The Parameterized Complexity of Some Minimum Label Problems

Michael R. Fellows[*] Jiong Guo[†] Iyad A. Kanj[‡]

Abstract

We study the parameterized complexity of several minimum label graph problems, in which we are given an undirected graph whose edges are labeled, and a property Π, and we are asked to find a subset of edges satisfying property Π that uses the minimum number of labels. These problems have a lot of applications in networking. We show that all the problems under consideration are W[2]-hard when parameterized by the number of used labels, and that they remain W[2]-hard even on graphs whose pathwidth is bounded above by a small constant. On the positive side, we prove that most of these problems are FPT when parameterized by the solution size, that is, the size of the sought edge set. For example, we show that computing a maximum matching or an edge dominating set that uses the minimum number of labels, is FPT when parameterized by the solution size. Proving that some of these problems are FPT is nontrivial, and requires interesting and elegant algorithmic methods that we develop in this paper.

1 Introduction

In this paper we consider several *minimum label graph problems* that are defined as follows:

> **Input**: A graph $G = (V, E)$ whose edges are associated with labels or colors specified by a function $\mathcal{C} : E \to C$, where C denotes the set of labels (also referred to as colors in this paper), a graph property Π, and an integer d.
> **Output**: A set $E' \subseteq E$ such that the subgraph of G consisting of the set of edges in E' satisfies Π, and the number of labels/colors used by the edges in E' is at most d.

Minimum label problems have been extensively studied in the last few years. These problems are motivated by applications from telecommunication networks, electrical networks, and multi-modal transportation networks. For example, in communication networks, there are different types of communication media, such as optic fiber, cable, microwave, and telephone line. A communication node may communicate with different nodes by choosing different types of communication media. Given a set of communication network nodes, the problem of finding a connected communication network using as few types of communication media (i.e., labels/colors) as possible is exactly the Minimum Label Spanning Tree problem, in which the property Π is the property of being a spanning tree of G (see [5, 14] for more details). Among the minimum label problems that have been extensively studied, we mention the Minimum Label Spanning Tree problem [1, 2, 3, 5, 9, 11, 14, 15, 18, 19, 20], the Minimum Label Path problem [2, 4, 9, 17, 21] (where Π is the property of being a path between two designated vertices), the Minimum Label Cut problem [10, 21] (where Π is the property of being a cut between two designated vertices), and the Minimum Label Perfect Matching problem [12] (where Π is the property of being a perfect matching).

The previous work on minimum label problems mainly dealt with determining the classical complexity of these problems and studying their approximabilty. Some of the previous work, however, dealt with developing exact algorithms for these problems. For example, Broersma et al. [2] devised two exact algorithms for the Minimum Label Path and Minimum Label Cut problems with running time

[*]The University of Newcastle, Callaghan, NSW 2308, Australia. michael.fellows@newcastle.edu.au.
[†]Institut für Informatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, D-07743 Jena, Germany. guo@minet.uni-jena.de.
[‡]School of Computing, DePaul University, 243. S. Wabash Avenue, Chicago, IL 60604, USA. ikanj@cs.depaul.edu.

$O(n \cdot \min\{|C|^{d(s,t)}, 2^{|C|}\})$ and $O(n^2 \cdot |C|!)$, respectively, where C denotes the set of labels (colors), and $d(s,t)$ denotes the distance between the two designated vertices s and t.

In the current paper we study the parameterized complexity of several minimum label graph problems, with respect to two natural parameters: the number of used labels d, and the size of the solution $|E'|$. The problems under consideration are: Minimum Label Spanning Tree (MLST), Minimum Label Hamiltonian Cycle (MLHC) (where Π is the property of being a Hamiltonian cycle), Minimum Label Cut (MLC), Minimum Label Edge Domination Set (MLEDS) (where Π is the property of being an edge dominating set, that is, every edges in $E \setminus E'$ shares at least one endpoint with some edge in E'), Minimum Label Perfect Matching (MLPM), Minimum Label Maximum Matching (MLMM) (where Π is the property of being a maximum matching of G), and Minimum Label Path (MLP).

From some of the NP-hardness reductions for the above problems, we can derive parameterized intractability results with respect to the parameter d; for example, the NP-hardness reduction for Minimum Label Spanning Tree shows that this problem is W[2]-hard [11]. In this paper, we strengthen these intractability results by showing that, even on graphs whose pathwidth is at most a small constant, when parameterized by the number of used labels d, these problems remain W[2]-hard. These results are interesting, as very few natural parameterized problems are known to be (parameterized) intractable on graphs with bounded pathwidth. When parameterized by the solution size $|E'|$, we show that, with the only exceptions of Minimum Label Path and Minimum Label Cut, which are W[1]-hard, all other problems are fixed-parameter tractable (on general graphs). Showing that some of these problems are FPT is non-trivial, and requires elegant algorithmic methods that we develop in this paper.

All the hardness results will be presented in Section 2, while Section 3 contains all the fixed-parameter tractability results.

For the background and terminologies on graphs, we refer the reader to West [16], and for that on parameterized complexity, we refer the reader to Downey and Fellows' book [7].

2 Parameterized Hardness Results

First, we show that even on graphs whose pathwidth is at most a small constant, all the considered minimum label problems are W[2]-hard, when parameterized by the number of used labels d. These results are very interesting since there are few problems that are known to be W-hard on graphs of bounded pathwidth.

Theorem 2.1 *Parameterized by the number of used labels d:*

- Minimum Label Edge Dominating Set *and* Minimum Label Maximum Matching *are W[2]-hard on trees of pathwidth at most 1;*
- Minimum Label Spanning Tree *and* Minimum Label Path *are W[2]-hard on graphs with pathwidth at most 2;*
- Minimum Label Cut *and* Minimum Label Perfect Matching *are W[2]-hard on graphs with pathwidth at most 3; and,*
- Minimum Label Hamiltonian Cycle *is W[2]-hard on graphs with pathwidth at most 5.*

Proof. All the corresponding FPT-reductions are from the W[2]-hard Hitting Set (HS) problem, defined as follows. Given a ground set S, a collection $\mathcal{L}$ of subsets of S, and a nonnegative integer k, decide if there exists a subset S' of S of cardinality at most k, such that every subset in $\mathcal{L}$ has a non-empty intersection with S'. We only give one FPT-reduction showing that Minimum Label Spanning Tree (MLST) is W[2]-hard. The reductions for the other problems are similar.

For a given instance of HS, we construct a graph G where, for each subset c in $\mathcal{L}$, there is a star consisting of a root vertex and $|c|$ leaves. The edges in this star are labeled with the elements of c. Then, we connect the leaves of this star by a path whose edges have the same label x, where $x \notin S$. Finally, we connect all root vertices by a path whose edges have the same label x. Clearly, the resulting graph has pathwidth 2. Observe that every size-d solution of the HS-instance corresponds to a solution of the resulting MLST-instance using $d+1$ labels, and vice versa. This gives the W[2]-hardness of MLST. □

Next, we consider Minimum Label Cut (MLC) and Minimum Label Path (MLP) with the size of the set E' as the parameter.

Theorem 2.2 *Parameterized by the solution size $|E'|$:*

- Minimum Label Cut *is W[1]-hard on graphs with pathwidth at most 4, and*
- Minimum Label Path *is W[1]-hard on graphs with pathwidth at most 2.*

Proof. We give an FPT-reduction for MLC from the W[1]-hard Multicolored Clique problem [8]. Multicolored Clique has as input a graph G, together with a proper k-coloring of the vertices of G, and the question is whether there is a k-clique in G consisting of exactly one vertex from each color class. The parameter is the clique size k.

To construct an MLC-instance from a Multicolored Clique instance $(G = (V, E), k)$, we partition the edges in E into $\binom{k}{2}$ subsets, each containing the edges between two color classes. For each subset of edges, we create in the MLC-instance a path between two designated vertices s and t whose length is equal to the size of this subset; each edge of the path is in a one-to-one correspondence with an edge in this subset. Finally, we replace each edge of the path by two parallel length-2 paths, and these two length-2 paths are labeled by the two endpoints of the corresponding edge, respectively. In the resulting MLC-instance we ask for an s-t cut of size at most $2 \times \binom{k}{2}$, using at most k labels.

Since there are exactly $2 \times \binom{k}{2}$ edge-disjoint paths between s and t, every solution of the MLC-instance contains exactly $2 \times \binom{k}{2}$ edges whose labels correspond to k vertices from the Multicolored Clique instance. Those vertices must induce exactly $\binom{k}{2}$ many edges in G. The converse is also easy to check. Thus, there is a correspondence between the solutions of both instances. Moreover, the resulting MLC-instance is clearly a graph whose pathwidth is equal to 4.

The FPT-reduction for Minimum Label Path works analogously. □

3 Fixed-Parameter Tractability Results

Parameterized by the solution size, Minimum Label Spanning Tree, Minimum Label Perfect Matching, and Minimum Label Hamiltonian Cycle are all fixed-parameter tractable, since the instance size is bounded by a function of the parameter. However, it requires much more effort to show that Minimum Label Maximum Matching (MLMM) and Minimum Label Edge Dominating Set (MLEDS) are fixed-parameter tractable with respect to the same parameter. We note that we are mainly concerned with establishing the fixed-parameter tractability of MLMM and MLEDS. Consequently, the running time of the parameterized algorithms developed in this paper is not very practical, and can definitely be improved much further.

3.1 Minimum Label Maximum Matching (MLMM)

Let (G, k) be an instance of MLMM, where k is the size of a maximum matching in G. We denote by $e(G)$ and $n(G)$ the number of edges and vertices, respectively, in G. Let M be a maximal matching in G, $I = V(G) \setminus V(M)$, and note that I is an independent set in G. We denote by $G[M]$ the subgraph of G induced by the endpoints of the edges in M.

The algorithm is a search-tree based algorithm: it starts by growing a set of partial solutions, i.e., matchings, into an optimal solution, i.e., a maximum matching that uses the minimum number of colors. To do so, the algorithm branches on some vertices and edges in G to decide whether they belong to an optimal solution or not. Since the branching will consider all possibilities, we will maintain the invariant that at least one partial solution, among all partial solutions we keep, can be extended to an optimal solution. The algorithm can be split into several stages, each trying to simplify the resulting instance further by possibly performing more branchings. In order for the reader to get a feel of what these stages are trying to achieve, and how together they contribute to the final solution, we give an intuitive description of each stage first.

In Stage 1, we branch on the vertices and edges in $G[M]$ to determine which ones belong to an optimal solution. At the end of this stage, the edges in $G[M]$ will be removed, as well as some of its vertices. We will be left with a bipartite graph whose first partition S is a subset of vertices in $G[M]$, consisting of the endpoints of the edges that belong to an optimal solution (under the corresponding branching), and whose second partition is a subset of vertices in I. We note that during this stage some edges in $G[M]$ will be added to the partial solutions, and hence, their colors are decided to be used by the optimal solution. Moreover, the parameter k is decremented by a value equal to the number of edges added to the partial solution.

In Stage 2, we start with a bipartite graph $B = (S, I)$, and we would like to compute a maximum matching that matches S into I, and that uses the minimum number of colors, under the constraint that some colors have already been determined (from Stage 1) to be used by an optimal solution. In this stage we will simplify the instance further. We branch by enumerating all possible partitions of S into groups S_i, $i = 1, \ldots, \ell$, such that there is an optimal solution in which all vertices in S_i are matched using edges of the same color—we will call such a set of edges a *monochromatic matching*. For a fixed partitioning of S into groups, we compute, for each group S_i, the set $\mathcal{M}_i$ of monochromatic matchings that match S_i into I. If $|\mathcal{M}_i|$ is bounded above by a predefined function of k, then we can compute a matching in $\mathcal{M}_i$ that is part of an optimal solution by trying (branching on) all monochromatic matchings in $\mathcal{M}_i$, and subsequently remove S_i from S. If all monochromatic matchings in $\mathcal{M}_i$ use the same color, we branch on every vertex in $\mathcal{M}_i$ whose degree in $\mathcal{M}_i$ is larger than a predefined function of k.

In Stage 3, we can assume that, for each remaining group S_i, $|\mathcal{M}_i|$ is larger than a predefined function of k, and for each $\mathcal{M}_i$ whose monochromatic matchings all use the same color, the degree of every vertex in $\mathcal{M}_i$ is larger than a predefined function of k. We show in this case that an optimal solution can be computed easily (without any branching): a matching M' that matches S into I exists, such that the set of edges in M' incident on each group S_i is a monochromatic matching in $\mathcal{M}_i$.

Stage 1

Let M_{opt} be an optimal solution that we are trying to compute. For every edge e in $G[M]$ we branch as follows.

- e in M_{opt}: in this case we include e, decrement k by 1, and remove e and its endpoints from the graph. We also record that the color $\mathcal{C}(e)$ is used in the optimal solution.
- e is not in M_{opt}: in this case we simply remove e, that is, we set $G := G - e$.

For every remaining vertex v in $G[M]$ we branch as follows.

- v in M_{opt}: in this case we keep v in the graph.
- v is not in M_{opt}: in this case we remove v by setting $G := G - v$.

Let S be the set of remaining vertices in $G[M]$, and note that since all the edges in $G[M]$ have been removed during the branching, S is an independent set. Moreover, assuming that our partial solution (branching) is valid (i.e., leads to an optimal solution), every vertex in S must be an endpoint of an edge in the optimal solution M_{opt}. Without loss of generality, and since the parameter k can only decrease during the branching, we will denote the resulting parameter by k; this will simplify the notation in the remaining discussion. Assuming that our branching is valid, we have the following observation.

Observation 1 *The following are true:*

(a) $|S| = k$*, and hence,*

(b) for every $u \in I$*,* $deg(u) \leq k$*.*

Let $B = (S, I)$ be the resulting bipartite graph from G after the branching. The remaining task amounts to computing a matching with the minimum number of colors that matches S into I—and hence has size k, under the constraint that some of the colors have been used.

Analysis of the number of partial solutions enumerated in Stage 1

Since $|M| \leq k$, the number of vertices in $G[M]$ is at most $2k$, and the number of edges in $G[M]$ is at most $\binom{2k}{2} = k(2k-1)$.

The branching in Stage 1 can be implemented as follows. For each $i = 0, \ldots, k$, we choose a matching of size i from the edges in $G[M]$ to be included in M_{opt}. For each of the remaining at most $(2k-2i)$ vertices in $G[M]$, we branch on it as indicated above, thus creating at most 2^{2k-2i} partial solutions. Therefore, the number of partial solutions enumerated in Stage 1 is bounded above by:

$$\begin{aligned}
\sum_{i=0}^{k} \binom{k(2k-1)}{i} 2^{2k-2i} &= 4^k \sum_{i=0}^{k} \binom{k(2k-1)}{i} 1/4^i && (1)\\
&\leq 4^k \binom{k(2k-1)}{k} \sum_{i=0}^{k} 1/4^i && (2)\\
&\leq 4^k \cdot (e(2k-1))^k \cdot O(1) && (3)\\
&\leq 4^k \cdot (2ek)^k \cdot O(1) = O((8ek)^k).
\end{aligned}$$

Inequality (2) is justified by the fact that the coefficient $\binom{k(2k-1)}{k}$ is the largest coefficient in the summation. Inequality (3) uses the fact that $\binom{n}{k} \leq (en/k)^k$, where e is the base of the natural logarithm (for instance, see [6]). It follows that the number of partial solutions enumerated in Stage 1 is $O((8ek)^k)$.

Stage 2

Given the bipartite graph $B = (S, I)$ and the parameter k, we try in this stage to simplify the instance further by performing more branching. We say that a matching is *monochromatic* if all its edges have the same color. If M' is a monochromatic matching, we denote by $\mathcal{C}(M')$ the color of the edges in M'.

We would like to partition S into groups such that all the vertices in the same group are matched in M_{opt} by a monochromatic matching of a distinct color. For this purpose we try all possible partitions of S. For a fixed partition of S into ℓ groups $S_1, \ldots, S_\ell$, we work under the assumption that the vertices in each group are matched by a monochromatic matching in M_{opt} of a distinct color (with respect to the colors of the other groups). Clearly, there exists at least one partition of S for which this working hypothesis is true, namely the one induced by the color classes in M_{opt}.

Let $S_1, \ldots, S_\ell$ be a fixed partition of S into ℓ nonempty groups, where $1 \leq \ell \leq k$ is an integer. It is possible that a group S_i uses the color of an edge that was added to a partial solution in Stage 1. Therefore, for each (possibly empty) subset C_{used} of the set of colors of the edges added in Stage 1, we try all one-to-one mappings from C_{used} to $\{S_1, \ldots, S_\ell\}$. Fix such a mapping. Then some groups in $\{S_1, \ldots, S_\ell\}$ have been assigned colors, and hence the colors of the monochromatic matchings sought for these groups are fixed. Clearly, since we are trying all possible assignments of the used colors to the groups, there will be an assignment that corresponds to that of M_{opt}, and we are safe.

Let S_i, $i \in \{1, \ldots, \ell\}$, be a group. If S_i has a preassigned color, let c_i be this color and define $\mathcal{M}_i = \{M_i \mid M_i \text{ is a monochromatic matching that matches } S_i \text{ into } I \text{ and } \mathcal{C}(M_i) = c_i\}$. Otherwise, the color of S_i is undetermined yet, and in this case define $\mathcal{M}_i = \{M_i \mid M_i \text{ is a monochromatic matching that matches } S_i \text{ into } I\}$.

Let $h(k)$ be a function of k to be determined later, and let S_i, $i \in \{1, \ldots, \ell\}$, be a group. We perform more branching to simplify the instance as follows.

If $|\mathcal{M}_i| \leq h(k)$, we branch on every matching in $\mathcal{M}_i$ as the matching that matches S_i in M_{opt}. For each branch corresponding to a matching M_i in $\mathcal{M}_i$, we add the edges in M_i to the potential solution, decrement k by $|S_i|$, remove the vertices in $V(M_i)$ from the graph, and remove every edge whose color is $\mathcal{C}(M_i)$ from the graph (such an edge can no longer be used). Since we are trying all possible matchings M_i in $\mathcal{M}_i$, we are safe.

If the total number of colors used by the matchings in $\mathcal{M}_i$ is at most $h(k)$, we branch by trying all possible colors appearing in $\mathcal{M}_i$ to determine the color used in M_{opt} to match S_i (this color has to be one of the colors in $\mathcal{M}_i$). For each such color c, we remove all the edges in $\mathcal{M}_i$ whose colors are different from c. Again, since we are branching on all possible colors in $\mathcal{M}_i$, we are safe.

If all the edges of the matchings in $\mathcal{M}_i$ have the same color, and if there exists a vertex v in S_i with at most $h(k)$ edges incident on it in the matchings in $\mathcal{M}_i$, we branch on which edge in a matching in $\mathcal{M}_i$ matches v in M_{opt}. For each branch corresponding to an edge e_v, we add e_v to the potential solution, remove the endpoints of e_v from the graph, and decrement k by 1. We can now assume the following.

Assumption 3.1 *For each $i \in \{1, \ldots, \ell\}$:*

(i) $|\mathcal{M}_i| > h(k)$.

(ii) Either the number of colors appearing in $\mathcal{M}_i$ *is more than* $h(k)$*, or it is exactly 1.*

(iii) If $\mathcal{M}_i$ *has exactly one color appearing in it, then every vertex in* S_i *has more than* $h(k)$ *edges that are incident on it in the matchings in* $\mathcal{M}_i$.

In the next stage we show how, given the above assumption, we can easily compute a solution to the resulting instance.

Analysis of the number of partial solutions enumerated in Stage 2

Let c_{used} be the number of colors used in Stage 1. The number of partitions of S into ℓ groups is at most $\ell^{|S|} \leq \ell^k$. For each partition of S into ℓ groups, where $\ell \geq c_{used}$, we map the colors used in a one-to-one fashion to a subset of the ℓ groups. There are at most $\ell!/(\ell - c_{used})! \leq \ell!$ such mappings. Therefore, the total number of partitions of S in which some of the ℓ groups (exactly c_{used} many groups among them) have been assigned the used colors is at most $\sum_{\ell=1}^{k} \ell^k \ell! \leq k^{k+1}k!$.

Now for each S_i, $i \in \{1, \ldots, \ell\}$, we compute at most $h(k)+1$ monochromatic matchings $M_i \in \mathcal{M}_i$. To do so, we iterate over each color c, and compute at most $h(k)+1$ monochromatic matchings of color c. For a fixed color c, we consider the subgraph of B consisting only of the edges incident on S_i whose color is c. Note that each matching in this subgraph that matches S_i into I is a maximum matching. It was shown in [13] how, after computing a maximum matching in a bipartite graph, every other maximum matching can be computed in linear time in the number of vertices of the subgraph, per matching. Therefore, computing at most $h(k)+1$ monochromatic matchings of color c that match S_i into I can be done in time $O(e(G)\sqrt{n(G)} + n(G)h(k))$. As a matter of fact, since whenever we fix a color c for a group S_i we only look at the edges of color c incident on the vertices in S_i, and since we totally compute at most $h(k)+1$ matchings incident on the vertices in S_i, computing at most $h(k)+1$ monochromatic matchings (regardless of the color) incident on the vertices of S_i can be done in time $O(e(G)\sqrt{n(G)} + n(G)h(k))$. Since there are at most k groups, computing the sets $\mathcal{M}_i$, $i = 1, \ldots, \ell$, can be done in time $O(ke(G)\sqrt{n(G)} + kh(k)n(G))$.

To make the graph B satisfy the statements in Assumption 3.1, we do the following. After computing the set $\mathcal{M}_i$ for each group S_i as indicated above, we check if $|\mathcal{M}_i| \leq h(k)$. If it is, we branch on every monochromatic matching in $\mathcal{M}_i$. For each such matching M_i, we remove the endpoints of the edges in M_i, and hence the group S_i from the graph, and decrease the parameter by $|S_i|$. Since we are branching on every monochromatic matching in $\mathcal{M}_i$, we are safe. Since there are at most $h(k)$ matchings in $\mathcal{M}_i$, and at most k groups S_i, the total number of enumerations is at most $h(k)^k$.

Now we can assume that the cardinality of each set $\mathcal{M}_i$ is at least $h(k)+1$.

If there is a set $\mathcal{M}_i$ such that the total number of colors appearing in it is at most $h(k)$, then we branch by trying every color in $\mathcal{M}_i$ as the color used to match S_i in M_{opt}. For each such color c, we remove all the edges incident on S_i whose color is different from c, and we remove every edge whose color is c but is not incident on a vertex in S_i. The total number of enumerations is again at most $h(k)^k$.

Finally, if we have a set $\mathcal{M}_i$ such that all the matchings in this set have the same color c, then for every vertex v (if any) in S_i whose degree in $\mathcal{M}_i$ is at most $h(k)$, we branch on which edge in $\mathcal{M}_i$ is used to match v in M_{opt}. For each edge in $\mathcal{M}_i$ incident on v, we remove the endpoints of the edge from the graph and decrement k by 1. Since we are trying all possible edges incident on such a vertex v, we are safe. The total number of enumerations in this case is at most $h(k)^k$ (there are at most k vertices in S).

We can now assume that B satisfies the statements in Assumption 3.1. The total number of enumerations incurred to make B satisfy the statements in Assumption 3.1 is at most $h(k)^k \cdot h(k)^k \cdot h(k)^k = h(k)^{3k}$.

It follows that the number of partial solutions enumerated in Stage 2 is bounded above by the number of partitions of S, multiplied by the number of enumerations to make B satisfy the statements in Assumption 3.1. From the above discussion, it follows that the number of partial solutions enumerated in Stage 2 is $O(k^{k+1}k! + h(k)^{3k})$.

Stage 3

Given an instance $B = (S, I)$ and a parameter k such that S is partitioned into $S_1, \ldots, S_\ell$, where each set $\mathcal{M}_i$ associated with S_i, for $i = 1, \ldots, \ell$, satisfies the statements of Assumption 3.1, we show how to compute a matching M' that matches S into I, and such that the set of edges in M' incident on S_i is a monochromatic matching whose edges are edges from the matchings in $\mathcal{M}_i$.

Theorem 3.2 *Let $h(k) \geq k^2 + k$. Assuming that each $\mathcal{M}_i$, $i = 1, \ldots, \ell$, satisfies Assumption 3.1, then we can compute a matching M' that matches S into I, such that the set of edges in M' incident on S_i, for $i = 1, \ldots, \ell$, is a monochromatic matching whose edges are edges from the matchings in $\mathcal{M}_i$.*

Proof. Starting with S_1, we pick a monochromatic matching $M_1 \in \mathcal{M}_1$ that matches S_1 into I. Let $I_1 = V(M_1) \cap I$. Inductively, assume that we have determined a monochromatic matching M_j, where $1 \leq j < \ell$, such that the edges in M_j are edges from the matchings in $\mathcal{M}_j$, and such that $I_j = M_j \cap I$ is disjoint from $I_1 \cup \ldots \cup I_{j-1}$. We show how to determine a monochromatic matching M_{j+1} whose edges are edges from the matchings in $\mathcal{M}_{j+1}$, and such that $I_{j+1} = M_{j+1} \cap I$ is disjoint from $I_1 \cup \ldots \cup I_j$. We distinguish two cases:

Case 1. $\mathcal{M}_{j+1}$ contains more than $h(k)$ colors. By Observation 1-(b), each vertex in I has degree at most k. Since $|I_1 \cup \ldots \cup I_j| \leq |S| \leq k$, there are at most k^2 edges incident on $I_1 \cup \ldots \cup I_j$. Since $\mathcal{M}_{j+1}$ contains more than $h(k)$ monochromatic matchings of distinct colors, the number of monochromatic matchings in $\mathcal{M}_{j+1}$ whose edges are incident on $I_1 \cup \ldots \cup I_j$ is at most k^2. Therefore, the fact that $h(k) > k^2$ guarantees the existence of a monochromatic matching $M_{j+1} \in \mathcal{M}_{j+1}$ whose endpoints in I are disjoint from $I_1 \cup \ldots \cup I_j$. Consequently, I_{j+1} is disjoint from $I_1 \cup \ldots \cup I_j$.

Case 2. $\mathcal{M}_{j+1}$ contains a single color. By Assumption 3.1-(iii), every vertex in S_{j+1} has more than $h(k)$ edges incident on it in $\mathcal{M}_{j+1}$. As in **Case 1** above, the number of edges incident on $I_1 \cup \ldots \cup I_j$ is at most k^2. Since $h(k) \geq k^2 + k$, for every vertex in S_{j+1}, there are at least k edges incident on it in $\mathcal{M}_{j+1}$ such that none of them is incident on a vertex in $I_1 \cup \ldots \cup I_j$. Moreover, all these edges (for all $v \in S_{j+1}$) have the same color. By Hall's theorem [16] (note that $|S_{j+1}| \leq k$), there is a matching M_{j+1} whose edges are edges from the matchings in $\mathcal{M}_{j+1}$, and such that I_{j+1} is disjoint from $I_1 \cup \ldots \cup I_j$. □

Analysis of the running time of Stage 3

This stage involves no enumerations. Moreover, it is easy to see that, in both **Case 1** and **Case 2**, computing the matching M_j takes $O(|\mathcal{M}_j| \cdot |S_j|)$ time. Therefore, computing the matching M' takes $O(k^3)$ time.

Putting all together

The correctness of the algorithm follows from the fact that it is enumerating all possible branchings. For each possible branching, either we reject the instance, or we end up computing a maximum matching that uses a certain number of colors. The maximum matching we output at the end is the maximum matching with the minimum number of colors. The running time of the algorithm is bounded by the number of partial solutions enumerated, multiplied by the running time spent along each enumeration (path in the search tree). The number of partial solutions we enumerate is the product of those enumerated in Stage 1 ($O((8ek)^k)$) and Stage 2 ($O(k^{k+1}k! + h(k)^{3k})$), which is $O((8e)^k \cdot k^{7k})$ after choosing $h(k) = k^2 + k$. Along each path in the search tree we end up processing the graph G, which takes linear time in its number of vertices and edges, computing a maximum matching in G, which takes $O(e(G)\sqrt{n(G)})$, and computing the sets $\mathcal{M}_i$ in Stage 2, which takes $O(ke(G)\sqrt{n(G)} + k^3 n(G))$. Therefore, the running time of the algorithm is $O((8e)^k \cdot k^{7k+3} e(G)\sqrt{n(G)})$.

Theorem 3.3 Minimum Label Maximum Matcing *is FPT when parameterized by the size of the maximum matching in the graph.*

3.2 Minimum Label Edge Dominating Set (MLEDS)

The ideas used by the algorithm are similar in flavor to those used for the MLMM problem. Therefore, we will omit some details to avoid repetition. We start with the following easy observation.

Observation 2 *Let M be a matching in G, and let Q be an edge dominating set of G. Then $|Q| \geq |M|/2$.*

Let (G, k) be an instance of MLEDS. Let M be a maximal matching in G, $I = V(G) \setminus V(M)$, and note that I is an independent set in G. If $|M| > 2k$, then by Observation 2, G does not have an edge dominating set of size at most k, and we can reject the instance (G, k). Therefore, we may assume henceforth that $|M| \leq 2k$.

Similar to what we did for the MLMM problem, we will branch on the edges and vertices in M to determine which ones contribute to a solution Q_{opt}, which is an edge dominating set of G of size at most k that uses the minimum number of colors (if such a solution exists).

For an edge $e \in G[M]$, we branch on e as follows. If e is decided to be in Q_{opt}, we set $G = G - e$, decrement k by 1, mark all the edges incident on e in the graph as dominated, and label both endpoints of e with the label "IN_{used}" to indicate that they are in Q_{opt}, and are endpoints of some edge that is already decided to be in Q_{opt}. (We will use the label "IN" later to indicate that the vertex is decided to be in Q_{opt} but has no incident edge that was decided to be in Q_{opt} yet.) We also indicate that the color of e has been used by storing it in a set of colors C_{used}. On the other hand, if e is decided not be in Q_{opt}, we set $G = G - e$.

For a vertex $v \in G[M]$ whose status has not been determined yet by the above branching (i.e., v does not have the label IN_{used}), we branch on v as follows. If v is decided to be an endpoint of an edge in Q_{opt}, we label v as IN, and mark every edge incident on v as dominated. If v is decided not be an endpoint of an edge in Q_{opt}, we label it as OUT.

Note that since I is an independent set in G, every edge in G must be dominated by an edge in Q_{opt} with at least one endpoint in $G[M]$. In particular, this is true for every edge in $G[M]$. Therefore, after branching on the edges and vertices in $G[M]$, we need to check that, for every edge $e \in G[M]$ that was decided not to be in Q_{opt} and subsequently removed from G, at least one of its endpoints has label IN or IN_{used}. If this is not the case, then the partial solution that we have enumerated is not valid, and we reject it.

Noting that after the above branching all the edges of $G[M]$ were removed from G, we end up with a bipartite graph $B = (S, I)$, where S consists of the vertices in $G[M]$. Every vertex in S has one of the following labels: (1) IN_{used} indicating that the vertex is an endpoint of a known edge which was determined to be in Q_{opt}, (2) IN indicating that the vertex is the endpoint of some edge in Q_{opt} but this edge has not been determined yet, and (3) OUT indicating that the vertex is not an endpoint of an edge in Q_{opt}. The edges in B have one of two possible types: (1) dominated, those are the edges with at least one endpoint of label IN_{used} or IN, and (2) not dominated, and those are the edges whose endpoint in S is of label OUT.

Since we are trying all possible branches for the edges and vertices in $G[M]$, we are safe. The number of partial solutions enumerated by the branching can be upper bounded in a similar fashion to that in Stage 1 of the algorithm for MLMM. The only difference here is that the number of edges in the maximal matching M is at most $2k$, and hence, the number of vertices in $G[M]$ is at most $4k$, and consequently the number of edges in $G[M]$ is at most $2k(4k - 1)$. Using a similar analysis to that in Stage 1 of MLMM, we obtain that the number of partial solutions enumerated by the above branching is at most $(128ek)^k$.

Now given the instance $B = (S, I)$, and the parameter k (without loss of generality), we will branch further to simplify the instance. First, observe that since the number of edges in Q_{opt} is at most k, the number of vertices in S that are labeled with IN_{used} or IN is at most $2k$ (otherwise, we reject the enumeration).

Observation 3 *For every vertex w in I, the number of edges incident on w whose endpoint in S is labeled with IN_{used} or IN is at most $2k$.*

Note that, for every edge $e = \{u, v\}$ where $u \in S$ has label OUT, e needs to be dominated by an edge incident on v; therefore, the vertex v must be an endpoint of some edge in Q_{opt}. Since the number of edges in Q_{opt} is at most k, and B is bipartite, there can be at most k vertices in I that are neighbors of vertices in S of label OUT; let I_{in} be the set of such vertices. Since (by Observation 3) every vertex in I has at most $2k$ edges incident on it whose endpoint in S is labeled IN_{used} or IN, we can branch on every such edge incident on a vertex in I_{in} to determine if the edge is in Q_{opt} or not. For each such edge, if the edge is decided to be in Q_{opt}, we include the edge in the solution, label both its endpoints IN_{used}, we remove the edge, decrement k by 1, and update C_{used} appropriately; if the edge is decided not be in Q_{opt}, we simply remove it. After this branching, we check that for every vertex in I_{in} at least one of the edges incident on it was decided to be in Q_{opt}; otherwise, we reject the branch. The number of partial solutions generated by this branching is at most $(2k)^k$.

After branching on the edges incident on the vertices in I_{in} and removing them, the vertices in I_{in} and the vertices in S of label OUT can be removed. Every remaining vertex in S is either of label IN_{used} or IN.

Since a vertex in S of label IN_{used} is an endpoint of an edge already in Q_{opt}, every edge incident on a vertex in IN_{used} is dominated. Therefore, if for every vertex of label IN in S we determine one of its incident edges to be in Q_{opt}, we obtain an edge dominating set of B. On the other hand, our branching stipulates that from every vertex in S of label IN we must determine at least one edge incident on it to be in Q_{opt}. Therefore, our problem reduces to picking for every vertex of label IN in S exactly one edge incident on it, so that the total number of colors used is minimized. To do so, we first remove the vertices of label IN_{used} from S, since no edge incident on any of them needs to be considered. At this point S should have at most k vertices; otherwise, we can reject. Then for every color c in C_{used}, and for every vertex v of label IN in S, if there is an edge of color c incident on v, we include e in the solution, decrement k, and remove the vertex from B. (Note that edges whose color is in C_{used} are gained for free.)

After this step, every vertex in S is of label IN, and there is no edge incident on any vertex in S whose color appears in C_{used}. To compute a set of edges of minimum colors, such that for every vertex in S exactly one edge in this set is incident on it, we try each partition of S into ℓ groups, $\ell \in \{1, \ldots, k\}$, such that all vertices in the same group are incident on edges of the same color in Q_{opt} (as we did in Stage 2 of the MLMM problem). For each such partition and each group in this partition, we find a color c such that every vertex in this group is incident on an edge of color c. If such a choice is not possible for some group, then we reject the partition.

At the end, we end up with an edge dominating set for G. We output the edge dominating set of G of size at most k that uses the minimum number of colors, over all partial solutions generated from all branches.

Since S has at most k vertices at this stage, the total number of partitions of S is at most k^{k+1}.

It follows that the total number of partial solutions enumerated by the algorithm is $O((128ek)^k \cdot (2k)^k \cdot k^{k+1}) = O((256e)^k k^{3k+1})$. For each such partial solution we need to process the graph G during the branching, which takes time $O(n(G) + e(G))$. Therefore, the running time of the algorithm is $O((256e)^k k^{3k+1}(n(G) + e(G)))$.

Theorem 3.4 MINIMUM LABEL EDGE DOMINATING SET *is FPT when parameterized by the size of the edge dominating set.*

4 Concluding Remarks

In this paper, we considered some minimum label graph problems. We showed that, when parameterized by the number of used labels, most of these problems are intractable, even on graphs of bounded pathwidth. On the other hand, we showed that most of these problems become parameterized tractable when parameterized by the solution size.

We note that, recently, there has been a lot of interest in studying structured graph problems, such as problems on colored graphs, due to their applications in various fields such as networking and computational biology. (The convex recoloring problem is such an example in computational biology.) While these problems are practically very important, they are often computationally hard due to the structural requirement on the solution sought. Therefore, it is both natural and interesting to study whether these problems remain intractable with respect to different parameters, such as the number of colors, the pathwidth/treewidth of the graph, the solution size, or even with respect to more restrictive parameters, such as the vertex cover or the max leaf number. This paper follows this line of research.

Finally, it is interesting to study the parameterized complexity of other minimum label graph problems that have practical applications. A good candidate would be the Minimum Label Feedback Arc Set problem on directed graphs.

References

[1] H. Broersma and X. Li. Spanning trees with many or few colors in edge-colored graphs. *Discussiones Mathematicae Graph Theory*, 17(2):259–269, 1997.

[2] H. Broersma, X. Li, G. Woeginger, and S. Zhang. Paths and cycles in colored graphs. *Australasian Journal on Combinatorics*, 31:299–311, 2005.

[3] T. Brüggemann, J. Monnot, and G. Woeginger. Local search for the minimum label spanning tree problem with bounded color classes. *Operations Research Letters*, 31(3):195–201, 2003.

[4] R. Carr, S. Doddi, G. Konjevod, and M. Marathe. On the red-blue set cover problem. In *Proc. 11th ACM-SIAM SODA*, pages 345–353, 2000.

[5] R. Chang and S. Leu. The minimum labeling spanning trees. *IPL*, 63(5):277–282, 1997.

[6] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms, 2nd edition*. McGraw-Hill Book Company, Boston, MA, 2001.

[7] R. G. Downey and M. R. Fellows. *Parameterized complexity*. Monographs in Computer Science. Springer-Verlag, New York, 1999.

[8] M. Fellows, D. Hemmelin, and R. Rosamond. On the fixed-parameter intractability and tractability of multiple-interval graph properies, 2007. Manuscript.

[9] R. Hassin, J. Monnot, and D. Segev. Approximation algorithms and hardness results for labeled connectivity problems. *Journal of Combinatorial Optimization*, 14(4):437–453, 2007.

[10] S. Jha, O. Sheyner, and J. Wing. Two formal analyses of attack graphs. In *Proc. 15th IEEE Computer Security Foundations Workshop*, pages 49–63, 2002.

[11] S. Krumke and H. Wirth. On the minimum label spanning tree problem. *IPL*, 66(2):81–85, 1998.

[12] J. Monnot. The labeled perfect matching in bipartite graphs. *IPL*, 96(3):81–88, 2005.

[13] T. Uno. Algorithms for enumerating all perfect, maximum and maximal matchings in bipartite graphs. In *Proc. 8th ISAAC*, volume 1350 of *LNCS*, pages 92–101. Springer, 1997.

[14] S. Voss, R. Cerulli, A. Fink, and M. Gentili. Applications of the pilot method to hard modifications of the minimum spanning tree problem. In *Proc. 18th MINI EURO Conference on VNS*, 2005.

[15] Y. Wan, G. Chen, and Y. Xu. A note on the minimum label spanning tree. *IPL*, 84(2):99–101, 2002.

[16] D. West. *Introduction to graph theory*. Prentice Hall Inc., Upper Saddle River, NJ, 1996.

[17] H. Wirth. *Multicriteria Approximation of Network Design and Network Upgrade Problems*. PhD thesis, Department of Computer Science, Universität Würzburg, Germany, 2005.

[18] Y. Xiong. *The Minimum Labeling Spanning Tree Problem and Some Variants*. PhD thesis, Graduate School of the University of Maryland, USA, 2005.

[19] Y. Xiong, B. Golden, and E. Wasil. A one-parameter genetic algorithm for the minimum labeling spanning tree problem. *IEEE Transactions on Evolutionary Computation*, 9(1):55–60, 2005.

[20] Y. Xiong, B. Golden, and E. Wasil. Worst case behavior of the MVCA heuristic for the minimum labeling spanning tree problem. *Operations Research Letters*, 33(1):77–80, 2005.

[21] P. Zhang, L. Tang, W. Zhao, J. Cai, and A. Li. Approximation and hardness results for label cut and related problems, 2007. Manuscript.