PROCEEDINGS OF THE YEREVAN STATE UNIVERSITY

Physical and Mathematical Sciences 2021, **55**(3), p. 160–168

M a t h e m a t i c s

INTERVAL VERTEX-COLORINGS OF CACTUS GRAPHS WITH RESTRICTIONS ON VERTICES

A. Kh. 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 interval vertex-coloring of a graph G is a coloring of the vertices of the graph with intervals of integers such that the intervals of any two adjacent vertices do not intersect. In this paper we consider the case, where for each vertex v there is a length $l(v)$ and a set of colors $S(v)$, from which the colors should be and it is required to find an interval vertex-coloring γ such that for each vertex v the restrictions are met, i.e. $|\gamma(v)| = l(v), \gamma(v) \subseteq S(v)$. We will provide a pseudo-polynomial algorithm for cactus graphs. If it is impossible to have an interval vertex-coloring that satisfies all the restrictions, then the algorithm will tell that as well.

https://doi.org/10.46991/PYSU:A/2021.55.3.160

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

Keywords: cactus graphs, trees, interval vertex-coloring, list coloring, dynamic programming, pseudo-polynomial algorithm.

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 $d_G(v)$.

A cactus is a connected graph in which any two simple cycles have at most one vertex in common. Equivalently, it is a connected graph in which every edge belongs to at most one simple cycle [1]. Fig. 1 illustrates different examples of cactus and non-cactus graphs.

Let I_k be the set $\{1,\dots,k\}$ of integers and let 2^{I_k} be the set of all the subsets of I_k. We will denote by $\tau(I_k)$ the set of all the elements from 2^{I_k} that form an interval of integers. More formally $\tau(I_k) = \{s : s \in 2^{I_k}, s \text{ is a non empty interval of integers}\}$. An interval vertex-$k$-coloring of a graph G is a function $\gamma : V(G) \to \tau(I_k)$ such that $\gamma(u) \cap \gamma(v) = \emptyset$ for all the edges $(u,v) \in E(G)$.

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

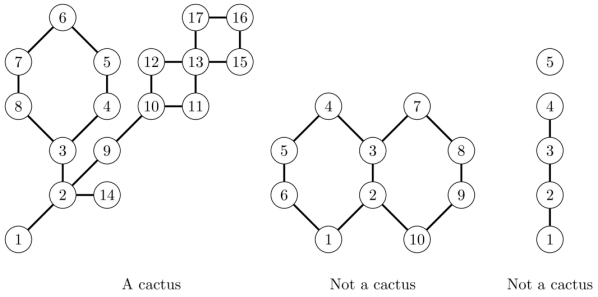

Fig. 1. Example of cactus and non-cactus graphs.

For a directed graph $\overrightarrow{G}$, if there is an edge from a vertex u to a vertex v, we will denote it by $u \to v$. 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 (a connected undirected acyclic graph) and its arbitrary 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 all vertices $v \in T_r$ there is a path in T_r from r to v. We will say that T_r is a rooted tree with the root r. Fig. 2 illustrates the rooted tree T_{v_1} with the root v_1.

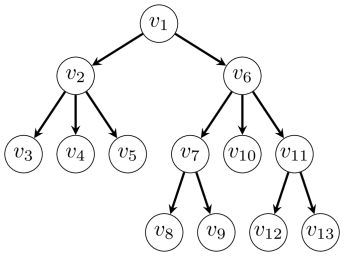

Fig. 2. A rooted tree T_{v_1} with the root v_1.

A vertex u is said to be the parent of the vertex v (denoted by $p(v)$) if $u \to v$ in T_r. 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 such vertices $w \in V(T_r)$ for which $v \to w$. A vertex having no children is said to be a leaf vertex.

Depth-first search (DFS) [2] is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root vertex (selecting some arbitrary vertex as the root vertex in the case of a graph) and explores as far as possible along

each branch before backtracking. In a depth-first search of an undirected graph G, every edge of G is either a tree edge or a back edge ([2], Theorem 22.10). If we start the DFS algorithm from a vertex r on a connected graph, then the DFS tree will be the rooted tree T_r that contains all the tree edges from the traversing. The DFS graph will be the directed graph that, in addition to the DFS tree, also includes the back edges.

In this paper, we will provide a pseudo-polynomial algorithm [3] for finding an interval vertex-k-coloring of the given cactus graph that meets the given restrictions. In [4] different cases of interval vertex-colorings were considered. In that paper, it was shown that the problem is NP-complete [5, 6] for complete and for bipartite graphs, and a polynomial solution was provided for star graphs. In [7] it was shown that for bipartite graphs, the edge 3-colorability problem is NP-complete even if there is at most one forbidden color for each vertex. In [8] the problem of interval edge-coloring was considered, where the edges should be colored with intervals of integers, and it was shown that the problem is NP-Complete even for star graphs with at most one forbidden color per edge. Interval vertex-colorings with restrictions is the general case of list colorings, in which each vertex is colored with one color instead of an interval. List colorings were first studied in the 1970s in the independent papers [9] and [10]. The list coloring problem is NP-complete for complete bipartite graphs, and can be solved in polynomial time for block graphs [11]. Interval edge-colorings with restrictions were also considered in different papers [12–15].

A Pseudo-polynomial Algorithm for Interval Vertex-k-coloring with Given Restrictions on a Cactus Graph.

Problem. *Given a cactus graph G and functions $l : V(G) \to I_k$ and $S : V(G) \to 2^{I_k}$. Find a function $\gamma : V(G) \to \tau(I_k)$ such that $|\gamma(v)| = l(v)$, $\gamma(v) \subseteq S(v)$ for every vertex $v \in V(G)$, and $\gamma(u) \cap \gamma(v) = \emptyset$ for any edge $(u,v) \in E(G)$.*

We will denote by $I(v,i)$ the interval of integers $[i, i + l(v) - 1]$. To solve the problem, we will construct the DFS graph of the given cactus (by selecting an arbitrary vertex as the root vertex) and later use dynamic programming to calculate the answer from the bottom to the top.

In the DFS graph of a cactus all the back edges will form the cycles of the cactus. We will draw the back edges with red color and the tree edges with black color. Fig. 3 illustrates the DFS graph of a cactus.

In Fig. 3 the edges $v_5 \to v_1, v_8 \to v_6$ and $v_{13} \to v_1$ are the back edges. There are three cycles formed by each of the back edges. Since the graph is a cactus no two cycles share an edge even though the cycles (v_1, v_2, v_5, v_1) and $(v_1, v_6, v_{11}, v_{13}, v_1)$ have a common vertex v_1. Moving forward, we will assume that we have already constructed the DFS graph of the given cactus G for some root vertex r, and all the statements and definitions will be based on that assumption. The DFS graph will be denoted by $\widetilde{G}$. For a vertex v, let $SG(v)$ be the subgraph of the graph $\widetilde{G}$ induced [1] by all the vertices w such that there is a path from v to w in the DFS tree. $SG(v)$ includes the back edges too. The vertex v is called the ancestor of the vertex u if $u \neq v$ and $u \in V(SG(v))$.

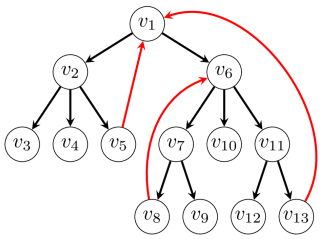

Fig. 3. The DFS graph of a cactus graph.
The red edges are the back edges.

Lemma. *In the DFS graph of a cactus graph G, for each tree edge $v \to u$, there can be at most one back edge connecting a vertex from the subgraph $SG(u)$ to an ancestor of the vertex u.*

Proof. Suppose that there are two back edges that connect the vertex $u_1 \in V(SG(u))$ to the ancestor vertex v_1 of u and the vertex $u_2 \in V(SG(u))$ to the ancestor vertex v_2 of u. Then the edge $v \to u$ will be inside the cycle formed by the path from v_1 to u_1 and the back edge $u_1 \to v_1$, and inside the cycle formed by the path from v_2 to u_2 and the back edge $u_2 \to v_2$. Since the two paths include the edge $v \to u$, it means the edge $v \to u$ is inside two different cycles, which can not happen for cactus graphs. □

Let $h(v)$ be the distance from the root vertex r to the vertex v in the DFS tree. If $v \to u$ is a back edge, then $h(v) > h(u)$. From Lemma we can say that for any vertex v, there can be at most one back edge that starts from that vertex; otherwise, the edge $p(v) \to v$ would be inside two different cycles. This means that $|E(\widetilde{G})| = O(|V(\widetilde{G})|)$ and since the graph G is the underlying graph of the DFS graph $\widetilde{G}$ it means $|E(G)| = O(|V(G)|)$ for the cactus graph G.

Let us define a function $B : V(\widetilde{G}) \to V(\widetilde{G})$ by the following way: if there is a back edge $v \to u$ that starts from the vertex v, then we let $B(v) = u$, otherwise, $B(v) = v$. Since there can be at most one back edge in the case of cactus graphs, the function B is uniquely defined for every vertex.

Let us also define a function $A : V(\widetilde{G}) \to V(\widetilde{G})$. For a vertex v if there is a vertex $u \in V(SG(v))$ such that $h(B(u)) < h(v)$ (the vertex $B(u)$ is the ancestor of the vertex v), then $A(v) = u$ (from the lemma there can be at most one such vertex). If there is no such vertex, then $A(v) = v$.

If $A(v) \neq v$, then there is a back edge, starting from the subgraph $SG(v)$, that connects to an ancestor of the vertex v and the back edge is $A(v) \to B(A(v))$. Note that the equality $A(v) = v$ still does not mean that there is no back edge from the subgraph $SG(v)$ that connects to an ancestor of the vertex v, because the back edge could start from the vertex v. If $h(B(A(v)) < h(v)$, then the edge $p(v) \to v$ belongs to the cycle formed by the edge $A(v) \to B(A(v))$ and the path from $B(A(v))$ to $A(v)$.

In Fig. 3, $B(v_{13}) = v_1$, $B(v_{11}) = v_{11}$, $A(v_{13}) = v_{13}$, $A(v_{11}) = v_{13}$, $A(v_2) = v_5$, $A(v_6) = v_{13}$, $A(v_3) = v_3$.

Suppose that we are coloring with intervals from the set $\tau(I_k)$. We will say that the interval vertex-k-coloring satisfies the restrictions in a subgraph of the graph $\widetilde{G}$ if the underlying graph of that subgraph satisfies the restrictions. For each vertex v and two integers $l_1 \in I_k, l_2 \in I_k$ we are going to calculate a value $colorable[v][l_1][l_2]$, which will be 1 if it is possible to have an interval vertex-k-coloring in the subgraph $SG(v)$ (including the back edges that connect the vertices of $V(SG(v))$) such that all the restrictions are met for $SG(v)$, the color of the vertex v is the interval $I(v,l_1)$, and the color of the vertex $A(v)$ is the interval $I(A(v),l_2)$, otherwise, $colorable[v][l_1][l_2]$ will be 0. If $A(v) = v$ and $l_1 \neq l_2$, then $colorable[v][l_1][l_2]$ would be 0. We will say that there is a back edge from the subgraph $SG(v)$ if $h(B(A(v))) < h(v)$.

In order to calculate $colorable[v][l_1][l_2]$ we need to calculate these values for all the children of v and then based on these values calculate the answers in the vertex v. Now suppose that for some vertex v we have already calculated the values for its children $u_1, \ldots, u_m$ and we have all the values $colorable[u_i][l_{i1}][l_{i2}]$ for every $1 \leq i \leq m$ and $1 \leq l_{i1}, l_{i2} \leq k$. How can we combine these results to calculate $colorable[v][l_1][l_2]$ for every $1 \leq l_1, l_2 \leq k$? For each vertex v it is possible to have at most one child vertex such that $A(u_i) = A(v)$, which would mean that the back edge from the subgraph $SG(v)$ starts from the subgraph $SG(u_i)$. It is possible for some child vertices to have $B(A(u_i)) = v$, which would mean that for some of the child vertices the back edges from their subgraphs end at the vertex v. Fig. 4 illustrates the general case, where the single back edge $A(v) \to B(A(v))$ starts from the subgraph $SG(u_m)$ and it also illustrates the case, where $B(A(u_1)) = v$.

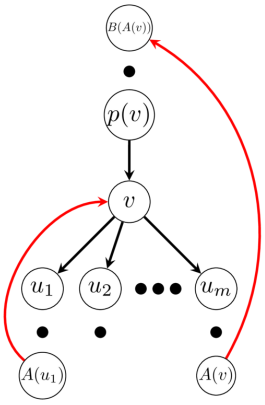

Fig. 4. Illustrating possible combinations of back edges.

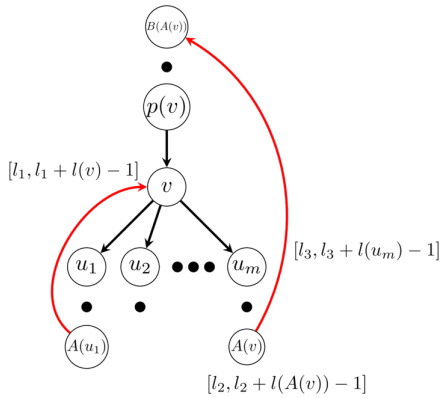

Fig. 5. Adding the child vertex u_m to the answer

There are a couple of different cases that we need to handle:

C1. $B(A(v)) = v$, which means there is no back edge from the subgraph $SG(v)$.

C2. $B(A(v)) \neq v$ and $v = A(v)$, which means the back edge from the subgraph $SG(v)$

starts from the vertex v.

C3. $B(A(v)) \neq v$ and $v \neq A(v)$, which means the back edge from the subgraph $SG(v)$ is the same as the back edge from one of its child vertex subgraphs. In that case without loss of generality, we can assume that this child is the child vertex u_m (the last child vertex in the list of child vertices). Hence in this case we get $A(v) = A(u_m)$. In the Fig. 4 this case is illustrated.

For the child vertices, there are also a couple of different options:

U1. $h(B(A(u_i))) > h(v)$, which means there is no back edge from the subgraph $SG(u_i)$ (the vertex u_2 in the Fig. 4). In this case the edge (v,u_i) is a bridge in the original cactus.

U2. $h(B(A(u_i))) = h(v)$ or equivalently $B(A(u_i)) = v$, which means the back edge from the subgraph $SG(u_i)$ ends in the vertex v (the vertex u_1 in the Fig. 4). There can be such multiple child vertices, because even though they share the vertex v, they do not share a common edge.

U3. $h(B(A(u_i))) < h(v)$, which means that the back edge from the subgraph $SG(u_i)$ is the same back edge from the subgraph $SG(v)$ (the vertex u_m in the Fig. 4). This is the same case as the C3, and there can be at most one such child vertex, in which case we will assume that it is the vertex u_m.

Let us handle each of the C1, C2, C3 cases separately. We are going to construct the $colorable[v][l_1][l_2]$ by iterating over the child vertices in the order of $u_1,\ldots,u_m$. Since the vertex u_m might require special attention, we will treat it differently, but all the other child vertices will be handled the same way for cases C1, C2 and C3.

When calculating the answers for the vertex v, we will store numbers $dp[i][l_1]$, which will be defined in the following way. If it is possible to find an interval vertex-k-coloring in the subgraph of the graph $\widetilde{G}$ induced by the vertices $\{v\} \cup V(SG(u_1)) \cup \cdots \cup V(SG(u_i))$ such that the vertex v is colored with $I(v,l_1)$ and all the restrictions are met in this subgraph, then $dp[i][l_1] = 1$, otherwise it is 0. If $i = 0$, then it means we only have the vertex v in the subgraph. Note that we also take into account the back edges for these subgraphs, to make sure the coloring is valid. When $i = m$ and we have the case C3 we will not calculate the $dp[m][l_1]$ and will calculate the $colorable[v][l_1][l_2]$ directly, otherwise, for the cases C1 and C2, $colorable[v][l_1][l_2] = dp[m][l_1]$ if the equality $l_1 = l_2$ is satisfied and 0 otherwise (because $A(v) = v$ and we do not need to store two intervals).

Let us first calculate $dp[0][l_1]$. For $l_1, 1 \leq l_1 \leq k$, $dp[0][l_1] = 1$ if and only if $I(v,l_1) \subseteq S(v)$ and $dp[0][l_1] = 0$ otherwise.

Now suppose that we have the values for $dp[i-1][l_1]$ and we want to calculate the values of $dp[i][l_1]$ for all l_1 after adding the vertex u_i (we assume u_i is not the case U3). In the case of $U1$, $dp[i][l_1]$ should be equal to 1 if $dp[i-1][l_1] = 1$ and there is an integer l_3 such that $colorable[u_i][l_3][l_3] = 1$ (since $A(u_i) = u_i$) and the intervals

$I(v,l_1)$ and $I(u_i,l_3)$ do not intersect, otherwise $dp[i][l_1]$ should be 0. In the case of U2, $dp[i][l_1]$ should be equal to 1 if $dp[i-1][l_1]=1$ and there are integers l_3,l_4 such that $colorable[u_i][l_3][l_4]=1$ and the interval $I(v,l_1)$ does not intersect with the intervals $I(u_i,l_3)$ and $I(A(u_i),l_4)$, since the vertex v is connected with the vertex u_i and the vertex $A(u_i)$, otherwise $dp[i][l_1]$ should be 0.

Now suppose that it is the case C3 and we calculated the values $dp[m-1][l_1]$. In this case $A(v)=A(u_m)$. $colorable[v][l_1][l_2]=1$ if $dp[m-1][l_1]=1$ and there exists an integer l_3 such that the intervals $I(v,l_1)$ and $I(u_m,l_3)$ do not intersect and $colorable[u_m][l_3][l_2]=1$. Fig. 5 illustrates this case.

We would calculate all these values from bottom to top in the tree. To find out if there is an interval vertex-coloring for the cactus that satisfies the restrictions we should find an integer l_1 such that $colorable[r][l_1][l_1]=1$. If we also store the intervals that we used for the child vertices during the calculation, we would be able to construct the answer from top to bottom.

Now let us estimate the complexity of the algorithm. Let $N=|V(G)|$ and we already know that $|E(G)|=O(N)$ for the cactus graphs. Constructing the DFS graph and calculating the values $A(v),B(v)$ for all the vertices can be done in $O(N)$. For each vertex v, we would need to iterate over l_1 from 1 to k, and for each child vertex, we would need to iterate over l_3, which means we would do $O(k^2)$ operations $d_G(v)$ times and since $\sum_{v\in V(G)} d_G(v)$ is $O(N)$ we would do $O(N\cdot k^2)$ operations for the case U1. For the case U2, we would need to choose the l_1,l_3 and l_4, and we need to do it once for every back edge (only for the end vertex of the back edge), so that would take $O(N\cdot k^3)$ operations in total. And for the case U3, we would need to choose l_1,l_2 and l_3, which could happen only once for each vertex, resulting in $O(N\cdot k^3)$ operations in total. In the case of trees, we would only have the case U1, and hence the complexity would be $O(N\cdot k^2)$

Received 30.06.2021
Reviewed 31.10.2021
Accepted 14.11.2021

REFERENCES

1. West D.B. *Introduction to Graph Theory*. New Jersey, Prentice-Hall (1996).
2. Cormen T.H. *Introduction to Algorithms*. Cambridge, Masachusetts, London, The MIT Press (2009).
3. Garey M.R., Johnson D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Series of Books in the Math. Sci., Freeman W.H. and Co. (1979).
4. Kubale M. Interval Vertex-coloring of a Graph with Forbidden Colors. *Discret. Math.* **74** (1989), 125–136. https://doi.org/10.1016/0012-365X(89)90204-5

5. Karp R. *Reducibility Among Combinatorial Problems*. In: Complexity of Computer Computations (eds. by R. Miller, J. Thatcher). Plenum Press (1972), 85–103 .
6. Cook S.A. The Complexity of Theorem Proving Procedures. *Proceedings of the Third Annual ACM Symposium*. New York (1971), 151–158.
7. Even S., Itai A., Shamir A. On the Complexity of Timetable and Multicommodity Flow Problems. *SIAM J. Comput.* **5** (1976), 691–703.
8. Kubale M. Interval Edge Coloring of a Graph with Forbidden Colors. *Discret. Math.* **121** (1993), 135–143. https://doi.org/10.1016/0012-365X(93)90546-6
9. Erdős P., Rubin A.L.,Taylor H. Choosability in Graphs. Proc. West Coast Conf. on Combinatorics. *Graph Theory and Computing, Congr. Numer.* **26** (1979), 125–157.
10. Vizing V.G. Vertex Colorings with Given Colors. *Metody Diskret. Analiz.* **29** (1976), 3–10 (in Russian).
11. Sahakyan A.K. List Coloring of Block Graphs and Complete Bipartite Graphs. *World Science* **8** : 69 (2021), 36–43. https://doi.org/10.31435/rsglobal_ws/30082021/7661
12. Sahakyan A.K., Kamalian R.R. Interval Edge-Colorings of Trees with Restrictions on the Edges. *Proceedings of the YSU A. Physical and Mathematical Sciences* **55** : 2 (2021), 113–122. https://doi.org/10.46991/PYSU:A/2021.55.2.113
13. Sahakyan A. Interval Edge-Coloring with Restrictions on Edges for Graphs with Special Structures of Cycles. *Norwegian Journal of Development of the International Science* **71** (2021), 13–19. https://doi.org/10.24412/3453-9875-2021-71-13-19
14. Sahakyan A.K. Interval Edge Coloring of Trees with Strict Restrictions on the Spectrums. *Science Review* **3** : 38 (2021), 27–29. https://doi.org/10.31435/rsglobal_sr/30072021/7592
15. Sahakyan A.K. Edge Coloring of Cactus Graphs with Given Spectrums. *International Academy Journal Web of Scholar* **2** : 52 (2021), 25–28. https://doi.org/10.31435/rsglobal_wos/30062021/7617

Ա. Խ. ՍԱՀԱԿՅԱՆ, Ռ. Ռ. ՔԱՄԱԼՅԱՆ

ԿԱԿՏՈՒՍ ԳՐԱՖՆԵՐԻ ՄԻՋԱԿԱՅՔԱՅԻՆ ԳԱԳԱԹԱՅԻՆ ՆԵՐԿՈՒՄՆԵՐ ԳԱԳԱԹՆԵՐԻ ՎՐԱ ՏՐՎԱԾ ՍԱՀՄԱՆԱՓԱԿՈՒՄՆԵՐՈՎ

G գրաֆի միջակայքային գագաթային ներկումը գրաֆի գագաթների այնպիսի ներկում է ամբողջ թվերի միջակայքերով, որ ցանկացած երկու կից գագաթների միջակայքերը չեն հատվում: Դիտարկվել է այն դեպքը, երբ կակտուսի ամեն v գագաթի համար տրված է $l(v)$ երկարություն և $S(v)$ գույների բազմություն, որոնք կարելի է օգտագործել և պահանջվում է գտնել այնպիսի γ միջակայքային գագաթային ներկում, որ ցանկացած v գագաթի համար $|\gamma(v)| = l(v), \gamma(v) \subseteq S(v)$: Կակտուս գրաֆների համար տրվել է ներկման գոյությունը սպուգող և կառուցող պսևդոբազմանդամային ալգորիթմ:

А. Х. СААКЯН, Р. Р. КАМАЛЯН

ИНТЕРВАЛЬНАЯ ВЕРШИННАЯ РАСКРАСКА КАКТУСОВ С ОГРАНИЧЕНИЯМИ НА ВЕРШИНАХ

Интервальная раскраска вершин графа G – это раскраска вершин графа такими интервалами целых чисел, что интервалы любых двух соседних вершин не пересекаются. В работе рассматривается случай, когда для каждой вершины v кактуса заданы длина $l(v)$ и множество цветов $S(v)$, из которого должны быть выбраны цвета. Требуется найти такую интервальную раскраску γ вершин кактуса, при которой для каждой вершины v ограничения соблюдаются, т.е. $|\gamma(v)| = l(v), \gamma(v) \subseteq S(v)$. Представлен псевдополиномиальный алгоритм решения задачи для кактусов.