

SPANNING TREES WITH MANY OR FEW COLORS
IN EDGE-COLORED GRAPHS

HAJO BROERSMA

Faculty of Applied Mathematics
University of Twente, P.O. Box 217
7500 AE Enschede, The Netherlands

AND

XUELIANG LI*

Department of Applied Mathematics
Northwestern Polytechnical University
Xi'an, Shaanxi 710072, P.R. China

Abstract

Given a graph $G = (V, E)$ and a (not necessarily proper) edge-coloring of G , we consider the complexity of finding a spanning tree of G with as many different colors as possible, and of finding one with as few different colors as possible. We show that the first problem is equivalent to finding a common independent set of maximum cardinality in two matroids, implying that there is a polynomial algorithm. We use the minimum dominating set problem to show that the second problem is *NP*-hard.

Keywords: edge-coloring, spanning tree, matroid (intersection), complexity, *NP*-complete, *NP*-hard, polynomial algorithm, (minimum) dominating set.

1991 Mathematics Subject Classification: 05C05, 05C85, 05B35.

*This research was carried out while the second author was visiting the Faculty of Applied Mathematics, University of Twente, supported by the Euler Institute for Discrete Mathematics and its Applications.

1. INTRODUCTION

In (real-world) problems that are modeled by graphs, the edges of the graphs often are assigned certain weights, labels or colors representing certain types (or costs) of (the) relations between the vertices. In many of these problems the goal is to find a subgraph in which a function of the weights, labels or colors of the edges attains some optimum value. Well-known examples are the minimum spanning tree problem and the traveling salesman problem. Other examples are problems in which one is interested in (the existence of) multicolored cycles, i.e. cycles in which all edges have different colors, or monochromatic cycles, i.e. cycles in which all edges have the same color. Here we focus on finding spanning trees with many or few different colors in (not necessarily properly) edge-colored graphs.

All graphs considered are finite and undirected, and contain no multiple edges or loops. We use [1] for basic graph theoretic terminology and notation. In describing problems and their complexity the terminology of [2] is applied.

Let $G = (V, E)$ be a connected graph. By a *coloring* of G we will mean a function $c : E \rightarrow \mathbb{N}$. If G is assigned such a coloring, then we say that G is a *colored graph*, and we call $c(e)$ the *color* of the edge $e \in E$. We note that c is not necessarily a proper edge-coloring, i.e., two adjacent edges may have the same color. For a subgraph H of G , we let $C(H) = \bigcup_{e \in E(H)} c(e)$ and $c(H) = |C(H)|$. A spanning tree T of G is called a *maximum tree* if $c(T)$ is maximum, i.e., if there is no spanning tree T' of G such that $c(T') > c(T)$. Similarly we say that T is a *minimum tree* if $c(T)$ is minimum.

In the sequel we will consider the complexity of finding a maximum tree in a colored graph G and of finding a minimum tree of G . In Section 2 we show that the problem of finding a maximum tree is equivalent to finding a common independent set of maximum cardinality in two matroids, implying that there is a polynomial algorithm for this problem. In Section 3 we use the minimum dominating set problem to show that finding a minimum tree is *NP*-hard.

2. MAXIMUM TREES

We start this section with a simple observation. But first we need some additional terminology.

Let F be a (spanning) forest of a connected graph $G = (V, E)$ with a coloring $c : E \rightarrow \mathbb{N}$. Then F is said to be an *optimal forest* if $c(F) = |E(F)|$ (the edges of F have different colors) and there is no forest F' of G such

that $c(F') > c(F)$. A spanning tree T of G is said to *correspond with* F if $C(T) = C(F)$ and T can be obtained from F by adding edges of G (with colors from $C(F)$).

Lemma 1. *T is a maximum tree of a connected colored graph G if and only if T corresponds with an optimal forest of G .*

Proof. Let G be a connected graph with a coloring $c : E(G) \rightarrow \mathbb{N}$. If T is a maximum tree with $c(T) = k$, then clearly T contains a spanning forest F of G with $C(F) = C(T)$ and $|E(F)| = k$. Assuming F is not optimal, some forest F' exists such that $c(F') > k$. Since G is connected this implies the existence of a spanning tree T' of G with $c(T') \geq c(F') > c(T)$, contradicting the maximality of T . Hence F is optimal, and T clearly corresponds with F .

Next assume T corresponds with an optimal forest F of G . Then $C(T) = C(F)$ and $c(T) = |E(F)|$. Assuming T is not maximum, consider a spanning tree T' of G with $c(T') > c(T)$. T' contains a spanning forest F' of G with $c(F') = |E(F')| = c(T') > c(T) = c(F)$, contradicting the optimality of F . ■

The above lemma shows that finding a maximum tree of a connected colored graph G is equivalent to finding an optimal forest of G . The latter problem can easily be described in terms of the so-called cardinality intersection problem of two matroids. We refer to [3, 4] for terminology and notation concerning matroids and combinatorial optimization. For convenience we repeat some of the definitions here.

A *matroid* $M = (S, \mathcal{I})$ is a structure in which S is a finite set of *elements* and \mathcal{I} is a family of subsets of S (called *independent sets*) such that

- (1) $\emptyset \in \mathcal{I}$ and all proper subsets of a set $I \in \mathcal{I}$ are in \mathcal{I} .
- (2) If $U, V \in \mathcal{I}$ with $|U| = |V| + 1$, then there exists an element $x \in U \setminus V$ such that $V \cup \{x\} \in \mathcal{I}$.

In the sequel we will often use $V + x$ instead of $V \cup \{x\}$, and $V + x - y$ instead of $(V \cup \{x\}) \setminus \{y\}$, et cetera.

The *cardinality intersection problem* for two matroids is as follows. Given two matroids $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ over the same set S , find a maximum cardinality intersection $I \in \mathcal{I}_1 \cap \mathcal{I}_2$. We note that this definition in [3] should not be confused with the intersection of two matroids defined in [4, p. 123]. The latter is a matroid. It is well-known that there exists a polynomial algorithm for the cardinality intersection problem for two matroids. A description of such an algorithm can be found in [3, p. 313–314]. We will now define two suitable matroids on the edge set of a graph $G = (V, E)$ with

a coloring $c : E \rightarrow \mathbb{N}$.

$$\begin{aligned} \text{Let } \mathcal{I}_1 &= \{E' \subseteq E \mid (V, E') \text{ is a forest}\} \quad \text{and} \\ \mathcal{I}_2 &= \{E' \subseteq E \mid c((V, E')) = |E'|\}. \end{aligned}$$

Lemma 2. (E, \mathcal{I}_1) and (E, \mathcal{I}_2) are matroids.

Proof. We check that (E, \mathcal{I}_1) and (E, \mathcal{I}_2) satisfy (1) and (2). For (E, \mathcal{I}_1) , clearly every subset, including \emptyset , of the edge set of a forest of G induces a forest of G ; moreover, if for two forests F_1 and F_2 of G , $|E(F_1)| = |E(F_2)| + 1$, then for some edge $e = uv \in E(F_1)$, u and v are in different components of F_2 , so that $F_2 + e$ is a forest of G . For (E, \mathcal{I}_2) , clearly every subset $E'' \subseteq E'$, including \emptyset , of some edge set $E' \subseteq E$ with $c((V, E')) = |E'|$ satisfies $c((V, E'')) = |E''|$; moreover, if for two subsets $E_1, E_2 \subseteq E$, $c((V, E_1)) = |E_1|$, $c((V, E_2)) = |E_2|$, and $|E_1| = |E_2| + 1$, then there is an edge $e \in E_1 \setminus E_2$ with $c(e) \notin C((V, E_2))$ such that $E_3 = E_2 \cup \{e\} \subseteq E$ and $c((V, E_3)) = |E_3|$. ■

Remark. (E, \mathcal{I}_1) is often called the *graphic matroid*; (E, \mathcal{I}_2) is a so-called *transversal matroid* with respect to the partition of edges in color classes.

Let $G = (V, E)$ be a graph, $c : E \rightarrow \mathbb{N}$ a coloring of G , and let $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ be two matroids associated with G , where \mathcal{I}_1 and \mathcal{I}_2 are defined as above.

Lemma 3. F is an optimal forest of G if and only if $E(F)$ is a maximum cardinality intersection of M_1 and M_2 .

Proof. By definition of M_1 and M_2 , any set $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ corresponds to a forest (V, I) of G such that $c((V, I)) = |I| = |E((V, I))|$ and vice versa. Clearly $|I|$ is maximum if and only if (V, I) is an optimal forest of G . ■

Combining Lemma 1 and Lemma 3 we conclude that T is a maximum tree of a connected colored graph G if and only if T corresponds to a solution I of the maximum cardinality intersection problem for the two given matroids defined on $E(G)$. Since the latter problem can be solved in polynomial time and T can be obtained from $(V(G), I)$ by successively adding edges of G avoiding cycles, it is clear that we can find a maximum tree of G in polynomial time. It is just a matter of translation to adapt and extend the general algorithm for the maximum cardinality intersection problem given in [3, p. 313–314] to a polynomial algorithm for finding a maximum tree. Moreover, the same technique can be applied to solve the problem of finding a base with maximum number of colors in a matroid each element of which

is colored. However, for the aforementioned maximum tree problem we can give an explicit discussion which we think is useful and easier to understand. First we give a characterization of optimal forests in terms of tree operations.

Let $G = (V, E)$ be a graph and $c : E \rightarrow \mathbb{N}$ a coloring of G , and consider some forest F of G such that $c(F) = |E(F)|$. We will describe a class \mathcal{F}_F of forests containing F in two ways. We start with an informal, more or less algorithmic description.

First set $\mathcal{F} = \{F\}$. For every $F' \in \mathcal{F}$, and for every $e \in E(G) \setminus E(F')$ such that $c(e) \notin C(F')$, consider all forests $\mathcal{F}_C = \{F' + e - f \mid F' + e - f$ is a forest for some $f \in E(F')\}$. Also, for every forest $F' \in \mathcal{F}$, consider all forests $\mathcal{F}' = \{F'' \mid E(F'') \subseteq E(G), c(F'') = |E(F'')|, C(F'') = C(F')\}$. Now replace \mathcal{F} by $\mathcal{F} \cup \mathcal{F}_C \cup \mathcal{F}'$, and repeat the process for elements of \mathcal{F} until no new forests are found. Let \mathcal{F}_F denote the final class of forests found by this process.

Another, formal way of describing the class \mathcal{F}_F is as follows. Define a relation R on the set of forests of G by: $F_1 R F_2$ if and only if F_1 and F_2 are forests of G such that $c(F_1) = |E(F_1)| = c(F_2) = |E(F_2)|$, and either

- (3) $C(F_1) = C(F_2)$ or
- (4) $F_2 = F_1 + e - f$ for some edges $e \in E(F_2) \setminus E(F_1)$ and $f \in E(F_1) \setminus E(F_2)$.

Clearly R is a reflexive and symmetric relation. Denote by $t(R)$ the transitive closure of R . Then $t(R)$ is an equivalence relation, and \mathcal{F}_F is the equivalence class containing F .

We note here that for two forests F_1 and F_2 of G with $c(F_1) = |E(F_1)| = c(F_2) = |E(F_2)|$ and $C(F_1) = C(F_2)$, one cannot always find an edge $e \in E(F_2) \setminus E(F_1)$ and $f \in E(F_1) \setminus E(F_2)$ such that $F_2 = F_1 + e - f$. We give an example.

Example. Let $G = (V, E)$ be a graph with $V = \{v_1, v_2, v_3, v_4\}$ and $E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_3v_4\}$ and coloring $c : E \rightarrow \mathbb{N}$ given by $c(v_1v_2) = c(v_3v_4) = 1$, $c(v_2v_3) = c(v_1v_4) = 2$, and $c(v_1v_3) = 3$. Consider two forests F_1 and F_2 of G defined by $F_1 = (V, E_1)$ and $F_2 = (V, E_2)$, where $E_1 = \{v_1v_2, v_1v_3, v_1v_4\}$ and $E_2 = \{v_2v_3, v_1v_3, v_3v_4\}$. Then F_1 and F_2 satisfy (3), but not (4). Moreover, if we omit (3) from the definition of R above, then the equivalence classes induced by the newly defined equivalence relation $t(R)$ are such that for this example F_1 and F_2 are in different classes.

Suppose that F_{opt} is an optimal forest of the colored graph G . Then we can characterize the optimal forests of G in the following way.

Theorem 4. F is an optimal forest of G if and only if $F_{\text{opt}} \in \mathcal{F}_F$.

Proof. First suppose $F_{\text{opt}} \in \mathcal{F}_F$. Noting that all elements of \mathcal{F}_F have the same number of edges, and all edges have different colors, we easily obtain that $c(F) = c(F_{\text{opt}})$ and $|E(F)| = |E(F_{\text{opt}})| = c(F_{\text{opt}}) = c(F)$. Since F_{opt} is an optimal forest, there exists no forest F' such that $c(F') > c(F_{\text{opt}}) = c(F)$. Hence F is an optimal forest of G .

Next suppose F is an optimal forest of G . We use induction on $k = |E(F_{\text{opt}}) \setminus E(F)|$ to prove that $F_{\text{opt}} \in \mathcal{F}_F$. If $k = 0$, there is nothing to prove. Suppose $k > 0$, and suppose that $|E(F_{\text{opt}}) \setminus E(F)| < k$ implies $F_{\text{opt}} \in \mathcal{F}_F$. If $C(F_{\text{opt}}) = C(F)$, then by definition of \mathcal{F}_F , $F_{\text{opt}} \in \mathcal{F}_F$. Hence assume there exists an edge $e \in E(F_{\text{opt}}) \setminus E(F)$ with $c(e) \notin C(F)$. Then $F + e$ contains a (unique) cycle C , otherwise $F + e$ is a forest with $c(F + e) = c(F) + 1 (= |E(F + e)|)$, a contradiction. Since F_{opt} is a forest and $e \in E(F_{\text{opt}})$, C contains an edge $f \in E(F) \setminus E(F_{\text{opt}})$. Denote $F_1 = F + e - f$. Clearly F_1 is an optimal forest of G and $F_1 \in \mathcal{F}_F$. This implies $F \in \mathcal{F}_{F_1}$, since \mathcal{F}_F is an equivalence class. Now, since $|E(F_{\text{opt}}) \setminus E(F_1)| < |E(F_{\text{opt}}) \setminus E(F)| = k$, from the induction hypothesis we obtain $F_{\text{opt}} \in \mathcal{F}_{F_1} = \mathcal{F}_F$. This completes the proof. ■

The above characterization of optimal forests, and the definition of the equivalence classes \mathcal{F}_F do not seem to be very useful in developing a polynomial algorithm for finding an optimal forest. Especially condition (3) — which could not be omitted — does not seem to make a polynomial algorithm based on this very likely. To describe such an algorithm we will now turn our attention to the way one can augment the number of colors of a forest.

For a graph $G = (V, E)$ and a coloring $c : E \rightarrow \mathbb{N}$ of G , let $\mathcal{F}_i = \{F \text{ is a forest of } G \mid c(F) = |E(F)| = i\}$, $i = 0, 1, \dots, c(F_{\text{opt}})$, where F_{opt} is an optimal forest of G . We first show that $\mathcal{F}_i = \mathcal{F}_F$ for some forest F of G , and then, more importantly, we show that for every forest $F_1 \in \mathcal{F}_i$ with $0 \leq i < c(F_{\text{opt}})$, there exists a forest $F_2 \in \mathcal{F}_{i+1}$ and a sequence of edges $e_1, e_2, \dots, e_k \in E(F_2) \setminus E(F_1)$ and $f_1, f_2, \dots, f_{k-1} \in E(F_1) \setminus E(F_2)$ such that $F_2 = F_1 + e_1 - f_1 + \dots + e_{k-1} - f_{k-1} + e_k$.

Lemma 5. $\mathcal{F}_i = \mathcal{F}_F$ for some forest F of G .

Proof. Let $F \in \mathcal{F}_i$. Then clearly $\mathcal{F}_F \subseteq \mathcal{F}_i$ since all forests of \mathcal{F}_F have the same number of edges, and all edges have different colors. We now show that $\mathcal{F}_i \subseteq \mathcal{F}_F$. Let $F' \in \mathcal{F}_i$, i.e., $c(F') = |E(F')| = i$. If $C(F') = C(F)$, then clearly $F' \in \mathcal{F}_F$. Next assume $C(F') \neq C(F)$, and consider some edge $e \in E(F') \setminus E(F)$ with $c(e) \notin C(F)$. No matter whether $F + e$ is a forest or not, we can find an edge $f \in E(F) \setminus E(F')$ such that $F + e - f$ is a forest and $F + e - f \in \mathcal{F}_F$. Moreover $|C(F + e - f) \setminus C(F')| < |C(F) \setminus C(F')|$,

and we can prove that $F' \in \mathcal{F}_F$ by induction on $|C(F) \setminus C(F')|$ in the obvious way. ■

Note that from the example it follows that the assertion in Lemma [5] is not true if we omit (3) in the definition of R .

Lemma 6. *Let $i \in \{0, 1, \dots, c(F_{opt}) - 1\}$. Then for every forest $F_1 \in \mathcal{F}_i$ there exists a forest $F_2 \in \mathcal{F}_{i+1}$ such that $F_2 = F_1 + e_1 - f_1 + \dots + e_{k-1} - f_{k-1} + e_k$ for some edges $\{e_1, \dots, e_k\} \subseteq E(F_2) \setminus E(F_1)$ and $\{f_1, \dots, f_{k-1}\} \subseteq E(F_1) \setminus E(F_2)$.*

Proof. By induction on $k = |E(F_2) \setminus E(F_1)|$. If $k = 1$, there is nothing to prove: $F_2 = F_1 + e$ for some $\{e\} = E(F_2) \setminus E(F_1)$. Now assume $k \geq 2$ and suppose the result holds if $|E(F_2) \setminus E(F_1)| < k$.

Since $|E(F_2)| = |E(F_1)| + 1$, there exists an edge $e_1 \in E(F_2) \setminus E(F_1)$ such that $F_1 + e_1$ is a forest. If $c(e_1) \notin C(F_1)$, we are done: then $F_1 + e_1 \in \mathcal{F}_{i+1}$. Next assume $c(e_1) \in C(F_1)$. Then some edge $f_1 \in E(F_1) \setminus E(F_2)$ exists with $c(f_1) = c(e_1)$. Now consider $F'_1 = F_1 + e_1 - f_1$. Clearly F'_1 is a forest, $F'_1 \in \mathcal{F}_i$, and $|E(F_2) \setminus E(F'_1)| < |E(F_2) \setminus E(F_1)| = k$. Hence by the induction hypothesis, $F_2 = F'_1 + e_2 - f_2 + \dots + e_{r-1} - f_{r-1} + e_r$ for some edges $\{e_2, \dots, e_r\} \subseteq E(F_2) \setminus E(F'_1)$ and $\{f_2, \dots, f_{r-1}\} \subseteq E(F'_1) \setminus E(F_2)$. Clearly $e_1 \notin \{e_2, \dots, e_r\}$ since $e_1 \in E(F'_1)$, and $f_1 \notin \{f_2, \dots, f_{r-1}\}$ since $f_1 \notin E(F'_1)$. Moreover, $\{e_2, \dots, e_r\} \subseteq E(F_2) \setminus E(F_1)$ and $\{f_2, \dots, f_{r-1}\} \subseteq E(F_1) \setminus E(F_2)$. Hence $F_2 = F_1 + e_1 - f_1 + \dots + e_{r-1} - f_{r-1} + e_r$ for some edges $\{e_1, \dots, e_r\} \subseteq E(F_2) \setminus E(F_1)$ and $\{f_1, \dots, f_{r-1}\} \subseteq E(F_1) \setminus E(F_2)$. ■

Based on the above observations and the Matroid Intersection Algorithm, we can formulate the following algorithm for finding an optimal forest in a graph $G = (V, E)$ with a coloring $c : E \rightarrow \mathbb{N}$.

Algorithm

- Step 0 (Start)
 - Let F be any forest of G with $c(F) = |E(F)|$, possibly with $E(F) = \emptyset$.
 - $A := \emptyset, Q := \emptyset, T := \emptyset$.

- Step 1 (Greedy augmentation)
 For all $e_i \in E \setminus E(F)$:
 if $c(e_i) \notin C(F)$ and $F + e_i$ is a forest, then $F := F + e_i$.
- Step 2 (Construction of the auxiliary digraph $D(F) = (E, A)$)
 For all $e_i \in E \setminus E(F)$:
 if $c(e_i) \notin C(F)$, then $Q := Q \cup \{e_i\}$; else for all e_j in F with $c(e_j) = c(e_i)$: $A := A \cup \{(e_j, e_i)\}$;
 if $F + e_i$ is a forest, then $T := T \cup \{e_i\}$; else for all e_j in $C_i - e_i$:
 $A := A \cup \{(e_i, e_j)\}$ (where C_i is the unique cycle in $F + e_i$).
- Step 3 (Augmentation)
 In $D(F)$, search for a shortest directed path $P(e, e')$ from an element $e \in Q$ to an element $e' \in T$ (by breadth-first search), put $F := F \oplus P(e, e')$, and go to Step 0. Stop if there is no such path; then F is an optimal forest of G .

The correctness of the algorithm is obvious, since it is a translation from the Matroid Intersection Algorithm. The complexity of the algorithm depends on the implementation, but it can be implemented to have an overall running time $O(n^4)$, where n is the number of vertices of G . In fact, for the time bound, we can have at most n augmentations. For each augmentation, the time requirements are dominated by the construction of the auxiliary digraph $D(F)$. For each $e_i \in E \setminus E(F)$, we find in F the edges with the same color as e_i , or a cycle C_i in $F + e_i$. Thus, the construction of $D(F)$, Q , and T can be carried out in $O(|E|n)$ time. Finally, searching the digraph for $P(e, e')$ takes $O(|E|n)$ time. Therefore, the total complexity is $O(n^2|E|)$, or $O(n^4)$.

Example. Consider the graph G of Figure 1a, in which the thicker edges belong to the forest F of G . We start our algorithm with F . Clearly $c(F) = |E(F)| = 4$. In Step 1 no greedy augmentation takes place. In Step 2 we obtain the auxiliary digraph $D(F)$ of Figure 1b, in which the vertices are labeled according to the color of the corresponding edges in G , with a subscript F added if and only if the corresponding edge belongs to F . In Step 3 we obtain $52_F 24_F 4$ as a shortest directed path in $D(F)$ between a vertex of Q and T . This leads to the spanning tree F' of G indicated by the thicker edges in Figure 1c.

Perhaps it is worth mentioning that, constructing a similar edge-colored graph by connecting $k \geq 3$ disjoint triangles in a ‘path fashion’ and assigning suitable colors to the edges, we can obtain a graph G on $3k$ vertices for which we need a directed path P of length $2k$ in $D(F)$ of G to obtain an augmentation in Step 3 of our algorithm. We can choose the colors in such

a way that P is not unique.

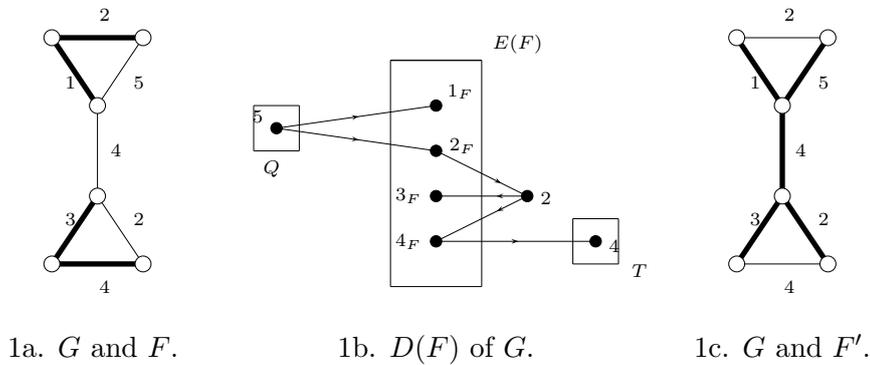


Figure 1. Construction of $D(F)$ to find an augmentation.

3. MINIMUM TREES

As we have seen in the previous section, there is a polynomial algorithm for finding a maximum tree of a connected colored graph. The situation is quite different for the related problem of finding a minimum tree. As we will show in the sequel, this problem is NP -hard, implying there is no polynomial algorithm for this problem unless $P = NP$.

We start with some additional terminology. Given a graph $G = (V, E)$, a set $S \subseteq V$ is called a *dominating set* if every vertex of $V \setminus S$ has a neighbor in S .

According to [2, GT2, p. 190] the following problem is NP -complete.

Problem 1

INSTANCE: Graph $G = (V, E)$, positive integer $K \leq |V|$.

QUESTION: Is there a dominating set of cardinality K or less for G ?

We will use this result to show that the following problem is also NP -complete.

Problem 2

INSTANCE: Graph $G = (V, E)$, coloring $c : E \rightarrow \mathbb{N}$, positive integer $K \leq c(G)$.

QUESTION: Is there a spanning tree T with $c(T) \leq K$ for G ?

Theorem 7. *Problem 2 is NP-complete.*

Proof. It is not difficult to see that Problem 2 is in *NP*. One way is to see that a nondeterministic algorithm needs only guess a spanning tree T of G and check in linear time whether $c(T) \leq K$.

Next we transform Problem 1 to Problem 2. Let $G = (V, E)$ be a graph and $K \leq |V|$ an integer. We shall construct a new graph $G' = (V', E')$ and a coloring $c : E' \rightarrow \mathbb{N}$ such that G' has a spanning tree T with $c(T) \leq K + 1$ if and only if G has a dominating set S with $|S| \leq K$, and such that the construction can be accomplished in polynomial time. Let $V = \{v_1, \dots, v_n\}$. Now construct G' as follows. Start with the vertex set V and for each $v_i \in V$ add a new vertex u_i and an edge $v_i u_i$ with color $c(v_i u_i) = i$. Moreover, for each edge $v_i v_j \in E$, add a new vertex u_{ij} and an edge $v_i u_{ij}$ with color $c(v_i u_{ij}) = j$. Finally add a new vertex v and all edges from v to the u -vertices, with the same color $n + 1$. Clearly the construction can be accomplished in polynomial time. We now finish the proof by showing that G has a dominating set S with $|S| \leq K$ if and only if G' has a spanning tree T with $c(T) \leq K + 1$.

First assume G has a dominating set S with $|S| \leq K$. For each vertex $v_i \in V$ choose one vertex $v_j \in S$ such that either $v_i = v_j$ or $v_i v_j \in E$. In G' choose the corresponding edge $v_i u_i$ with color $i = j$ if $v_i = v_j$, or the edge $v_i u_{ij}$ with color j if $v_i v_j \in E$. Hence in G' we choose precisely n edges with at most $|S| \leq K$ different colors. Adding all edges with color $n + 1$ we obtain the edge set of a spanning tree of G' with at most $K + 1$ different colors.

Next assume G' has a spanning tree T with $c(T) \leq K + 1$. Clearly T contains an edge with color $n + 1$. For all $v_i \in V'$, the tree T contains precisely one path of length 2 from v to v_i . Replace any other edges incident with v_i in T by edges incident with v and the same other end vertex. Then the new tree T' of G' also has $c(T') \leq K + 1$ and it is not difficult to see that the colors of all tree edges not incident with v correspond to a dominating set S of G with $|S| = c(T') - 1 \leq K$: if the edge in T' incident with v_i has color j , then v_i is dominated by $v_j \in S$. This completes the proof. ■

The following consequence of Theorem 7 is immediate.

Corollary 8. *Finding a minimum tree in a connected colored graph is an NP-hard problem.*

Acknowledgments

We thank the anonymous referee for pointing out a mistake in the first version of the algorithm.

REFERENCES

- [1] J.A. Bondy and U.S.R. Murty, Graph Theory with Applications (MacMillan-Elsevier, London-New York, 1976).
- [2] M.R. Garey and D.S. Johnson, Computers and Intractability (Freeman, New York, 1979).
- [3] E.L. Lawler, Combinatorial Optimization, Networks and Matroids (Holt, Rinehart and Winston, New York, 1976).
- [4] D.J.A. Welsh, Matroid Theory (Academic Press, London-New York-San Francisco, 1976).

Received 23 November 1995

Revised 30 January 1997