) - \underline{S_\alpha}(v) + 1 = d_T(v)$ and all the colors from $\underline{S_\alpha}(v)$ to $\overline{S_\alpha}(v)$ occur in $S_\alpha(v)$).

We can assume $N \geq 2$, since the case $N = 1$ is obvious. In that case every tree with $N \geq 2$ has a vertex v with $d_T(v) = 1$ (a leaf vertex). Let v_0 be an arbitrary leaf vertex. Consider the rooted tree T_{v_0}. Let the vertex connected to v_0 be v_1. In that case T_{v_0} would look like the tree shown in Fig. 3.

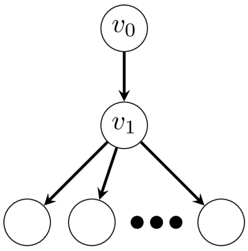

Fig. 3. The rooted tree T_{v_0}.

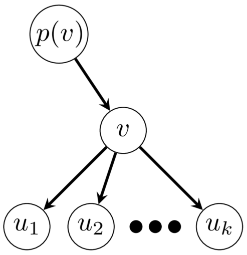

Fig. 4. The subtree $U(v)$ with children $u_1, \dots, u_k$.

By definition, $v_0 = p(v_1)$ and $U(v_1)$ would be the subtree induced by $V(S(v_1)) \cup \{v_0\}$, hence $U(v_1)$ would be the entire T_{v_0}. We are going to solve the problem using a dynamic programming on trees. For each vertex v and each color $c \in \{1, \dots, N\}$ we are going to calculate a value $colorable[v][c]$, which will be 1, if it is possible to have an interval coloring on $U(v)$ such that the edge connecting v and $p(v)$ is colored with the color c and all the restrictions are met on $U(v)$. We are not going to calculate the value of $colorable[v_0][c]$, because v_0 does not have a parent. For the entire tree we can say that it is possible to have an interval coloring that meets all the restrictions, if there is a color c, for which $colorable[v_1][c] = 1$. If for some color c, $colorable[v_1][c] = 1$ it means that the tree $U(v_1)$ has an interval coloring with the edge $v_0 \to v_1$ having the color c. It means the entire tree has an interval coloring that meets the restrictions.

In order to calculate $colorable[v_1][c]$ we need to calculate these values for all the children of v_1 and then based on these values calculate the answers in the vertex v_1. Now suppose that for some vertex v we already calculated the values for its children $u_1, \dots, u_k$ ($k = d_T(v) - 1$ and we have all the values $colorable[u_i][c]$ for every $1 \leq i \leq k$ and $1 \leq c \leq N$). How can we combine these results to calculate $colorable[v][c]$ for every color c? Fig. 4 illustrates that subgraph.

For every vertex u that has a parent $p(u)$ we will denote the edge $p(u) \to u$ by e_u. Consider all possible combinations of a color c for the edge e_v and all the possible colors $c_1, \dots, c_k$ for the edges $e_{u_1}, \dots, e_{u_k}$ such that $l(e_v) \leq c \leq r(e_v)$ and $colorable[u_i][c_i] = 1$ as illustrated in Fig. 5. If the colors $c, c_1, \dots, c_k$ are different from each other and form an interval of integers, then we can say that $colorable[v][c] = 1$. Since the restrictions for the edge e_v are satisfied, the colors around the vertex v form an interval of integers, and it is be possible to have interval colorings for the subtrees $U(u_1), \dots, U(u_k)$ with the colors $c_1, \dots, c_k$. Fig. 6 illustrates the subtrees that have interval colorings.

This means that for every color c such that $l(e_v) \leq c \leq r(e_v)$ we just need to find some colors $c_1, \ldots, c_k$ such that $colorable[u_1][c_1] = 1, \ldots, colorable[u_k][c_k] = 1$ and the colors $c, c_1, \ldots, c_k$ form an interval of integers.

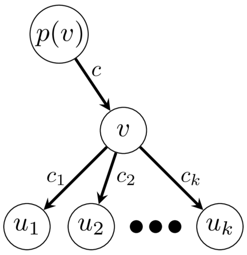

Fig. 5. Combination of possible colors around the vertex v.

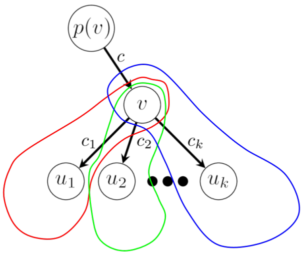

Fig. 6. The subtrees that we already know have interval colorings.

Suppose that the minimal color that we want to use for the vertex v is L. Then $c, c_1, \ldots, c_k$ should all be colors from $[L, L+k]$. It means we have to match a color from $[L, L+k]$ that is not c to a vertex u_k such that all the vertices have different colors. Let the colors from L to $L+k$ without the color c be $g_1, \ldots, g_k$. We can select a color g_j for a vertex u_i only if $colorable[u_i][g_j] = 1$. We can create a bipartite graph F with vertices $u_1, \ldots, u_k$ on the left side and colors $g_1, \ldots, g_k$ on the right side and connect a vertex on the left side to a vertex on the right side with an edge, if $colorable[u_i][g_j] = 1$ as illustrated on Fig. 7.

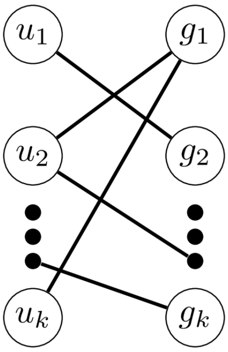

Fig. 7. The bipartite graph F constructed by the child vertices and their possible colors.

Suppose that we found a perfect matching $M \subseteq E(F)$ in this graph, then we can represent the matching as a function m with $m(u_i) = g_j$ if $(u_i, g_j) \in M$. In that case, if we color the edge e_v with the color c and the edge e_{u_i} with the color g_j, then all the restrictions will be met, because $colorable[u_i][m(u_i)] = 1$ and the colors

$c, m(u_1), \dots, m(u_k)$ form the interval $[L, L+k]$. Note that in order to find the perfect matching we fixed the L, but to calculate for each c such that $l(e_v) \leq c \leq r(e_v)$ we need to try to find a perfect matching for all possible L such that $1 \leq L \leq c \leq L+k \leq N$, and if for some L we find a perfect matching it means we found a way of coloring the subtrees $S(u_i)$ for the children of v in such a way that all the restrictions are met and the coloring is an interval coloring.

For each vertex v, we need to calculate maximum matching $O(N \cdot d_T(v))$ times and the matching algorithm will run for $O(d_T(v))$ vertices. If we use Kuhn's algorithm [20] for the maximum bipartite matching, then it will take $O(d_T(v)^3)$ every time we run the algorithm, that gives $O(N \cdot \sum\limits_{v \in V(T)} d_T(v)^4)$ for the entire algorithm. Note that $\sum\limits_{v \in V(T)} d_T(v) = 2 \cdot |E(T)| \leq 2 \cdot N$.

It is easy to prove, that if $\sum\limits_{i=1}^{N} x_i = N$ and $x_i \geq 0$, then $\sum\limits_{i=1}^{N} x_i^4 \leq N^4$, because $N^4 = \left(\sum\limits_{i=1}^{N} x_i\right)^4 \geq \sum_{i=1}^{N} x_i^4$ (if we open the brackets in $\left(\sum\limits_{i=1}^{N} x_i\right)^4$ it will be the sum of x_i^4 and other positive members).

This means the complexity of the algorithm is $O(N^5)$. If we also want the coloring to be an interval t-coloring, then it can be done in $O(N^7)$ by fixing the color 1 and the color t as explained in the proof of the Theorem 1.

Received 18.05.2021
Reviewed 19.07.2021
Accepted 23.07.2021

REFERENCES

1. West D.B. *Introduction to Graph Theory*. Prentice-Hall, New Jersey, 1996.
2. Asratian A.S., Kamalian R.R. Interval Colorings of Edges of a Multigraph. *Appl. Math.* **5** (1987), 25–34 (in Russian).
3. Kamalian R.R. Interval Colorings of Complete Bipartite Graphs and Trees. *Preprint of the Computing Centre of the Academy of Sciences of Armenia*. Yer. (1989).
4. Kamalian R.R. *Interval Edge-colorings of Graphs*. Doctoral Thesis. Novosibirsk (1990).
5. Axenovich M.A. On interval colorings of planar graphs. *Congr. Numer.* **159** (2002) 77–94.
6. Kamalian R.R., Petrosyan P.A. A note on interval edge-colorings of graphs. *Math. Probl. Comput. Sci.* **36** (2012) 13–16.
7. Petrosyan P.A. Interval edge-colorings of complete graphs and n-dimensional cubes. *Discrete Math.* **310** (2010) 1580–1587. https://doi.org/10.1016/j.disc.2010.02.001
8. Petrosyan P.A., Khachatrian H.H., Tananyan H.G. Interval edge-colorings of Cartesian products of graphs I. *Discuss. Math. Graph Theory* **33** (2013) 613–632. https://doi.org/10.7151/dmgt.1693

9. Khachatrian H. H., Petrosyan P. A. Interval edge-colorings of complete graphs. *Discrete Mathematics*, **339(9)**, (2016) 2249–2262. https://doi.org/10.1016/j.disc.2016.04.002
10. Kamalian R. R., Petrosyan, P. A. On Lower Bound for $W(K_{2n})$. *Mathematical Problems of Computer Science*, Vol. **23**, (2004), pp. 127–129
11. Kamalian R.R., Petrosyan P.A., On interval edge colorings of Harary graphs $H_{2n-2,2n}$. *Mathematical Problems of Computer Science*, Vol. **24**, (2005), pp. 86-88
12. Kamalian R. R., Petrosyan P. A. On Interval Colorings of Complete k-partite Graphs K_n^k. *Mathematical Problems of Computer Science*, Vol. **26**, (2006), pp. 28–32
13. Erdős P., Rubin A.L.,Taylor H. Choosability in graphs, *Proc. West Coast Conf. on Combinatorics*, Graph Theory and Computing, Congr. Numer. **26**, (1979), pp. 125-157.
14. Vizing V. G. Vertex colorings with given colors, *Metody Diskret. Analiz.* (in Russian), **29**: 3–10 (1976)
15. Caro Y., Schönheim J. Generalized 1-factorization of Trees. *Discrete Mathematics* **33** (1981), 319–321. https://doi.org/10.1016/0012-365X(81)90275-2
16. Even S., Itai A., Shamir A. On the Complexity of Timetable and Multicommodity Flow Problems. *SIAM J. Comput.* **5** (1976), 691–703.
17. Karp R. Reducibility among combinatorial problems. *In R. Miller & J. Thatcher (ed.), Complexity of Computer Computations* (pp. 85–103) . Plenum Press . (1972)
18. Cook S. A. The complexity of theorem proving procedures. *Proceedings of the Third Annual ACM Symposium* (p./pp. 151–158), New York. (1971)
19. Asratian A.S. *Investigation of Some Mathematical Model of Scheduling Theory*. Doctoral Thesis, M., MGU (1980) (in Russian).
20. Kuhn H. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly* **2** (1955), 83–97. https://doi.org/10.1002/nav.3800020109

Ա. Խ. ՍԱՀԱԿՅԱՆ, Ռ. Ռ. ՔԱՄԱԼՅԱՆ

ԾԱՌԵՐԻ ՄԻՋԱԿԱՅՔԱՅԻՆ ԿՈՂԱՅԻՆ ՆԵՐԿՈՒՄՆԵՐ ԿՈՂԵՐԻ ՎՐԱ ՏՐՎԱԾ ՍԱՀՄԱՆԱՓԱԿՈՒՄՆԵՐՈՎ

G գրաֆի կողային ներկումը հաջորդական բնական թվերով $c_1,\ldots,c_t$ կոչվում է միջակայքային t-ներկում, եթե բոլոր գույները օգտագործված են և ցանկացած գագաթին կից կողերը ներկված են տարբեր գույներով և կազմում են բնական թվերի միջակայք: G գրաֆը միջակայքային ներկելի է, եթե ինչ-որ t բնական թվի համար գոյություն ունի միջակայքային t-ներկում: Հոդվածում դիտարկվել է այն դեպքը, երբ ծառի կողերի վրա տրված են սահմանափակումներ և տրվել է այդ սահմանափակումներին բավարարող միջակայքային ներկման գոյությունը ստուգող բազմանդամային ալգորիթմ:

А. Х. СААКЯН, Р. Р. КАМАЛЯН

ИНТЕРВАЛЬНАЯ РЕБЕРНАЯ РАСКРАСКА ДЕРЕВЬЕВ С ОГРАНИЧЕНИЯМИ НА РЕБРАХ

Раскраска ребер графа G последовательными целыми числами $c_1, ..., c_t$ называется интервальной t-раскраской, если используются все цвета и цвета ребер, инцидентных любой вершине графа G, различны и образуют интервал целых чисел. Граф G является интервально раскрашиваемым, если он имеет интервальную t-раскраску для некоторого натурального t. В работе рассматривается задача существования интервальной раскраски дерева с заданными ограничениями на его ребрах. Представлен полиномиальный алгоритм проверки существования интервальной раскраски, удовлетворяющей этим ограничениям.