Maximum colored trees in edge-colored graphs

V.Borozan* W. Fernandez de La Vega* Y. Manoussakis* C. Martinhon†
R. Muthu* R. Saad‡§

Abstract

We consider maximum properly edge-colored trees in edge-colored graphs G^c. We also consider the problem where, given a vertex r, determine whether the graph has a spanning tree rooted at r, such that all root-to-leaf paths are properly colored. We consider these problems from graph-theoretic as well as algorithmic viewpoints. We prove their optimization versions to be NP-hard in general and provide algorithms for graphs without properly edge-colored cycles. We also derive some nonapproximability bounds. A study of the trends random graphs display with regard to the presence of properly edge-colored spanning trees is presented.

Keywords: c-edge-colored graph, colored spanning tree, maximum colored tree.

1 Introduction, Notation and Terminology

Here we consider the problem of properly colored spanning trees of graphs colored with any number of colors, and also a related question which allows a slight relaxation. We look at these questions from a graph-theoretic as well as an algorithmic perspective. They are also useful in many practical applications. Much of the earlier related work was with *rainbow* spanning trees, i.e., spanning trees in which each pair of edges differ in color (see [3, 7, 8, 17]). Some work on algorithmic aspects of paths and cycles in edge-colored graphs can be found in [2, 4, 5, 15, 16]. A recent survey paper on heterochromatic graphs (rainbow colored graphs) can be found in [14].

Formally, let $\chi_c = \{1, 2, \ldots, c\}$ be a given set of colors, $c \geq 2$. Throughout the paper, G^c denotes an edge-colored simple graph, where each edge is assigned some color $i \in \chi_c$. The vertex and edge-sets of G^c are denoted $V(G^c)$ and $E(G^c)$, respectively. The order n, of G^c, is the number of its vertices. The size m, of G^c, is the number of its edges. For a given color i, $E^i(G^c)$ denotes the set of edges of G^c colored i. When no confusion arises, we write V, E and E^i instead of $V(G^c), E(G^c)$ and $E^i(G^c)$, respectively. For edge-colored complete graphs, we write K_n^c instead of G^c. If H is a subgraph of G^c, then $N_H^i(x)$ denotes the set of vertices of H, joined to x with an edge in color i. The colored i-degree of x in H, denoted by $d_H^i(x)$, corresponds to $|N_H^i(x)|$, i.e., the cardinality of $N_H^i(x)$. Whenever $H \cong G^c$, for simplicity, we write $N^i(x)$ (resp. $d^i(x)$) instead of $N_{G^c}^i(x)$ (resp. $d_{G^c}^i(x)$).

An edge between two vertices x and y is denoted by xy, its color by $c(xy)$ and its weight (if any) by $weight(xy)$. The weight of a subgraph is the sum of the weights of its edges. A subgraph of G^c is said

*University of Paris-XI, OrsayLRI, Bât. 490, 91405 Orsay Cedex, France

†Fluminense Federal University, Institut of Computation, Niteroi, RJ, 24210-240, Brazil. Sponsored by Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq

‡CMR Saint-Jean, Campus Saint-Jean, C.P.100 Succursale Bureau-chef, Richelain (QC) J0J 1R0, Canada.

§E-mails: yannis@lri.fr, mart@dcc.ic.uff.br, muthu@lri.fr, r.saad@st-jean.rmc.ca

to be properly edge-colored (or *just* proper) if any two incident edges in this subgraph differ in color. A proper path (trail) is any proper subgraph whose underlying non-colored graph is a path (trail). The length of a path (trail) is the number of its edges. A proper cycle in G^c is a proper subgraph whose underlying non-colored graph is a cycle. An edge-colored graph is said to be acyclic if it does not contain proper cycles.

A tree in G^c is a subgraph such that its underlying non-colored graph is connected and acyclic. A spanning tree is one covering all vertices of G^c. Following the convention introduced in the previous paragraph, a proper tree is one such that no two incident edges are of the same color. A tree T in G^c with fixed root r is said to be weakly proper if every path in T, from the root r to any leaf, is a proper one. To facilitate discussions, in the sequel, a weakly proper tree will be called a weak tree. Notice that in the case of weak trees, incident edges may have the same color, while this may not happen for proper trees. We use PT and WT to denote respectively, proper tree and weak tree. PST and WST refer to the corresponding spanning trees, while MPT and MWT represent maximum proper and weak trees, respectively. A proper spanning subgraph, whose underlying non-colored graph is a forest is called a proper spanning forest PSF.

A tree-cycle factor $C_0, C_1, C_2, \ldots, C_k$ is a collection of $k+1$ pairwise vertex-disjoint subgraphs of G^c such that C_0 is a proper tree and all other subgraphs C_i, $i \neq 0$, are properly edge-colored cycles in G^c satisfying $\bigcup_i V(C_i) = V(G^c)$. Similarly, a forest-cycle factor $T_1, \ldots, T_l, C_1, \ldots, C_k$ is a collection of $l+k$ pairwise vertex-disjoint subgraphs of G^c, such that $T_1, \ldots, T_l$ are proper trees and the remaining subgraphs $C_1, \ldots, C_k$, are proper cycles in G^c, satisfying $\bigcup_i V(C_i) = V(G^c)$. We utilise these concepts to develop some of our results in Section 5.

We state below a theorem due to Yeo [18] which we use in Section 3. This theorem characterizes precisely the class of edge-colored graphs, on any number of colors, which contain no properly colored cycles. A simpler version for graphs whose edges are colored using two colors was obtained by Grossman and Häggkvist [12].

Theorem 1.1 (Yeo). *If G^c is an edge-colored acyclic graph then it has a vertex u, such that the edges between u and any component of $G^c \setminus \{u\}$ are monochromatic.*

The paper is organised as follows. In Section 2, we prove the NP-completeness of various problems of colored trees in edge-colored graphs, and also derive some non-approximability bounds. In Section 3 we provide algorithms to solve the PST and WST problems on acyclic graphs. Our algorithms in that section yield maximum sized trees, in case a spanning one does not exist. In Section 4 we provide an algorithm that produces a PSF with the maximum possible number of edges. The difference from the earlier section is only in the case when the graph does not contain a PST. In Section 5 we present some precise mathematical characterizations of edge-colored complete graphs which contain a PST. Section 6 studies the trends random graphs display with respect to containment of a PST. We conclude with a summary and open problems in Section 7.

2 NP-completeness, nonapproximability results for the Colored Tree Problems

Initially consider a c-edge-colored graph G^c on n vertices. If the number of colors c is constant, it is easy to see that the PST problem is NP-hard since it generalises the Maximum Degree-Constrained Tree problem (MDCT) (see [11]), which in turn generalizes the Longest Path problem. In this case, the number of edges with different colors incident to node $v \in V$, denotes the maximum degree d_v associated with node v in the MDCT problem. The next result shows that the proper tree problem remains NP-hard even for graphs with $c = \Omega(n^2)$ colors.

Theorem 2.1. *The maximum proper tree problem on G^c is NP-hard even for $c = \Omega(n^2)$.*

Proof. Let G^c be an instance of the MPT problem with n nodes and c colors. Construct a complete graph $K_n^{c'}$ with n nodes and color each of its edges with a different color. Add new edges between some fixed vertex of $K_n^{c'}$ and all vertices of G^c and give them all the same color different from those used on G^c and $K_n^{c'}$. Clearly, this new graph has $c = \Omega(n^2)$ colors and contains a proper tree on $n + t$ nodes if and only if G^c has one on t nodes. □

Next we consider the MWT problem.

Theorem 2.2. *Given a vertex r in G^c, $c \geq 2$, the* MWT *problem rooted at r is NP-hard.*

Proof. The MWT problem obviously belongs to NP. To show that MWT is NP-hard we construct a reduction from the MAX-3-SAT problem as follows. Consider a boolean expression B in the CNF with variables $x_1, \ldots, x_s$ and clauses $c_1, \ldots, c_t$. In addition, suppose that B constains exactly 3 literals per clause (actually, we may also consider clauses of arbitrary size). We show how to construct a 2-edge-colored graph G^c associated with any such formula B, such that, there exists a truth assignment to the variables of B satisfying t' clauses if and only if G contains a WT with root r, covering $2s + t' + 1$ nodes. The vertex set $V(G^c)$ consists of $2s + t + 1$ nodes and is defined as:

$V = \{r, a_{11}, a_{12}, a_{21}, a_{22}, \ldots, a_{s1}, a_{s2}, c'_1, \ldots, c'_t\}$.

The vertex r is the root, vertices a_{i1}, a_{i2} for $i = 1, \ldots, s$ are associated respectively with variables x_i of B, and all vertices c'_j (for $j = 1, \ldots, t$) are associated with the clauses c_j of B.

The edge set $E(G^c)$ is constructed in the following way. All edges between the root r and the vertices a_{i1}, a_{i2}, for each $i \in \{1, \ldots, s\}$, are added with color blue. Each pair a_{i1}, a_{i2}, for each $i \in \{1, \ldots, s\}$, is connected by a red edge. For each occurrence of x_i in the positive form in the clause c_j we add a blue edge $a_{i2}c'_j$. Analogously, for each ocurrence of x_i in the negative form in the clause c_j we add a red edge $a_{i2}c'_j$. See the example of Figure 1 (a).

Therefore given a truth assignment for B, we obtain a Weak Tree T in G^c as follows. For each variable x_i which is true, we select edges ra_{i1}, $a_{i1}a_{i2}$ and all blue edges incident to a_{i2}. Similarly, for each variable x_i which is false, we select edges ra_{i2} and all red edges incident to a_{i2}.

Conversely, if T is a Weak Tree rooted at r covering $t' + 2s + 1$ nodes, an assigment for all variables of B is obtained as follows. Observe first, that by our construction, every weak tree rooted at r can be

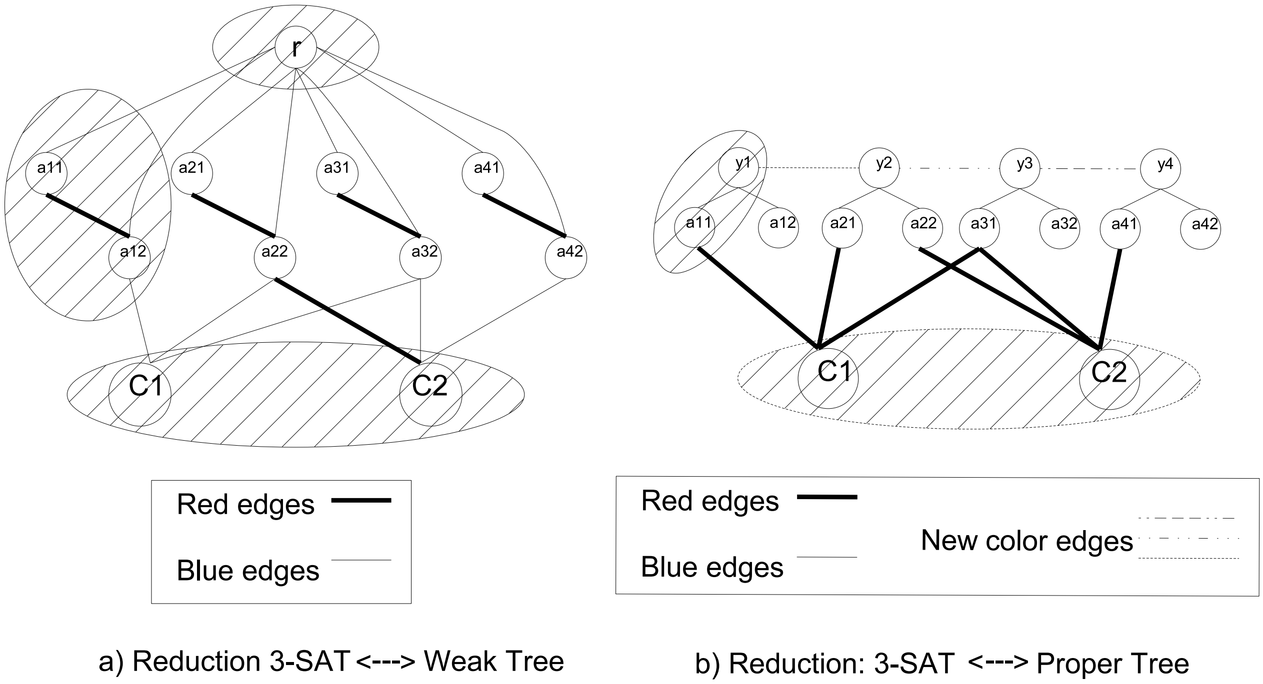


Figure 1: Redution 3-SAT formula $B = (x_1 \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee x_3 \vee x_4)$ to WST (fig.(a)) and PST (fig.(b)).

extended to include all the nodes $a_{i1}, a_{i2}, i \in \{1, \ldots, s\}$, and thus covers at least $2s + 1$ nodes. In such an extended tree, if the last edge on the path from the root r to some c'_j, namely $a_{i2}c'_j$ is colored blue, then the corresponding variable x_i is set to true, otherwise it is set to false. $\square$

In the rest of this section, we present nonapproximability results for the MWT and MPT problems. Recall that in the MWT, the objective is to maximize the number of nodes covered by a tree T with root r. Initially, consider the following auxiliary result relating the maximum number of covered vertices $Opt(G)$ and the maximum number of satisfied clauses $Opt(B)$.

Lemma 2.3. $Opt(G) = Opt(B) + 2s + 1$.

Proof. Consider the construction described in the proof of the previous theorem, correlating a 3-CNF-formula and a corresponding graph. As in that proof, for every assignment of values to the variables of the formula, we have a weak tree rooted at r, covering all the $a_{11}, a_{12}, \ldots, a_{s1}, a_{s2}$ and all c'_j associated with the set of satisfied clauses of B. Conversely, given any weak tree rooted at r, we obtain an assignment of values to the variables in B, such that we satisfy at least as many clauses as the number of paths from r to c'_j. It follows that $Opt(G) = Opt(B) + 2s + 1$. $\square$

Theorem 2.4. *The* MWT *problem is nonapproximable within* $63/64 + \epsilon$*, for* $\epsilon > 0$*, unless* $P = NP$*.*

Proof. Again, consider a boolean expression B with s variables and t clauses. In addition, suppose that B constains exactly 3 literals per clause. Then, by the gap reduction technique we prove that:

1) if $Opt(B) \geq t$ then $Opt(G) \geq f(s,t)$, where $f(s,t) = 2s + t + 1$ and,
2) if $Opt(B) < (7/8 + \epsilon)t$ then $Opt(G) < (63/64 + \epsilon)f(s,t)$, for $\epsilon > 0$.
Observe, in this case, that we are using a classical negative result for the MAX-3-SAT problem. As proved in Hastad [13], the MAX-3-SAT cannot be approximated within $7/8 + \epsilon$, unless P=NP.

The first condition follows directly since $Opt(G) = Opt(B) + 2s + 1$ by Lemma 2.3. Thus, consider $Opt(B) < (7/8+\epsilon)t$. From this inequality, and Lemma 2.3, it follows that $Opt(G) < (7/8+\epsilon)t+2s+1 = (7/8 + \epsilon)t + f(s,t) - t$. Therefore $Opt(G) < (\epsilon - 1/8)t + f(s,t)$.

Now, from the definition of the 3-SAT problem it follows that $3 \leq s \leq 3t$. Therefore, $f(s,t) \leq 2s + t + s/3 = 7s/3 + t \leq 8t$. Thus, for $0 < \epsilon < 1/8$, it follows that $Opt(G) < (\epsilon - 1/8)f(s,t)/8 + f(s,t)$ $= (\epsilon/8 - 1/64 + 1)f(s,t)$. Finally, $Opt(G) < (63/64 + \epsilon')f(s,t)$ where $\epsilon' = \epsilon/8$. □

Now, we deal with the MPT problem.

Lemma 2.5. $Opt(G) = Opt(B) + 2s$

Proof. We construuct a tree associated with any formula in 3-CNF as follows. We have three vertices y_i, a_{i1}, a_{i2}, for each variable $x_i, 1 \leq i \leq s$, in the formula. We also have vertices $c'_1, \ldots, c'_t$ corresponding to each of the t clauses in the formula. We have a red edge between a_{i1} and c'_i if variable x_j occurs in positive form in the clause C_j. In case the variable x_i occurs in the negative form in clause C_j, we put a red edge between a_{i2} and c'_j. We have blue edges between y_i and a_{i1} and y_i and a_{i2}, for each $i \in \{1, \ldots, s\}$. The vertex pairs y_i and y_{i+1} are connected each by different colors other than red and blue for each $i \in \{1, \ldots, s-1\}$.

It can be easily deduced by looking at Figure 1 (b), that for any $i \in \{1, \ldots, s\}$, at most two of y_i, a_{i1}, a_{i2} can be covered by a proper tree. The vertices corresponding to the satisfiied clauses can always be covered. Thus, clearly $Opt(G) \geq Opt(B) + 2s$.

Now suppose that we are able to also cover a vertex corresponding to an unsatisfied clause. This implies that the vertex, say c'_j is connected to some a_{i1} or a_{i2}, only one of which is present in the tree. Also, every satisfied clause is covered, and this is independent of the assignment made to the variable x_i. This means we can satisfy an extra clause contradicting the optimality of the assignment. Thus $Opt(G) \leq Opt(B) + 2s$. □

Theorem 2.6. *The* MPT *problem is non approximable within* $55/56 + \epsilon$*, for* $\epsilon > 0$*, unless* $P = NP$*.*

Proof. Consider a boolean expression B with s variables, t clauses and exactly 3 literals per clause. We want to show that:
1) if $Opt(B) \geq t$, then $Opt(G) \geq f(s,t)$, where $f(s,t) = 2s + t$ and,
2) if $Opt(B) < (7/8 + \epsilon)t$, then $Opt(G) < (53/54 + \epsilon)f(s,t)$, for $\epsilon > 0$.

Case (1) follows immediately from Lemma 2.5.

For case (2), we again apply the gap reduction technique using the MAX-3-SAT problem. Clearly, $Opt(G) = Opt(B) + 2s < (7/8 + \epsilon)t + 2s = (7/8 + \epsilon)t + (f(s,t) - t) = (\epsilon - 1/8)t + f(s,t)$. Now, from the definition of the 3-SAT problem it follows that $3 \leq s \leq 3t$. Therefore, $f(s,t) \leq 2s + t \leq 6t + t = 7t$.

Thus, for $0 < \epsilon < 1/8$, it follows that $Opt(G) \leq (\epsilon - 1/8)f(s,t)/7 + f(s,t) = (\epsilon/7 - 1/56)f(s,t) + f(s,t)$. Finally, $Opt(G) < (55/56 + \epsilon')f(s,t)$ where $\epsilon' = \epsilon/7$. $\square$

3 Maximum colored trees in acyclic edge-colored graphs

As mentioned and proved in an earlier section, the problem of determining whether an edge-colored graph has a PST is NP-Complete. This is also the case with the WST problem. In this section, we prove that these problems can be solved efficiently when we restrict our attention to the class of colored acyclic graphs. When there is no spanning tree, our algorithms can be directly adapted to find a tree of maximum cardinality.

While some ideas are common, the proofs and conditions for the PST and WST problems on acyclic edge-colored graphs, differ significantly. Consequently, we have divided the presentation into two seperate subsections, for the sake of clarity.

3.1 PST

We begin with a very simple theorem which shows, when a given acyclic graph has a PST. The algorithm which results from this proof has complexity $O(cn^{2.5})$. Subsequently, we present an alternative, more complicated, proof *but* with a much better algorithmic complexity of $O(n^{2.5})$. Since that proof is more involved, we divide it into two parts. We prove the result first for acyclic complete graphs and then generalize it to all acyclic graphs. In addition to the superior running time, in case the graph does not contain a PST this latter algorithm can be modified to produce a MPT. Finally, it is possible that the ideas used in the latter results can be adapted to develop approximation algorithms for the MPT problem on general graphs.

Theorem 3.1. *An acyclic graph G^c has a* PST *if and only if the union of maximum matchings of each of the colors in χ_c contains exactly $(n-1)$ edges.*

Proof. We use, here, $\mathcal{M}_i$ to denote a maximum matching in color i and $\mathcal{T}_i$ to denote the edge set of color i in a PST. Any subgraph induced by $\bigcup_{i=1}^{c} \mathcal{M}_i$ is clearly a proper one. Thus if $|\bigcup_{i=1}^{c} \mathcal{M}_i| \geq n$, then the subgraph contains a proper cycle, contradicting the acyclicity of G^c. Now suppose $|\bigcup_{i=1}^{c} \mathcal{M}_i| < (n-1)$. Clearly, if G^c has a PST then for some color i, $|\mathcal{M}_i| < |\mathcal{T}_i|$, since $|\bigcup_{i=1}^{c} \mathcal{M}_i| < (n-1) = |\bigcup_{i=1}^{c} \mathcal{T}_i|$. This is a contradiction of the maximality of $\mathcal{M}_i$. $\square$

We now provide alternative results specifying conditions under which an acyclic graph has a PST. We also develop an efficient algorithm to construct one if it exists. If the given graph does not have an PST, then our algorithm can be adapted in a straightforward manner to produce a *proper spanning forest* (PSF) with the smallest possible number of components (trees). Among others, this forest also contains an MPT. This adaptation is just a greedy procedure, which finds a tree of largest size, and then repeats the procedure on the subgraph induced by the residual vertices. We first prove all these results for acyclic edge-colored *complete* graphs and then extend them for *general* acyclic edge-colored graphs.

Let $G_1, \ldots, G_p$, $p \geq 1$ be the components of $G^c \setminus \{u\}$. By Theorem 1.1 if G^c is acyclic, for some vertex u, the edges between u and G_i are monochromatic, for all $i \in \{1, \ldots, p\}$. We call such a vertex u, a *yeo-vertex*. We refer to a yeo-vertex which uses different colors for the edges to different components as a *rainbow-yeo-vertex*. We call any other yeo-vertex *non-rainbow-yeo*, if the need arises to distinguish them.

We first prove an elementary result on colored acyclic graphs, which we use later.

Lemma 3.2. *For any colored acyclic graph G^c, $c \leq n-1$.*

Proof. Assume, G^c is an acyclic colored graph with $c \geq n$. Consider a subgraph induced by a set of n edges of distinct colors. Since, this subgraph has at most n vertices, the presence of n edges implies a cycle. It is a properly colored cycle as the set of selected edges are all of distinct colors. This contradicts the assumption that G^c is acyclic. □

Based on Theorem 1.1, we state below an easy lemma.

Lemma 3.3. *If an acyclic edge-colored graph contains a non-rainbow-yeo vertex, then it has no* PST.

Proof. This is because there are at least two components in the graph obtained by deleting this vertex which must be connected to each other in a potential PST by a path of length two through this vertex. These edges are necessarily of the same color and hence the resulting spanning tree is not a proper one. □

In Theorem 3.5 we characterize acyclic edge-colored complete graphs having proper spanning trees. With a view to proving that result, we first give a structural characterization of acyclic edge colored complete graphs.

Proposition 3.4. *An acyclic edge-colored complete graph consists of a unique sequence of induced cliques (called blocks) $\mathcal{B}_1, \ldots, \mathcal{B}_k$ such that:*
i) each $\mathcal{B}_i$ is a maximal monochromatic clique in the subgraph induced by $\bigcup_{j=i}^{k} \mathcal{B}_j$,
ii) all edges between each $\mathcal{B}_i$ and $\bigcup_{j=i+1}^{k} \mathcal{B}_j$ are monochromatic in the same color as the edges of $\mathcal{B}_i$.

Proof. For a complete graph, the induced subgraph obtained by deleting any vertex has exactly one component, because it is again a complete graph with one fewer vertex. Thus for any such graph which is acyclic, a yeo-vertex is necessarily a rainbow-yeo-vertex. In other words it is a monochromatic vertex (all edges incident to it are of the same color). Thus if there is more than one yeo-vertex in the given graph, then they are monochromatic in the same color, since they also have a direct edge between themselves (as the graph is complete).

Thus, the entire set of yeo-vertices of an acyclic edge-colored complete graph induce a monochromatic clique, and additionally, they are all connected to every other vertex in the graph by edges of this same color.

Note that if we remove this entire clique of vertices, the resultant smaller graph is also an acyclic edge-colored complete graph on fewer vertices.

Thus, it follows, that there exists a similar clique (of a color distinct from the one just removed) in the remainder of the graph. We delete this clique and place the constituent vertices in a second group. We repeat this procedure of collecting vertices in this type of groups and obtaining a smaller graph until all the vertices have been removed. The partial order we use in finding a PST, is precisely the order determined by these groupings.

□

We now state a theorem, which given an acyclic edge-colored complete graph, determines whether or not it contains a PST. The decision is made on the basis of the cardinalities of the blocks $\mathcal{B}_1, \ldots, \mathcal{B}_k$ in the ordering described in Proposition 3.4. To facilitate the description of our results we let the color of the ith block be denoted c_i. For any fixed i, $1 \leq i \leq k$, we let t_i denote the total number of vertices in the set of blocks from $\mathcal{B}_i$ to $\mathcal{B}_k$, i.e., $t_i = |V(\mathcal{B}_i)| + \ldots + |V(\mathcal{B}_k)|$. We define $\mathcal{S}_i = \{\mathcal{B}_j | i \leq j \leq k; c_j = c_i\}$. Define $\mathcal{S}_i^{'} = \bigcup x \in \mathcal{S}_i$. Now, we define $t_i^{c_i} = \left|\mathcal{S}_i^{'}\right|$

With the terminology above, we may state the following.

Theorem 3.5. *An acyclic edge-colored complete graph has a* PST *if and only if :*
i) The last block $\mathcal{B}_k$ has two vertices, and
ii) for each $i < k$, if block $\mathcal{B}_i$ has the same color as the last one $\mathcal{B}_k$, then $t_i^{c_i} \leq \frac{t_i}{2} + 2$. Else $t_i^{c_i} \leq \frac{t_i}{2}$.

Proof. First assume that the given graph has a PST, and let $\mathcal{T}$ represent any such possible PST. Notice that in the partial ordering of the vertices according to Proposition 3.4, the group of vertices in block $\mathcal{B}_i$ are necessarily leaves in the subtree induced by the vertices of $\bigcup_{j=i}^{k} \mathcal{B}_j$. Moreover, each of these vertices are attached to distinct internal vertices of the subtree (since all edges incident to these vertices in the subgraph induced by $\bigcup_{j=i}^{k} \mathcal{B}_j$ are of the same color). A necessary condition is, thus, that there is a proper tree $\mathcal{T}_i$ spanning $\bigcup_{j=i}^{k} \mathcal{B}_j$ for each $i \in \{1, \ldots, k\}$.

Thus, in order to add the vertices of $\mathcal{B}_{i+1}$ as leaves to the subtree $\mathcal{T}_i$, to construct $\mathcal{T}_{i+1}$, there must be at least $|\mathcal{B}_{i+1}|$ vertices not having any edge of color c_i incident to them in $\mathcal{T}_i$. However, all vertices, in blocks $i, \ldots, k$ whose color is c_{i+1} necessarily use an edge of this color in the tree. Additionally, each of them (except if they are in the last block $\mathcal{B}_k$), are also appended to some other vertex in the tree $\mathcal{T}_i$, with color c_i. The vertices of $\mathcal{B}_{i+1}$ must therefore be attached as leaves to $|\mathcal{B}_{i+1}|$ distinct vertices different from the vertices accounted for above. This proves the second condition of the theorem.

The first condition states that the last block $\mathcal{B}_k$ must be of size exactly two. From the earlier arguments, these vertices must contain a proper tree spanning them. Since they induce a monochromatic clique, it is possible only if there are exactly two vertices.

Conversely, if the conditions are on the cardinalities are satisfied, we show that a PST exists. We construct a partial tree consisting of the edge between the two vertices of the last block $\mathcal{B}_k$. Subsequently, we consider in order the vertices of the blocks $\mathcal{B}_{k-1}, \ldots, \mathcal{B}_1$ and at each stage, we pick a set of vertices of size $|\mathcal{B}_i|$ in the partial tree $\mathcal{T}_{i-1}$, which are free of the color c_i and attach the vertices of $\mathcal{B}_i$ as leaves to distinct vertices in this set, to augment the tree to $\mathcal{T}_i$. □

Now based on Theorem 3.5 we describe, an algorithm which computes a PST for any acyclic edge-colored complete graph, if one exists. If one does not exist then the algorithm can easily be adapted to produce a PSF with the minimum possible number of components.

We now describe our algorithm to construct the PST.

Algorithm 1 PST for K_n^c-acyclic

1: **compute** the order described above
2: **if** last block $\mathcal{B}_k$ has more than two vertices **then return** "No PST"
3: **if** last block $\mathcal{B}_k$ has two vertices **then** connect the two vertices of $\mathcal{B}_k$ to get an initial Proper Tree
4: **for** $i = k - 1$ to 1 **do**
5:    **if** condition 2 of Lemma 3.5 is true **then**
6:       join the vertices of $\mathcal{B}_i$ as leaves, to distinct vertices already incorporated in the tree which have not used an edge of color c_i in the partial proper tree obtained in the previous iteration.
7:    **else**
8:       return "NO PST"
9:    **end if**
10: **end for**
11: **return** the PST

The running time of the above algorithm is $O(n^2)$. This is the cost using Breadth-First-Search (BFS), to compute the order in Step 1. The rest of the algorithm consists in finding for each new block of vertices, a certain type of matching saturating them, which enables them to be attached as leaves of the partial tree. If there is no PST, our algorithm finds a maximum size proper tree, and then repeats the procedure on the subgraph induced by the residual vertices, resulting in an SSF with the fewest possible number of trees.

We now show how Theorem 3.5 as well as the algorithm can be extended to find a PST in a general acyclic graph, if one exists. We define a canonical auxiliary tree associated with any acyclic edge-colored graph. It is similar to the linear order for acyclic edge-colored complete graphs. We assume that the graph has no non-rainbow-yeo-vertex. The root of the auxiliary tree is any yeo-vertex of the graph. The number of children of the root is the number of components resulting from the deletion of this vertex from the graph. The root of the auxiliary tree is connected to each subtree using an edge of the same color as the edge between the original yeo-vertex and the corresponding component. The root of the subtrees are likewise computed recursively. Thus we get an auxiliary rooted edge-colored tree $\mathcal{T}$.

First we need to define some associated auxiliary graphs, which we use to characterize acyclic edge-colored graphs that have a PST.

Definition 3.6 (auxiliary tree)**.** *Given an acyclic edge-colored graph G^c, we define an associated auxiliary tree $\mathcal{T}$ as follows.*
i) $n(\mathcal{T}) = n(G^c)$.
ii) The root $r(\mathcal{T})$ is associated with a yeo-vertex v_0 of G^c.
iii) There is one subtree corresponding to each component $G^c \setminus \{v_0\}$.
iv The roots of these subtrees are computed recursively and are attached as children of v_0.
v) The color of the edge between two nodes of $\mathcal{T}$ is the same as the color of the edges between the corresponding component and yeo-vertex in G^c.
vi) Note that the presence of an edge between two nodes in $\mathcal{T}$ does not imply the presence of an edge between the corresponding vertices in G^c.

We now define a set of bipartite graphs one for each color in $\{1, \dots, c\}$.

Definition 3.7. *Associated with each color $l \in \chi_c$ we have a bipartite graph as follows.*
i)If a node in the auxiliary tree (of Definition 3.6) corresponding to a vertex has a child of color l then it is placed in the left part of the bipartite graph corresponding to color l.
ii) Any other vertex with an edge of color l incident to it in the graph G^c is placed in the right part.
iii) The edges are all those in the original graph of the color l, crossing this partition.

Theorem 3.8. *An acyclic edge-colored graph has a* PST *if and only if for each color in χ_c the corresponding bipartite graph has a matching saturating the vertices of the left partite set.*

Proof. By Lemma 3.3, if there exists a non-rainbow-yeo-vertex, then we immediately conclude that the given graph has no PST. It follows that we need only consider the case where all the yeo-vertices are of the rainbow type. Assume that v_0 is one such vertex. Let the components of $G^c \setminus \{v_0\}$ be $C_1, \dots, C_t$. It is straightforward to verify that G^c has a PST if and only if the subgraph of G^c induced by the vertex set $V(C_i) \cup \{v_0\}$ has a PST for each $i \in \{1, \dots, t\}$.

We are in effect able to use a rainbow-yeo-vertex to divide the problem into smaller and independent subproblems. This immediately suggests a recursive appproach to solving the problem. We use the recursion tree of Definition 3.6 and the corresponding bipartite graphs of Definition 3.7 to divide the problem.

In fact, if any C_i does not have a PST then G^c also does not have a PST. We conclude that G^c has a PST if and only if each C_i has a PST which can be extended to include the vertex v_0. The connection of component C_i to v_0 must be by an edge of color i. The feasibility of this is checked using cardinalities, like in the case of complete acyclic graphs. We do not have a simple linear structure here, unlike in that case, and hence check the condidions using matching in the auxiliary bipartite graph instead. The existence and computation of the matching can be done using any of the standard algorithms. The subtrees rooted at the children of v_0 are computed recursively in the same way.

Thus, using the recursion tree obtained, we construct the bipartite graphs described above and then solve the PST problem by transforming it to a series of matching problems. $\square$

Suppose the graph G^c does not have a PST. This either means that there is no rainbow yeo-vertex, or the cardinality conditions fail.

In the former case, let us denote the components of $G^c \setminus v_0$, by $C_1, \dots, C_t$. We find the maximum proper trees for each C_i. We also compute the maximum proper trees for each $G_i \cup v_0$, which use touch vertex v_0. From the latter set of trees, we pick the ones of largest cardinality corresponding to distinct colors of the edge incident to v_0 and take their union. A maximum proper tree is the largest among the first set of trees and the tree obtained by combining the second set as described above.

The second case is merely a special case of the previous one, wherein, when we compute the tree combining the second sequence, we consider all the components.

The next theorem follows easily from the preceeding analysis and results.

Theorem 3.9. *Given an acyclic edge-colored graph G^c, a* PST*, if any, can be found in G^c in time $O(n^{2.5})$.*

Proof. This cost is dominated by the time to compute matchings in graphs bipartite graphs, which we use in the algorithm for PST. □

3.2 WST

We now show how to construct a WST, if one exists, in an acyclic edge-colored graph G^c, with a given root vertex r. Like we did for the case of the PST, we define here as well, an associated auxiliary tree.

Definition 3.10 (auxiliary tree)**.** *Given an acyclic edge-colored graph G^c, and a specified vertex r, we define an associated auxiliary tree $\mathcal{T}'$ as follows.*
i) The tree $\mathcal{T}$ is computed according to the Definition 3.6.
ii) The tree $\mathcal{T}'$ is then obtained by re-rooting $\mathcal{T}$ at the node R, corresponding to the vertex r in G^c.

Theorem 3.11. *An acyclic edge-colored graph G^c, has a* WST *if and only if in the auxiliary tree $\mathcal{T}'$ every path from the root R to any leaf is properly colored.*

Proof. The above auxiliary tree $\mathcal{T}'$, obtained by re-rooting at the vertex corresponding to the specified root of the WST provides a direct way to compute a WST if one exists. A WST rooted at r is computed for each subgraph induced by $\{r\} \cup C_i$, $i \in \{1, \ldots, t\}$. Here, $C_1, \ldots, C_t$ are the components of $G^c \setminus \{r\}$. These trees are then merely fused together at their only common vertex r, to get a WST. Such a tree is a weak one rooted at r, because in the auxiliary tree any path from R to any leaf is a properly colored one. □

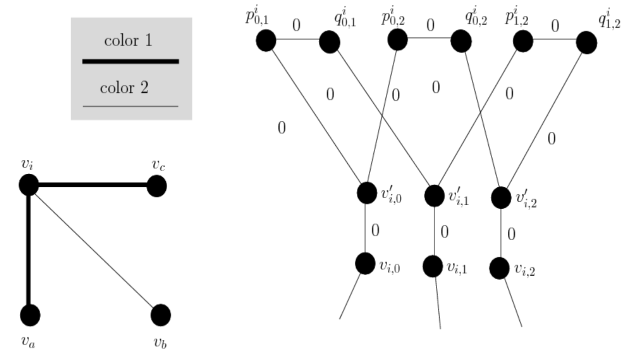
Theorem 3.12. *Given an acyclic edge-colored graph G^c, a* WST*, if any, can be found in G^c in time $O(n^{2.5})$.*

Proof. It is almost identical to the proof of Theorem 3.9. □

4 Proper spanning forests in acyclic graphs

In this section, we show how to find in polynomial time, a maximum proper spanning forest (MPSF) in G^c, the maximality being in terms of the number of edges. If the graph has a PST, then naturally, one is produced by our algorithm for maximum forest. This section differs from the previous one, for graphs which do not contain a PST. In the previous section, the algorithm we describe produces a largest possible tree, whereas, here the algorithm produces a forest with, possibly, many trees, in such a way that the total number of edges is maximized. We conclude the section by showing how to decide, in polynomial time, whether G^c contains a proper spanning forest (PSF), satisfying given degree constraints on each vertex. We also show how to construct the forest if one exists in this latter case.

Basically, for the MPSF problem, the idea is to construct a new colored multigraph $G^{c'}$, with color set $\chi_{c'} = \chi_c \cup \{0\}$ (where 0 is a new color) and an associated non-colored weighted graph G, which always contains a perfect matching. The multigraph $G^{c'}$, has multiplicity at most two. Moreover, the color of two edges between the same pair of vertices always differ. If there are two edges between a pair of vertices, then one of them is always colored 0. We prove that a maximum weight perfect matching in



(a) Neighborhood of v_i in $G^{c'}$ (b) Gadget G_i of G (associated to $v_i \in V(G^{c'})$)

Figure 2: Redution 3-SAT α WST problem. Example using $B = (x_1 \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee x_3 \vee x_4)$

G, whose construction is described below, is associated with a set of properly edge-colored closed trails in $G^{c'}$, such that the number of edges with a color from χ_c is maximized. As a result, deleting all edges colored 0 in this set, directly produces a maximum proper spanning forest in G^c.

$G^{c'}$ is constructed by taking the union of G^c with a complete monochromatic graph (in color 0) on the same vertex set. Let K_n^0 denote the resulting monochromatic complete subgraph of $G^{c'}$ with the edges of color 0. Thus, we have $V(G^{c'}) = V(G^c) = V(K_n^0)$ and $E(G^{c'}) = E(G^c) \cup E(K_n^0)$.

Before we describe the construction of G, we first define gadgets G_i associated with each vertex v_i of G^c, as depicted in the sequel (see Figure 2). We use these gadgets in our construction of G. Formally:

- $V(G_i) = (\bigcup_{\gamma \in \chi_{c'}} \{v_{i,\gamma}, v'_{i,\gamma} : N^{\gamma}_{G^{c'}}(v_i) \neq \varnothing\}) \cup (\bigcup_{\alpha,\beta \in \chi_{c'}} \{p^i_{\alpha,\beta}, q^i_{\alpha,\beta} : \alpha < \beta,\ N^{\alpha}_{G^{c'}}(v_i) \neq \varnothing$ and $N^{\beta}_{G^{c'}}(v_i) \neq \varnothing\})$
- $E(G_i) = (\bigcup_{\gamma \in \chi_{c'}} \{v_{i,\gamma} v'_{i,\gamma} : N^{\gamma}_{G^{c'}}(v_i) \neq \varnothing\}) \cup (\bigcup_{\alpha,\beta \in \chi_{c'}} \{v'_{i,\alpha} p^i_{\alpha,\beta}, p^i_{\alpha,\beta} q^i_{\alpha,\beta}, q^i_{\alpha,\beta} v'_{i,\beta} : \alpha < \beta\})$

Now, the weighted non-colored graph $G = (V', E')$ with $w : E' \to \{0,1\}$ is constructed as follows:

- $V' = \bigcup_{v_i \in V(G^{c'})} V(G_i)$, and
- $E' = (\bigcup_{v_i \in V(G^{c'})} E(G_i)) \cup (\bigcup_{\gamma \in \chi_{c'}} \{v_{i,\gamma} v_{j,\gamma} : v_i v_j \in E^{\gamma}(G^{c'})\})$
- $w(v_{i,\gamma} v_{j,\gamma}) = 1$, for every $v_{i,\gamma} v_{j,\gamma} \in E'$ with $\gamma \in \chi_c$. The remaining edge weights of E' will be settled to 0, *i.e.,* $w(v_{i,0} v_{j,0}) = 0$ for every $v_{i,0} v_{j,0} \in E'$ and $w(xy) = 0$, for every $xy \in E(G_i)$.

After constructing $G^{c'}$ and G as above, we solve the maximum perfect matching problem over G (see [9]). Observe that graph G always contains a perfect matching. To see that, it suffices to exhibit a perfect matching (with weight 0) by only choosing edges of $E(G_i)$. Thus, if M_i denotes a perfect matching in G_i (which is unique in this case), the subset $M = \bigcup_{v_i \in V(G^{c'})} M_i$ obviously defines a perfect matching in G. Therefore, we can establish the following result:

Theorem 4.1. *Let G^c be an acyclic c-edge-colored graph. Then, the maximum proper spanning forest problem can be solved in time $O(n^{7.5})$.*

Proof. Basically, the idea is to prove that maximum proper spanning forests in G^c are associated with maximum perfect matchings in G, and vice-versa. Initially, suppose we have a proper spanning forest T^* in G^c with the maximum number of edges. Let $T_1^*, T_2^*, \ldots, T_k^*$ (for $k \geq 1$) be the subtrees of T^*. Note that, for every subtree T_i^* of T^*, the number of vertices of T_i^* with odd degree is even. As a consequence of that, the total number of vertices with odd degrees in T^* is also even. Let P_{T^*} be this subset of vertices. Now, consider the multigraph $G^{c'}$ as above, obtained after the addition of edges with color 0 to G^c. Further, consider H^0 (with all edges colored 0) the complete subgraph of $G^{c'}$ induced by the vertices of P_{T^*}. Let $M^0 \subseteq E(H^0)$ be an arbitrary perfect matching of H^0 and $H^{c'} = (\overline{V}, \overline{E})$ with $\overline{V} = V(G^{c'})$ and $\overline{E} = M^0 \cup E(T^*)$, the associated subgraph. Note that subgraph $H^{c'}$ is not necessarily connected and all edges incident to $v \in \overline{V}$ have a different color. Further, notice that all vertices of $H^{c'}$ have an even degree and each connected component, say CT_i (for $1 \leq i \leq k'$ and $k' \leq k$), contains a properly edge-colored closed trail, *i.e.*, each CT_i defines an eulerian trail.

Now, given $\overline{E}$, we construct a perfect matching M^* in $G = (V', E')$ as described in the sequel. Initially, we set $M^* = \emptyset$ and add to M^* all edges $v_{i,0}v_{j,0}$ of E' with $v_iv_j \in M^0$. Now, we increase M^* by choosing all edges $v_{i,\gamma}v_{j,\gamma}$ of E' with $v_iv_j \in E^{\gamma}(T^*)$ and $\gamma \in \chi_c$. The remaining edges in the gadgets G_i (with weight 0) are now directly obtained. Note that $c(M^*) = |E(T^*)|$. Finally, we prove that M^* is a maximum perfect matching in G. Suppose, by contradiction, we have some new perfect matching M' with weight $c(M') > c(M^*)$ and an associated proper edge-colored subgraph H' of G^c with $c(H') > |E(T^*)|$. In this case, there are two possibilities: a) If H' defines a new proper spanning forest, this contradicts the fact that T^* is a maximum proper forest in G^c; b) If H' does not define a proper spanning forest, we would have some properly edge-colored cycle in H', contradicting the fact that G^c is acyclic.

Conversely, consider the weighted graph $G = (V', E')$ associated to G^c as above, and M^* a maximum perfect matching with weight $c(M^*)$ in G. Let $\overline{M} = (E' \setminus E(G_i)) \cap M^*$ be a subset of M^*. Now, to obtain $G^{c'}$ from G it suffices to color edges $v_{i,\gamma}v_{j,\gamma}$ of G with $\gamma \in \chi_{c'}$ and contract all gadgets G_i to vertex v_i. Note that all edges of $G^{c'}$ associated to $\overline{M}$ define a subset of proper spanning closed trails in G^c (denoted by $CT_1, \ldots, CT_{k'}$) with the maximum number of edges with colors in χ_c. Hence, since G^c is acyclic, all properly edge-colored cycles in CT_i (for $i \in \{1, \ldots, k'\}$) contain at least one edge-colored 0. After deleting all these edges, one directly obtains a proper spanning forest $T^* = \bigcup_{\ell=1}^{k} T_\ell$ in G^c with weight $c(T^*) = c(M^*)$ and $k \geq k'$. Finally, note that T^* contains a maximum number of edges since no edges with unitary weights in G were eliminated in the process.

Now, we show that the algorithm above has a polynomial time complexity in the order of G^c. Initially, observe that each gadget G_i of G (with $c \geq 2$) contains at most $c(c+1)$ vertices. From Lemma 3.2, we know that $c = O(n)$. Thus, the non-colored graph G contains $\Theta(n^3)$ vertices, in the worst-case. However, note that a maximum perfect matching in any graph G can be obtained in time $O(|V(G)|^{2.5})$ (see [9] for details). Therefore, the algorithm for the proper spanning forest has total complexity equal to $O(n^{7.5})$. ◻

Now, we conclude with the following result regarding proper spanning forests with given degrees.

Theorem 4.2. *Let G^c be a c-edge-colored acyclic graph and $d : V(G^c) \rightarrow \{0, 1, \ldots, n-1\}$, an integer function. In addition, consider $0 \leq d(v_i) \leq |\{\gamma \in \chi_c : N^{\gamma}_{G^c}(v_i) \neq \emptyset\}|$. Then we can find in polynomial time, provided that one exists, a proper spanning forest in G^c satisfying d.*

Proof. Initially, given an acyclic edge-colored graph G^c, we construct a non-colored graph G, as described in the sequel. For each $v_i \in V(G^c)$ with $d(v_i) > 0$ we define gadgets G_i in the following manner:

- $V(G_i) = (\bigcup_{\gamma \in \chi_c} \{v_{i,\gamma}, v'_{i,\gamma} : N^{\gamma}_{G^c}(v_i) \neq \varnothing\}) \cup \{v^i_1, \ldots, v^i_{d(v_i)}\}$
- $E(G_i) = (\bigcup_{\gamma \in \chi_c} \{v_{i,\gamma} v'_{i,\gamma} : N^{\gamma}_{G^c}(v_i) \neq \varnothing\}) \cup (\bigcup_{\gamma \in \chi_c} \{v'_{i,\gamma} v^i_j : j = 1, \ldots, d(v_i)\}$

Above, if $d(v_i) = 0$ for some $v_i \in V(G^c)$, we just set $V(G_i) = (\bigcup_{\gamma \in \chi_c} \{v_{i,\gamma}, v'_{i,\gamma} : N^{\gamma}_{G^c}(v_i) \neq \varnothing\})$ and $E(G_i) = (\bigcup_{\gamma \in \chi_c} \{v_{i,\gamma} v'_{i,\gamma} : N^{\gamma}_{G^c}(v_i) \neq \varnothing\})$.

Now, a non-colored graph $G = (V', E')$ is constructed as follows:

- $V' = \bigcup_{v_i \in V(G^c)} V(G_i)$, and
- $E' = (\bigcup_{v_i \in V(G^c)} E(G_i)) \cup (\bigcup_{\gamma \in \chi_c} \{v_{i,\gamma} v_{j,\gamma} : v_i v_j \in E^{\gamma}(G^c))$

We show that G^c contains a proper spanning forest T satisfying the degree constraint d, if and only if, $G = (V', E')$ contains a perfect matching. Hence, our result follows since the perfect matching problem can be solved polynomial time.

Initially, consider M^*, a perfect matching in G, if any. In this case, we can obtain a proper spanning forest $H^{c'}$ in G^c (for $c' \leq c$) in the following manner. Initially, let $\overline{M} = (E' \setminus E(G_i)) \cap M^*$ be a subset of M^*. Now, we color all edges $v_{i,\gamma} v_{j,\gamma} \in E'$ with color $\gamma \in \chi_c$ and contract all gadgets G_i to vertex v_i. Finally, we construct $H^{c'}$ by choosing all edges of G^c associated to $\overline{M}$. Notice by the construction of G_i, that we have exactly $d(v_i) > 0$ edges incident to v_i in the resulting edge-colored subgraph $H^{c'}$. Further, all edges incident to v_i have a different color and v_i is isolated if $d(v_i) = 0$. Therefore, since G^c is an acyclic edge-colored graph, the subgraph $H^{c'}$ contains no properly colored cycles and defines a proper spanning forest in G^c satisfying d.

Conversely, consider $H^{c'}$ a proper spanning forest in G^c, and the graph G as above. Initially, set $M^* = \emptyset$ in G. Now, we obtain the associated matching M^* in two steps:

(1) For every edge $v_i v_j \in E^{\gamma}(H^{c'})$ with $\gamma \in \chi_c$ we add edges $v_{i,\gamma} v_{j,\gamma}$ to M^*;

(2) The remaining edges of M^* present in the gadgets G_i are now directly obtained.

It is easy to see that M^* constructed as above defines a perfect matching in G. $\square$

5 Proper trees in edge-colored complete graphs

Recall that the WST problem is trivial for complete graphs, since such a graph always has a star rooted at any vertex, constituting a WST. As for the PST problem on edge-colored complete graphs, the NP-completeness proved in Section 2 for general edge-colored graphs, holds here as well. In this section, we prove this hardness result. We provide a nice graph-theoretic characterization for edge-colored complete graphs K_n^c which have proper spanning trees. However, in view of the afore-mentioned hardness result, the conditions implied by this characterization have only mathematical interest, so they cannot be computed in polynomial time.

Recall that when we restrict the focus to acyclic edge-colored complete graphs, the problem becomes tractable, as proved in Section 3. Also, the problem is polynomial for $K_n^c,\ c = 2$. This latter case, is the same as the proper hamiltonian path problem, which is known to be efficiently solvable [5].

Theorem 5.1. *The* PST *is NP-complete for complete graphs K_n^c, colored with $c \geq 3$ colors.*

Proof. Let G^c be an instance of the PST problem with n nodes and $c \geq 2$ colors. Construct a new colored complete graph K_{2n}^{c+1} on $2n$ vertices and $(c+1)$-colors, as follows. Add all edges in the complement graph $\overline{G^c}$ using a new color, and retain the edges of G^c with their original color. Also, use the extra color to form a complete graph on a new set of n vertices as well as a complete bipartite graph between the old vertices and new vertices.

Observe, that all the new vertices, being monochromatic, are necessarily leaves in any PST. Additionally, no two of them may be adjacent to the same vertex, since all edges incident to the entire set of new vertices are of the same color. Thus, it is necessary for the original graph G^c to have a PST. It is also a sufficient condition, since a PST of G^c does not use any edges of the new color, and hence the set of new vertices can be attached as leaves to such a tree to get a PST of K_{2n}^{c+1}.

Since the PST problem is NP-complete for arbitrary graphs colored with two or more colors, it is also NP-complete for complete graphs colored with three or more colors. ☐

The following two lemmas both of algorithmic nature, are simple but of fundamental importance. These lemmas, proved by some authors of this paper, were first announced in [1], but for the sake of completeness, we include their proofs here. By repeated application of the second lemma inductively, we obtain a theorem which provides an interesting mathematical characterization of complete graphs with a PST.

Lemma 5.2. *Let T be a proper tree and C be a properly edge-colored cycle, such that T and C are vertex-disjoint in K_n^c. Assume that for some edge $e = xy$ of T, $c(x, C) = c(e)$ and for some vertex $z \in C,\ c(yz) = c(e)$. Then K_n^c admits a proper tree with vertex set $V(T) \cup V(C)$.*

Proof. Set $C : x_1x_2 \cdots x_i(= z) \cdots x_{k-1}x_kx_1$. Let T_1 (resp. T_2) denote the subtree of $T - e$ with root x (resp. y). Assume first that either $c(x_{i-1}x_i) = c(e)$ or $c(x_ix_{i+1}) = c(e)$, say $c(x_ix_{i+1}) = c(e)$. Then the tree $T_1 \cup [xx_{i+1}x_{i+2} \cdots x_iy] \cup T_2$ is a proper one. Assume next $c(x_{i-1}x_i) \neq c(e)$ and $c(x_ix_{i+1}) \neq c(e)$. Let x_rx_{r+1} be an edge on C (not incident to x_i) such that $c(x_rx_{r+1}) \neq c(e)$. Clearly such an edge exists, since C is proper. Consider first the proper tree $T_1 \cup [xx_{r+1}x_r \cdots x_ix_{i-i} \cdots x_{r+2}]$. Now join T_2 to this tree by using the edge x_iy. The resulting proper tree is the required one. ☐

Lemma 5.3. *Assume that the vertices of K_n^c are covered by a proper tree T and a properly edge-colored cycle C, such that T and C are pairwise vertex-disjoint in K_n^c. Then K_n^c admits a proper spanning tree.*

Proof. Set $C : x_1x_2 \cdots x_i \cdots x_{k-1}x_kx_1$ within a clockwise orientation. Let t_1, t_p be two leaves of T, and let $t_1t_2 \cdots t_{p-1}t_p$ denote the (unique) path from t_1 to t_p on that tree. To facilitate discussions let us set $e_i = t_it_{i+1}$, $i = 1, 2, \cdots, p-1$. Observe first that $c(t_1C) = c(e_1)$, for otherwise if for some vertex x_i of C, $c(t_1x_i) \neq c(e_1)$, then obviously we may join T to C through the edge t_1x_i and going around the cycle C clockwise or anticlockwise depending on whether $c(t_1x_i) \neq c(x_ix_{i-1})$ or $c(t_1x_i) \neq c(x_ix_{i-1})$, respectively (all indices are considered modulo k). It follows that $c(t_1C) = c(e_1)$, thus by Lemma 5.2, applied on e_1, for any vertex x_i of C, $c(x_it_2) \neq c(e_1)$, for otherwise we are finished. Note here, that if $d(t_2) \geq 3$, then, a similar argument can be applied starting from all leaves reachable from t_2, not using the edge t_2t_3. In other words, all edges between C and t_2 are on a same color different from the color of the edge e_1, and the colors of all other edges incident to t_1, except e_2. Thus, we will prove, in fact, that $c(t_2, C) = c(e_2)$. Assume by contradiction that $c(t_2, C) \neq c(e_2)$. Let x_ix_{i+1} be an edge of C such that $c(x_ix_{i+1}) \neq c(t_2x_i)$. Clearly such an edge exists since C is properly edge-colored. But then, the tree T together with the segment $[t_2x_ix_{i+1} \cdots x_{i-1}]$ define a proper spanning tree in G^c. Consequently assume that $c(t_2C) = c(e_2)$. Now by replacing edge e_i by e_{i+1} and applying all above arguments we may conclude that all edges between t_k and C are on a same color and $c(t_kC) \neq c(t_{k-1}t_k)$. At this final step, it suffices to join apropriately T and C by using any arbitrary edge between t_k and C. □

Theorem 5.4. *K_n^c has a proper spanning forest with at most p trees, if and only if its vertices are covered by $p \geq 1$ proper trees $T_1, \ldots, T_p$ and a set of $k \geq 1$ properly edge-colored cycles, say $C_1, C_2 \cdots, C_k$ all these components being pairwise vertex-disjoint in K_n^c.*

Proof. By Induction on k. Case $k = 1$, is solved by Lemma 5.3, since the cycle can be merged with any tree T_i to get a new tree T_i' and the number of cycles in the decomposition now becomes $k - 1$. By induction, $C_1, C_2 \cdots, C_{k-1}$ and T may be merged successively with any tree in the collection, the number of trees always being p, so that eventually we have a proper spanning forest with exactly p trees. □

Unfortunately, as a consequence of the NP-completeness result for PST in complete graphs, it follows that we are unlikely to be able to find an algorithm to compute the tree-cycle factor inpolynomial time. There is still, however, scope to develop approximation algorithms. However we are able to obtain the following.

Corollary 5.5. *There is a polynomial algorithm for finding in K_n^c a proper tree of order $min(n, M+2)$, where M denotes the order of a maximum proper cycle-factor F of K_n^c.*

Proof. It is well known that finding a maximum proper cycle-factor in an edge-colored graph is polynomial (see [4]). If F is perfect or almost perfect (it spans $n-1$ vertices), then by previous theorem we may find a proper spanning tree in K_n^c. On the other hand, if F spans less than $n-2$ vertices, then consider an edge in $K_n^c - F$ and then apply again previous theorem. □

Theorem 5.6. *K_n^c has a* PST *if and only if its vertices are covered by a proper tree T and a set of properly edge-colored cycles, say $C_1, C_2 \cdots, C_k$ all these components being pairwise vertex-disjoint in K_n^c.*

Proof. This is a straightforward consequece of the previous theorem. □

6 Random graphs

Let $c \geq 3$ be a fixed integer and let $G(n,p)$ be the random graph on n vertices where each edge is present with independent probability p. We define $G^c(n,p)$ as the edge union of c independent copies of $G(n, \frac{p}{c})$ where for each $j \in \{1, \ldots, c\}$, the edges of the j^{th} copy are colored by j.

Theorem 6.1. *If $p = \frac{\lambda \log n}{n}$, where λ is a sufficiently large constant, then with probability tending to 1 as n tends to infinity, $G^c(n,p)$ contains a proper spanning tree.*

Proof. Set $G = G^c(n,p)$. Let c_0 be a fixed color. We describe an algorithm to prospectively construct a proper spanning tree in our graph and prove that it works with probability close to 1. The algorithm proceeds in two stages.

(1) Construct a sequence of proper trees $T_0, T_1, T_2, ...T_t$. Here, T_0 is a star with an arbitrary root and with ν edges with colors pairwise distinct and different from c_0. We define t and ν later. For each $i \in \{1, \ldots, t\}$, T_i is obtained from T_{i-1} by expanding a pendant vertex with edges of all the colours distinct from c_0, and also distinct from the color of the edge incident to this vertex in T_{i-1}.

(2) From T_t obtain a proper spanning tree by adding a pairing of the internal vertices of T_t other than the root with the remaining vertices of G using edges of color c_0.

This concludes the description of the algorithm.

Note that T_t has a total of $t(c-2)+\nu+1$ vertices among which exactly t are internal vertices, distinct from the root. We want to match these t vertices with the remaining vertices of G, that is, we want

$$\begin{aligned} t &= n - t(c-2) - \nu - 1 \\ \text{or, } t &= \frac{n-\nu-1}{c-1}, \end{aligned}$$

and this will be an integer for precisely one value of ν with $1 \leq \nu \leq c-1$. We fix this value for ν.

To prove the correctness of the first stage of the algorithm it suffices to prove that at each point in the construction we can find with probability $1 - o(1/n)$ an edge with any particular fixed colour linking a point v, say, already in the current tree and the external set which has size at least t. Now, when we look for such an edge of colour j, say, we have not looked previously at the edges of colour j linking v to the remaining vertices. Hence, the conditional distribution of these edges given the previous steps of the algorithm is the same as their unconditional distribution, namely, each is present with probability $\frac{\lambda \log n}{cn}$ and they are independent. Therefore the probability that at least one is present is at least

$$\begin{aligned} 1 - (1 - \frac{\lambda \log n}{cn})^t &\geq 1 - \exp(-\frac{\lambda \log n}{c(c-1)}) \\ &= 1 - o(1/n) \end{aligned}$$

if $\lambda \geq c^2$.

We turn now to the second stage. Since we have not looked yet at the edges of colour c_0, we just have to check that with high probability, there exists in a random graph with edge probability p, a pairing between two given disjoint sets with sizes t. This amounts to asking for a perfect matching in the random bipartite graph $B(t,p)$. Now $p = \frac{\lambda \log n}{cn} \geq \frac{\theta \log n}{t}$ with $\theta = \frac{2\lambda}{3c(c-1)}$. This graph is known to have, almost surely, a perfect matching for any fixed θ greater than 1/2 and thus for $\lambda \geq c(c-1)/2$ (see Corollary 13, page 159 in [6]). Putting together this estimate with the bound already found, we infer that the theorem holds for $\lambda \geq c^2$. On the other direction, the theorem does *not* hold for $\lambda \leq c$ by connectivity considerations. $\square$

7 Conclusions

In this paper we recapitulate the notion of various types of colored trees in edge-colored graphs. We obtain results reflecting the computational difficulties involved in their solution and provide efficient algorithms for the specific family of acyclic graphs. We give a mathematical characterization of complete graphs which contain a PST. We study the trends of random graphs with reference to the problem of PST.

We list here some possible future directions for research in this area.

(1) Algorithms to solve the PST and WST problems on other special classes of graphs like planar graphs or hypercubes.

(2) We conjecture the existence of approximation algorithms with performance guarantee of at least logarithm of the optimal solution.

(3) Given an edge-colored (complete) graph, determine the minimum number of edges whose colors need to be changed in order to render the graph acyclic. This problem resembles the feedback arc set problem on digraphs.

8 Acknologements

We thank the CNPq/Brazil and the French Ministry of Education for their financial support.

References

[1] A. Abouelaoualim, Exploration des graphs aretes-colorees: topologie, algorithmes, complixite et (non)-approximabilite, Ph.D. Thesis, Universite Paris Sud, 2007.

[2] A. Abouelaoualim, K. Ch. Das, L. Faria, Y. Manoussakis, C. Martinhon, R. Saad, Paths and trails in edge-colored graphs,*Theoretical Computer Science,* **409**, 2008, 497-510.

[3] N. Alon, R. A. Brualdi, B. L. Shader, Multicolored forests in bipartite decompositions of graphs, *Journal of Combinatorial Theory*, Series B, **53**, 1991, 143-148.

[4] J. Bang-Jensen, G. Gutin, Digraphs: Theory, Algorithms and Applications, *Springer*, 2002.

[5] A. Benkouar, Y. Manoussakis, V. T. Paschos, R. Saad, On the complexity of some hamiltonian and eurelian problems in edge-colored complete graphs *RAIRO - Operations Research*, **30**, 1996, 417-438.

[6] Bollobás, Random graphs, *Academic Press*, 1985

[7] R. A. Brualdi, S. Hollingsworth, Multicolored trees in complete graphs, *Journal of Combinatorial Theory*, Series B, **68**, 1996, 310-313.

[8] R. A. Brualdi, S. Hollingsworth, Multicolored trees in complete bipartite graphs, *Discrete Mathematics*, **240**, 2001, 239-245.

[9] S. Even, O. Kariv, An $(O(n^{2.5})$ algorithm for maximum matching in general graphs, *Proceeding of the 16th Annual Symposium on Foundations of Computer Science, Berkeley, CA*, IEEE Computer Society Press, New York, 1975, 100-112.

[10] H.N. Gabow, Z. Galil, S. Micali, An $O(EVlogV)$ Algorithm for Finding a Maximal Weighted Matching in General Graphs, *Network Models*,SIAM Journal on Computing, **15(1)**, 1986, 120-130.

[11] M. R. Garey David S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, *W. H. Freeman*, 1979.

[12] J. Grossman and R. Häggkvist, Properly Edge Colored Cycles in Edge-Partioned Graphs, *Journal of Combinatorial Theory, Series B*, **34**, 1983, 77-81.

[13] J. Hästad, Some optimal inapproximability results, *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, El Paso, Texas, 1997, 1-10.

[14] M. Kano, X. Li, Monochromatic and Heterochromatic Subgraphs in Edge-Colored Graphs - A Survey, Graphs and Combinatorics, **24(4)**, Springer-Verlag, 2008, 237–263.

[15] Y. Manoussakis, Alternating paths in edge-colored complete graphs, Discrete Applied Mathematics, **56**, 1995, 297-309.

[16] R. Saad, Finding a longest properly edge-colored hamiltonian cycle in an edge colored complete graph is not hard, *Combinatorics, Probability and Computing*, **5**, 1996, 297-306.

[17] K. Suziki, A Necessary and Sufficient Condition for the Existence of a Heterochromatic Spanning Tree in a Graph, *Graphs and Combinatorics*, Springer-Verlag, **22**, 2006, 261-269.

[18] A. Yeo, A Note on Alternating Cycles in Edge-colored Graphs, *Journal of Combinatorial Theory, Series B,* **69**, 1997, 222-225.