Complexity of Computing the Anti-Ramsey Numbers for Paths

Saeed Akhoondian Amiri
University of Cologne, Germany
amiri@informatik.uni-koeln.de

Alexandru Popa
University of Bucharest, Romania
National Institute of Research and Development in Informatics, Bucharest, Romania
alexandru.popa@fmi.unibuc.ro

Mohammad Roghani
Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany
Sharif University of Technology, Teheran, Iran
mohammadroghani43@gmail.com

Golnoosh Shahkarami
MPI for Informatics, Saarland Informatics Campus, Graduate School of Computer Science, Saarbrücken, Germany
gshahkar@mpi-inf.mpg.de

Reza Soltani
Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany
Sharif University of Technology, Teheran, Iran
rsoltani97@gmail.com

Hossein Vahidi
MPI for Informatics, Saarland Informatics Campus, Graduate School of Computer Science, Saarbrücken, Germany
hovahidi@mpi-inf.mpg.de

Abstract

The anti-Ramsey numbers are a fundamental notion in graph theory, introduced in 1978, by Erdös, Simonovits and Sós. For given graphs G and H the *anti-Ramsey number* $\mathrm{ar}(G, H)$ is defined to be the maximum number k such that there exists an assignment of k colors to the edges of G in which every copy of H in G has at least two edges with the same color.

Usually, combinatorists study extremal values of anti-Ramsey numbers for various classes of graphs. There are works on the computational complexity of the problem when H is a star. Along this line of research, we study the complexity of computing the anti-Ramsey number $\mathrm{ar}(G, P_k)$, where P_k is a path of length k. First, we observe that when k is close to n, the problem is hard; hence, the challenging part is the computational complexity of the problem when k is a fixed constant.

We provide a characterization of the problem for paths of constant length. Our first main contribution is to prove that computing $\mathrm{ar}(G, P_k)$ for every integer $k > 2$ is NP-hard. We obtain this by providing several structural properties of such coloring in graphs. We investigate further and show that approximating $\mathrm{ar}(G, P_3)$ to a factor of $n^{-1/2-\epsilon}$ is hard already in 3-partite graphs, unless $P = NP$. We also study the exact complexity of the precolored version and show that there is no subexponential algorithm for the problem unless ETH fails for any fixed constant k.

Given the hardness of approximation and parametrization of the problem, it is natural to study the problem on restricted graph families. Along this line, we first introduce the notion of color connected coloring, and, employing this structural property, we obtain a linear time algorithm to compute $\mathrm{ar}(G, P_k)$, for every integer k, when the host graph, G, is a tree.

2012 ACM Subject Classification Mathematics of computing → Combinatorics; Mathematics of computing → Graph theory; Theory of computation → Computational complexity and cryptography; Theory of computation → Graph algorithms analysis

45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020).
Editors: Javier Esparza and Daniel Kráľ; Article No. 6; pp. 6:1–6:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Keywords and phrases Coloring, Anti-Ramsey, Approximation, NP-hard, Algorithm, ETH

Digital Object Identifier 10.4230/LIPIcs.MFCS.2020.6

Related Version A full version of the paper is available at [3] https://arxiv.org/abs/1810.08004.

1 Introduction

For given graphs G and H, the *anti-Ramsey number* $\mathrm{ar}(G, H)$ is defined to be the maximum number k such that there exists an assignment of k colors to the edges of G in which every copy of H in G has at least two edges with the same color. Classically, the graph G is a large complete graph and the graph H is from a particular graph class.

The study of anti-Ramsey numbers was initiated by Erdös, Simonovits and Sós in 1975 [10]. Since then, there have been a large number of papers on the topic. There are papers that study the case when $G = K_n$ and H is a: cycle, e.g., [10, 21, 5], tree, e.g., [20, 19], clique, e.g., [14, 10, 6], matching, e.g., [22, 8, 17] and others, e.g., [10, 4].

The anti-Ramsey numbers are connected with the rainbow number [15] $rb(G, H)$, which is defined as the minimum number k such that in *any* coloring of the edges of G with k colors, there exists a rainbow copy of H. Thus, $ar(G, H) = rb(G, H) - 1$. We call a coloring without a rainbow copy of H, an *H-free coloring*.

Various combinatorial works studied the case when H is a path or a cycle. For instance, the work of Simonovits and Sos [24] shows that there exists a constant t such that for a sufficiently long path $ar(K_n, P_t) \in O(t \cdot n)$. The combinatorial analysis of the problem is extremely difficult when instead of K_n we use an arbitrary graph as the *host* graph. For a more detailed exposition of the combinatorial results on anti-Ramsey numbers, we refer the reader to the following surveys: [23, 15].

Besides the extremal results, the anti-Ramsey numbers have been studied from the computational point of view in several papers. The anti-Ramsey numbers when G is an arbitrary graph was studied for the case when H is a star. The problem was introduced by Feng et al. [11, 12, 13], motivated by applications in wireless mesh networks and was termed the *maximum edge q-coloring*.

They provide a 2-approximation algorithm for $q = 2$ and a $(1 + \frac{4q-2}{3q^2-5q+2})$-approximation for $q > 2$. They show that the problem is solvable in polynomial time for trees and complete graphs in the case $q = 2$. Later, Adamaszek and Popa [2] show that the problem is APX-hard and present a 5/3-approximation algorithm for graphs with a perfect matching. For more results related to the maximum edge q-coloring, the reader can refer to [1].

To improve our understanding on such problems, we continue the recent line of study of the computational complexity of the problem. Similar to previous works we restrict H to a basic class of graphs, paths. We let G be either an arbitrary graph or a restricted family of graphs such as trees or bipartite graphs. We provide a big picture on what is tractable and what is not tractable when we are dealing with anti-Ramsey numbers on paths. Namely we prove the following.

Our Results

1. First, we show that computing the value of $ar(G, P_k)$ is NP-hard for every $k > 2$ via a reduction from the maximum independent set problem. Namely, we prove the following theorem.

   ▶ **Theorem 1.** *For every $k > 2$, P_k-FREE COLORING PROBLEM is* NP-*hard.*

The above theorem basically states that there is no XP algorithm, parameterized by k, for the problem unless $P = NP$. The reduction is multi stage: firstly we distinguish between the odd and even values of k. Then for each parity of k, given an instance of independent set, we construct an auxiliary graph and prove several structural lemmas on that graph to establish a one to one mapping between the maximum independent set in the original graph and the maximum anti-Ramsey coloring on the auxiliary graph. By a more careful analysis of the above proof for the special case of $k = 3$, we show the problem is inapproximable by a factor $n^{-1/2-\epsilon}$, even on 3-partite graphs, unless $P = NP$. Given the hardness of the problem, it is natural to investigate what would be the best exponential algorithm for the problem. We study the running time of the exact algorithm for a slight variant of the problem, namely, Precolored P_k-free coloring. We prove that the problem does not admit an exact algorithm with running time $2^{o(|E(G)|)}$ assuming ETH.

▶ **Theorem 2.** *There is no $2^{o(|E(G)|)}$ algorithm for Precolored $ar(G, P_k)$, for any fixed k, unless ETH fails.*

To obtain such a reduction, we provide a graph construction with low edge density gadgets. This is unlike standard hardness proofs where it is possible to blow up the graph by any polynomially bounded size.

2. Given the above hardness results, even for small values of k, it is natural to explore the tractability of problem when the host graph has a nice structural property. We first introduce a generic algorithmic idea, of *color connected coloring* and we exploit this to develop a linear time algorithm on trees.

▶ **Theorem 3.** *For a tree T, there is an exact linear time algorithm that computes $ar(T, P_k)$ for every constant integer k; the algorithm runs in time $O(|V(T)|k^4)$.*

Our algorithm is based on dynamic programming on trees, however, unlike most problems in trees, this one is not that straightforward and we employed several techniques to solve the problem. There are known combinatorial results for cycles of length three on outerplanar graphs [16] and the algorithm for trees for 3-consecutive coloring of [7]. Our algorithm is independent of the latter; however, if we set $k = 3$ our algorithm solves the aforementioned problem, while the other direction does not work.

The paper is organized as follows. In Section 2, we introduce preliminaries. Then, we prove the NP-hardness of computing $ar(G, P_k)$ in Section 3 and next, we show the hardness of inapproximability for P_3-free coloring. In Section 4 we show the exact complexity result for Precolored P_3-free coloring. In Section 5, we provide an exact polynomial time algorithm for trees. Finally, in Section 6, we summarize the results and present directions for future work.

2 Preliminaries

We use $\mathbb{N}$ to denote the set of natural numbers and we write $[n]$ to denote the set $\{1, \ldots, n\}$. We refer the reader to [9] for basic notions related to graph theory. All the graphs considered in this paper are simple and undirected.

Let G be a graph, we write $V(G)$ for its vertices and $E(G)$ for its edges. For $k \in \mathbb{N}^+$ we denote by P_k a path with $k+1$ vertices. The length of P_k is k, the number of its edges. Also let p be a P_k, depending on the context we may write $p = (e_1, \ldots, e_k)$ where $e_i \in E(p)$ or $p = (v_1, \ldots, v_{k+1})$ where $v_i \in V(p)$ to describe a path.

▶ **Definition 4** (Coloring). *Given an undirected graph $G = (V, E)$, a coloring of the edges of G is a function $c : E \to \mathbb{N}$. Similarly for any subset $A \subseteq E$ we define $c(A) = \bigcup_{e \in A} c(e)$.*

We call a coloring of the edges of a graph G a *rainbow coloring* if for every pair of edges $e \neq e' \in E$ we have $c(e) \neq c(e')$. Let G, H be two graphs, an edge coloring c of G is *H-free coloring* if there is no rainbow subgraph of G isomorphic to H. We denote the number of distinct colors used in c by $c_{G,H}$. Let $\mathcal{C}$ be the set of all H-free colorings of G. The anti-Ramsey number of G is $ar(G, H) = \max_{c \in \mathcal{C}} c_{G,H}$. We observe that if k is part of the input, then the problem of computing $ar(G, P_k)$ is at least as hard as finding a Hamiltonian path.

▶ **Observation 5.** *Computing $ar(G, P_{|V(G)|-1})$ is NP-hard.*

Proof. $ar(G, P_{|V(G)|-1}) = |E|$ if and only if G does not have a Hamiltonian Path. ◀

In the above we can replace Hamiltonian Path in the proof with longest path and in addition use the length of this path as parameter to prove the hardness for large values of k.

3 Hardness of P_k Anti-Ramsey Coloring

In this section for every $k > 2$, we prove the hardness by a reduction from the maximum independent set (MIS) problem.

Proof Sketch. We construct a new graph G' from a graph G such that from a maximum P_k-free coloring of G', we can derive the size of the maximum independent set of G. To obtain the desired result, we divide the problem into three subproblems. We use the reduction with different approaches for

1. $k = 4$,
2. every even $k > 4$,
3. every odd $k > 1$.

Roughly speaking, we replace every vertex and edge with specific gadgets; this depends on the parity of k. Afterward, in each case, intuitively, we prove that if a vertex belongs to an independent set, its corresponding gadget can be colored with more distinct colors than a vertex that does not belong to an independent set. On the other hand, for each case, we design edge gadgets such that their coloring can be (almost) fixed in advance, despite the choice of colors for the vertex gadgets. The crucial part of the proofs lies in the analysis of a structure of the maximum P_k-free coloring of G' and, exploiting the dependency between vertex gadgets.

In the following, we provide a short version of the proof for odd and even values of k; for detailed proof, we refer the reader to the full version [3]. Besides, by a slight modification to the proof of odd values of k, we obtain an approximation hardness for the case of $k = 3$. Every missing proof is available in the full version [3].

Hardness of the Problem for Odd $k > 1$

Assumption I. In this part we assume $k > 1$ is an odd integer.

In the following, we first present an upper bound on the number of colors when the graph H is a path. For certain technical reasons that we will see in the proofs, we define a constant c_k depending only on k with a particular lower bound.

▶ **Lemma 6.** *$ar(G, P_k) \leq c_k|V(G)|$ for some $c_k \in \Theta(k\sqrt{\log k})$ and $c_k > 3k\sqrt{\log k}$.*

Assumption II. In this section, c_k is what we used in Lemma 6. Whenever we write I it means the maximum independent set in the graph G.

Given an undirected graph G, we construct a graph G' as follows:

1. For each $v \in V(G)$ we introduce two new vertices $s_v, t_v \in V(G')$ and $(f_k+1)c_k|V(G)|$ internally disjoint paths of length $k-1$, $\mathcal{P}^v = \{P_1^v, \dots, P^v_{(f_k+1)c_k|V(G)|}\}$, connecting s_v to t_v. Later in Lemma 12 we determine the value of f_k.
2. For each edge $\{v,u\} \in E(G)$, add 4 new edges in $E(G')$: $\{s_v,t_u\}$, $\{t_v,s_u\}$, $\{t_v,t_u\}$, $\{s_v,s_u\}$. Let us define the union of all such edges in the entire graph G' as E_t^s, more formally $E_t^s = \bigcup_{\{u,v\}\in E(G)} \{\{s_v,t_u\},\{t_v,s_u\},\{t_v,t_u\},\{s_v,s_u\}\}$.

An edge coloring is *valid* if it is a P_k-free coloring. We start by providing some lemmas and observations on the structure of valid colorings of G' to establish a connection between such a coloring and an independent set in G.

▶ **Lemma 7.** *In any P_k-free coloring of G' the edges in E_t^s will receive at most $2c_k|V(G)|$ distinct colors.*

The next lemma bounds the number of distinct colors of each individual $\mathcal{P}^v$.

▶ **Lemma 8.** *If G is a cycle of length $2(k-1)$ then $ar(G,P_k) = 2(k-2)$.*

▶ **Lemma 9.** *Let H be a graph isomorphic to $\mathcal{P}^v$ for any $v \in V(G)$. Then there is a valid coloring of H with $(k-2)\cdot(f_k+1)c_k|V(G)|$ distinct colors.*

▶ **Lemma 10.** *There is no valid coloring of G' with more than $(k-2)\cdot(f_k+1)c_k|V(G)|$ distinct colors in one $\mathcal{P}^v$ for $v \in V(G), |V(G)| \geq 2$.*

▶ **Definition 11** (Family of Distinct Colored Paths)**.** *A set of paths $\mathcal{P}$ is a family of distinct colored paths if the following conditions hold:*

1. *Their union is a graph with a valid P_k-free coloring.*
2. *For every $P \neq Q \in \mathcal{P}$ and, for every $e \in P, e' \in Q$ we have that $c(e) \neq c(e')$.*

Note that from the above Definition 11, it is clear that the set of paths should be pairwise edge disjoint (otherwise it does not meet the second condition), also one path may repeat some of its own colors.

The following lemma, basically states that we cannot have two adjacent nodes u, v in G such that their corresponding paths receive many distinct colors in G'. We employ this key property later in the hardness proof to obtain an MIS based on the size of the family of distinct colored paths.

▶ **Lemma 12.** *Let $\{v,u\} \in E(G)$, then there is a constant f_k (this is what we used to construct G'), depending only on k, such that, in any valid coloring of G' if there are families of distinct colored paths $\mathcal{P} \subseteq \mathcal{P}^v, \mathcal{Q} \subseteq \mathcal{P}^u$, such that each $P \in \mathcal{P} \cup \mathcal{Q}$ is colored with at least $k-2$ distinct colors, then $\min\{|\mathcal{P}|, |\mathcal{Q}|\} < f_k$.*

For a better understanding of the above lemma see Figure 1. The following establishes a lower bound on the number of distinct colors w.r.t. the size of a maximum independent set I.

▶ **Lemma 13.** $ar(G',P_k) > |I|(k-2)(f_k+1)c_k|V(G)| + (|V(G)|-|I|)(k-3)(f_k+1)c_k|V(G)|$

Now we can prove the hardness for every odd $k > 1$. You can find the complete proof in [3], however the idea is to use Lemma 12: given a coloring, we cannot have many blow up vertices that are colored with many colors by Lemma 12. Basically such vertices form an independent set in the original graph, the second part employs Lemma 13 at its heart. We have to be careful in our counting arguments. We suggest the reader see several important details in the full proof in [3].

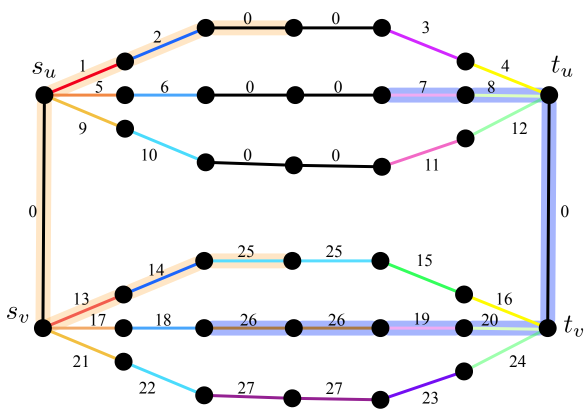

Figure 1 The coloring scheme of vertex gadgets for P_7-free coloring. Colors are represented by numbers. To simplify the visualization, some connector edges and some parallel paths are not drawn. For $v \in I$ each path gets $k-2=5$ colors and for $u \in V \setminus I$ each path gets $k-3=4$ colors. Two paths of length 7 are highlighted, neither of them are rainbow.

▶ **Lemma 14.** *For every odd $k > 1$, P_k-FREE COLORING PROBLEM is* NP*-hard.*

With a slight twist we get the following, for its proof please see [3].

▶ **Theorem 15.** *Unless $P = NP$, for any fixed $\delta > 0$, there is no polynomial time $\frac{1}{\sqrt{|V(G)|}^{1-\delta}}$-approximation for P_3-free coloring even in 3-partite graphs.*

Hardness of the Problem for Even Values of $k > 2$

Assumption: In this part we assume $k = 2t, t > 2$.

▶ **Definition 16** ($S(d)$). *For an integer $d \geq 1$, let $S(d)$ be a subdivided star, i.e., $S(d)$ is obtained by subdividing every edge of $K_{1,d}$. We call the corresponding vertex of $K_{1,d}$ in the partition with size one, as the center of $S(d)$. Every subdivided edge of $K_{1,d}$ is a branch. Therefore, $S(d)$ has exactly d branches.*

▶ **Definition 17** (wasted edge). *In a coloring of G, we choose one arbitrary edge from each color and call each unchosen edge of G a wasted edge.*

Therefore, if D is a set of all wasted edges of a maximum H-free coloring of G, then $|D| + ar(G, H) = |E(G)|$.

▶ **Definition 18** ($D_{l,w}$). *We construct an edge gadget $D_{l,w}$ as follows. Let $u_1, u_2, ..., u_{l+1}$ be $l+1$ distinct vertices. Then for every $i \in [l]$, we connect u_i to u_{i+1} by w internally disjoint paths each of length two.*

We call u_1 *head* and u_{l+1} *tail* of $D_{l,w}$.
Graph Construction Given a graph G, we construct a graph G' as follows.

1. For each vertex $v \in V(G)$ with degree d_v, we add one $S(d_v)$, named S_v, to G'. Each branch of S_v corresponds to one of the incident edges of v.
2. For every edge $e = \{u, v\} \in E(G)$, we add a $D_{t-2,4|E(G)|+8}$ to G', named D_e, such that its head is the leaf of the corresponding branch of e in S_u and its tail is the leaf of the corresponding branch of e in S_v.

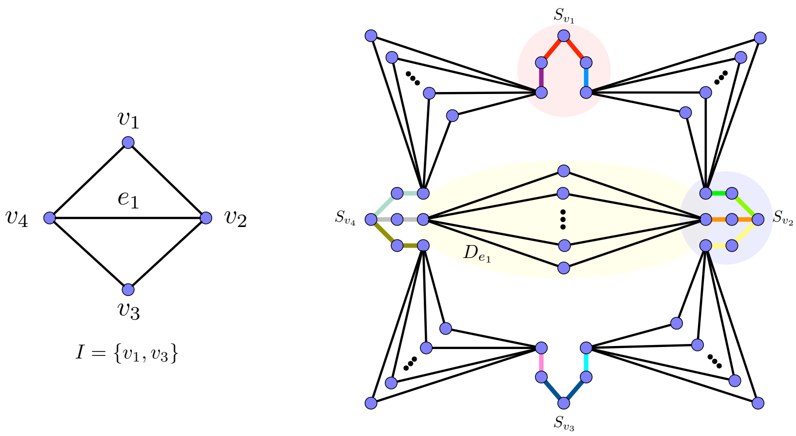

Figure 2 Illustration of graph construction for $k = 6$. The left figure shows the graph G, and the right figure shows its corresponding G'. All black edges of G' have some unique new color. The coloring is the maximum P_6-free coloring of G'.

For a better understanding of the graph construction see Figure 2.

▶ **Lemma 19.** *In any maximum P_k-free coloring c of G', for every D_e, $e \in E(G)$, there exist at least eight edge disjoint paths, each of length $2t - 4$ between its head and tail such that their union is rainbow.*

▶ **Lemma 20.** *In any maximum P_k-free coloring of G', in each S_v for $v \in V(G)$, there are at least $d_v - 1$ wasted edges.*

▶ **Lemma 21.** *In any maximum P_k-free coloring of G', for any $v \in V(G)$ if S_v has $d_v - 1$ wasted edges, then its coloring has the following properties: 1) all incident edges of the center vertex of S_v have the same color and 2) each remaining edge of S_v has a distinct color.*

▶ **Lemma 22.** *Let $u, v \in V(G)$ and $e = \{u, v\} \in E(G)$. In any maximum P_k-free coloring of G', S_v has at least d_v wasted edges or S_u has at least d_u wasted edges.*

▶ **Lemma 23.** *Let I be a maximum independent set of G and let D be the set of all wasted edges in a maximum P_k-free coloring of G', then $|I| = 2|E(G)| - |D|$.*

Proof. We provide a coloring c as follows. For every $v \in I$, color S_v with $d_v - 1$ wasted edges as explained in the Lemma 21. For every $u \in V(G) \setminus I$, for each branch b of S_u, we color both of its edges with a new color, $c_{v_{b_e}}$. For every $e \in E(G)$, we color D_e as a rainbow with new distinct colors. See Figure 2 for a better understanding of the coloring c.

First, we claim that c is a maximum P_k-free coloring of G' and then we show that $|I|$ can be derived from the size of c, or equivalently from $ar(G', P_k)$.

To show that c is a P_k-free coloring we perform a case distinction for every path of length k in G', in the following u, v are two arbitrary adjacent vertices in the graph G:

1. A path P between the center of S_u to the center of S_v for $\{u, v\} \in E(G)$.
2. A path P that contains center of S_v as one of its non-leaf vertices.

For the first case, as $e = \{u, v\}$ by Lemma 22 w.l.o.g. we can suppose S_u has been colored with at least d_u wasted edges. Therefore, the first two edges of P starting from the center of S_u belong to a branch b of S_u, have the same color c_{u_b} in c, so P is not a rainbow path.

For the second case, the path P has at least one branch, b, of S_v and at least one incident edge to the center of S_v in another branch b' of S_v. Hence, if we colored S_v with $d_v - 1$ wasted edges, then by Lemma 21 two edges of P that are incident to the center of S_v have the same color. Otherwise, if S_v is colored with d_v wasted edges, both edges of b have the same color c_{v_b}, therefore P is not a rainbow path.

Now we show that c is a maximum P_k-free coloring of G'. Note that by Lemma 20, the minimum number of wasted edges in an individual S_v for $v \in V(G)$ is at least $d_v - 1$. Observe that by Lemma 22, number of S_v's for $v \in V(G)$ with $d_v - 1$ wasted edges is at most $|I|$. Moreover, in c, number of such S_v's is exactly $|I|$ which is the maximum possible number of them. Also, for each remaining vertex, $v \in V(G)$, S_v has exactly d_v wasted edges (the minimum number of possible wasted edges other than $d_v - 1$). Also, c does not have any wasted edge in the rest of G'. Therefore, c has the least number of wasted edges. Hence, c has the maximum number of distinct colors in any P_k-free coloring of G'.

Total number of wasted edges in c is $|D| = \sum_{v \in I}(d_v - 1) + \sum_{v \notin I} d_v$. Hence, we get that $|I| = 2|E(G)| - |D|$ as claimed. ◀

Hence, we get the following.

▶ **Lemma 24.** *For every even $k > 4$, P_k-FREE COLORING PROBLEM is* NP*-hard.*

▶ **Lemma 25.** *For $k = 4$, P_k-FREE COLORING PROBLEM is* NP*-hard.*

Proof of Theorem 1. By Lemma 24, Lemma 25, and Lemma 14 we show that for every integer $k > 2$ the problem is hard. ◀

4 Precoloring $ar(G, P_k)$ Has No Subexponential Algorithm for all $k > 2$

In this section, we study the complexity of exact algorithms computing the anti-Ramsey number $ar(G, P_k)$ where P_k is a path with k edges. We now consider a variant of the problem for the exact time complexity of the problem.

▶ **Problem 26** (Precolored $ar(G, H)$)**.** *The input consists of a graph $G = (V, E)$ where $E = E_1 \cup E_2$. The edges in E_1 have assigned a color while the edges in E_2 are uncolored. Color the edges in E_2 with as many new colors as possible such that there is no rainbow copy of H in G.*

For this problem, we provide a *fine grained reduction* from 3SAT to show the hardness of the problem. That is, we provide an instance of Precolored $ar(G, P_k)$ problem (for a constant $k > 2$). Due to the page limits, you can see the entire proof and gadget constructions in [3] and here we just show an example of a clause gadget and explain the main idea behind the proof by this example.

The example clause is actually $(x \vee y \vee \bar{z})$. The bottom edges in the Figure 3 are actually literal gadgets (a single edge), so each variable gets exactly one color (all other clauses are connected to it, we did not draw all of them). Later we will see a color of a literal gadget is either T_{k-2} or F_{k-2} which later determines value of the variable in the SAT formula. The construction of the clause gadget is such that among all uncolored edges, only the edges $\{v_{k-1}, x_1\}, \{v_{k-1}, y_1\}, \{u_{k-1}, z_1\}$ are able to get a new color (a color that is not in the set of predefined colors). It is possible to show that these 3 edges together can afford only one new color for the corresponding clause gadget. This new color enforces the coloring of other uncolored edges, and in particular determines whether the corresponding literal gadget will get the color T_{k-2} or F_{k-2}. Hence, from a coloring that assigns one new color per clause, we can find the satisfying assignment and vice versa.

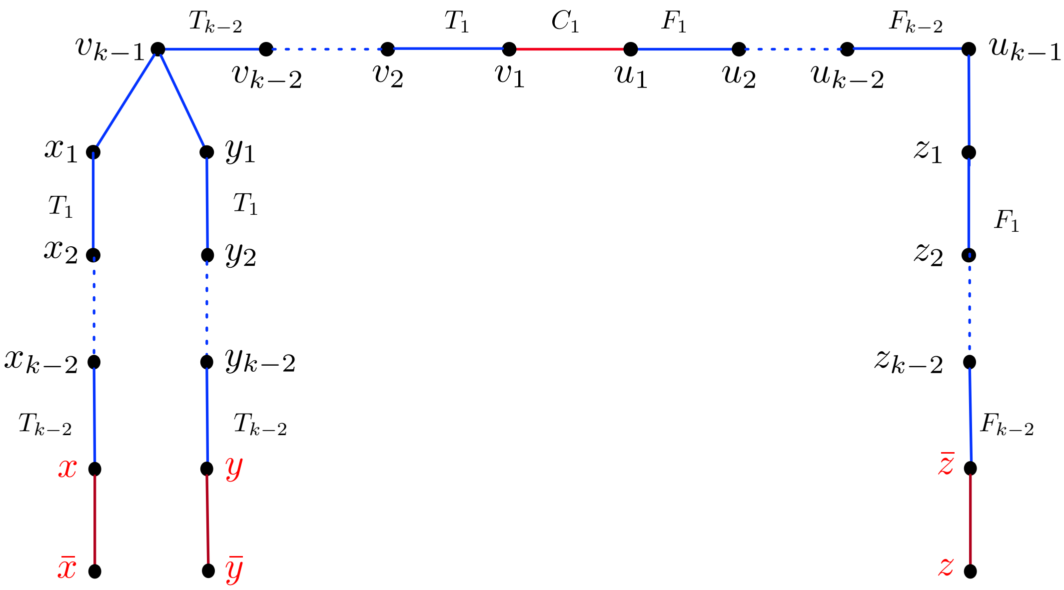

Figure 3 Sketch of a gadget where the clause has exactly one negative literal. Edge $\{u_1, v_1\}$ is forced to get the color of one of its neighbors. The gadget is sparse and has $5k - 7$ edges.

The above line of analysis shows the problem is NP-hard in this graph. However our gadgets are light weight: each of them has $O(k)$ edges, hence the constructed graph is sparse. Given the sparsification lemma [18] and the fact that our constructed graph has linear size w.r.t. the size of the 3-SAT instance, we conclude that there is no $2^{o(|E(G)|)}$ time algorithm for Precolored $ar(G, P_k)$ assuming ETH. Hence, we can prove the following theorem. See [3] for a complete proof.

▶ **Theorem 2.** *There is no $2^{o(|E(G)|)}$ algorithm for Precolored $ar(G, P_k)$, for any fixed k, unless ETH fails.*

5 Color Connected Coloring and its Applications

In this section, we introduce the notion of color connected coloring and using that we provide a polynomial time algorithm to compute $ar(T, P_k)$, where T is a tree. Roughly speaking, in a color connected coloring we try to color the graph with the maximum number of colors so that the set of edges of every color class induces a connected subgraph. The main result of this section is the following theorem.

▶ **Theorem 3.** *For a tree T, there is an exact linear time algorithm that computes $ar(T, P_k)$ for every constant integer k; the algorithm runs in time $O(|V(T)|k^4)$.*

Let c be a P_k-free coloring of a graph G and let c_1 be one of such colors used in c. Then, we call the induced graph $G[\{v \mid \exists u \in V(G), e = \{u, v\} \in E(G), c(e) = c_1\}]$ as an induced c_1-graph and we write it $G[c_1]$. If $G[c_1]$ is connected then we say c_1 is a *connected color*; otherwise, it is a *disjoint color*.

▶ **Definition 27** (Color Connected Coloring)**.** *Given a graph G, a P_k-free coloring c of G is a color connected coloring if for every color c_i used in c, $G[c_i]$ is a connected component.*

In the rest of this section, we assume that T is a rooted tree with r_T as its root. We define T_v as the largest subtree with $v \in V(T)$ as its root. Depth of a vertex $v \in V(T)$, H_v, is the number of edges between v and the root. Furthermore, we define $C(v)$ as the set of children of v in a rooted tree. As we can color the graph with at most $|E|$ many colors, in this proof we use a palette of colors $\mathcal{C} = \{c_e \mid e \in E(T)\}$. That is whenever we color an edge e with a new color, its color will be c_e, otherwise, e will get a color of one of the already colored edges.

▶ **Lemma 28.** *There exists a maximum P_k-free coloring of T, which is color connected.*

Proof. Let c be a maximum P_k-free coloring of T with the minimum number of color connected components. If for every c_i, $T[c_i]$ has one connected component we are done. Otherwise, towards the contradiction, let c_1 be a color used in c, for which $T[c_1]$ has more than one connected components, $\{T_1, \dots, T_r\}$ for some $r > 1$. W.l.o.g. suppose T_1 is the component of $T[c_1]$ with the deepest root, in other words $\text{argmax}_{i\in[r]} \min_{u\in V(T_i)} H_u$ equals to one. Since $r > 1$, the root of subtree T_1, v, has a parent. Let e be the edge between v and its parent. We recolor all of $E(T_1)$ with color $c(e)$. This clearly creates a new coloring c' with the same set of colors as c; however, it has one less color connected component than c which contradicts our minimality assumption on c. To complete the contradiction, it is sufficient to show that c' is a P_k-free coloring.

Towards the contradiction, let P be a rainbow P_k in c'. We perform a case distinction on $|E(P) \cap E(T_1)|$ to derive a contradiction.

1. $|E(P) \cap E(T_1)| = 0$: In this case, the coloring of P in c and c' is identical. Moreover, P is not rainbow in c, hence P is not rainbow in c' either, a contradiction.
2. $|E(P) \cap E(T_1)| = 1$: In this case, let $e' \in E(P) \cap E(T_1)$ be the only edge of P that is recolored in c'. There must exist another edge e'' of P which is colored by c_1. We know that $e'' \notin E(T_1)$, so e'' is not incident to v. We claim that $e'' \notin E(T_v)$. Suppose by contradiction, $e'' \in E(T_v)$. Since $e'' \notin E(T_1)$, w.l.o.g. assume $e'' \in E(T_2)$. Since T_1 and T_2 are two disjoint connected components in T_v and $v \in V(T_1)$, $\min_{u\in V(T_1)} H_u < \min_{u\in V(T_2)} H_u$ which contradicts the fact that T_1 is the component of $T[c_1]$ with deepest root. We showed that $e'' \notin E(T_v)$. Since $|E(P) \cap E(T_1)| = 1$, its obvious that $e \in E(P)$. $c'(e) = c'(e')$, a contradiction.
3. $|E(P) \cap E(T_1)| > 1$: In this case, at least two edges of P have the same color $c(e)$, hence P is not rainbow, a contradiction. ◀

The purpose of our algorithm is to find a maximum P_k-free color connected coloring of a tree, T, since by Lemma 28 it is a maximum P_k-free coloring of T.

▶ **Definition 29** (L_1^v, L_2^v)**.** *For a color connected coloring c of T, we define L_1^v to be a longest rainbow path in T_v starting from v. Moreover, let L_2^v be the longest rainbow path such that L_1^v and L_2^v are edge disjoint and $L_1^v \cup L_2^v$ is also rainbow.*

▶ **Lemma 30.** *A color connected coloring c of T is P_k-free if and only if $|E(L_1^v)|+|E(L_2^v)| < k$, for all $v \in V(T)$.*

Proof. If there exist $v \in V(T)$ such that $|E(L_1^v)| + |E(L_2^v)| \geq k$, c is not a P_k-free coloring, since $L_1^v \cup L_2^v$ is a rainbow path.

To prove the other direction of the lemma, first we need to prove the following claim.

▷ Claim 30.1. For any $v \in V(T)$, $L_1^v \cup L_2^v$ is a maximum length rainbow path including v in T_v.

Proof of Claim 30.1. We prove the claim by contradiction, suppose there is a rainbow path which can be partitioned as $L_3 \cup L_4$, each starting from v, such that $|E(L_3)| + |E(L_4)| > |E(L_1^v)|+|E(L_2^v)|$. Since L_1^v is a longest rainbow path we have that $|E(L_3)|, |E(L_4)| > |E(L_2^v)|$. Hence, L_3 and L_4 must have a common color with L_1^v. We know that the incident edge of v in each path L_1^v, L_3, L_4 must have the same color, since c is a color connected coloring. But we assumed that $L_3 \cup L_4$ is rainbow, a contradiction. Hence, the claim is proved. ◁

Now we can prove the remaining direction of the lemma. Suppose P is a rainbow path in T_v. Thus, P can be partitioned as $P_1 \cup P_2$, each starting from $u \in V(T_v)$. Note that $|E(L_1^u)| + |E(L_2^u)| < k$ by the lemma statement. Also, by the above claim, we know $|E(P)| \leq |E(L_1^u)|+|E(L_2^u)|$. Therefore, $|E(P)| < k$ for any arbitrary rainbow path in T_v. ◀

▶ **Definition 31** ($D(v, i, j)$). *Let $i \geq j$, $i + j < k$, and $v \in V(T)$, we define $D(v, i, j)$ to be the number of distinct colors in a color connected maximum P_k-free coloring of T_v such that $|E(L_1^v)| = i$ and $|E(L_2^v)| = j$.*

For $e = \{u, v\}$ where v is the parent of u, we define T_e to be a subgraph of T_v with $E(T_u) \cup e$ as its edge set, that is a subgraph of T_v that is hanging from e.

Proof of Theorem 3. By Definition 31, we know that $ar(T, P_k) = \max\{D(r_T, i, j)|i+j < k\}$. We show that $D(v, i, j)$ can be computed using the values of $D(u, \cdot)$ for $u \in V(T_v) \setminus \{v\}$. Hence, $D(\cdot)$ can be computed by a post-order traversal of T.

To compute $D(v, i, j)$, if v is a leaf of T, the only valid case is $D(v, 0, 0)$, since there is no edge in T_v. Hence, in the remaining, we suppose that v is not a leaf. We proceed by case distinction based on types of children of v. A child u of v is of the following types:

1. $u \in L_1^v$,
2. $u \in L_2^v$,
3. $u \notin L_1^v \cup L_2^v$

Now for each child u of v and $z \in [3]$, such that $e = \{v, u\} \in E(T)$, we define $A_{u,z}$ as the maximum number of distinct colors in T_e if u belongs to case z, such that it does not violate the definition of $D(v, i, j)$. Note that only one child of v belongs to the first case. Also, for $j > 0$, there is only one child of v in the second case. Moreover, for $j = 0$ there is not any child in the second case. All other children of v belong to the third case. Therefore, we can compute $D(v, i, j)$ by Equation (1) and Equation (2), for $j > 0$ and $j = 0$, respectively.

$$D(v, i, j) = \max\{A_{u_1,1} + A_{u_2,2} + \sum_{u \in C(v) \setminus \{u_1,u_2\}} A_{u,3} | u_1, u_2 \in C(v), u_1 \neq u_2\}, \quad (1)$$

$$D(v, i, 0) = \max\{A_{u_1,1} + \sum_{u \in C(v) \setminus \{u_1\}} A_{u,3} | u_1 \in C(v)\}. \quad (2)$$

In what follows, we show how to compute the value of $A_{u,z}$.

a) $\mathbf{u \in L_1^v}$**.** Let $e = \{u, v\} \in E(T)$ and $u \in L_1^v$. Then we have that $E(L_1^v) \setminus \{\{v, u\}\}$ is a rainbow path of length $i - 1$. Observe that, since $c(e)$ is in at most one of $c(E(L_1^u))$ or $c(E(L_2^u))$, hence by appending e to their tails, at least one of the two paths, L_1^u or L_2^u, extends to a longer rainbow path. If L_1^u extends to a longer rainbow path, we have $|E(L_1^u)| = i - 1$. Otherwise, $c(e) \in c(E(L_u^1))$ and by Definition 29 every rainbow path with greater length than L_u^2 starting from u in T_u has a common color with L_1^u. Moreover the common color is $c(e)$, since the coloring is color connected. Hence, L_2^u is the longest rainbow path in T_u that extends to a longer rainbow path which results in $|E(L_2^u)| = i - 1$. Therefore, $|E(L_1^u)| = i - 1$ or $|E(L_2^u)| = i - 1$. Thus, $A_{u,1}$ equals to the maximum value obtained from these two cases.

1. $\boldsymbol{|E(L_1^u)| = i-1}$: In this case, e can get a new color c_e. Hence, the maximum number of distinct colors used in T_e for $D(v,i,j)$ is $\max_{x<i} D(u,i-1,x)+1$.
2. $\boldsymbol{|E(L_2^u)| = i-1, |E(L_1^u)| > i-1}$ **:** Then $c(\{v,u\}) \in c(E(L_1^u))$, since the length of the longest rainbow path must not exceed i. Also, e must have the same color as the incident edge of u in L_1^u, since the coloring is color connected. However, in this case, $P := L_2^u \cup e$ forms a rainbow path, since $c(e) \in c(E(L_1^u))$ and $|c(E(L_1^u)) \cap c(E(L_2^u))| = 0$. Moreover, P is the longest rainbow path of T_v starting with e, since every other path with longer length has a common color with L_1^u and we are looking for a color connected coloring, thus this color is $c(e)$. So the maximum number of distinct colors used in T_e for $D(v,i,j)$ in this case is $\max_{x \geq i} D(u,x,i-1)$.

b) $\mathbf{u \in L_2^v}$**.** $A_{u,2}$ can be computed similar to the previous case.

c) $\mathbf{u \notin L_1^v \cup L_2^v}$**.** In the following let $e_1 = \{v,u_1\} \in L_1^v$ and $e_2 = \{v,u_2\} \in L_2^v$. For every child u of v such that $u \notin \{u_1,u_2\}$, suppose that $x = |E(L_1^u)|, y = |E(L_2^u)|$. Also, let $e = \{u,v\}$. Hence, $A_{u,3}$ is equal to the maximum value obtained from the following cases by iterating over all combination of x and y such that $x+y<k$ and $x \geq y$.

1. $x < j$: In this case, e can get a new color c_e. Therefore, the optimal solution for this case of T_e is $D(u,x,y)+1$.
2. $j \leq x < i$: In this case, e can not get the new color c_e. For the contradiction, suppose that e has the new color c_e. Therefore, L_1^u will extend to a longer rainbow path with length $x+1$ which starts from v. Moreover, we are looking for color connected coloring, thus the extended path has not any common color with L_1^v. Since $x+1>j$, it leads to a contradiction to the assumption that L_2^v is the longest path such that $L_1^v \cup L_2^v$ is rainbow. Thus, e cannot have a new color c_e. Hence, the optimal solution for this case of T_e is at most $D(u,x,y)$. Let $c(e) = c(e_1)$, then any rainbow path starting from e in T_e has length less than or equal to L_1^v and has a common color with L_1^v. Therefore, the optimal solution for this case of T_e is exactly $D(u,x,y)$.
3. $i \leq x$ and $y < j$: In this case, $c(e) \in c(E(L_1^u))$, otherwise the concatenation of e and L_1^u creates a rainbow path of length $x+1$ which is larger than length of L_1^v. Hence, e must have the same color as the first edge of the path L_1^u starting from u, since the coloring is color connected. Therefore, the optimal solution for this case of T_e is $D(u,x,y)$.
4. $i \leq x$ and $j \leq y < i$: In this case, $c(e) \in c(E(L_1^u))$, otherwise the concatenation of e and L_1^u creates a rainbow path longer than L_1^v, a contradiction. Let suppose e_3 be the first edge of the path L_1^u which is incident to u. Hence, e must have the same color as e_3, since we are looking for a color connected coloring. In addition, e must have the same color as e_1, otherwise, L_2^u extends to a rainbow path of length $y+1$ which is longer that L_2^v and it does not have any common color with L_1^v, a contradiction. Hence, e, e_1, and e_3 must have the same color. We have counted the color of e_1 as a distinct color before. On the other hand, we count the color of e_3 in the calculation of $D(u,x,y)$. Therefore, we have to subtract it by one to avoid duplication. Hence, the optimal solution for this case is at most $D(u,x,y)-1$. Consider the coloring of T_u that results $D(u,x,y)$ distinct colors. Let us recolor all edges in $T_u[c(e_3)]$ by $c(e_1)$. Also, let $c(e) = c(e_1)$. Length of the longest rainbow path starting from v in T_e in the proposed coloring is $y+1$ which is not more than i. Furthermore, all rainbow paths starting from v in T_e have a common color with L_1^v, hence they do not violate the definition of L_2^v. Therefore, the optimal solution for this case of T_e is exactly $D(u,x,y)-1$.

5. $i \le x$ and $i \le y$: In this case, as $i < y + 1$, at least one of the $L_1^u \cup \{e\}$ or $L_2^u \cup \{e\}$ is a longer rainbow path than L_1^v, a contradiction to the choice of L_1^v. Therefore, this case is not possible and does not take part in the calculation of the value of the $D(v, i, j)$.

Notice that we only defined $D(v, i, j)$ for $i + j < k$. Hence, by Lemma 30, our coloring for every $D(v, i, j)$ is P_k-free color connected coloring.

▷ Claim 32. Let A, B, C be three arrays of length n. There is an $O(n)$ algorithm for finding $\max\{A_s + B_t + \sum_{r \in [n] \setminus \{s,t\}} C_r | s \neq t, \{s, t\} \subseteq [n]\}$.

According to the previous cases, we can compute $A_{u,z}$ for all $z \in [3]$ and $u \in C(v)$ in $O(k^2)$. Moreover, by Equation (1), Equation (2), and the above claim we can compute $D(v, i, j)$ in $O(deg(v))$, if we use dynamic programming approach. Therefore, the total time complexity of our algorithm is $O(|V(T)|k^4)$, since there are $O(|V(T)|k^2)$ values of $D(\cdot)$ that we need to compute. ◀

6 Conclusions and Open Problems

We studied the complexity of computing the anti-Ramsey number for simple paths. We proved that computing the $\mathrm{ar}(G, P_k)$ is hard for every constant integer $k > 2$, and for $k = 3$, the problem is hard to approximate to a factor of $n^{-1/2-\epsilon}$. To analyze the exact complexity of the problem, we provided a fine grain reduction, for a slight variation of it. It remains unanswered whether the inapproximability result extends to all paths of length at least 3.

On the positive side, we provided a linear time algorithm for trees. Color connected coloring does not apply to bounded treewidth graphs. However, we believe our techniques can be extended to provide an approximation algorithm for these graphs. We covered paths in depth, another natural class of graphs to be considered might be complete graphs or cycles.

References

1. A. Adamaszek and A. Popa. Approximation and hardness results for the maximum edge q-coloring problem. *J. Discrete Algorithms*, 38-41:1–8, 2016.
2. Anna Adamaszek and Alexandru Popa. Approximation and hardness results for the maximum edge q-coloring problem. In *ISAAC (2)*, pages 132–143, 2010.
3. Saeed Akhoondian Amiri, Alexandru Popa, Mohammad Roghani, Golnoosh Shahkarami, Reza Soltani, and Hossein Vahidi. Complexity of computing the anti-ramsey numbers for paths, 2018. `arXiv:1810.08004`.
4. Maria Axenovich and Tao Jiang. Anti-Ramsey numbers for small complete bipartite graphs. *Ars Comb.*, 73, 2004.
5. Maria Axenovich, Tao Jiang, and André Kündgen. Bipartite anti-Ramsey numbers of cycles. *Journal of Graph Theory*, 47(1):9–28, 2004.
6. Aart Blokhuis, Ralph J. Faudree, András Gyárfás, and Miklós Ruszinkó. Anti-Ramsey colorings in several rounds. *J. Comb. Theory, Ser. B*, 82(1):1–18, 2001. `doi:10.1006/jctb.2000.2016`.
7. Csilla Bujtás, E. Sampathkumar, Zsolt Tuza, Charles Dominic, and L. Pushpalatha. 3-consecutive edge coloring of a graph. *Discrete Mathematics*, 312(3):561–573, 2012. `doi:10.1016/j.disc.2011.04.006`.
8. He Chen, Xueliang Li, and Jianhua Tu. Complete solution for the rainbow numbers of matchings. *Discrete Mathematics*, 309(10):3370–3380, 2009. `doi:10.1016/j.disc.2008.10.002`.
9. Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

10 P. Erdös, M. Simonovits, and V. T. Sós. Anti-Ramsey theorems. *Infinite and finite sets (Colloq., Keszthely, 1973; dedicated to P. Erdös on his 60th birthday)*, pages 633–643, 1975.

11 W. Feng, L. Zhang, W. Qu, and H. Wang. Approximation algorithms for maximum edge coloring problem. In *TAMC*, pages 646–658, 2007.

12 Wangsen Feng, Ping Chen, and Bei Zhang. Approximate maximum edge coloring within factor 2: a further analysis. In *ISORA*, pages 182–189, 2008.

13 Wangsen Feng, Li'ang Zhang, and Hanpin Wang. Approximation algorithm for maximum edge coloring. *Theoretical Computer Science*, 410(11):1022–1029, 2009.

14 Alan Frieze and Bruce Reed. Polychromatic hamilton cycles. *Discrete Mathematics*, 118(1):69–74, 1993.

15 Shinya Fujita, Colton Magnant, and Kenta Ozeki. Rainbow generalizations of Ramsey theory: A dynamic survey. *Theory and Applications of Graphs*, 0, 2014.

16 Wayne Goddard and Honghai Xu. Vertex colorings without rainbow subgraphs. *Discussiones Mathematicae Graph Theory*, 36(4):989–1005, 2016. `doi:10.7151/dmgt.1896`.

17 Ruth Haas and Michael Young. The anti-Ramsey number of perfect matching. *Discrete Mathematics*, 312(5):933–937, 2012. `doi:10.1016/j.disc.2011.10.017`.

18 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. `doi:10.1006/jcss.2001.1774`.

19 Tao Jiang. Edge-colorings with no large polychromatic stars. *Graphs and Combinatorics*, 18(2):303–308, May 2002.

20 Tao Jiang and Douglas B. West. Edge-colorings of complete graphs that avoid polychromatic trees. *Discrete Mathematics*, 274(1-3):137–145, 2004. `doi:10.1016/j.disc.2003.09.002`.

21 J.J. Montellano-Ballesteros and V. Neumann-Lara. An anti-Ramsey theorem on cycles. *Graphs and Combinatorics*, 21(3):343–354, September 2005.

22 Ingo Schiermeyer. Rainbow numbers for matchings and complete graphs. *Discrete Mathematics*, 286(1-2):157–162, 2004.

23 Ingo Schiermeyer. Rainbow colourings. Invited papers from RIMS, Kyoto University, 2007.

24 Miklós Simonovits and Vera T. Sós. On restricted colourings of K_n. *Combinatorica*, 4(1):101–110, March 1984.