Approximating Maximum-Size Properly Colored Forests

Yuhang Bai

School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an, Shaanxi, China
Xi'an-Budapest Joint Research Center for Combinatorics, China

Kristóf Bérczi

MTA-ELTE Matroid Optimization Research Group and HUN-REN–ELTE Egerváry Research Group, Department of Operations Research, ELTE Eötvös Loránd University, Budapest, Hungary

Gergely Csáji

Department of Operations Research, ELTE Eötvös Loránd University, Budapest, Hungary
HUN-REN Centre for Economic and Regional Studies, Budapest, Hungary

Tamás Schwarcz

MTA-ELTE Matroid Optimization Research Group and HUN-REN–ELTE Egerváry Research Group, Department of Operations Research, ELTE Eötvös Loránd University, Budapest, Hungary

Abstract

In the *Properly Colored Spanning Tree* problem, we are given an edge-colored undirected graph and the goal is to find a properly colored spanning tree. The problem is interesting not only from a graph coloring point of view, but is also closely related to the *Degree Bounded Spanning Tree* and $(1,2)$-*Traveling Salesman* problems. We propose an optimization version called *Maximum-size Properly Colored Forest* problem, which aims to find a properly colored forest with as many edges as possible. We consider the problem in different graph classes and for different numbers of colors, and present polynomial-time approximation algorithms as well as inapproximability results for these settings. We also consider the *Maximum-size Properly Colored Tree* problem asking for the maximum size of a properly colored tree not necessarily spanning all the vertices. We show that the optimum is significantly more difficult to approximate than in the forest case, and provide an approximation algorithm for complete multigraphs.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Approximation algorithms analysis

Keywords and phrases Approximation algorithm, $(1,2)$-traveling salesman problem, Degree bounded spanning tree, Properly colored forest

Digital Object Identifier 10.4230/LIPIcs.ESA.2024.14

Related Version *Full Version*: `https://arxiv.org/abs/2402.00834` [2]

Funding The research was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021, by the Ministry of Innovation and Technology of Hungary – grant number ELTE TKP 2021-NKTA-62, and by Dynasnet European Research Council Synergy project – grant number ERC-2018-SYG 810115.
Yuhang Bai: Supported by the National Natural Science Foundation of China – NSFC, grant numbers 12071370 and 12131013 and by Shaanxi Fundamental Science Research Project for Mathematics and Physics – grant number 22JSZ009.
Gergely Csáji: Supported by the Hungarian Scientific Research Fund – grant number K143858, and by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation fund – grant number KDP-2023/C2258525.
Tamás Schwarcz: Supported by the ÚNKP-23-3 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

1 Introduction

Throughout the paper, we consider loopless graphs that might contain parallel edges. A *k-edge-colored graph* is a graph $G = (V, E)$ with a coloring $c\colon E \to [k]$ of its edges by k colors. We refer to a graph that is k-edge-colored for some $k \in \mathbb{Z}_+$ as *edge-colored*. A subgraph H of G is called *rainbow colored* if no two edges of H have the same color, and *properly colored* if any two adjacent edges of H have distinct colors. Since rainbow colored forests form the common independent sets of two matroids, i.e., the partition matroid defined by the color classes and the graphic matroid of the graph, a rainbow colored forest of maximum size can be found in polynomial time using Edmonds' celebrated matroid intersection algorithm [11]. However, much less is known about the properly colored case. In [5], Borozan, de La Vega, Manoussakis, Martinhon, Muthu, Pham, and Saad initiated the study of properly edge-colored spanning trees of edge-colored graphs and investigated the existence of such a spanning tree, called the *Properly Colored Spanning Tree* problem (PST). This problem generalizes the well-known bounded degree spanning tree problem for uncolored graphs as the number of colors bounds the degree of each vertex, as well as the properly colored Hamiltonian path problem when the number of colors is restricted to two. Since both of these problems are NP-complete, finding a properly colored spanning tree is hard in general.

The aim of this paper is to study the problem from an approximation point of view. Accordingly, we define the *Maximum-size Properly Colored Forest* problem (Max-PF) in which the goal is to find a properly colored forest of maximum size in an edge-colored graph, and discuss the approximability of the problem in various settings. Throughout the paper, by the *size* of a tree or a forest we mean the number of its edges. From an *application point of view*, the problem arises naturally in practice in the context of conflict-free scheduling. Consider a communication network where nodes represent switches or routers and edges represent communication lines between those. An edge-coloring of the graph might represent different channels or time slots for data transmission. A properly colored spanning tree then provides a set of communication paths without redundancy where no conflicts appear at the vertices. From a *theoretical point of view*, the proposed problem and the results may be interesting not only for the graph coloring but also for the optimization community.

1.1 Related work and connections

Finding properly colored spanning trees in graphs is closely related to constrained spanning tree problems, or in a more general context, to the problem of finding a basis of a matroid subject to further matroid constraints. In what follows, we give an overview of questions that motivated our investigations.

Properly colored trees. Properly colored spanning trees were first considered in Borozan et al. [5] where their existence was studied from both a graph-theoretic and an algorithmic perspective. They showed that finding a properly colored spanning tree remains NP-complete when restricted to complete graphs. Deciding the existence of a properly colored spanning tree is hard in general, hence a considerable amount of work has focused on finding sufficient conditions [8, 19, 20]. Since a properly colored spanning tree may not exist, it is natural to ask for the maximum size of a properly colored tree not necessarily spanning all the vertices, called the *Maximum-size Properly Colored Tree* problem (Max-PT). Borozan et al. [5] proved that Max-PT is hard to approximate within a factor of $55/56 + \varepsilon$ for any $\varepsilon > 0$, while they provided polynomial algorithms for graphs not containing properly edge-colored cycles. Hu, Liu and Maezawa [18] proved that the maximum size of a properly colored tree in an edge-colored connected graph is at least $\min\{|V| - 1, 2\delta^c(G) - 1\}$.

Degree bounded spanning trees. In the *Minimum Bounded Degree Spanning Tree* problem (Min-BDST), we are given an undirected graph $G = (V, E)$ with $|V| = n$, a cost function $c\colon E \to \mathbb{R}$ on the edges, and degree upper bounds $g\colon V \to \mathbb{Z}_+$ on the vertices, and the task is to find a spanning tree of minimum cost that satisfies all the degree bounds. There is an extensive list of results on variants of the problem [6, 7, 9, 14, 16, 17, 25, 29, 30]. When the degree bounds are the same for every vertex and the edge-costs are identically 1, we get the *Uniformly Bounded Degree Spanning Tree* problem.

Degree bounded matroids and multi-matroid intersection. Király, Lau and Singh [24] studied a matroidal extension of the Min-BDST problem. In their setting, a matroid with a cost function on its elements, and a hypergraph on the same ground set with lower and upper bounds $f(e) \le g(e)$ for each hyperedge e. The task is to find a minimum cost basis of the matroid which contains at least $f(e)$ and at most $g(e)$ elements from each hyperedge e. If we choose the matroid to be the graphic matroid of a graph $G = (V, E)$ and the hyperedges to be the sets $\delta(v)$ for $v \in V$, we get back the Min-BDST Tree problem with the value of Δ being 2. In [31], Zenklusen considered a different generalization of the Min-BDST problem where for every vertex v, the edges adjacent to v have to be independent in a matroid $\boldsymbol{M}_v$. This model was further extended by Linhares, Olver, Swamy and Zenklusen [26] who studied the problem of finding a minimum cost basis of a matroid $\boldsymbol{M}_0$ that is independent in other matroids $\boldsymbol{M}_1, \dots, \boldsymbol{M}_q$.

$(1,2)$-traveling salesman problem. Karp [21] showed that the metric Traveling Salesman Problem is NP-hard even in the special case when all distances between cities are either 1 or 2, called the *Traveling Salesman Problem with Distances 1 and 2* ($(1,2)$-TSP). This result was further strengthened by Papadimitriou and Yannakakis [28] who showed that $(1,2)$-TSP is in fact hard to approximate and MAX-SNP-hard. The currently best known inapproximability bound of 535/534 is due to Karpinski and Schmied [22]. The performance of local search-based approximations was studies by many [4, 23, 32]; Adamaszek, Mnich and Paluch [1] presented an 8/7-approximation algorithm with running time $O(n^3)$.

The problem Max-PF is closely related to the problems listed above.

- Max-PF provides a relaxation of both the PST and Max-PT problems.
- For an arbitrary graph G, let G' be the k-edge-colored multigraph obtained by taking k copies of each edge of G colored by different colors. Then, G has a uniformly bounded degree spanning tree with upper bound k if and only if G' admits a properly colored spanning tree.
- For a k-edge-colored graph $G = (V, E)$, let $\boldsymbol{M}$ be the graphic matroid of G. Furthermore, define a hypergraph on E as follows: for each vertex $v \in V$ and color $i \in [k]$, let $e_{v,i} \coloneqq \{e \in E \mid c(e) = i, e \text{ is incident to } v\}$ be a hyperedge with upper bound 1. Then, G has a properly colored spanning tree if and only if $\boldsymbol{M}$ admits a degree bounded basis.
- For a k-edge-colored graph $G = (V, E)$, let $\boldsymbol{M}_0$ be the graphic matroid of G. Furthermore, for each vertex $v \in V$ and color $i \in [k]$, let $\boldsymbol{M}_{v,i}$ be a rank-1 partition matroid whose ground set is the set of edges incident to v having color i. Then, G has a properly colored spanning tree if and only if the multi-matroid intersection problem $\boldsymbol{M}_0, \{\boldsymbol{M}_{v,i}\}_{v \in V, i \in [k]}$ admits a solution of size $|V| - 1$.
- Consider an instance of $(1,2)$-TSP on n vertices and let G denote the subgraph of edges of length 1. Since any linear forest of G of size x can be extended to a Hamiltonian cycle of length $2n - x$, one can reformulate $(1,2)$-TSP as the problem of finding a maximum linear forest in G. This problem reduces to Max-PF in 2-edge-colored graphs, see [2, Section 3.1] for further details.

Given the close connection to earlier problems, the reader may naturally wonder whether existing methods are applicable to the proposed problem. Consider an instance of Max-PF, that is, an edge-colored graph G and let Opt denote the maximum size of a properly colored forest in G. One can obtain a forest F of G of size at least Opt in which every color appears at most twice at every vertex, either by the approximation algorithm of [24] for the bounded degree matroid problem, or by the approximation algorithm of [26] for the multi-matroid intersection problem. Deleting conflicting edges from F greedily results in a properly colored forest of size at least $|F|/2 \geq \text{Opt}/2$, thus leading to a 1/2-approximation for Max-PF. The reason for providing a detailed overview of previous results and techniques was to emphasize that those approaches do not help to get beyond the approximation factor of 1/2. Our main motivation was to improve the approximation factor and to understand the inapproximability of the problem.

1.2 Our results

We use the convention that, by an *α-approximation algorithm*, for minimization problems we mean an algorithm that provides a solution with objective value at most α times the optimum for some $\alpha \geq 1$, while for maximization problems we mean an algorithm that provides a solution with objective value at least α times the optimum for some $\alpha \leq 1$.

We initiate the study of properly colored spanning trees from an optimization point of view and focus on the problem of finding a properly colored forest of *maximum size*, i.e., containing a maximum number of edges. We discuss the problem for several graph classes and numbers of colors, and provide approximation algorithms as well as inapproximability bounds for these problems. The results are summarized in Table 1.

Table 1 Complexity landscape of Max-PF.

Graphs	Number of colors: $k = 2$	$k = 3$	$k \geq 4$
Simple graphs	MAX-SNP-hard [2, Thm. 3.3]		
	3/4-approx. (Thm. 3.13)	5/8-approx. (Thm. 3.14)	4/7-approx. (Thm. 3.7)
Multigraphs	MAX-SNP-hard [2, Thm. 3.3]		
	3/5-approx. (Thm. 3.10)	4/7-approx. (Thm. 3.7)	5/9-approx. (Thm. 3.3)
Complete graphs	P (Thm. 3.2)	MAX-SNP-hard [2, Thm. 3.4]	
		5/8-approx. (Thm. 3.14)	4/7-approx. (Thm. 3.7)
Complete multigraphs		MAX-SNP-hard [2, Thm. 3.4]	
		4/7-approx. (Thm. 3.7)	5/9-approx. (Thm. 3.3)

We also consider Max-PT, that is, when a properly colored tree (not necessarily spanning) of maximum size is to be found. We give a strong inapproximability result in general, together with an approximation algorithm for complete multigraphs. The results are summarized in Table 2.

Studying the problem on complete graphs is interesting since the vast majority of previous work on finding properly colored (spanning) trees has focused on complete graphs. Designing approximation algorithms for complete graphs was motivated also by the result of Borozan et al. [5] (Theorem 2.2). For Max-PT, they provided a 55/56-inapproximability bound that we improve in our paper. Also, it is worth emphasizing that we prove a better approximation guarantee for complete graphs than the inapproximability bound for general graphs.

Table 2 Complexity landscape of MAX-PT.

Graphs	Number of colors $k = 2$	Number of colors $k \geq 3$
Simple graphs	$1/n^{1-\varepsilon}$-inapprox. for $\varepsilon > 0$ [2, Thm. 3.6]	
Multigraphs	$1/n^{1-\varepsilon}$-inapprox. for $\varepsilon > 0$ [2, Thm. 3.6]	
Complete graphs	P [3]	MAX-SNP-hard [2, Thm. 3.7] $1/\sqrt{(2+\varepsilon)n}$-approx. for any $\varepsilon > 0$ (Thm. 3.18)
Complete multigraphs	P [3]	MAX-SNP-hard [2, Thm. 3.7] $1/\sqrt{(2+\varepsilon)n}$-approx. for any $\varepsilon > 0$ (Thm. 3.18)

1.3 Our techniques

Most of the previous work on the Minimum Bounded Degree Spanning Tree, Degree Bounded Matroids and Multi-matroid Intersection problems was based on polyhedral approaches, combined with variants of iterative rounding. Polyhedral methods are indeed standard in approximation algorithms for related problems. Nevertheless, these techniques do not seem to be sufficient for beating the approximation factor of 1/2 for MAX-PF, see also the beginning of Section 3.3. In contrast, in the current paper, we take a different approach that relies on the following technical ingredient. Consider the matching matroids formed by edges of each color, and take the union – also called sum – of these matroids. If U is a maximum sized independent set of vertices in the matroid thus obtained, then we show that any properly colored forest spanning U provides a 1/2-approximation for MAX-PF. Since the maximum size of a properly colored forest is clearly bounded by the number of vertices, the factor 1/2 is tight only if each component of the returned forest has two vertices. However, if each component has, say, size three, then we would get a constant factor improvement and get a 2/3-approximation. Our algorithms focus on these small components and make local improvements to reduce the components of size two or to get an improved bound.

Paper organization

The paper is organized as follows. In Section 2, we introduce basic definitions and notation, and overview some results of matroid theory that we will use in our proofs. The rest of the paper is devoted to presenting approximation algorithms mainly for MAX-PF in various settings. In Section 3.1, we show that the vertex set of the graph can be assumed to be coverable by monochromatic matchings of the graph, and that such a reduction can be found efficiently using techniques from matroid theory. We then give a polynomial algorithm for 2-edge-colored complete multigraphs in Section 3.2. Our main result is a 5/9-approximation algorithm for the problem in k-edge-colored multigraphs, presented in Section 3.3. In Section 3.4, we explain how the approximation factor can be improved if the graph is simple or the number of colors is at most three. We further improve the approximation factor for 2- and 3-edge-colored simple graphs in Section 3.5. Finally, an approximation algorithm is given for MAX-PT in Section 3.6. Due to space constraints, the statements and proofs of our hardness results are deferred to the full version [2] of the paper.

2 Preliminaries

Basic notation. We denote the set of *nonnegative integers* by $\mathbb{Z}_+$. For a positive integer k, we use $[k] := \{1, \ldots, k\}$. Given a ground set S, the *difference* of $X, Y \subseteq S$ is denoted by $X \setminus Y$. If Y consists of a single element y, then $X \setminus \{y\}$ and $X \cup \{y\}$ are abbreviated as $X - y$ and $X + y$, respectively.

We consider loopless undirected graphs possibly containing parallel edges. A graph is *simple* if it has no parallel edges, and it is called a *multigraph* if parallel edges might be present. A simple graph is *complete* if it contains exactly one edge between any pair of vertices. By a *complete multigraph*, we mean a multigraph containing at least one edge between any pair of vertices. A graph is *linear* if each of its vertices has degree at most 2 in it. Let $G = (V, E)$ be a graph, $F \subseteq E$ be a subset of edges, and $X \subseteq V$ be a subset of vertices. The *subgraph of* G and *set of edges induced by* X are denoted by $G[X]$ and $E[X]$, respectively. The *graph obtained by deleting* F *and* X is denoted by $G - F - X$. We denote the *vertices of the edges in* F by $V(F)$, and the *vertex sets of the connected components of the subgraph* $(V(F), F)$ by $\mathrm{comp}(F) \subseteq 2^{V(F)}$. We denote the *set of edges in* F *having exactly one endpoint in* X by $\delta_F(X)$ and define the *degree of* X *in* F as $d_F(X) \coloneqq |\delta_F(X)|$. We dismiss the subscript if $F = E$. A *matching* is a subset of edges $M \subseteq E$ satisfying $d_M(v) \leq 1$ for every $v \in V$. We say that F *covers* X if $d_F(v) \geq 1$ for every $v \in X$, or in other words, if $X \subseteq V(F)$.

Let $c\colon E \to [k]$ be an edge-coloring of G using k colors. The function c is extended to subsets of edges where, for a subset $F \subseteq E$ of edges, $c(F)$ denotes the set of colors appearing on the edges of F. For an edge-colored graph $G = (V, E)$, we use $E_i = \{e \in E \mid c(e) = i\}$ to denote the edges of color i. Without loss of generality, we assume throughout that E_i contains no parallel edges. We call a subset of vertices $U \subseteq V$ *matching-coverable* if there exist matchings $M_i \subseteq E_i$ for $i \in [k]$ such that $\bigcup_{i=1}^{k} M_i$ covers U. A *properly colored 1-path-cycle factor* of a graph G is a spanning subgraph consisting of a properly colored path C_0 and a (possibly empty) collection of properly colored cycles $C_1, \ldots, C_q$ such that $V(C_i) \cap V(C_j) = \emptyset$ for $0 \leq i < j \leq q$. We will use the following result of Bang-Jensen and Gutin [3], extended by Feng, Giesen, Guo, Gutin, Jensen, and Rafiey [13].

▶ **Theorem 2.1** (Bang-Jensen and Gutin [3]). *A 2-edge-colored complete graph G has a properly colored Hamiltonian path if and only if G contains a properly colored 1-path-cycle factor. Furthermore, any properly colored 1-path-cycle factor can be transformed into a properly colored Hamiltonian path in polynomial time.*

For our approximation algorithm for Max-PT in complete graphs, we will rely on the following result of Borozan et al. [5].

▶ **Theorem 2.2** (Borozan et al. [5]). *Let $G = (V, E)$ be an edge-colored complete multigraph. Then, there exists an efficiently computable partition $V_1 \cup V_2$ of V such that Max-PT can be solved in polynomial-time in both $G[V_1]$ and $G[V_2]$. Furthermore, the optimal solution F_1 in $G[V_1]$ is a properly colored spanning tree of $G[V_1]$.*

Matroids. For basic definitions on matroids and on matroid optimization, we refer the reader to [15,27]. A *matroid* $\boldsymbol{M} = (E, \mathcal{I})$ is defined by its *ground set* E and its *family of independent sets* $\mathcal{I} \subseteq 2^E$ that satisfies the *independence axioms*: (I1) $\emptyset \in \mathcal{I}$, (I2) $X \subseteq Y,\ Y \in \mathcal{I} \Rightarrow X \in \mathcal{I}$, and (I3) $X, Y \in \mathcal{I},\ |X| < |Y| \Rightarrow \exists e \in Y \setminus X\ s.t.\ X + e \in \mathcal{I}$. Members of $\mathcal{I}$ are called *independent*, while sets not in $\mathcal{I}$ are called *dependent*. The *rank* $r_{\boldsymbol{M}}(X)$ of a set X is the maximum size of an independent set in X.

The *union* or *sum* of k matroids $\boldsymbol{M}_1 = (E, \mathcal{I}_1), \ldots, \boldsymbol{M}_k = (E, \mathcal{I}_k)$ over the same ground set is the matroid $\boldsymbol{M}_\Sigma = (E, \mathcal{I}_\Sigma)$ where $\mathcal{I}_\Sigma = \{I_1 \cup \cdots \cup I_k \mid I_i \in \mathcal{I}_i \text{ for each } i \in [k]\}$. Edmonds and Fulkerson [12] showed that the rank function of the sum is $r_{\boldsymbol{M}_\Sigma}(Z) = \min\{\sum_{i=1}^{k} r_i(X) + |Z - X| \mid X \subseteq Z\}$, and provided an algorithm for finding a maximum sized independent set of $\boldsymbol{M}_\Sigma$, together with its partitioning into independent sets of the matroids appearing in the sum, assuming an oracle access[1] to the matroids $\boldsymbol{M}_i$.

[1] In matroid algorithms, it is usually assumed that the matroid is given by a *rank oracle* and the running time is measured by the number of oracle calls and other conventional elementary steps. For a matroid $\boldsymbol{M} = (E, \mathcal{I})$ and set $X \subseteq E$ as an input, a rank oracle returns $r_{\boldsymbol{M}}(X)$.

For an undirected graph $G = (V, E)$, the *matching matroid* of G is defined on the set of vertices V with a set $X \subseteq V$ being independent if there exists a matching M of G such that $X \subseteq V(M)$, that is, M covers all the vertices in X. Determining the rank function of the matching matroid is non-obvious since it requires the knowledge of the Berge-Tutte formula on the maximum cardinality of a matching in a graph. Nevertheless, the rank of a set can still be computed in polynomial time, see [12] for further details.

3 Approximation algorithms

In this section, we provide approximation algorithms for Max-PF and Max-PT in various settings. First, in Section 3.1, we establish a connection between Max-PF and the sum of matching matroids defined by the color classes of the coloring of the graph. In Section 3.2, we discuss 2-edge-colored complete multigraphs and show that Max-PF is solvable in polynomial time for this class. Our main result is a general 5/9-approximation algorithm for Max-PF in multigraphs, presented in Section 3.3. In Section 3.4, we explain how the approximation factor can be improved if the graph is simple or the number of colors is at most three, and then we further improve the approximation factor for 2- and 3-edge-colored simple graphs in Section 3.5. Finally, an approximation algorithm for Max-PT is given in Section 3.6. We denote by $\textsc{Opt}[G]$ the size of an optimal solution for the underlying problem, i.e., Max-PF or Max-PT, in graph G throughout.

3.1 Preparations

For analyzing the proposed algorithms, we need some preliminary observations. Consider an instance of Max-PF, that is, a k-edge-colored graph $G = (V, E)$ on n vertices. Recall that E_i denotes the set of edges colored by i and that a subset of vertices $U \subseteq V$ is called matching-coverable if there exist matchings $M_i \subseteq E_i$ for $i \in [k]$ such that $\bigcup_{i=1}^{k} M_i$ covers U. Using the matroid terminology, this is equivalent to U being independent in the sum of the matching matroids defined by the color classes. The next lemma shows that it suffices to restrict the problem to a maximum sized matching-coverable set.

▶ **Lemma 3.1.** *For any matching-coverable set $U \subseteq V$, there exists a maximum-size properly colored forest F_{opt} in G such that $d_{F_{opt}}(u) \geq 1$ for every $u \in U$. Furthermore, if U is a maximum-size matching-coverable set, then $\textsc{Opt}[G] = \textsc{Opt}[G[U]]$.*

Proof. Let $U \subseteq V$ be a matching-coverable set and let $M_1, \ldots, M_k$ be matchings satisfying $M_i \subseteq E_i$ and $U \subseteq V(\bigcup_{i=1}^{k} M_i)$. Let F_{opt} be a maximum-size properly colored forest in G that has as many edges in common with $M_1 \cup \cdots \cup M_k$ as possible. We claim that F_{opt} covers U. Suppose indirectly that there exists a vertex $u \in U$ that is not covered by F_{opt}. For any edge $e \in M_1 \cup \cdots \cup M_k$ incident to u, $F_{opt} + e$ is still a forest by the indirect assumption. Moreover, F_{opt} contains at most one edge adjacent to e having the same color as e. Since F_{opt} has maximum size, there exists exactly one such edge f. However, as f has the same color as $e \in M_1 \cup \cdots \cup M_k$, we get that $f \notin M_1 \cup \cdots \cup M_k$. Therefore, $F_{opt} - f + e$ is a maximum-size properly colored forest containing more elements from $M_1 \cup \cdots \cup M_k$ than F_{opt}, a contradiction. This proves the first half of the lemma.

To see the second half, let U be a maximum-size matching-coverable set and F_{opt} be a maximum-size properly colored forest covering U, implying $U \subseteq V(F_{opt})$. Note that $N_i = E_i \cap F_{opt}$ is a matching for every $i \in [k]$, hence $V(F_{opt})$ is also a matching-coverable set. By the maximality of U, we get $U = V(F_{opt})$, concluding the proof. ◀

Algorithm 1 Algorithm for MAX-PF in 2-edge-colored complete multigraphs.

Input: A 2-edge-colored complete multigraph $G = (V, E)$.
Output: A properly colored forest F.

1 Find maximum matchings $M_1 \subseteq E_1, M_2 \subseteq E_2$ maximizing $|V(M_1 \cup M_2)|$.
2 Let $F := M_1 \cup M_2$ and $U := V(F)$.
3 Let $\mathcal{P}$ and $\mathcal{C}$ denote the path and cycle components in $\mathrm{comp}(F)$, respectively.
4 **if** $\mathcal{P} = \emptyset$ **then**
5 Delete any edge of F, transform the remaining set of edges into a properly colored Hamiltonian path P using Theorem 2.1, and update $F \leftarrow P$.
6 **else**
7 Let $P \in \mathcal{P}$ arbitrary and let $F' := F[P \cup \bigcup_{C \in \mathcal{C}} C]$. Transform F' into a properly colored Hamiltonian path P' using Theorem 2.1 and update $F \leftarrow (F \setminus F') \cup P'$.
8 **return** F

▶ Remark 3.1. Since a rank oracle for the matching matroid of a graph can be constructed in polynomial time, a maximum-size matching-coverable set U can be found by using the matroid sum algorithm of Edmonds and Fulkerson [12]. The algorithm also provides a partition $U = U_1 \cup \cdots \cup U_k$ where U_i is independent in the matching matroid defined by E_i. For each U_i, one can find a matching $M_i \subseteq E_i$ that covers U_i using Edmonds' matching algorithm [10]. Furthermore, each matching M_i can be chosen to be a maximum matching in E_i, due the the underlying matroid structure. Concluding the above, a maximum-size matching-coverable set U together with maximum matchings $M_1, \ldots, M_k$ with $M_i \subseteq E_i$ and $V(\bigcup_{i=1}^{k} M_i) = U$ can be found in polynomial time.

3.2 2-edge-colored complete multigraphs

Though MAX-PF is hard even to approximate in general, the problem turns out to be tractable for 2-edge-colored complete multigraphs. Our algorithm is presented as Algorithm 1.

▶ **Theorem 3.2.** *Algorithm 1 outputs a maximum-size properly colored forest for 2-edge-colored complete multigraphs in polynomial time.*

Proof. Note that each component of $M_1 \cup M_2$ is either a path or a cycle whose edges are alternating between M_1 and M_2. If $M_1 \cup M_2$ is the union of cycles, then Algorithm 1 gives a properly colored Hamiltonian path in $G[U]$ by Step 5. By Lemma 3.1, $G[U]$ contains a maximum-size properly colored forest and hence $\mathrm{OPT} \leq |U| - 1$, implying that F is optimal.

If $M_1 \cup M_2$ has a path component, then Step 7 of Algorithm 1 does not reduce the number of edges, i.e., the output F of the algorithm has size $|M_1| + |M_2|$. Since M_1 and M_2 were chosen to be maximum matchings in E_1 and E_2, respectively, the sum of their sizes is clearly an upper bound on the maximum size of a properly colored forest, implying that F is optimal. The overall running time of the algorithm is polynomial by Theorem 2.1 and Remark 3.1. ◀

3.3 General case

This section is dedicated for the proof of our main result, a general approximation algorithm for MAX-PF. The high-level idea of our approach is as follows. With the help of Lemma 3.1, we restrict the problem to a subgraph $G[U]$ where U is a maximum-size matching-coverable set. Throughout the algorithm, we maintain matchings $M_i \subseteq E_i$ for $i \in [k]$ such that

Algorithm 2 Approximation algorithm for MAX-PF in multigraphs.

Input: A multigraph $G = (V, E)$ with edge-coloring $c \colon E \to [k]$.
Output: A properly colored forest F in G.

1 Find matchings $M_i \subseteq E_i$ for $i \in [k]$ maximizing $|\bigcup_{i=1}^k V(M_i)|$. // Preprocessing steps.
2 Let $F \coloneqq \bigcup_{i=1}^k M_i$ and $U \coloneqq V(F)$.
3 $U_s \coloneqq \bigcup\{C \in \text{comp}(F) \mid |C| = 2\}$. // Union of size-two components.
4 $U_r \coloneqq U \setminus U_s$. // Remaining vertices.
5 Take a maximum forest F° in $F[U_r]$ and set $F \leftarrow (F \setminus F[U_r]) \cup F^\circ$. // Maximum forest in U_r.
6 **for** $uv \in E \setminus F$ ***with*** $c(uv) \notin c(\delta_F(u) \cup \delta_F(v))$ **do** // Trying to add single edges.
7   If u and v are in different components of F, then $F \leftarrow F + uv$ and go to Step 3.
8 Let $E' \coloneqq E[U_s] \cup \{vw \in E \mid v \in U_s, w \in U_r, c(vw) \notin c(\delta_F(w))\}$. // Candidate edges for extending F°.
9 Let $E'_i \coloneqq E' \cap E_i$.
10 **for** $uv \in E'$ ***with*** $u \in U_s, v \in U_r$ **do** // Trying to improve using single edges.
11   If there exist matchings $N_i \subseteq E'_i$ for $i \in [k]$ such that $uv \in N_{c(uv)}$ and $U_s + v \subseteq V(\bigcup_{i=1}^k N_i)$, then $F \leftarrow (F \setminus F[U_s]) \cup (\bigcup_{i=1}^k N_i)$ and go to Step 3.
12 **for** $uv_1, uv_2 \in E[U_s]$ ***with*** $v_1 \neq v_2$, $c(uv_1) \neq c(uv_2)$ **do** // Trying to improve using pairs of edges.
13   If there exist matchings $N_i \subseteq E'_i[U_s]$ for $i \in [k]$ such that $uv_1 \in N_{c(uv_1)}$, $uv_2 \in N_{c(uv_2)}$ and $U_s \subseteq V(\bigcup_{i=1}^k N_i)$, then $F \leftarrow (F \setminus F[U_s]) \cup (\bigcup_{i=1}^k N_i)$ and go to Step 3.
14 Take a maximum forest $F^\bullet$ in $F[U_s]$ and set $F \leftarrow (F \setminus F[U_s]) \cup F^\bullet$. // Getting rid of parallel edges.
15 **return** F

$F = \bigcup_{i=1}^k M_i$ covers U. We then try to improve the structure of F by decreasing the number of its components of size 2 by local changes. These local improvement steps consist of adding one or two appropriately chosen edges. If no improvement is found, then a careful analysis of the structure of the current solution gives a better-than-1/2 guarantee for the approximation factor.

Before stating the algorithm and the theorem, let us remark that there are several ways of getting a 1/2-approximation for MAX-PF in general. As it was mentioned already in Section 1, the algorithms of [24] and [26] provide such a solution. However, there is a simple direct approach as well: find matchings $M_i \subseteq E_i$ for $i \in [k]$ maximizing the size of $U \coloneqq V(\bigcup_{i=1}^k M_i)$, and take a maximum forest F in $\bigcup_{i=1}^k M_i$. This provides a 1/2-approximation by Lemma 3.1, since $|F| \geq |U|/2 \geq \textsc{Opt}[G[U]]/2 \geq \textsc{Opt}[G]$ holds. However, improving the 1/2 approximation factor is non-trivial and requires new ideas. Our main contribution is to break the 1/2 barrier and show that the problem can be approximated within a factor strictly better than 1/2. The algorithm is presented as Algorithm 2.

▶ **Theorem 3.3.** *Algorithm 2 provides a 5/9-approximation for MAX-PF in multigraphs in polynomial time.*

Proof. First, we show that if the algorithm terminates, then it returns a properly colored forest F in G. Throughout the algorithm, the edge set F is the union of matchings of color i for $i \in [k]$, hence it is properly colored. By Steps 5 and 14, the algorithm outputs the union of a forest F° covering U_r and a forest $F^\bullet$ covering U_s, which is a forest. These prove the feasibility.

Now we turn to the approximation factor. Let F, U_s, U_r and E' denote the corresponding sets at the termination of the algorithm, and let $G' := (U, E')$ and $G'' = (U, E[U] \setminus E')$. By Lemma 3.1, we have

$$\textsc{Opt}[G] = \textsc{Opt}[G[U]] \le \textsc{Opt}[G'] + \textsc{Opt}[G'']. \tag{1}$$

We give upper bounds on $\textsc{Opt}[G']$ and $\textsc{Opt}[G'']$ separately.

▷ Claim 3.4. $\textsc{Opt}[G'] = |F[U_s]| = |U_s|/2$.

Proof. Clearly, $\textsc{Opt}[G'] \ge |U_s|/2$ as the output of Algorithm 2 has these many edges in $E[U_s] \subseteq E'$.

Let F' be a maximum-size properly colored forest of G' that covers every vertex in U_s; note that such a forest exists by Lemma 3.1. Suppose to the contrary that $|F'| > |U_s|/2$. Then, either there is an edge $e = uv \in F' \setminus E[U_s]$, or there are edges $e_1 = uv_1$ and $e_2 = uv_2$ with $c(e_1) \ne c(e_2)$ such that $e_1, e_2 \in F' \cap E[U_s]$. In particular, there are matchings $N_1, \dots, N_k$ with $N_i \subseteq E'_i$ such that they either cover every vertex in $U_s + v$ and $uv \in N_{c(uv)}$, or they cover U_s and $e_1 \in N_{c(e_1)}$ and $e_2 \in N_{c(e_2)}$. Both cases lead to a contradiction, since the algorithm would have found such matchings $N_1, \dots, N_k$ in Step 11 or Step 13. Therefore, $\textsc{Opt}[G'] = |U_s|/2$ indeed holds. ◁

We use the following simple observation to bound $\textsc{Opt}[G'']$.

▷ Claim 3.5. If an edge $e \in E[U] \setminus E'$ connects two components of F, then there exists an edge in $F[U_r]$ which is adjacent to e and has the same color.

Proof. Since $E[U_s] \subseteq E'$, e has at least one endpoint in U_r. If $e = vw$ such that $v \in U_s$ and $w \in U_r$, then $c(e) \in c(\delta_F(w))$ by $e \notin E'$ and the definition of E'. Otherwise, e is spanned by U_r, and since it was not added to F in Step 7, it is adjacent to an edge of F having the same color. ◁

With the help of the claim, we can bound $\textsc{Opt}[G'']$.

▷ Claim 3.6. $\textsc{Opt}[G''] \le 3 \cdot |F[U_r]|$.

Proof. Let F'' be a maximum-size properly colored forest of G''. For each edge $f \in F[U_r]$, F'' has at most two edges adjacent to f having color $c(f)$. Then, Claim 3.5 implies that F'' has at most $2 \cdot |F[U_r]|$ edges connecting different components of F. As F'' is a forest, it has at most $|F[U_r]|$ edges spanned by a component of $F[U_r]$, thus $|F''| \le 3 \cdot |F[U_r]|$ follows. ◁

Using (1), Claim 3.4, and Claim 3.6, we get $\textsc{Opt}[G] \le \textsc{Opt}[G'] + \textsc{Opt}[G''] \le |F[U_s]| + 3 \cdot |F[U_r]| = |F| + 2 \cdot |F[U_r]|$, which yields

$$|F| \ge \textsc{Opt}[G] - 2 \cdot |F[U_r]|. \tag{2}$$

Using that $|U| \ge \textsc{Opt}[G[U]] = \textsc{Opt}[G]$, we get

$$2 \cdot |F| = |U_s| + 2 \cdot |F[U_r]| = |U| - |U_r| + 2 \cdot |F[U_r]| \ge \textsc{Opt}[G] - |U_r| + 2 \cdot |F[U_r]|. \tag{3}$$

Since each component of $F[U_r]$ has size at least three, we have $|F[U_r]| \ge 2/3 \cdot |U_r|$. Thus (3) implies

$$8|F| \ge 4 \cdot \textsc{Opt}[G] - 4 \cdot |U_r| + 8 \cdot |F[U_r]| \ge 4 \cdot \textsc{Opt}[G] + 2 \cdot |F[U_r]|. \tag{4}$$

By adding (2) and (4), we obtain $9 \cdot |F| \ge 5 \cdot \textsc{Opt}[G]$, proving the approximation factor.

By Remark 3.1, each step of the algorithm can be performed in polynomial time, and the total number of for loops in Steps 10 and 12 is also clearly polynomial in the number of edges of the graph. Hence it remains to show that the algorithm makes polynomially many steps back to Step 3. This follows from the fact that whenever the algorithm returns to Step 3, a local improvement was found and so the sum $|U_s| + |\operatorname{comp}(F)|$ strictly decreases that can happen at most $2n$ times. This concludes the proof of the theorem. ◀

The analysis in Theorem 3.3 is tight for k-edge-colored multigraphs if $k \geq 4$; see [2, Figure 4a] for an example.

3.4 Simple graphs and multigraphs with small numbers of colors

While Algorithm 2 provides a 5/9-approximation in general, the approximation factor can be improved if the graph is simple or the number of colors is small. In what follows, we show how to get better guarantees when G is simple or $k \leq 3$.

▶ **Theorem 3.7.** *Algorithm 2 provides a 4/7-approximation for* Max-PF *in simple graphs and in 3-edge-colored multigraphs.*

Proof. We use the notation and extend the proof of Theorem 3.3. Consider an instance where G is simple or $k = 3$; this assumption is in fact used only in the next simple observation.

▷ Claim 3.8. Let C be a component of F with $|C| = 3$. If $|F''[C]| = 2$, then there exist $e \in F''[C]$ and $f \in F[U_r]$ such that $c(e) = c(f)$ and e and f has at least one common endpoint.

Proof. If G is simple, then $|E[C]| \leq 3$, thus $|F''[C] \cap F[U_r]| \geq 1$. If $k = 3$, then $|c(F''[C]) \cap c(F[C])| \geq 1$, that is, $c(e) = c(f)$ for some $e \in F''[C]$ and $f \in F[C]$. Since $|C| = 3$, e and f have at least one common endpoint. ◁

Let $m_3 := |\{C \in \operatorname{comp}(F) \mid |C| = 3\}|$. Using Claim 3.8, we strengthen Claim 3.6 as follows.

▷ Claim 3.9. $\textsc{Opt}[G''] \leq 3 \cdot |F[U_r]| - m_3$.

Proof. Let $\gamma := |\{C \in \operatorname{comp}(F) \mid |C| = 3, |F''[C]| = 2\}|$. Let F''_1 denote the set of edges $uv \in F''$ such that u and v are in different components of F, and let $F''_2 := F'' \setminus F''_1$. Claim 3.5 and Claim 3.8 imply that F'' has at least $|F''_1| + \gamma$ edges e for which there exists $f \in F[U_r]$ such that $c(e) = c(f)$ and e and f has at least one common endpoint. For each $f \in F[U_r]$, F'' has at most two edges having the same color as f and sharing at least one common endpoint with f, implying $2 \cdot |F[U_r]| \geq |F''_1| + \gamma$. Since F has $m_3 - \gamma$ size-three components spanning at most one edge of F''_2, we have $|F''_2| \leq |F[U_r]| - (m_3 - \gamma)$. Then, $|F''| = |F''_1| + |F''_2| \leq (2 \cdot |F[U_r] - \gamma) + (|F[U_r] + \gamma - m_3) = 3 \cdot |F[U_r]| - m_3$, and the claim follows. ◁

Using (1), Claim 3.4 and Claim 3.9, we get $\textsc{Opt}[G] \leq \textsc{Opt}[G'] + \textsc{Opt}[G''] \leq |F[U_s]| + 3 \cdot |F[U_r]| - m_3 = |F| + 2 \cdot |F[U_r]| - m_3$, that is,

$$|F| \geq \textsc{Opt}[G] - 2 \cdot |F[U_r]| + m_3. \tag{5}$$

Since $F[U_r]$ has m_3 components of size three and the other components of $F[U_r]$ has size at least four, we have $|F[U_r]| \geq 2m_3 + 3/4 \cdot (|U_r| - 3m_3) = 3/4 \cdot |U_r| - 1/4 \cdot m_3$. Then, (3) implies

$$6 \cdot |F| \geq 3 \cdot \textsc{Opt}[G] - 3 \cdot |U_r| + 6 \cdot |F[U_r]| \geq 3 \cdot \textsc{Opt}[G] + 2 \cdot |F[U_r]| - m_3. \tag{6}$$

By adding (5) and (6), we obtain $7 \cdot |F| \geq 4 \cdot \textsc{Opt}[G]$, proving the approximation factor. ◀

The analysis in Theorem 3.7 is tight for 3-edge-colored multigraphs and for k-edge-colored simple graphs for $k \geq 4$; see [2, Figures 4b and 4c] for examples.

▶ **Theorem 3.10.** *Algorithm 2 provides a 3/5-approximation for* Max-PF *in 2-edge-colored multigraphs.*

Proof. We use the notation and extend the proof of Theorem 3.3 assuming that $k = 2$. For $e \in F''$, define

$$x(e) \coloneqq |\{f \in F[U_r] \mid c(e) = c(f),\ e \text{ and } f \text{ has at least one common endpoint}\}|.$$

For a subset $S \subseteq F''$, we use the notation $x(S) \coloneqq \sum_{e \in S} x(e)$.

▷ Claim 3.11. $x(F''[C]) \geq |F''[C]| - 1$ for every even component $C \in \text{comp}(F[U_r])$, and $x(F[C]) \geq |F''[C]|$ for every odd component $C \in \text{comp}(F[U_r])$.

Proof. Let $\ell \coloneqq |C|$. Since $k = 2$, $F[C]$ is an alternating path, let $v_1, v_2, \ldots, v_\ell$ denote its vertices and $f_1, \ldots, f_{\ell-1}$ denote its edges such that $f_i = v_i v_{i+1}$ for $i \in [\ell - 1]$. For each edge $e \in F''[U_r]$ we have $x(e) \geq 1$ unless $e = v_1 v_\ell$ and $c(e) \neq c(f_1) = c(f_{\ell-1})$. This proves the claim since $c(f_1) \neq c(f_{\ell-1})$ if $\ell = |C|$ is odd. ◁

Let $m_3 \coloneqq |\{C \in \text{comp}(F) \mid |C| = 3\}|$. Using Claim 3.11, we strengthen Claim 3.6 as follows.

▷ Claim 3.12. $\text{Opt}[G''] \leq |F[U_r]| + |U_r| + m_3$.

Proof. $F[U_r]$ has $|U_r| - |F[U_r]|$ components, thus it has at most $|U_r| - |F[U_r]| - m_3$ even components. Claim 3.5 implies that $x(e) \geq 1$ holds for each edge $e \in F''$ connecting two components of F. Using Claim 3.11, it follows that $x(F'') \geq |F''| - (|U_r| - |F[U_r]| - m_3)$. For each edge $f \in F[U_r]$, F'' has at most two edges having the same color as f and at least one common endpoint of f, thus $x(F'') \leq 2|F[U_r]|$. Then, $|F''| \leq x(F'') + |U_r| - |F[U_r]| - m_3 \leq |F[U_r]| + |U_r| - m_3$, and the claim follows. ◁

Using (1), Claim 3.4 and Claim 3.12, we get $\text{Opt}[G] \leq \text{Opt}[G'] + \text{Opt}[G''] \leq |F[U_s]| + |F[U_r]| + |U_r| - m_3 = |F| + |U_r| - m_3$, that is,

$$|F| \geq \text{Opt}[G] - |U_r| + m_3. \tag{7}$$

As in the proof of Theorem 3.7, $|F[U_r] \geq 3/4 \cdot |U_r| - 1/4 \cdot m_3$, thus (3) implies

$$4 \cdot |F| \geq 2 \cdot \text{Opt}[G] - 2|U_r| + 4|F[U_r]| \geq 2 \cdot \text{Opt}[G] + |U_r| - m_3. \tag{8}$$

By adding (7) and (8), we get $5 \cdot |F| \geq 3 \cdot \text{Opt}[G]$, proving the approximation factor. ◀

The analysis in Theorem 3.10 is tight for 2-edge-colored multigraphs; see [2, Figure 4d] for an example.

3.5 Simple graphs with small numbers of colors

For simple graphs, the algorithm can be significantly simplified while leading to even better approximation factors if the number of colors is small. The modified algorithm is presented as Algorithm 3. First, we consider the case $k = 2$.

▶ **Theorem 3.13.** *Algorithm 3 provides a $\frac{3}{4}$-approximation for* Max-PF *in 2-edge-colored simple graphs in polynomial time.*

Algorithm 3 Approximation algorithm for MAX-PF in simple graphs.

Input: A simple graph $G = (V, E)$ with edge-coloring $c\colon E \to [k]$.
Output: A properly colored forest F in G.

1 Find maximum matchings $M_i \subseteq E_i$ for $i \in [k]$ maximizing $|\bigcup_{i=1}^k V(M_i)|$.
2 Let $F' := \bigcup_{i=1}^k M_i$.
3 Take a maximum forest F in F'.
4 **return** F

Proof. Let M_1 and M_2 denote the maximum matchings found in Step 1 of the algorithm. Then in Step 2, F' is a properly colored edge set which is the vertex-disjoint union of paths and even cycles. As the graph is simple, every cycle has length at least 4. In Step 3, we delete no edge from the paths and exactly one edge from each cycle. Since every cycle had length at least 4, we deleted at most $1/4 \cdot (|M_1| + |M_2|)$ edges and hence the algorithm outputs a solution of size $|F| \geq 3/4 \cdot (|M_1| + |M_2|)$. On the other hand, $\text{OPT}[G] \leq |M_1| + |M_2|$ clearly holds since every properly colored forest of G decomposes into the union of a matching in E_1 and a matching in E_2. This concludes the proof of the theorem. ◀

The analysis in Theorem 3.13 is tight for 2-edge-colored simple graphs; see [2, Figure 5a] for an example.

▶ Remark 3.2. Note that the proof of Theorem 3.13 only uses that M_1 and M_2 are maximum matchings and does not rely on the fact that $|V(M_1 \cup M_2)|$ is maximized.

Now we discuss the case when $k = 3$.

▶ **Theorem 3.14.** *Algorithm 3 provides a $\frac{5}{8}$-approximation for MAX-PF in 3-edge-colored simple graphs in polynomial time.*

Proof. Let M_1, M_2 and M_3 denote the maximum matchings found in Step 1 of the algorithm. Then in Step 2, F' is a properly colored edge set in which every vertex has degree at most 3.

▷ Claim 3.15. $|F'(C)| = 1$ for every component $C \in \text{comp}(F')$ of size 2.

Proof. The statement follows by the assumption that G is simple. ◁

▷ Claim 3.16. $|F'(C)| \leq 3/2 \cdot |C|$ for every even component $C \in \text{comp}(F')$.

Proof. The statement follows from the fact that each vertex has degree at most 3 in F'. ◁

▷ Claim 3.17. $|F'(C)| \leq 3/2 \cdot (|C| - 1)$ for every odd component $C \in \text{comp}(F')$.

Proof. Suppose to the contrary that $|F'(C)| > 3/2 \cdot (|C| - 1)$. Since every vertex has degree at most 3 in F', C either contains $|C| - 2$ vertices of degree 3 and two vertices of degree 2, or $|C| - 1$ vertices of degree 3 and one vertex u of degree at least one in F'. However, the former case cannot happen as C is an odd component and the sum of the degrees of the vertices is an even number, namely $2|F'|$. Let $e \in F'$ be an edge incident to u. Since every vertex in $C - u$ has degree exactly 3, each vertex in C is incident to an edge of color $c(e)$ in F'. However, F' is a properly colored edge set, hence the edges in $F'(C)$ colored by $c(e)$ form a perfect matching of C, contradicting $|C|$ being odd. ◁

For $i \in [n]$, let m_i denote the number of components in $\text{comp}(F')$ containing i vertices. Furthermore, let $m \coloneqq \sum_{i=2}^{\lfloor n/2 \rfloor} m_{2i}$, that is, m is the number of even components in F' of size at least four. Using Claim 3.15, Claim 3.16 and Claim 3.17, we get

$$\begin{aligned}
2 \cdot |F'| &\leq \sum_{\substack{C \in \text{comp}(F') \\ |C|=2}} 2 + \sum_{\substack{C \in \text{comp}(F') \\ C \text{ is even} \\ |C| \geq 4}} 3 \cdot |C| + \sum_{\substack{C \in \text{comp}(F') \\ C \text{ is odd}}} 3 \cdot (|C|-1) \\
&= \sum_{C \in \text{comp}(F')} 3 \cdot (|C|-1) - m_2 + 3m \\
&= 3 \cdot |F| - m_2 + 3m.
\end{aligned}$$

Note that $\text{OPT}[G] \leq |M_1| + |M_2| + |M_3| = |F'|$ clearly holds since every properly colored forest of G decomposes into the union of a matching in E_1, a matching in E_2, and a matching in E_3. Then, by rearranging the previous inequality, we get

$$3 \cdot |F| \geq 2 \cdot \text{OPT}[G] + m_2 - 3m. \tag{9}$$

Let $U \coloneqq \bigcup_{i=1}^{3} V(M_i)$. Since each matching-coverable set can be covered by maximum matchings, U is a maximum-size matching-coverable set, thus $\text{OPT}[G] = \text{OPT}[G[U]]$ holds by Lemma 3.1. Now F is a forest, thus $|F| = |F[U]| = |U| - \sum_{i=2}^{n} m_i = |U| - m_2 - m - \sum_{j=1}^{\lfloor (n-1)/2 \rfloor} m_{2j+1}$, that is, $\sum_{j=1}^{\lfloor (n-1)/2 \rfloor} m_{2j+1} = |U| - m_2 - m - |F|$. Using this equation and the fact that U is the union of the components of F with size at least two, we have

$$\begin{aligned}
2 \cdot |U| = 2 \cdot \sum_{i=2}^{n} i \cdot m_i &\geq 4m_2 + 8m + 6 \cdot \sum_{j=1}^{\lfloor (n-1)/2 \rfloor} m_{2j+1} \geq 4m_2 + 8m + 5 \cdot \sum_{j=1}^{\lfloor (n-1)/2 \rfloor} m_{2j+1} \\
&= 4m_2 + 8m + 5 \cdot (|U| - m_2 - m - |F|) = 5 \cdot |U| - m_2 + 3m - 5 \cdot |F|.
\end{aligned}$$

Rearranging and using $|U| \geq \text{OPT}[G[U]] = \text{OPT}[G]$, we obtain

$$5 \cdot |F| \geq 3 \cdot \text{OPT}[G] - m_2 + 3m. \tag{10}$$

By adding (9) and (10), we get $8 \cdot |F| \geq 5 \cdot \text{OPT}[G]$, proving the approximation factor. ◀

The analysis in Theorem 3.14 is tight for 3-edge-colored simple graphs; see [2, Figure 5b] for an example.

▶ Remark 3.3. A key ingredient of Algorithm 3 is that it starts with maximum matchings $M_i \subseteq E_i$, which makes it possible to compare the size of the solution output by the algorithm against $\text{OPT} \leq \sum_{i=1}^{k} |M_i|$. In contrast, Algorithm 2 starts with arbitrary matchings $M_i \subseteq E_i$ maximizing $|V(\bigcup_{i=1}^{k} M_i)|$. The reason why that algorithm operates with matchings instead of maximum matchings is that in certain steps we need to find matchings containing some fixed edges, hence they cannot necessarily be chosen to be maximum matchings.

3.6 Approximating Max-PT

Finally, for any $\varepsilon > 0$ we give an $1/\sqrt{(2+\varepsilon)(n-1)}$-approximation algorithm for MAX-PT in complete multigraphs. The approximation factor is far from being constant; still, the algorithm is of interest since its approximation guarantee is better than the general upper bound on the approximability of MAX-PT.

Our algorithm for MAX-PT in complete multigraphs is presented as Algorithm 4.

Algorithm 4 Approximation algorithm for MAX-PT in complete multigraphs.

Input: A complete multigraph $G = (V, E)$ with edge-coloring $c\colon E \to [k]$ and $\varepsilon > 0$.
Output: A properly colored tree F in G such that $|F| \geq \text{OPT}/\sqrt{(2+\varepsilon)(n-1)}$.

1 **if** $\exists v, w \in V, |E[\{v, w\}]| \geq n$ **then**
2   Choose n parallel edges between v and w arbitrarily and delete the remaining ones.
3 Let $F := \emptyset$ and $n_\varepsilon := (\varepsilon^2 + 9\varepsilon + 18)/\varepsilon^2$.
4 **if** $n < n_\varepsilon$ **then**
5   Compute all properly colored trees in G and let F_{opt} be one with maximum size.
6     $F \leftarrow F_{opt}$
7 **else**
8   Compute V_1, V_2 and optimal properly colored tree F_i of $G[V_i]$ for $i \in [2]$ as in Theorem 2.2.
9   Let $E' := \{vw \mid v \in V_1, w \in V_2, c(vw) \notin c(\delta_{F_1}(v))\}$.
10   Compute a properly colored forest $F_{12} \subseteq E'$ that covers a maximum number of vertices in V_2 and $|\delta_{F_{12}}(v)| \leq 1$ for each $v \in V_2$.
11   **if** $|F_1| + |F_{12}| \geq |F_2|$ **then**
12         $F \leftarrow F_1 \cup F_{12}$
13   **else**
14         $F \leftarrow F_2$
15 **return** F

▶ **Theorem 3.18.** *For complete multigraphs on n vertices and for any fixed constant $\varepsilon > 0$, Algorithm 4 provides a $1/\sqrt{(2+\varepsilon)(n-1)}$-approximation for MAX-PT in polynomial time.*

Proof. First, we show that deleting edges in Step 2 does not decrease the size of the optimal solution. Indeed, for any optimal solution F_{opt}, if $e = vw \in F_{opt}$ but e is deleted, then there are at least n parallel edges between v and w having different colors. As the degrees of v and w are at most $n-1$ in F_{opt}, there is always at least one edge f among those parallel ones such that $F_{opt} - e + f$ is a properly colored tree again. Note that after the deletion of unnecessary parallel edges, the total number edges of the graph is bounded by n^3.

Let $\varepsilon > 0$ be the parameter of the algorithm. If $n < n_\varepsilon = (\varepsilon^2 + 9\varepsilon + 18)/\varepsilon^2$, then the output is clearly optimal. Furthermore, the number of possible solutions is bounded by $\binom{n_\varepsilon^3}{n_\varepsilon}$ which is a constant, hence the runtime is constant.

Assume now that $n \geq n_\varepsilon$. Let $V_1 \cup V_2$ be the partition of V as in Theorem 2.2. We may assume that $V_1, V_2 \neq \emptyset$ since otherwise the algorithm clearly gives an optimal solution. Let F_1 and F_2 be maximum-size properly colored trees in $G[V_1]$ and $G[V_2]$, respectively. Let $n_1 := |V_1|$, $n_2 := |V_2|$ and $x_1 := \text{OPT}[G[V_1]] = |F_1| = n_1 - 1$, $x_2 := \text{OPT}[G[V_2]] = |F_2|$. The forest F_{12} in Step 10 can be determined using a maximum bipartite matching algorithm in a bipartite graph $H = (S, T; W)$ defined as follows. The vertex set S contains a vertex (v, i) for each $v \in V_1$ and color $i \in [k]$ such that v has no incident edges in F_1 having color i, that is, $S = \{(v, i) \mid v \in V_1, i \notin c(\delta_{F_1}(v))\}$. The vertex set T contains a copy of each vertex in V_2, that is, $T = \{v \mid v \in V_2\}$. Finally, there is an edge added between $(v, i) \in S$ and $u \in T$ in W if $uv \in E$ has color i in G. It is not difficult to check that a maximum matching of H gives a properly colored forest F_{12} that can be added to F_1 and with respect to that, covers as many vertices in V_2 as possible.

For the output F of Algorithm 4, either we have $|F| = x_2$ or $|F| = x_1 + y$, where $y = |F_{12}|$. Recall that $n_1 \geq 1$ by our assumption, hence $x_1 + y \geq 1$. Indeed, this clearly holds if $n_1 \geq 2$, while if $n_1 = 1$ then $y \geq 1$ by the completeness of the multigraph. Let F_{opt} be an optimal properly colored tree in G. We claim that $\textsc{Opt}[G] = |F_{opt}| \leq 3x_1 + y + (2x_1 + y)x_2$. To see this, let $U \coloneqq \{u \in V_2 \mid \text{there exists } uv \in F_{opt} \text{ with } v \in V_1\}$ and set $U' \coloneqq \{u \in U \mid c(uv) \in c(\delta_{F_1}(v)) \text{ for every } uv \in F_{opt} \text{ with } v \in V_1\}$. Since every edge of F_1 is adjacent to at most two edges in F_{opt} having the same color, we have $|U'| \leq 2x_1$. Moreover, by the choice of F_{12}, we have $|U \setminus U'| \leq y$. These together imply $|U| \leq 2x_1 + y$. Now $F_{opt} \setminus E[V_1 \cup U]$ is the union of properly colored trees in $G[V_2]$, all of which can have size at most x_2. By the above, there are at most $|U| = 2x_1 + y$ such components as F_{opt} is connected, leading to $|F_{opt} \setminus E[V_1 \cup U]| \leq (2x_1 + y)x_2$. Finally, observe that $|F_{opt} \cap E[V_1 \cup U]| \leq 3x_1 + y$ by $|V_1| \leq x_1 + 1$ and $|U| \leq 2x_1 + y$. Since F_{opt} has at most $|V| - 1$ edges, these together show $\textsc{Opt}[G] = |F_{opt}| \leq \min\{n - 1, 3x_1 + y + (2x_1 + y)x_2\}$.

The approximation factor of Algorithm 4 is hence at least $\max\{x_1 + y, x_2\}/\min\{3x_1 + y + (2x_1 + y)x_2, n - 1\}$. To lower bound this expression, let $x_1' \coloneqq x_1 + y$. Then, it suffices to show that

$$\frac{\max\{x_1', x_2\}}{\min\{n - 1, 3x_1' + 2x_1'x_2\}} \geq \frac{1}{\sqrt{(2 + \varepsilon)(n - 1)}}$$

for $1 \leq x_1' \leq n - 1$ and $0 \leq x_2 \leq n - 1$, since the value on the left hand side is a lower bound on the approximation factor. Assume that this is not the case, and in particular, we have $x_2 < \sqrt{n-1}/\sqrt{2+\varepsilon}$ and $x_1'/(3x_1' + 2x_1'x_2) < 1/\sqrt{(2+\varepsilon)(n-1)}$ for some $n \geq n_\varepsilon$. From the latter inequality, we get $\sqrt{(2+\varepsilon)(n-1)}/2 - 3/2 < x_2$. Therefore, $\sqrt{(2+\varepsilon)(n-1)}/2 - 3/2 < x_2 < \sqrt{n-1}/\sqrt{2+\varepsilon}$. However, $\sqrt{(2+\varepsilon)(n-1)}/2 - 3/2 \geq \sqrt{n-1}/\sqrt{2+\varepsilon}$ whenever $n \geq n_\varepsilon$, a contradiction.

We conclude that Algorithm 4 provides a $1/\sqrt{(2+\varepsilon)(n-1)}$-approximation. Also, by Theorem 2.2 and the fact that a maximum-size matching can be computed in polynomial time, the running time is polynomial. This concludes the proof of the theorem. ◀

▶ Remark 3.4. For $\varepsilon = 2$, the algorithm provides a $1/(2\sqrt{n-1})$-approximation and $n_2 = 10$. That is, the brute force approach of Step 5 is only executed for $n \leq 9$. However, any properly colored tree with two edges gives a $1/(2\sqrt{n-1})$-approximation, and deciding the existence of such a tree requires $\binom{|E|}{2}$ steps.

References

1. Anna Adamaszek, Matthias Mnich, and Katarzyna Paluch. New approximation algorithms for (1,2)-TSP. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICS.ICALP.2018.9.
2. Yuhang Bai, Kristóf Bérczi, Gergely Csáji, and Tamás Schwarcz. Approximating maximum-size properly colored forests. *arXiv preprint*, 2024. arXiv:2402.00834.
3. Jørgen Bang-Jensen and Gregory Gutin. Alternating cycles and paths in edge-coloured multigraphs: a survey. *Discrete Mathematics*, 165–166:39–60, 1997. doi:10.1016/s0012-365x(96)00160-4.
4. Piotr Berman and Marek Karpinski. 8/7-approximation algorithm for (1, 2)-TSP. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, pages 641–648, 2006.
5. Valentin Borozan, W Fernandez de La Vega, Yannis Manoussakis, C Martinhon, Rahul Muthu, Hong Phong Pham, and Rachid Saad. Maximum colored trees in edge-colored graphs. *European Journal of Combinatorics*, 80:296–310, 2019. doi:10.1016/j.ejc.2018.02.027.

6 Kamalika Chaudhuri, Satish Rao, Samantha Riesenfeld, and Kunal Talwar. What would Edmonds do? Augmenting paths and witnesses for degree-bounded MSTs. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 26–39. Springer, Berlin, Heidelberg, 2005. doi:10.1007/11538462_3.

7 Kamalika Chaudhuri, Satish Rao, Samantha Riesenfeld, and Kunal Talwar. What would Edmonds do? Augmenting paths and witnesses for degree-bounded MSTs. *Algorithmica*, 55(1):157–189, 2009. doi:10.1007/s00453-007-9115-5.

8 Yangyang Cheng, Mikio Kano, and Guanghui Wang. Properly colored spanning trees in edge-colored graphs. *Discrete Mathematics*, 343(1):111629, 2020. doi:10.1016/j.disc.2019.111629.

9 Artur Czumaj and Willy-B Strothmann. Bounded degree spanning trees. In *Algorithms—ESA'97: 5th Annual European Symposium Graz, Austria, September 15–17, 1997 Proceedings 5*, pages 104–117. Springer, Berlin, Heidelberg, 1997. doi:10.1007/3-540-63397-9_9.

10 Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17:449–467, 1965. doi:10.4153/cjm-1965-045-4.

11 Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and Their Applications*, pages 69–87. Gorden and Breach, New York, 1970. (Also in *Combinatorial Optimization — Eureka, You Shrink!*, pages 11–26, Springer, Berlin, Heidelberg, 2003.).

12 Jack Edmonds and Delbert Ray Fulkerson. Transversals and matroid partition. *Journal of Research of the National Bureau of Standards (B)*, 69:147–153, 1965.

13 Jinfeng Feng, Hans-Erik Giesen, Yubao Guo, Gregory Gutin, Tommy Jensen, and Arash Rafiey. Characterization of edge-colored complete graphs with properly colored Hamilton paths. *Journal of Graph Theory*, 53(4):333–346, 2006. doi:10.1002/jgt.20188.

14 Ted Fischer. Optimizing the degree of minimum weight spanning trees. Technical report, Department of Computer Science, Cornell University, 1993.

15 András Frank. *Connections in Combinatorial Optimization*, volume 38 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2011.

16 Martin Fürer and Balaji Raghavachari. Approximating the minimum degree spanning tree to within one from the optimal degree. In *Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms (Orlando, FL, 1992)*, pages 317–324. ACM, New York, 1992.

17 Michel X Goemans. Minimum bounded degree spanning trees. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 273–282. IEEE, 2006. doi:10.1109/FOCS.2006.48.

18 Jie Hu, Hao Li, and Shun-ichi Maezawa. Maximum properly colored trees in edge-colored graphs. *Journal of Combinatorial Optimization*, 44(1):154–171, 2022. doi:10.1007/s10878-021-00824-z.

19 Mikio Kano, Shun-ichi Maezawa, Katsuhiro Ota, Masao Tsugaki, and Takamasa Yashima. Color degree sum conditions for properly colored spanning trees in edge-colored graphs. *Discrete Mathematics*, 343(11):112042, 2020. doi:10.1016/j.disc.2020.112042.

20 Mikio Kano and Masao Tsugaki. Rainbow and properly colored spanning trees in edge-colored bipartite graphs. *Graphs and Combinatorics*, 37(5):1913–1921, 2021. doi:10.1007/s00373-021-02334-5.

21 Richard M Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer, Boston, MA, 1972. doi:10.1007/978-1-4684-2001-2_9.

22 Marek Karpinski and Richard Schmied. On approximation lower bounds for TSP with bounded metrics. *arXiv preprint*, 2012. arXiv:1201.5821.

23 Sanjeev Khanna, Rajeev Motwani, Madhu Sudan, and Umesh Vazirani. On syntactic versus computational views of approximability. *SIAM Journal on Computing*, 28(1):164–191, 1998. doi:10.1137/s0097539795286612.

24 Tamás Király, Lap Chi Lau, and Mohit Singh. Degree bounded matroids and submodular flows. *Combinatorica*, 32(6):703–720, 2012. doi:10.1007/s00493-012-2760-6.
25 J Könemann and R Ravi. A matter of degree: Improved approximation algorithms for degree-bounded minimum spanning trees. *SIAM Journal on Computing*, 31(6):1783–1793, 2002. doi:10.1137/S009753970036917X.
26 André Linhares, Neil Olver, Chaitanya Swamy, and Rico Zenklusen. Approximate multi-matroid intersection via iterative refinement. *Mathematical Programming*, 183:397–418, 2020. doi:10.1007/s10107-020-01524-y.
27 James Oxley. *Matroid Theory*, volume 21 of *Oxford Graduate Texts in Mathematics*. Oxford University Press, Oxford, second edition, 2011. doi:10.1093/acprof:oso/9780198566946.001.0001.
28 Christos H Papadimitriou and Mihalis Yannakakis. The traveling salesman problem with distances one and two. *Mathematics of Operations Research*, 18(1):1–11, 1993.
29 Ramamoorthi Ravi, Madhav V Marathe, Sekharipuram S Ravi, Daniel J Rosenkrantz, and Harry B Hunt III. Many birds with one stone: Multi-objective approximation algorithms. In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 438–447. Association for Computing Machinery, New York, 1993. doi:10.1145/167088.167209.
30 Mohit Singh and Lap Chi Lau. Approximating minimum bounded degree spanning trees to within one of optimal. *Journal of the ACM*, 62(1):1–19, 2015. doi:10.1145/2629366.
31 Rico Zenklusen. Matroidal degree-bounded minimum spanning trees. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1512–1521. ACM, New York, 2012. doi:10.1137/1.9781611973099.120.
32 Xianghui Zhong. On the approximation ratio of the k-Opt and Lin-Kernighan algorithm. In *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, pages 83:1–83:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.ESA.2020.83.