Parameterized Algorithms and Kernels for Rainbow Matching

Sushmita Gupta[1], Sanjukta Roy[2], Saket Saurabh[3], and Meirav Zehavi[4]

1   University of Bergen, Norway
    `Sushmita.Gupta@uib.no`
2   The Institute of Mathematical Sciences, HBNI, Chennai, India
    `sanjukta@imsc.res.in`
3   University of Bergen, Norway, and
    The Institute of Mathematical Sciences, HBNI, Chennai, India
    `saket@imsc.res.in`
4   University of Bergen, Norway
    `Meirav.Zehavi@uib.no`

Abstract

In this paper, we study the NP-complete *colorful* variant of the classical MATCHING problem, namely, the RAINBOW MATCHING problem. Given an edge-colored graph G and a positive integer k, this problem asks whether there exists a matching of size at least k such that all the edges in the matching have distinct colors. We first develop a deterministic algorithm that solves RAINBOW MATCHING on paths in time $\mathcal{O}^{\star}\left(\left(\frac{1+\sqrt{5}}{2}\right)^{k}\right)$ and polynomial space. This algorithm is based on a curious combination of the method of bounded search trees and a "divide-and-conquer-like" approach, where the branching process is guided by the maintenance of an auxiliary bipartite graph where one side captures "divided-and-conquered" pieces of the path. Our second result is a randomized algorithm that solves RAINBOW MATCHING on general graphs in time $\mathcal{O}^{\star}(2^{k})$ and polynomial-space. Here, we show how a result by Björklund et al. [JCSS, 2017] can be invoked as a black box, wrapped by a probability-based analysis tailored to our problem. We also complement our two main results by designing kernels for RAINBOW MATCHING on general and bounded-degree graphs.

Keywords and phrases Rainbow Matching, Parameterized Algorithm, Bounded Search Trees, Divide-and-Conquer, 3-Set Packing, 3-Dimensional Matching

1   Introduction

The classical notion of matching has been extensively studied for several decades in the area of Combinatorial Optimization [6, 14]. Given an undirected graph G, a set of edges is called a *matching* if the edges are pairwise non-adjacent. That is, no two edges share a common vertex. In the MAXIMUM MATCHING problem, the objective is to find a matching of maximum size. The first polynomial time algorithm for MAXIMUM MATCHING was given by Edmonds [6] in his classic paper *Paths, Trees and Flowers*. It is important to remark that this is the paper which underlined the importance of study of polynomial time algorithms for the first time. After a series of improvements, the current fastest algorithm for MAXIMUM MATCHING was given by Micali and Vazirani and it runs in time $\mathcal{O}(m\sqrt{n})$ [15]. However, finding a matching that satisfies some additional constraints often immediately becomes NP-complete, where three notable examples are MINIMUM MAXIMAL MATCHING [18], INDUCED

Matching [17] and Multiple Choice Matching [11]. In this paper, we study the NP-hard variant of Maximum Matching called Multiple Choice Matching from the viewpoint of parameterized complexity.

The Multiple Choice Matching problem, also called Rainbow Matching, is one of the NP-hard variant of Maximum Matching mentioned in the classical book by Garey and Johnson [9, Problem GT55]. In this work, we will stick to the name Rainbow Matching. This problem is formally defined as follows.

Rainbow Matching   **Parameter:** k
Input: An undirected graph G, a coloring function $\chi : E(G) \to \{1, \dots, q\}$ and a positive integer k.
Question: Does there exist a matching of size at least k such that all the edges in the matching have distinct colors?

A matching where all the edges have distinct colors will be called *colorful matching*. Itai et al. [11] showed, already in 1978, that Rainbow Matching is NP-complete on (edge-colored) bipartite graphs. Close to three decades later, Le and Pfender [13] revisited the computational complexity of this problem. Specifically, they showed that the Rainbow Matching problem is NP-complete even on (edge-colored) paths, complete graphs, P_8-free trees in which every color is used at most twice, P_5-free linear forests in which every color is used at most twice, and P_4-free bipartite graphs in which every color is used at most twice. In this paper, we consider this problem from the parameterized rather than classical complexity perspective.

A *parameterization* of a problem is the association of an integer k with each input instance, which results in a *parameterized problem*. For our purposes, we need to recall three central notions that define the parameterized complexity of a parameterized problem. The first one is the notion of a *kernel*. Here, a parameterized problem is said to admit a *kernel* of size $f(k)$ for some function f that depends *only* on k if there exists a polynomial-time algorithm, called a *kernelization algorithm*, that translates any input instance into an equivalent instance of the same problem whose size is bounded by $f(k)$ and such that the value of the parameter does not increase. In case the function f is polynomial in k, the problem is said to admit a *polynomial kernel*. Hence, kernelization is a mathematical concept that aims to analyze the power of preprocessing procedures in a formal, rigorous manner. The second notion that we use is the one of *fixed-parameter tractability (FPT)*. Here, a parameterized problem Π is said to be FPT if there is an algorithm that solves it in time $f(k) \cdot |I|^{\mathcal{O}(1)}$, where $|I|$ is the size of the input and f is a function that depends only on k. Such an algorithm is called a *parameterized algorithm*. In other words, the notion of FPT signifies that it is not necessary for the combinatorial explosion in the running time of an algorithm for Π to depend on the input size, but it can be confined to the parameter k. Finally, we recall that Parameterized Complexity also provides tools to refute the existence of polynomial kernels and parameterized algorithms for certain problems (under plausible complexity-theoretic assumptions). We refer the reader to the books [3, 5] for more information on these notions in particular, and on Parameterized Complexity in general. The notation $\mathcal{O}^{\star}(\cdot)$ is used to hide factors polynomial in the input size.

1.1 Our Contribution

Our starting point is the FPT algorithm mentioned in the article of Le and Pfender [13]. This algorithm is based on the connection between Rainbow Matching and 3-Set Packing. In the 3-Set Packing problem, we are given a universe U, a set family $\mathcal{F}$ consisting of subsets

of U of size at most 3 and an integer k, and the objective is to check whether there exists a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ containing at least k pairwise-disjoint sets. Observe that given an instance $I = (G, \chi, k)$ of RAINBOW MATCHING, we can view I as an instance of 3-SET PACKING by setting $U = V(G) \cup \{1, \ldots, q\}$, and letting $\mathcal{F}$ contain every set $\{u, v, \chi(e)\}$ corresponding to an edge $e = uv \in E(G)$. Now, observe that (G, χ, k) is a yes-instance of RAINBOW MATCHING if and only if $(U, \mathcal{F}, k)$ is a yes-instance of 3-SET PACKING. This immediately implies that known algorithms for 3-SET PACKING can be employed to solve RAINBOW MATCHING. In particular, using the known algorithms for 3-SET PACKING, we obtain the following algorithms for RAINBOW MATCHING: (1) a deterministic algorithm running in time $\mathcal{O}^\star(8.097^k)$ [19]; (2) a randomized algorithm running in time $\mathcal{O}^\star(1.4953^{3k}) = \mathcal{O}^\star(3.3434^k)$ [2].

Rainbow Matching on Paths.   Our first contribution concerns the RAINBOW MATCHING problem on paths. We obtain the following algorithm, which is faster than the one that we design later for general graphs.

▶ **Theorem 1.** *There exists a deterministic algorithm for* RAINBOW MATCHING *on paths that runs in time* $\mathcal{O}^\star\left(\left(\dfrac{1+\sqrt{5}}{2}\right)^k\right)$ *and uses polynomial space.*

The proof of Theorem 1 is based on a combination of the classical method of bounded search trees [3, 5, 7, 8] together with a "divide-and-conquer-like" approach. The algorithm always maintains a family of vertex-disjoint paths $\mathcal{S}$, and the objective is to find a colorful matching of size k that uses exactly one edge from each path in $\mathcal{S}$ and $k - |\mathcal{S}|$ edges from P. We call this variant of RAINBOW MATCHING the DISJOINT SET RAINBOW MATCHING problem. Observe that when $\mathcal{S} = \emptyset$, then DISJOINT SET RAINBOW MATCHING is precisely RAINBOW MATCHING. To compactly represent potential partial solutions to our problem at every step of the recursion, that is, partial witnesses that there may indeed exist a colorful matching that uses exactly one edge from each path in $\mathcal{S}$, our algorithm works as follows. The algorithm uses an auxiliary bipartite graph where we maintain a partial solution to our problem, in terms of a matching in this bipartite graph. This has the additional benefit that the measure becomes very simple: we just measure the size left to cover, *i.e.* $k - t$, where $t = |\mathcal{S}|$ denotes the size of the partial solution. (We remark that we are able to construct a solution in the same time as it takes to solve the decision version of the DISJOINT SET RAINBOW MATCHING problem.)

Rainbow Matching on General Graphs.   Our second contribution is an algorithm on general graphs that is better than the known algorithms for 3-SET PACKING. In particular, we obtain the following result.

▶ **Theorem 2.** *There exists a randomized algorithm for* RAINBOW MATCHING *with constant, one-sided error that runs in time* $\mathcal{O}^\star(2^k)$ *and uses polynomial space.*[1]

The proof of Theorem 2 is based on the general method described in [2] for solving various packing and matching problems. We tailor the analysis of Bjorklund et al. [2] to the RAINBOW MATCHING problem. This gives us the desired saving over the algorithm for 3-SET PACKING.

---

[1] Specifically, if the algorithm determines that an input instance is a yes-instance, then this answer is necessarily correct.

Kernelization.   Finally, we turn to consider the question of kernelization. Here, we exploit the connection between our problem and 3-SET PACKING to design a kernelization algorithm. We also design a smaller kernel for RAINBOW MATCHING on paths, or more generally, for graphs of bounded degree.

▶ **Theorem 3.** RAINBOW MATCHING *admits a kernel of size* $\mathcal{O}(k^3)$ *on general graphs. Moreover, it admits a kernel of size* $\mathcal{O}(dk^2)$ *on graphs of maximum degree d.*

1.2   Related Work

Le and Pfender [13] gave a factor $(\frac{2}{3} - \epsilon)$ approximation algorithm for RAINBOW MATCHING on general graphs for every $\epsilon > 0$. They also designed a few polynomial time algorithms when the instances of RAINBOW MATCHING are restricted to special graph classes. As stated before, Le and Pfender [13] also related this problem to 3-SET PACKING, and showed that the problem is FPT. Moreover, they showed that the problem is FPT on P_5 free forests parameterized by the number of components. Colorful matchings have also been studied from graph theoretic and combinatorial perspectives. For example, they are related to Ryser's famous conjecture regarding Latin transversal [16]. In the language of colorful matchings, the conjecture says that every proper edge coloring of the complete bipartite graph $K_{2n+1,2n+1}$ with $2n + 1$ colors contains a rainbow matching with $2n + 1$ edges. Additional examples are studies of sufficient conditions for edge-colored graphs to guarantee the existence a colorful matching of a certain size. Moreover, previous studied also examined what is the size of the largest colorful matching in an edge-colored graph with additional restrictions. For more information on these topics, we refer to [13] and references therein. Finally, let us mention that colorful matchings belong to a family of problems called rainbow subgraph problems. A *rainbow subgraph* of an edge-colored graph is a subgraph whose edges have distinct colors. We refer to [12] for a survey containing results and questions regarding rainbow subgraps.

1.3   Preliminaries

Let $[n]$ denote the set $\{1, \ldots, n\}$. For a graph G, we let $V(G)$ and $E(G)$ denote its vertex set and its edge set, respectively. For two vertices $u, v \in V(G)$, we use uv to denote an edge between u and v.

Reduction Rule.   To design our kernelization algorithm, we rely on the notion of a *reduction rule*. A reduction rule is a polynomial-time procedure that replaces an instance $(\mathcal{I}, k)$ of a parameterized problem Π (where k is the parameter) by a new instance $(\mathcal{I}', k')$ of Π. The rule is said to be *safe* if $(\mathcal{I}, k)$ is a yes-instance if and only if $(\mathcal{I}', k')$ is a yes-instance. As customary in the field we allow reduction rule to return the answer yes or the answer no (see [3, 5]).

2   Algorithm for Rainbow Matching on Paths

In this section we give a deterministic algorithm for RAINBOW MATCHING on paths that is faster than the algorithm we will present for RAINBOW MATCHING on general graphs in the next section. Towards that we will solve the following general problem, and from there solve RAINBOW MATCHING in paths.

DISJOINT SET RAINBOW MATCHING   **Parameter:** k
Input: A path P with edge coloring $\chi : E(G) \to [q]$, a collection $\mathcal{S}$ of (vertex disjoint) paths (vertex disjoint from P) of arbitrary lengths, and a positive integer k.
Question: Does there exist a colorful matching of size k that uses exactly one edge from each path in $\mathcal{S}$ and $k - |\mathcal{S}|$ edges from P?

Note that RAINBOW MATCHING is a special case of DISJOINT SET RAINBOW MATCHING, where P is the input graph (a path), $\mathcal{S} = \emptyset$ and k is the parameter. Thus, solving DISJOINT SET RAINBOW MATCHING will yield an algorithm for RAINBOW MATCHING.

Overview.   To solve DISJOINT SET RAINBOW MATCHING, whenever possible we apply reduction rules, or solve the instance in polynomial time. In the absence of either of these, the algorithm branches on an edge, based on whether it is part of the solution or not.

Measure.   We associate the measure $\mu(P, \mathcal{S}, k) = k - |\mathcal{S}|$ to the instance $(P, \mathcal{S}, k)$. We will use this measure to bound the number of nodes in the search tree. When the instance is clear from the context, we will simply use μ.

Auxiliary bipartite graph.   At every step of the search we maintain a bipartite graph $\mathcal{B}(\mathcal{S})$ on the vertex set $([q], \mathcal{S})$ and edge set containing pairs $cP' \in [q] \times \mathcal{S}$ such that color c appears on an edge in the path P' in (the collection) $\mathcal{S}$.

We first prove a lemma which allows us to solve the DISJOINT SET RAINBOW MATCHING problem when the measure is at most one.

▶ **Lemma 4.** *If $\mu(P, \mathcal{S}, k) \leq 1$, then we can test if $(P, \mathcal{S}, k)$ is a yes-instance in polynomial time.*

Proof.   We divide the proof based on whether $\mu(P, \mathcal{S}, k) < 0$ or $\mu(P, \mathcal{S}, k) = 0$ or $\mu(P, \mathcal{S}, k) = 1$. If $\mu(P, \mathcal{S}, \mathcal{C}, k) < 0$, then $k < |\mathcal{S}|$ and so clearly no matching of size k can exist which chooses exactly one edge from each path in $\mathcal{S}$.

If $\mu(P, \mathcal{S}, k) = 0$, then $k = |\mathcal{S}|$. Let $Q_1, \ldots, Q_k$ denote the paths in $\mathcal{S}$. We will show that there exists a colorful matching of size k that uses exactly one edge from each path in $\mathcal{S}$ if and only if there is a matching in $\mathcal{B}(\mathcal{S})$ that saturates $\mathcal{S}$. Let $\mathcal{M}$ be a colorful matching of size k that uses exactly one edge from each path in $\mathcal{S}$. Furthermore, for each i $(1 \leq i \leq k)$ let $m_i \in \mathcal{M}$ be the edge that is part of path Q_i. Then $\{\chi(m_i)Q_i \mid i \in [k]\}$ forms a matching that saturates $\mathcal{S}$ in $\mathcal{B}(\mathcal{S})$. In the reverse direction given a matching $\mathcal{M}'$ in $\mathcal{B}(\mathcal{S})$ that saturates $\mathcal{S}$, we obtain a colorful matching $\mathcal{M}$ that uses exactly one edge from each path Q_i in $\mathcal{S}$, as follows. Since $\mathcal{M}'$ saturates $\mathcal{S}$ we have that for every path $Q_i \in \mathcal{S}$ there is an edge jQ_i for some $j \in [q]$. This implies that there is an edge, say m_i on Q_i such that $\chi(m_i) = j$, *i.e.* m_i has color j. Since, the paths in $\mathcal{S}$ are pairwise vertex disjoint, the set $\mathcal{M}^\star = \{m_i \mid i \in [k]\}$ forms a matching in the graph P. Recall that $\mathcal{M}'$ is a matching in $\mathcal{B}(\mathcal{S})$ in which one of the endpoints of the edges are from the set $[q]$, thus, it follows that $\mathcal{M}^\star$ is a colorful matching of size k that uses exactly one edge from each path in $\mathcal{S}$. Thus, implying that in this case we can check whether or not $(P, \mathcal{S}, k)$ is a yes-instance in polynomial time by checking if the bipartite graph $\mathcal{B}(\mathcal{S})$ has a matching that saturates $\mathcal{S}$ [10].

If $\mu(P, \mathcal{S}, k) = 1$, then $k = |\mathcal{S}| + 1$. In this case, we consider every edge $e = uv$ on P. Let us now consider a specific iteration concerning an edge $e = uv$ on P. Then, we construct $\mathcal{B}(\mathcal{S} \cup \{uv\})$. Similar to the case of $\mu(P, \mathcal{S} \cup \{uv\}), k) = 0$, we have now reduced the problem into checking whether or not, there is a matching in $\mathcal{B}(\mathcal{S} \cup \{uv\}))$ that saturates $\mathcal{S} \cup \{uv\}$.

This condition can be tested in polynomial time. If in at least one iteration, we found a saturating matching then we have a yes-instance, and otherwise we have no-instance. This completes the proof. ◄

Lemma 4 yields the following reduction rule.

▶ **Reduction Rule 5.** *If $\mu(P, \mathcal{S}, k) \leq 1$, then using Lemma 4 test whether or not $(P, \mathcal{S}, k)$ is a yes-instance. If Lemma 4 returns yes, then return that $(P, \mathcal{S}, k)$ is a yes-instance; else, return that $(P, \mathcal{S}, k)$ is a no-instance.*

The safeness of Reduction Rule 5 follows from Lemma 4. The next reduction rule allows us to identify a prefix of the path P such that there exists a colorful matching of size at least k that contains *exactly* one edge from the prefix.

▶ **Reduction Rule 6.** *In the instance $(P, \mathcal{S}, k)$, let $P = v_1, v_2, \ldots, v_{n-1}, v_n$. Suppose that for every index $i \in [n-1]$ the following property is true: when the subpaths $P_{i-1} = v_1, \ldots, v_{i-1}$ and $P' = v_i, v_{i+1}$ of P are added to $\mathcal{S}$, the size of a maximum matching in the new bipartite graph $\mathcal{B}(\mathcal{S} \cup \{P_{i-1}, P'\})$ (obtained after the addition of P_{i-1} and P' to $\mathcal{S}$, and suitable edges) is at most $|\mathcal{S}| + 1$. Then, return that $(P, \mathcal{S}, \mathcal{C}, k)$ is a no-instance.*

Next we show that the correctness of Reduction Rule 6.

▶ **Lemma 7.** *Reduction Rule 6 is safe.*

Proof. For the sake of contradiction, we assume that $(P, \mathcal{S}, k)$ is a yes-instance. In this case we will show that there exists an index $i \in [n-1]$ with the following property: when the subpaths $P_{i-1} = v_1, \ldots, v_{i-1}$ and $P' = v_i, v_{i+1}$ of P are added to $\mathcal{S}$, the size of a maximum matching in the new bipartite graph $\mathcal{B}(\mathcal{S} \cup \{P_{i-1}, P'\})$ is $|\mathcal{S}| + 2$. This will be contradiction, and thereby prove the lemma.

Since $(P, \mathcal{S}, k)$ is a yes-instance there exists a colorful matching of size k, denoted by $\mathcal{M}$, that uses k colors from $[q]$, exactly one edge from each path in $\mathcal{S}$, and $k - |\mathcal{S}|$ edges from P. Thus, there is a maximum matching in the bipartite graph on $\mathcal{B}(\mathcal{S})$ that saturates every vertex in the $\mathcal{S}$-side. It is obtained by taking edges that connect a vertex in the $\mathcal{S}$-side (*i.e.* a path in the collection $\mathcal{S}$) with the color that appears on the matching edge in $\mathcal{M}$ that is part of the same path. We use $\mathcal{M}'$ to denote this bipartite matching.

Let $\{e_{j_1}, e_{j_2}, \ldots, e_{j_{k-|\mathcal{S}|}}\}$ denote the matching edges in $\mathcal{M}$ as they appear left to right in P. For some $i \geq 3$, let $v_i v_{i+1}$ denote the edge e_{j_2} (the matching edge with the second smallest index in P). It follows that edge e_{j_1} appears in the subpath $P_{i-1} = v_1, \ldots, v_{i-1}$. Note that the vertex $\chi(e_{j_2}) \in [q]$ is not saturated by $\mathcal{M}'$ in $\mathcal{B}$ because e_{j_2} is part of the matching $\mathcal{M}$ while it is inside P. Similarly, the vertex $\chi(e_{j_1}) \in [q]$ is also not saturated by $\mathcal{M}'$ in $\mathcal{B}$. Hence, when paths P_{i-1} and P' are added to the bipartite graph, $\mathcal{M}'$ can be extended by exactly two more edges: edges between $\chi(e_{j_1})$ and P_{i-1} and $\chi(e_{j_2})$ and P'. Thus, the new bipartite graph has a matching of size $|\mathcal{S}| + 2$. ◄

The safeness of Reduction Rule 6, leads us to the following conclusion.

▶ **Lemma 8.** *If $(P, \mathcal{S}, k)$ is a yes-instance on which Reduction Rules 5 and 6 are not applicable, then there exists an index $i \in [n-1]$ such that there exists a colorful matching of size k that uses exactly one edge from the subpath $P_i = v_1, \ldots, v_i$ of P. Furthermore, such an index i can be found in polynomial time.*

Proof. Since Reduction Rules 5 and 6 are not applicable we have that $\mu(P, \mathcal{S}, k) \geq 2$. This implies that $k \geq |\mathcal{S}| + 2$. Let i denote the *smallest* integer in $[n-1]$ for which the following holds:

Property (⋆⋆) when the subpaths $P_{i-1} = v_1, \ldots, v_{i-1}$ and $P' = v_i, v_{i+1}$ of P are added to $\mathcal{M}$, the size of a maximum matching in the new bipartite graph $\mathcal{B}(\mathcal{S} \cup \{P_{i-1}, P'\})$ (obtained after the addition of P_{i-1} and P' to $\mathcal{S}$, and suitable edges) is $|\mathcal{S}| + 2$.

Observe that since $k \geq |\mathcal{S}| + 2 \geq 2$, such an integer i must exist.

Note that any colorful matching of size k uses at most *one* edge from P_i, else, it contradicts the fact that i is the smallest integer that satisfies Property (⋆⋆). Also note that since we have a colorful matching of size at least 2 in P_{i+1}, so $i \geq 3$. However, since there always exists a colorful matching of size k that uses one of the first two edges of the path P; hence, the matching must use one of the edges on P_i. This implies that there exists a colorful matching of size k that uses exactly one edge on P_i. Clearly, we can find the smallest integer described in the statement of the lemma in polynomial time. This concludes the proof. ◄

Lemma 8 yields a branching rule that can be described as follows. Let P_i be the subpath of P (given by Lemma 8) such that there exists a colorful matching of size k that uses exactly one edge from it. We recursively solve two subproblems one where we assume that edge $v_i v_{i+1}$ is in the colorful matching of size k we are constructing, and the other where we assume that edge $v_i v_{i+1}$ is not part of the solution we are constructing. Note that this rule is exhaustive because an edge (in particular, $v_i v_{i+1}$) can either belong to a matching, or it does not.

Algorithm. Now we can describe the branching rule in details along with the recursive call to a subproblem. Let $\mathcal{I} = (P = v_1, \ldots, v_n, \mathcal{S}, k)$ be the instance of DISJOINT SET RAINBOW MATCHING, where none of the Reduction Rules 5 or 6 are applicable. Let P_i be the subpath of P as described in Lemma 8.

Branch 1: (**The edge $v_i v_{i+1}$ belongs to a colorful matching of size k.**)
  We recursively solve the problem on the instance $(P \setminus \{v_1, \ldots, v_{i+1}\}, \mathcal{S} \cup \{P_{i-1}\} \cup \{[v_i v_{i+1}]\}, k)$. Since the size of $\mathcal{S}$ increases by 2, the measure μ decreases by 2. Observe that by Lemma 8 we know that there exists a colorful matching of size k that uses exactly one edge from the subpath P_i. However, since we have assumed that the edge $v_i v_{i+1}$ belongs to the colorful matching of size k, thus, edge $v_{i-1} v_i$ cannot be part of the same matching, and so one of the edges in P_{i-1} must be part of the matching as well. Thus, in this case we know that two of the edges in our matching are due to the edge $v_i v_{i+1}$ and the other is from P_{i-1}.

Branch 2: (**The edge $v_i v_{i+1}$ does not belong to a colorful matching of size k.**)
  In this case we recursively solve the problem on $(P \setminus \{v_1, \ldots, v_i\}, \mathcal{S} \cup \{P_i\}, k)$. Since the size of $\mathcal{S}$ increases by 1 and k remains the same, the measure μ decreases by 1. The correctness of this step follows from Lemma 8.

If either of the branches returns "yes", we return the same. Else, we return that the given instance is a no-instance.

The resulting branching vector for this algorithm is $(2, 1)$. Thus, solving the polynomial $x^2 \geq x + 1$ for a positive root yields $x \geq \frac{1}{2}(1 + \sqrt{5}) = 1.6181$. This upper bounds the running time of our algorithm. The correctness of the algorithm follows from Lemmas 4, 7 and 8.

Recall that as explained at the very onset of our discussion: Since RAINBOW MATCHING is a special case of DISJOINT SET RAINBOW MATCHING, hence our algorithm can solve RAINBOW MATCHING by using the algorithm for the latter on the instance $(G, \mathcal{S} = \emptyset, k)$. This completes the proof of Theorem 1.

3 FPT Algorithm for Rainbow Matching on General Graphs

This section is inspired by the proof of Theorem 4 in [2], which solves 3-SET PACKING in time $\mathcal{O}^\star(3.3434^k)$. We show that by an analysis tailored to RAINBOW MATCHING, we improve upon the time complexity $\mathcal{O}^\star(3.3434^k)$. More precisely, the objective of this section is to prove Theorem 2.

Towards the proof of Theorem 2, we need to consider a problem called 3-SET PREPACKING, which was introduced in [2]. The input of this problem consists of an n-element universe U, an n_1-element subuniverse $U_1 \subseteq U$, a family $\mathcal{F}$ of 3-sets, a positive integer k, and three non-negative integers p_0, p_1 and p_2 whose sum is k. The task is to determine whether there exists a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ of size k such that the 3-sets in $\mathcal{F}'$ are pairwise-disjoint, and for all $i \in \{0, 1, 2\}$, there exist exactly p_i sets S in $\mathcal{F}'$ such that $|S \cap U_1| = i$. We would need to rely on the following result.

▶ **Proposition 9.** *There exists a randomized algorithm for* 3-SET PREPACKING *with constant, one-sided error that runs in time* $\mathcal{O}^\star(2^{3p_0+2p_1+p_2})$ *and uses polynomial space. Specifically, if the algorithm determines that an input instance is a yes-instance, then this answer is necessarily correct.*

Let us denote the algorithm given by Proposition 9 by PrepackAlg. We present a reduction from our problem to 3-SET PREPACKING. For this purpose, we describe a procedure Reduce that given an instance $(G, \chi : V(G) \to [q], k)$ of RAINBOW MATCHING, constructs an instance $\text{reduce}(G, \chi, k) = (U, U_1, \mathcal{F}, k, p_0, p_1, p_2)$ of 3-SET PREPACKING with the same parameter k. Let us denote $n_1 = |V(G)| = n - q$, where n would denote $|U|$. First, Reduce sets $U = V(G) \cup [q]$, $\mathcal{F} = \{\{u, v, \chi(uv)\} : uv \in E(G)\}$, $p_0 = 0$, $p_1 = 0$ and $p_2 = k$. Second, Reduce sets $U_1 = V(G)$. Let us now argue that we obtain an equivalent instance.

▶ **Lemma 10.** *Let* $(G, \chi : V(G) \to [q], k)$ *be an instance of* RAINBOW MATCHING. *Then,* $reduce(G, \chi, k) = (U, U_1, \mathcal{F}, k, p_0, p_1, p_2)$ *is a yes-instance of* 3-SET PREPACKING *if and only if* $(G, \chi : V(G) \to [q], k)$ *is a yes-instance of* RAINBOW MATCHING.

Proof. In the first direction suppose that $\text{reduce}(G, \chi, k) = (U, U_1, \mathcal{F}, k, p_0, p_1, p_2)$ is a yes-instance of 3-SET PREPACKING. In particular, we then have that there exists a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ of size k such that the 3-sets in $\mathcal{F}'$ are pairwise-disjoint. Let us denote $\mathcal{M} = \{uv \in E(G) : \exists S \in \mathcal{F}' \text{ s.t. } \{u, v\} \subseteq S\}$. Since $|\mathcal{F}'| = k$, we have that $|\mathcal{M}| = k$, and since the 3-sets in $\mathcal{F}'$ are pairwise-disjoint, we have that $\mathcal{M}$ is a colorful matching. Thus, $(G, \chi : V(G) \to [q], k)$ is a yes-instance of RAINBOW MATCHING.

In the other direction suppose $(G, \chi : V(G) \to [q], k)$ is a yes-instance of RAINBOW MATCHING. Then there exists a colorful matching $\mathcal{M}$ of size k. Let us denote $\mathcal{F}' = \{\{u, v, \chi(uv)\}| \ uv \in \mathcal{M}\}$. Since the size of $\mathcal{M}$ is k, we have that the size of $\mathcal{F}'$ is k and since $\mathcal{M}$ is a colorful matching we have that the sets in $\mathcal{F}'$ are pairwise disjoint. Notice that every set in $\mathcal{F}'$ exactly two elements from U_1. Therefore, for all $i \in \{0, 1, 2\}$, there exist exactly p_i sets S in $\mathcal{F}'$ such that $|S \cap U_1| = i$. Thus, $(U, U_1, \mathcal{F}, k, p_0, p_1, p_2)$ is a yes-instance of 3-SET PREPACKING. ◀

Let us now prove Theorem 2.

Proof of Theorem 2. Given an instance $(G, \chi : V(G) \to [q], k)$ of RAINBOW MATCHING, we construct the instance $\text{reduce}(G, \chi, k) = (U, U_1, \mathcal{F}, k, p_0, p_1, p_2)$ of 3-SET PREPACKING. Then, we run the algorithm given by Proposition 9. We accept if and only if the algorithm from Proposition 9 accepted. The correctness follows from Proposition 9 and Lemma 10. Since, $p_0 = p_1 = 0$ and $p_2 = k$, by Proposition 9, the total running time is $\mathcal{O}^\star(2^k)$. ◀

4   Kernelization Algorithms

In this section we give a proof for Theorem 3. We first describe a kernel on general graphs. The kernelization algorithm on general graphs is actually a known kernel for 3-SET PACKING given in [3, Theorem 12.20] (also see [1, 4]). The best known kernel for 3-SET PACKING is given by Abu-Khzam [1] and it has $\mathcal{O}(k^2)$ elements and $\mathcal{O}(k^3)$ sets. However, as we explain now, we cannot use the kernel given by Abu-Khzam [1] directly for our purposes. This is in contrast to the fact that one can use the best known parameterized algorithms for 3-SET PACKING to design parameterized algorithms for RAINBOW MATCHING. Given an instance (G, χ, k) of RAINBOW MATCHING, we can transform it to an instance $(U, \mathcal{F}, k)$ of 3-SET PACKING as explained in the introduction. Now if we apply a kernelization algorithm for 3-SET PACKING, then it will return an equivalent instance $(U', \mathcal{F}', k')$ of 3-SET PACKING and *not of* RAINBOW MATCHING. A priori, it is not clear how we can transform $(U', \mathcal{F}', k')$ to an instance of RAINBOW MATCHING without increasing the size bounds on the kernel we obtain for RAINBOW MATCHING. Thus, to design a kernelization algorithm for RAINBOW MATCHING we give a kernelization algorithm for 3-SET PACKING such that it is easy to transform an instance of the latter to an instance of RAINBOW MATCHING. This kernel is given here mainly for *completeness*.

4.1   Kernelization for Rainbow Matching on general graphs

Now we give the kernelization algorithm alluded to in the first part of Theorem 3. Towards that we will use the sunflower lemma – a classical result of Erdős and Rado. We first define the terminology used in the statement of the lemma. A *sunflower* with k petals and a *core* Y is a collection of sets $S_1, \ldots, S_k$ such that $S_i \cap S_j = Y$ for all $i \neq j$; the sets $S_i \setminus Y$ are petals and we require *none of them to be empty*. Note that a family of pairwise disjoint sets is a sunflower (with an empty core).

▶ **Proposition 11.** [3, pg 38] (Sunflower lemma) *Let $\mathcal{A}$ be a family of sets (without duplicates) over a universe U, such that each set in $\mathcal{A}$ has cardinality exactly d. If $|\mathcal{A}| > d!(k-1)^d$, then $\mathcal{A}$ contains a sunflower with k petals and such a sunflower can be computed in time polynomial in $|\mathcal{A}|$, $|U|$, and k.*

Proof of first part of Theorem 3. Given an instance (G, χ, k) of RAINBOW MATCHING, we view this as an instance $\mathcal{J}$ of 3-SET PACKING as follows: $U = V(G) \cup \{1, \ldots, q\}$ and $\mathcal{F}$ consists of a set $\{u, v, \chi(uv)\}$ corresponding to every edge $e = uv \in E(G)$.

▶ **Reduction Rule 12.** *Let $(U, \mathcal{F}, k)$ be an instance of 3-SET PACKING and suppose that $\mathcal{F}$ contains a sunflower $S = \{S_1, \ldots, S_{3(k-1)+2}\}$ of cardinality $3k-1$ with core Y. Then, return $(U', \mathcal{F}', k)$, where $U' = \bigcup_{X \in \mathcal{F}'} X$, and $\mathcal{F}' = (\mathcal{F} \setminus S_1)$ is obtained by deleting a set S_1 from $\mathcal{F}$.*

To show the correctness of Reduction Rule 12, we need to show the following lemma.

▶ **Lemma 13.** *Reduction Rule 12 is safe.*

Proof. We will prove that $(U, \mathcal{F}, k)$ is a yes-instance of 3-SET PACKING if and only if $(U', \mathcal{F}', k)$ is a yes-instance of 3-SET PACKING. It is clear that if $(U', \mathcal{F}', k)$ is a yes-instance of 3-SET PACKING then so is $(U, \mathcal{F}, k)$; so the backward direction holds straightaway.

For the forward direction, we assume that we have a solution $\mathcal{S}$ to $(U, \mathcal{F}, k)$, *i.e.*, a set of k pairwise disjoint sets. If $\mathcal{S}$ does not contain S_1, then it is also a solution for $(U', \mathcal{F}', k)$. So let us assume that $S_1 \in \mathcal{S}$. Observe that the number of elements appearing in the sets in $\mathcal{S}$, apart from those present in S_1, is $3(k-1)$. Also, note that no set in $\mathcal{S} \setminus \{S_1\}$ intersects the

core Y. Thus, the number of sets in the sunflower S that intersects the elements present in the sets of $\mathcal{S}$ is upper bounded by $1 + 3(k-1)$ (the first one for S_1). This implies there exists a set $S^\star \in S$ that is pairwise disjoint with every set in $\mathcal{S} \setminus \{S_1\}$. Thus, $(\mathcal{S} \setminus \{S_1\}) \cup S^\star$ is a solution of size k for $(U', \mathcal{F}', k)$. This completes the proof.   ◄

Now, we are ready to describe the kernelization algorithm. If the number of sets in $\mathcal{F}$ is more than $6(3k-2)^3$, then the kernelization algorithm applies the sunflower lemma to find a sunflower of size $3k-1$, and applies Reduction Rule 12 on this sunflower.

The algorithm applies this procedure exhaustively, and obtains a new family of sets $\mathcal{F}'$ of size at most $6(3k-2)^3$. This concludes the size bound on the family $(U', \mathcal{F}', k)$ of 3-Set Packing. Observe that throughout the process, we have never reduced the size of any set in the family and each set in the family still corresponds to an edge and its color. Thus, given $(U', \mathcal{F}', k)$, let W be the vertices present in U'. Then we return $(G[W], \chi', k)$, where edge coloring χ' is the restriction of χ to the edges present in $G[W]$. This concludes the description of the kernelization algorithm.   ◄

4.2   A Kernel on graphs of bounded degree

In this section we design a small kernel for Rainbow Matching on graphs of bounded degree. Let (G, χ, k) be an instance of Rainbow Matching. Throughout this section we assume that the *maximum degree* of G is upper bounded by a fixed constant d.

For $i \in [q]$, let $E_i = \{e \in E(G) \mid \chi(e) = i\}$. We call the set of edges E_i as a *color class with color i*. Next we give reduction rule that bounds the size of each color class.

▶ **Reduction Rule 14.** *If there exists $i \in [q]$ such that $|E_i| \geq 2d(k-1) + 1$ then delete E_i and reduce k by 1. That is, we obtain an instance $(G', \chi', k-1)$. Here, G' is obtained by deleting all the edges in E_i and edge coloring χ' is obtained by restricting χ to the edges in G'.*

▶ **Lemma 15.** *Reduction Rule 14 is safe.*

Proof. We will prove that G has a colorful matching of size k if and only if G' has a colorful matching of size $k-1$. We first prove the forward direction. If G has a colorful matching $\mathcal{M}$ of size k that contains an edge $uv \in E_i$, then $\mathcal{M} \setminus \{uv\}$ is a colorful matching of size $k-1$ in the graph G'. If $\mathcal{M}$ does not have an edge of color i, then $\mathcal{M}$ itself is a colorful matching of size at least $k-1$ for G'.

For the backward direction, let $\mathcal{M}'$ be a colorful matching of size $k-1$ of G'. Observe that $\mathcal{M}'$ has $2(k-1)$ distinct vertices and each vertex has degree at most d. If every edge in $\mathcal{M}'$ has an endpoint that is adjacent to an edge in E_i, then the vertices in $\mathcal{M}'$ can share at most $2d(k-1)$ vertices. That is, at most $2d(k-1)$ edges from E_i can share vertices with $\mathcal{M}'$. Since, $|E_i| \geq 2d(k-1) + 1$, E_i has at least one edge that does not share a vertex with $\mathcal{M}'$. Let that edge be uv. Then, it follows that $\mathcal{M}' \cup \{uv\}$ is a k size colorful matching of G, and our proof is complete.   ◄

We apply Reduction Rule 14 exhaustively. If the premise of the rule is not satisfied, then for each color i, we have that $|E_i| \leq 2d(k-1)$. Next we give a polynomial time procedure that either outputs a colorful matching of size at least k or bounds the number of colors.

▶ **Lemma 16.** *Let (G, χ, k) be an instance of Rainbow Matching for which Reduction Rule 14 is not applicable. Then, in polynomial time either we can conclude that (G, χ, k) is a yes-instance or the number of distinct colors in the instance is upper bounded by $2d(k-1)$.*

Proof. We iteratively try to build a colorful matching of size k. If we fail to do so, then it will enable us to bound the number of color classes. Let $\mathcal{M}$ be an empty set. We repeat the below procedure until the graph is empty.

1. Pick an edge uv of G arbitrarily, and add it to $\mathcal{M}$. Let the edges incident on u have colors $c_u^1, c_u^2, \ldots, c_u^\ell$ and the edges incident on v have colors $c_v^1, c_v^2, \ldots, c_v^p$.
2. Delete all the edges in $\bigcup_{i=1}^\ell E_{c_u^i}$ and $\bigcup_{i=1}^p E_{c_v^i}$. Let the resulting graph be also called G.

If we can continue the above process for k steps (*i.e.* $|\mathcal{M}| \geq k$) then $\mathcal{M}$ is a colorful matching of size at least k. In this case we output $\mathcal{M}$ as the desired colorful matching. To see its correctness, observe that in every iteration we deleted the edges incident on both the endpoints of the added to $\mathcal{M}$. So, the edges we added to $\mathcal{M}$ are indeed pairwise vertex disjoint. Also, note that we delete all the edges with colors that are used on the edges that are incident to the edges that were added to $\mathcal{M}$. Hence, the edges in $\mathcal{M}$ have distinct color.

Otherwise, our procedure ends within at most $k-1$ steps, and so $|\mathcal{M}| \leq k-1$. Now, let us count the number of color classes we delete in each iteration. In other words, we count the number of color classes that are deleted each time we add an edge to M. If all the edges incident on u have distinct colors then $\ell \leq d$ because degree of u is at most d. Similarly, we are argue that $p \leq d$. Together they imply that we delete at most $2d$ color classes in each iteration. Hence, in at most $k-1$ iterations we delete at most $2d(k-1)$ color classes. Following this we are left with an an empty graph. Thus, we have shown that in this case, we can have at most $2d(k-1)$ color classes.   ◄

Lemmas 15 and 16 together prove second part of Theorem 3.

5   Conclusion, Discussion and Open Problems

In this paper, we considered RAINBOW MATCHING from the viewpoint of parameterized complexity, and designed faster parameterized algorithms as well as kernels for this problem. RAINBOW MATCHING is easily seen as a generalization of another well studied problem in parameterized algorithms, namely 3-DIMENSIONAL MATCHING, when we allow the input graph to be a multigraph. In this problem, we are given a set family $(U, \mathcal{F})$, together with a partition $U = \uplus_{i=1}^3 U_i$ and a positive integer k. Here, every set $F \in \mathcal{F}$ has the property that for all $i \in [3]$, $|F \cap U_i| = 1$. The question is whether there exists a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ containing k pairwise-disjoint sets. We first show that RAINBOW MATCHING is indeed a generalization of 3-DIMENSIONAL MATCHING. Towards proving this, we give a polynomial time parameter-preserving reduction from 3−DIMENSIONAL MATCHING to RAINBOW MATCHING. That is, we give the following ppt reduction, 3−DIMENSIONAL MATCHING $\leq_{\text{ppt}}$ RAINBOW MATCHING. Here, let us only present a rough sketch of the proof. The idea of the construction is as follows. In the bipartite graph of the constructed instance of RAINBOW MATCHING, one side represents the elements of U_1, and the other side represents the elements of U_2. Then, for every set $\{u_1, u_2, u_3\}$ in $\mathcal{F}$, where $u_i \in U_i$ for all $i \in [3]$, we add an edge between u_1 and u_2 whose color is u_3. It is easy to see that a solution for the original problem instance can be directly translated to a solution for the new problem instance, and vice versa. Moreover, the parameter k in both instances is set to be the same.

We also saw that there is a ppt reduction from RAINBOW MATCHING to 3-SET PACKING and thus we have the following chain of reductions.

$$3-\text{DIMENSIONAL MATCHING} \leq_{\text{ppt}} \text{RAINBOW MATCHING} \leq_{\text{ppt}} 3-\text{SET PACKING}.$$

It is known that 3-Dimensional Matching admits a randomized algorithm with running time $\mathcal{O}^\star(2^k)$ [2] and a deterministic algorithm with running time $\mathcal{O}^\star(2.5961^{2k}) = \mathcal{O}^\star(6.7398^k)$ [19]. We gave in the introduction a deterministic algorithm for Rainbow Matching that is the same as the one for 3-Set Packing. However, we remark that the algorithm for 3-Dimensional Matching given in [19] can actually be used to solve Rainbow Matching in $\mathcal{O}^\star(6.7398^k)$ time. Can we design a faster randomized or deterministic algorithm for Rainbow Matching or even 3-Dimensional Matching?

We gave an $\mathcal{O}(k^2)$ kernel on paths. Does there exist a linear kernel on paths? Could we get improved kernel for Rainbow Matching on simple family of graphs such as trees, graphs of constant treewidth or planar graphs. Could we show that $\mathcal{O}(k^3)$ size bound on the kernel for Rainbow Matching is optimal?

Finally, by a direct application of our randomized parameterized algorithm for Rainbow Matching running in time $\mathcal{O}^\star(2^k)$, we have that there exists a randomized algorithm for Rainbow Matching running in time $\mathcal{O}^\star(2^{n/2}) = \mathcal{O}^\star(1.4143^n)$. Here, n is the number of vertices in the input graph and the $n/2$ is an upper bound on the maximum size of a colorful matching in a graph. Using a simple dynamic programming algorithm, it is possible to design a $\mathcal{O}^\star(2^n)$ algorithm for Rainbow Matching. Designing a deterministic algorithm for Rainbow Matching running in time $(2 - \epsilon)^n$ for some fixed $\epsilon > 0$ is another interesting open problem.

References

1 F. N. Abu-Khzam. An improved kernelization algorithm for r-set packing. *Information Processing Letters*, 110:621–624, 2010.

2 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Narrow sieves for parameterized paths and packings. *Journal of Computer and System Sciences*, 87:119–139, 2017.

3 M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

4 H. Dell and D. Marx. Kernelization of packing problems. In *SODA'12*, 2012.

5 R. G. Downey and M. R. Fellows. *Fundamentals of parameterized complexity*. Springer, 2013.

6 Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17(3):449–467, 1965.

7 Fedor V. Fomin, Fabrizio Grandoni, and Dieter Kratsch. A measure & conquer approach for the analysis of exact algorithms. *J. ACM*, 56(5):25:1–25:32, 2009.

8 Fedor V. Fomin and Dieter Kratsch. *Exact Exponential Algorithms*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2010.

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

10 J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Computing*, 2:225–231, 1973.

11 Alon Itai, Michael Rodeh, and Steven L. Tanimoto. Some matching problems for bipartite graphs. *J. ACM*, 25(4):517–525, 1978.

12 Mikio Kano and Xueliang Li. Monochromatic and heterochromatic subgraphs in edge-colored graphs-a survey. *Graphs and Combinatorics*, 24(4):237–263, 2008.

13 Van Bang Le and Florian Pfender. Complexity results for rainbow matchings. *Theor. Comput. Sci.*, 524:27–33, 2014.

14 László Lovász and Michael D Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.

15 Silvio Micali and Vijay V. Vazirani. An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs. In *21st Annual Symposium on Foundations of Computer Science, Syracuse, New York, USA, 13-15 October 1980*, pages 17–27, 1980.

16 Herbert J Ryser. Neuere probleme der kombinatorik. *Vorträge über Kombinatorik, Oberwolfach*, pages 69–91, 1967.

17 Larry J. Stockmeyer and Vijay V. Vazirani. Np-completeness of some generalizations of the maximum matching problem. *Inf. Process. Lett.*, 15(1):14–19, 1982.

18 Mihalis Yannakakis and Fanica Gavril. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, 38(3):364–372, 1980.

19 Meirav Zehavi. Mixing color coding-related techniques. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, volume 9294 of *Lecture Notes in Computer Science*, pages 1037–1049, 2015.