

Restricted online Ramsey numbers of matchings and trees

Joseph Briggs

Department of Mathematical Sciences
Technion Israel Institute of Technology
Haifa, Israel

briggs@campus.technion.ac.il

Christopher Cox*

Department of Mathematical Sciences
Carnegie Mellon University
Pittsburgh, PA, U.S.A.

cocox@andrew.cmu.edu

Submitted: Apr 9, 2019; Accepted: Aug 16, 2020; Published: Sep 4, 2020

© The authors. Released under the CC BY-NC-SA license (International 4.0).

Abstract

Consider a two-player game between players Builder and Painter. Painter begins the game by picking a coloring of the edges of K_n , which is hidden from Builder. In each round, Builder points to an edge and Painter reveals its color. Builder's goal is to locate a particular monochromatic structure in Painter's coloring by revealing the color of as few edges as possible. The fewest number of turns required for Builder to win this game is known as the restricted online Ramsey number. In this paper, we consider the situation where this "particular monochromatic structure" is a large matching or a large tree. We show that in any t -coloring of $E(K_n)$, Builder can locate a monochromatic matching on at least $\frac{n-t+1}{t+1}$ edges by revealing at most $O(n \log t)$ edges. We show also that in any 3-coloring of $E(K_n)$, Builder can locate a monochromatic tree on at least $n/2$ vertices by revealing at most $5n$ edges.

Mathematics Subject Classifications: 05C57

1 Introduction

For families of graphs $\mathcal{G}_1, \dots, \mathcal{G}_t$, the *Ramsey number*, $R(\mathcal{G}_1, \dots, \mathcal{G}_t)$, is the least integer n such that any t -coloring of $E(K_n)$ contains a copy of some $G_i \in \mathcal{G}_i$ in color i for some $i \in [t]$. If $\mathcal{G}_1 = \mathcal{G}_2 = \dots = \mathcal{G}_t = \mathcal{G}$, then we abbreviate $R_t(\mathcal{G}) = R(\mathcal{G}_1, \dots, \mathcal{G}_t)$. Furthermore, if $\mathcal{G}_i = \{G_i\}$, then we write $R(G_1, \dots, G_t) = R(\mathcal{G}_1, \dots, \mathcal{G}_t)$. Determining the growth of the Ramsey number $R_2(K_n)$ is a central, wide-open problem in extremal combinatorics.

*Supported in part by U.S. taxpayers through NSF CAREER grant DMS-1555149.

Decades of study have spawned a myriad of generalizations and variants of the Ramsey numbers. In this paper, we consider a variant called the *restricted online Ramsey numbers*, which is defined through a two-player game between players Builder and Painter.

Let $\mathcal{G}_1, \dots, \mathcal{G}_t$ be families of graphs and fix $n \geq R(\mathcal{G}_1, \dots, \mathcal{G}_t)$. Painter begins the game by picking a coloring of the edges of K_n , which is hidden from Builder. In each round, Builder points to an edge of K_n and Painter reveals the color of that edge. Builder wins the game once she locates a copy of some $G_i \in \mathcal{G}_i$ in color i of Painter's coloring.

Builder's questions are called *queries* and edges that Builder has queried previously in the game are called *exposed*. The restricted online Ramsey number, $\tilde{R}(\mathcal{G}_1, \dots, \mathcal{G}_t; n)$, is the least integer ℓ for which Builder can guarantee winning the game by querying at most ℓ edges of K_n , regardless of Painter's strategy.

The restricted online Ramsey numbers are themselves a variant on the *online Ramsey numbers*, which were introduced independently by Beck [1] and Kurek and Ruciński [9]. The (unrestricted) online Ramsey numbers are defined through the same Builder–Painter game as above, but do not restrict the number of vertices in play; thus, $\tilde{R}(\mathcal{G}_1, \dots, \mathcal{G}_t; \infty)$ can be used to denote the online Ramsey number. Restricted online Ramsey numbers were originally alluded to in [4] and were recently studied in their own right by Gonzalez, He and Zheng [7].

For any $n \geq N := R(\mathcal{G}_1, \dots, \mathcal{G}_t)$, including $n = \infty$, we arrive at the following straightforward bounds:

$$\frac{N}{2} \leq \tilde{R}(\mathcal{G}_1, \dots, \mathcal{G}_t; n) \leq \binom{N}{2} \quad (1)$$

It is therefore natural to wonder whether $\tilde{R}(\mathcal{G}_1, \dots, \mathcal{G}_t; n)$ is linear in N , quadratic in N , or somewhere in between.

In the case of cliques, setting $N = R_2(K_k)$, Conlon [3] showed that $\tilde{R}_2(K_k; \infty) \leq (1 - \epsilon)^k \binom{N}{2}$ for some fixed $\epsilon > 0$ and k sufficiently large; however, in the restricted setting, the best known upper bound is $\tilde{R}_2(K_k; N) \leq \binom{N}{2} - \Omega(N \log N)$, due to Gonzalez, He and Zheng [7]. This suggests that, in general, determining restricted online Ramsey numbers is substantially more difficult than determining their unrestricted counterparts.

In this paper, we focus on the restricted online Ramsey numbers of matchings and trees. Let rK_2 denote the matching on r edges and let \mathcal{T}_n denote the family of all trees on n vertices. Recall the following well-known Ramsey-type results.

Theorem 1 (Cockayne and Lorimer [2]). *For $t \geq 2$ and positive integers r_1, \dots, r_t ,*

$$R(r_1K_2, \dots, r_tK_2) = \max_i r_i + 1 + \sum_{i=1}^t (r_i - 1).$$

Theorem 2 (Folklore). *For any graph G , either G or the complement of G is connected. In other words, $R_2(\mathcal{T}_n) = n$.*

Theorem 3 (Gerencsér and Gyárfás [6]). *$R_3(\mathcal{T}_n) = 2n - 2$ if n is even and $R_3(\mathcal{T}_n) = 2n - 1$ if n is odd.*

Observe that for any r, t , we trivially have $\tilde{R}_t(rK_2; \infty) = \tilde{R}_t(\mathcal{T}_r; \infty) = (t-1)(r-1) + 1$; however, the restricted version of the problem is more challenging. Despite this, we show that the restricted online Ramsey numbers for these graphs are indeed linear in their respective Ramsey numbers.

Here and throughout this paper, we use $\lg = \log_2$ and employ the convention that $\lg 0 = 0$ so that our results can be stated uniformly.

Theorem 1'. *Fix $t \geq 2$ and positive integers r_1, r_2, \dots, r_t . If $n \geq R(r_1K_2, \dots, r_tK_2)$, then*

$$\tilde{R}(r_1K_2, \dots, r_tK_2; n) \leq \frac{2t - 1 + (t - 3) \lg(t - 2)}{t + 1} n.$$

In particular, Builder can win the game using n queries for two colors, $\frac{5}{4}n$ queries for three colors, and in general, Builder can win the game using $(2 + \lg(t - 2))n$ queries for t colors.

Turning now to trees, the following result was communicated to us by Micek and Pegden.

Theorem 2' (Micek and Pegden [10]). *If $n \geq 2$, then $\tilde{R}_2(\mathcal{T}_n; n) = 2n - 3$.*

Since the above result is unpublished, we present a proof in Section 3.1 for completeness. We extend this result to 3-colorings in Section 3.2. Define $k(n) := \frac{n}{2} + 1$ if $n \equiv 2 \pmod{4}$ and $k(n) := \lceil \frac{n}{2} \rceil$ otherwise. Observe that Theorem 3 states that any 3-coloring of $E(K_n)$ contains a monochromatic tree on $k(n)$ vertices.

Theorem 3'. *If $n \geq 3$, then $\tilde{R}_3(\mathcal{T}_{k(n)}; n) \leq 5(n - 1)$.*

The proofs of Theorems 1', 2' and 3' do not rely on Theorems 1, 2 and 3, and thus provide self-contained proofs of the existence of the respective monochromatic graphs as well.

With the exception of Theorem 2', our upper bounds are not necessarily tight. Indeed, in Theorem 1' with $t = 4$, it is actually possible for Builder to locate one of these monochromatic matchings by querying only $\frac{7}{5}n$ edges, as opposed to the claimed bound of $\frac{8}{5}n$ queries. We do not know whether or not the bound in Theorem 3' is tight.

However, we can prove a different form of tightness by extending the restricted online Ramsey numbers to consider the Builder–Painter game on the edges of K_n when $n < R(\mathcal{G}_1, \dots, \mathcal{G}_t)$. We consider two natural extensions.

The locating game. In this game, Builder must either locate one of the monochromatic graphs or determine that Painter's coloring *cannot* contain any of these graphs.

Suppose that after ℓ queries, Builder has exposed color classes C_1, \dots, C_t . Builder has won the game if either

1. For some $i \in [t]$, C_i contains a copy of some $G_i \in \mathcal{G}_i$, or
2. For every $\chi: E(K_n) \rightarrow [t]$ with $C_i \subseteq \chi^{-1}(i)$, the coloring χ does *not* contain a copy of any $G_i \in \mathcal{G}_i$ in color i for any $i \in [t]$.

Define $\tilde{R}(\mathcal{G}_1, \dots, \mathcal{G}_t; n)$ to be the smallest ℓ for which Builder can guarantee to win the locating game on K_n by querying at most ℓ edges. This is an immediate extension of the original definition of \tilde{R} since Case 2 can never occur if $n \geq R(\mathcal{G}_1, \dots, \mathcal{G}_t)$. Observe, however, that the lower bound in (1) does not necessarily hold when $n < R(\mathcal{G}_1, \dots, \mathcal{G}_t)$.

The cornering game. In this game, Painter is required to guarantee the existence of one of the monochromatic graphs and Builder must simply determine which color contains it.

Call a coloring $\chi: E(K_n) \rightarrow [t]$ *valid* if it contains a copy of some $G_i \in \mathcal{G}_i$ in color i for some $i \in [t]$. Suppose that after ℓ queries, Builder has exposed color classes C_1, \dots, C_t . Builder has won the game if there is some fixed $c \in [t]$ such that every valid coloring χ with $C_i \subseteq \chi^{-1}(i)$ contains a copy of some $G_c \in \mathcal{G}_c$ in color c . In other words, Builder has won the game if she has determined that Painter's coloring *must* contain one of the monochromatic graphs in color c .

Define $\tilde{R}'(\mathcal{G}_1, \dots, \mathcal{G}_t; n)$ to be the smallest ℓ for which Builder can guarantee to win the cornering game on K_n by querying at most ℓ edges. It is not immediately clear that \tilde{R}' is an extension of \tilde{R} , but we will show in Section 4 that $\tilde{R}'(\mathcal{G}_1, \dots, \mathcal{G}_t; n) = \tilde{R}(\mathcal{G}_1, \dots, \mathcal{G}_t; n)$ whenever $n \geq R(\mathcal{G}_1, \dots, \mathcal{G}_t)$.

Observe that we always have the inequality $\tilde{R}(\mathcal{G}_1, \dots, \mathcal{G}_t; n) \geq \tilde{R}'(\mathcal{G}_1, \dots, \mathcal{G}_t; n)$. With these extensions in hand, we show the following.

Theorem 1''. *Fix $t, r \geq 2$. If $n = R_t(rK_2) - 1$, then $\tilde{R}_t(rK_2; n) = \binom{n}{2}$ and $\tilde{R}'_t(rK_2; n) \geq \Omega(n^2)$.*

Theorem 3''. *If $n \geq 3$, then $\tilde{R}_3(\mathcal{T}_{k(n)+1}; n) \geq \tilde{R}'_3(\mathcal{T}_{k(n)+1}; n) \geq \Omega(n^2)$.*

This paper is organized as follows. In Section 2, we prove Theorems 1' and 1''. We then prove Theorem 2' in Section 3.1 and prove Theorems 3' and 3'' in Section 3.2. In each of these sections, we focus first on describing Builder's strategy and then describe Painter's strategy. We conclude with a list of open questions in Section 4.

2 Monochromatic matchings

Builder's strategy. We begin by presenting the key lemma which motivates Builder's strategy. Throughout this paper, a *forest* is assumed to have no isolated vertices, i.e. every connected component is a tree with at least one edge. For a forest F , we denote the set of connected components of F by $\text{comp}(F)$.

Definition 4. Let F be a forest and let $\chi: E(F) \rightarrow [t]$. F is said to be a *good forest* (with respect to χ) if

- χ is a proper edge-coloring of F , and
- there is some color $c \in [t]$ such that every component of F contains an edge of color c .

Lemma 5. Fix $t \geq 2$ and let r_1, \dots, r_t be positive integers. Suppose that F is a forest and $\chi: E(F) \rightarrow [t]$ is a t -coloring. If F is a good forest with respect to χ and $|V(F)| \geq \max_i r_i + \sum_i (r_i - 1)$, then, for some $i \in [t]$, F contains a matching of size r_i in color i .

Proof. Denote by m_i the number of edges of color i in F , so $\sum_i m_i = e(F)$. Since F is a good forest, χ is a proper coloring of $E(F)$, so the largest matching in color i in F has size precisely m_i . Hence, we need only show that $m_i \geq r_i$ for some $i \in [t]$.

By assumption, there is some color $c \in [t]$ which appears in each component of F , so $m_c \geq |\text{comp}(F)|$. Therefore,

$$m_c + \sum_i m_i \geq |\text{comp}(F)| + e(F) = |V(F)| \geq \max_i r_i + \sum_i (r_i - 1),$$

so the claim follows from the pigeonhole principle. □

With this in mind, Builder's strategy is to locate a good forest in Painter's coloring which covers all but at most one vertex. The following lemma presents the main tool employed by Builder to accomplish this.

Lemma 6. Let $\chi: E(K_n) \rightarrow [t]$, and suppose that Builder has exposed all edges of some tree $T \subseteq E(K_n)$ on $m \geq 1$ edges. Suppose T is properly edge-colored under χ and that $xy \in E(K_n)$ is an exposed edge completely disjoint from T with $\chi(xy) \notin \chi(T) := \{\chi(f) : f \in E(T)\}$.

There exists a procedure $\text{TREEEXTEND}(\chi, T, xy)$ that, through querying at most $1 + \lceil \lg(\text{diam}(T) - 1) \rceil$ extra edges, returns a tree T^* with the following properties:

1. $V(T^*) \subseteq V(T) \cup \{x, y\}$,
2. $e(T^*) \in \{m + 1, m + 2\}$,
3. $\chi(T^*) \supseteq \chi(T) \cup \{\chi(xy)\}$, and
4. T^* is properly edge-colored under χ .

Proof. We first define TREEEXTEND and then prove the claimed properties.

Recall that a vertex v of a tree T is called a *center* if v is at a distance at most $\lceil \text{diam}(T)/2 \rceil$ from every other vertex of T , where $\text{diam}(T)$ is the diameter of T . Note that there will be one such vertex when $\text{diam}(T)$ is even, and two such vertices if $\text{diam}(T)$ is odd. We denote the center vertex of T by $\text{center}(T)$, where an arbitrary choice is made if there are two such vertices. Additionally, for an edge $xy \in E(T)$, define $T(x, y)$ to be the subtree of T which is formed by rooting T at y and removing all descendants of x .

- 1: **procedure** $\text{TREEEXTEND}(\chi, T, xy)$
- 2: Fix a proper 2-coloring $\eta: V(T) \rightarrow \{x, y\}$ of T ▷ For $v \in V(T)$, write $\eta_v = \eta(v)$
- 3: $T' \leftarrow T$
- 4: $v \leftarrow \text{center}(T)$
- 5: **loop**

```

6:      Query the edge  $v\eta_v$ 
7:      if  $\chi(v\eta_v) = \chi(xy)$  then
8:          return  $T + v\eta_v$ 
9:      else if  $T + v\eta_v$  is properly colored then
10:         return  $T + v\eta_v + xy$ 
11:      else
12:         There is some edge  $vv' \in E(T')$  with  $\chi(vv') = \chi(v\eta_v)$ 
13:         if  $v'$  is a leaf of  $T$  then
14:             return  $T - vv' + v\eta_v + xy$ 
15:         else if  $v'$  is a leaf of  $T'$  then
16:             return  $T - vv' + v\eta_v + xy + v'\eta_{v'}$ 
17:         else
18:              $T' \leftarrow T'(v, v')$ 
19:              $v \leftarrow \text{center}(T'(v, v'))$ 
20:         end if
21:     end if
22: end loop
23: end procedure

```

We show first that $\text{TREEEXTEND}(\chi, T, xy)$ does in fact return T^* and bound the number of queries made in the process.

A new edge is queried only when reaching the beginning of the loop. If $\text{diam}(T') \in \{1, 2\}$, then the procedure will return T^* before reaching Line 17, thus requiring only one additional query. Furthermore, if we reach Line 17, then we will have $\text{diam}(T'(v, v')) \leq \lceil \frac{\text{diam}(T')}{2} \rceil$ since $v = \text{center}(T')$. From this we conclude that $\text{TREEEXTEND}(\chi, T, e)$ returns T^* by querying at most $1 + \lfloor \lg(\text{diam}(T) - 1) \rfloor$ extra edges, recalling that $\lg 0 = 0$.

We now verify the claimed properties of T^* . Set $c = \chi(xy)$ and consider the four situations in which $\text{TREEEXTEND}(\chi, T, xy)$ can return T^* .

- T^* is returned on Line 8. Here $V(T^*) = V(T) \cup \{\eta_v\} \subseteq V(T) \cup \{x, y\}$ and $e(T^*) = m + 1$. Additionally, $\chi(v\eta_v) = c$, so since $c \notin \chi(T)$, we know that T^* is properly edge-colored and $\chi(T^*) = \chi(T) \cup \{c\}$. Finally, T^* is in fact a tree since adding the edge $v\eta_v$ does not create a cycle.
- T^* is returned on Line 10. Here $V(T^*) = V(T) \cup \{x, y\}$ and $e(T^*) = m + 2$. Additionally, we know that $\chi(v\eta_v) \neq c$, so since $T + v\eta_v$ is properly edge-colored, we know that T^* is also properly edge-colored and $\chi(T^*) = \chi(T) \cup \{\chi(v\eta_v), c\} \supseteq \chi(T) \cup \{c\}$. Finally, T^* is a tree since we do not create a cycle upon adding the edges $v\eta_v$ and xy .
- T^* is returned on Line 14. Here we have $V(T^*) = (V(T) \setminus \{v'\}) \cup \{x, y\}$ and $e(T^*) = m + 1$. Now, since we have $\chi(vv') = \chi(v\eta_v)$ for some $vv' \in E(T')$, we know that, since T' is a subtree of T , $\chi(vv') = \chi(v\eta_v) \neq c$ since $c \notin \chi(T)$. Thus, T^* is properly edge-colored since we removed the edge vv' , and also $\chi(T^*) = \chi(T) \cup \{c\}$.

since $\chi(v\eta_v) = \chi(vv')$. Finally, T^* is a tree since v' is a leaf of T , so $T - vv'$ is still at tree, and then adding the edges $v\eta_v$ and xy does not create a cycle.

- T^* is returned on Line 16. Here we have $V(T^*) = V(T) \cup \{x, y\}$ and $e(T^*) = m + 2$. Now, since v' is a leaf of T' but not a leaf of T , this means that we must have previously queried the edge $v'\eta_{v'}$ and found that $\chi(v'\eta_{v'}) = \chi(vv')$. Since η is a proper 2-coloring of $V(T)$ and $vv' \in E(T)$, we must have $\eta_v \neq \eta_{v'}$, so T^* is indeed properly edge-colored, and $\chi(T^*) = \chi(T) \cup \{c\}$. Finally, T^* is again a tree since we added the path $v\eta_v\eta_{v'}v'$ and removed the edge vv' . \square

We state a consequence for later reference.

Corollary 7. *Let χ, T, xy be as in Lemma 6 and set $T^* = \text{TREEEXTEND}(\chi, T, xy)$.*

- *If $e(T) = 1$, then T^* either has $|\chi(T^*)| = 2$ and is a path with 2 edges, or has $|\chi(T^*)| = 3$ and is a path with 3 edges.*
- *If $e(T) = 2$, then T^* either has $|\chi(T^*)| \geq 4$, or has $|\chi(T^*)| = 3$ and is a star with 3 edges.*

We now have all of the necessary tools to describe Builder's full strategy and prove Theorem 1'.

Proof of Theorem 1'. Builder maintains and grows a good forest F . While there are still at least 2 vertices x, y uncovered by F , Builder queries the edge xy . If its color $\chi(xy)$ is already present among all components of F , Builder adds this edge (as a 2-vertex component) to F , and repeats. Otherwise, there is some connected component $T \in \text{comp}(F)$ which does not have an edge of color $\chi(xy)$.

Here, Builder uses $\text{TREEEXTEND}(\chi, T, xy)$ to return a tree T^* and replaces F by $F - T + T^*$. By Lemma 6, $F - T + T^*$ is also a good forest and covers at least one more vertex than F , so this process must eventually terminate. Furthermore, the process terminates when F covers all but at most one vertex of K_n , and thus, by Lemma 5, Builder has located a copy of $r_i K_2$ in color i for some $i \in [t]$. Let F^* be the forest found by Builder.

We now count the total number of queries used to build F^* . For integers $m, k \geq 2$, define $q(m, k) := 2k - 1 + (k - 3) \lg(m - 2)$ and define also $q(1, 1) := 1$. For $T \in \text{comp}(F^*)$, let $q(T)$ denote the number of queries used by Builder to construct T .

Claim 8. *If $e(T) = m$ and $|\chi(T)| = k$, then $q(T) \leq q(m, k)$.*

Proof. If $k = 1$, then also $e(T) = 1$, so certainly $q(T) = 1 = q(1, 1)$. Thus suppose $k \geq 2$.

Since TREEEXTEND always appends at least one new color to a tree, we see that, for some $\ell \leq k$, there were trees T_1, \dots, T_ℓ and edges e_1, \dots, e_ℓ with T_1 being a single edge, $T_\ell = T$ and $T_{i+1} = \text{TREEEXTEND}(\chi, T_i, e_i)$ for all $i \in [\ell - 1]$. Certainly $q(T_1) = 1$, and, by Lemma 6, if $d_i := \text{diam}(T_i)$, then

$$q(T_{i+1}) \leq q(T_i) + 1 + (1 + \lfloor \lg(d_i - 1) \rfloor),$$

where the extra +1 comes from querying the edge e_i . Therefore,

$$q(T) \leq 1 + \sum_{i=1}^{\ell-1} (2 + \lfloor \lg(d_i - 1) \rfloor) = 2\ell - 1 + \sum_{i=2}^{\ell-1} \lfloor \lg(d_i - 1) \rfloor.$$

By Lemma 6, we know that $e(T_{i+1}) \in \{e(T_i) + 1, e(T_i) + 2\}$, so certainly $d_i \leq e(T_i) \leq m - 1$ for all $i \in [\ell - 1]$.

We now break into two cases based on T_2 :

- $|\chi(T_2)| = 2$: Since $T_2 = \text{TREEEXTEND}(\chi, T_1, e_1)$, we know that T_2 must be a path on 2 edges by Corollary 7; thus $d_2 = 2$. As such,

$$q(T) \leq 2\ell - 1 + \sum_{i=3}^{\ell-1} \lfloor \lg(d_i - 1) \rfloor \leq 2k - 1 + (k - 3) \lg(m - 2) = q(m, k).$$

- $|\chi(T_2)| \geq 3$: Here, again by Corollary 7, we must actually have $|\chi(T_2)| = 3$ and T_2 is a path on 3 edges, so $d_2 = 3$. Additionally, in this situation, we must also have $\ell \leq k - 1$, and so we bound

$$q(T) \leq 2\ell + \sum_{i=3}^{\ell-1} \lfloor \lg(d_i - 1) \rfloor \leq 2k - 2 + (k - 4) \lg(m - 2) \leq q(m, k). \quad \square$$

For positive integers m, k , let $F_{m,k}$ denote the forest formed by all trees $T \in \text{comp}(F^*)$ with $e(T) = m$ and $|\chi(T)| = k$. By Lemma 6 and Corollary 7, the only values of (m, k) for which $F_{m,k}$ can be nonempty are: $(1, 1)$, $(2, 2)$, $(3, 3)$ and (m, k) where $4 \leq k \leq m \leq 2k - 3$.

For $4 \leq k \leq m \leq 2k - 3$, a quick calculation shows that

$$\frac{q(m, k)}{m + 1} \geq \frac{q(m + 1, k)}{m + 2} \implies \max_{m:k \leq m \leq 2k-3} \frac{q(m, k)}{m + 1} = \frac{q(k, k)}{k + 1}.$$

Finally, another short calculation yields $\frac{q(k,k)}{k+1} \leq \frac{q(t,t)}{t+1}$ for all $k \in [t]$.

Thus, by Claim 8, we find that the total number of queries used to locate F^* , and thus the monochromatic matching, is bounded above by

$$\begin{aligned} \sum_{T \in \text{comp}(F^*)} q(T) &\leq \sum_{m,k} |\text{comp}(F_{m,k})| q(m, k) = \sum_{m,k} |V(F_{m,k})| \frac{q(m, k)}{m + 1} \\ &\leq \frac{q(t, t)}{t + 1} \sum_{m,k} |V(F_{m,k})| \leq \frac{1}{t + 1} (2t - 1 + (t - 3) \lg(t - 2)) n. \quad \square \end{aligned}$$

Remark 9. For any fixed $t \geq 4$, one can embark on a more sensitive analysis to improve the upper bound of $\frac{1}{t+1} (2t - 1 + (t - 3) \lg(t - 2)) n$. For example, when $t = 4$, we can arrive at an upper bound of $\frac{7}{5} n$ as opposed to $\frac{8}{5} n$ by working through Claim 8 more carefully. However, as t grows, it becomes increasingly difficult to carry out such an analysis.

Remark 10. We can improve slightly the upper bound of n queries when $t = 2$. Indeed, with the strategy described above, the number of queries is at most $\frac{1}{2}|V(F_{1,1})| + |V(F_{2,2})|$, which is equal to n only when $|V(F_{2,2})| = n$. However, $|V(F_{2,2})|$ is a multiple of 3, so the bound can be improved to $n - 1$ whenever $3 \nmid n$.

For $t = 3$, similar reasoning will improve also the upper bound of $\frac{5}{4}n$ by an additive constant depending on the value of $n \pmod{4}$.

Painter's strategy. We now describe Painter's strategy in both the locating and cornering games for hiding a monochromatic rK_2 from Builder when $n = R_t(rK_2) - 1 = (t + 1)r - t$.

Proof of Theorem 1''. Consider partitioning $V(K_n) = V_1 \sqcup \dots \sqcup V_t$ where $|V_1| = 2r - 1$ and $|V_i| = r - 1$ for all $i \geq 2$. Let $\chi: E(K_n) \rightarrow [t]$ be given by $\chi(xy) = \max\{i : V_i \cap \{x, y\} \neq \emptyset\}$.

Certainly χ does not contain a monochromatic rK_2 . For an edge $e \in E(K_n)$ and a color $c \in [t]$, let $\chi_{e,c}$ denote the coloring of $E(K_n)$ obtained by giving e color c and coloring the rest of the edges as in χ . Notice that if e is not completely contained in V_1 , then $\chi_{e,1}$ has a monochromatic rK_2 in color 1, and if e is completely disjoint from V_c for some $c \geq 2$, then $\chi_{e,c}$ has a monochromatic rK_2 in color c .

Consider first the locating game wherein Painter is not required to guarantee the existence of a monochromatic rK_2 . As Builder queries edges, Painter colors the edge as in χ until there is only one unexposed edge, call it e . Certainly there is some $c \in [t]$ such that $\chi_{e,c}$ has an rK_2 in color c , so since χ does not have any monochromatic rK_2 , Builder must query all $\binom{n}{2}$ edges of K_n in order to determine whether or not Painter's coloring has a monochromatic matching of size r .

Consider now the cornering game wherein Painter is required to guarantee the existence of a monochromatic rK_2 and Builder needs only determine which color has said matching. Again, Painter will color the edges that Builder queries as in χ until the very last edge, which she then gives a color which will form a monochromatic rK_2 . However, against this strategy, Builder can sometimes deduce which color will have this matching before reaching the very last edge.

If $t = 2$, then Painter's coloring has an rK_2 in color 2 if and only if some edge in V_1 gets color 2 (and otherwise the coloring must have an rK_2 in color 1). Therefore, to determine which color contains the matching, Builder must either query every edge in V_1 or query every edge not completely contained in V_1 . Hence, Builder must query at least

$$\min\left\{\binom{|V_1|}{2}, \binom{n}{2} - \binom{|V_1|}{2}\right\} = \frac{1}{9}(2n + 1)(n - 1)$$

edges.

For $t = 3$, Builder must query all edges *not* in $E[V_2, V_3]$ (the edges with one vertex in V_2 and the other in V_3); otherwise, there is some unexposed edge which is either completely contained in V_1 or meets V_1 in only one vertex and is disjoint from either V_2 or V_3 . Therefore, Builder must query at least

$$\binom{n}{2} - |V_2||V_3| = \frac{1}{16}(7n + 1)(n - 1)$$

edges.

Lastly, for $t \geq 4$, since every edge is disjoint from at least two of V_1, V_2, V_3, V_4 , Builder must query all $\binom{n}{2}$ edges to determine which color contains the matching. \square

3 Monochromatic trees

3.1 Two colors

Here we give the proof of Theorem 2', which was communicated to us by Micek and Pegden [10].

Builder's strategy. Let $\chi: E(K_n) \rightarrow [2]$. Builder begins by choosing $v \in V(K_n)$ arbitrarily and queries all $n - 1$ edges incident to v . For $i \in \{1, 2\}$, set $C_i^0 = \{v\} \cup \{u \in V(K_n) : \chi(uv) = i\}$. Builder proceeds recursively as follows: if there is some $x \in C_1^r \setminus C_2^r$ and $y \in C_2^r \setminus C_1^r$, Builder queries the edge xy and sets $C_{\chi(xy)}^{r+1} = C_{\chi(xy)}^r \cup \{x, y\}$ and $C_{3-\chi(xy)}^{r+1} = C_{3-\chi(xy)}^r$.

Notice that, for every r , C_i^r is a connected component in color i , so if ever $|C_i^r| = n$ for either $i = 1$ or $i = 2$, then Builder has located a spanning tree in color i . Set $a(r) = |C_1^r| + |C_2^r|$, so $a(0) = n + 1$ and if $a(r) \geq 2n - 1$, then it must be the case that $|C_i^r| = n$ for either $i = 1$ or $i = 2$. If Builder has not located a monochromatic spanning tree by step r , then since $C_1^r \cup C_2^r = V(K_n)$, we must have $a(r + 1) = a(r) + 1$. We conclude that Builder can locate a monochromatic spanning tree using at most $(n - 1) + (n - 2) = 2n - 3$ queries. \square

Painter's strategy. Painter will color the first $n - 2$ queried edges with color 1, color the next $n - 2$ queried edges with color 2, and then color the remaining edges arbitrarily. Since a spanning tree has $n - 1$ edges, Builder cannot have located a monochromatic spanning tree within the first $2(n - 2)$ queries, and thus must query at least $2n - 3$ edges. \square

3.2 Three colors

Builder's strategy. We begin with the key lemma which motivates Builder's strategy. Recall that $k(n) := \frac{n}{2} + 1$ if $n \equiv 2 \pmod{4}$ and $k(n) := \lceil \frac{n}{2} \rceil$ otherwise.

Lemma 11. For $n \geq 3$, suppose that there are $U_1, \dots, U_6 \subseteq V(K_n)$ (some of which may be empty) with $E(K_n) = \bigcup_{i=1}^6 \binom{U_i}{2}$. If $U_1 \cap U_2 = \emptyset$, then $|U_i| \geq k(n)$ for some $i \in [6]$.

Proof. Assume for the sake of contradiction that $|U_i| < k := k(n)$ for each $i \in [6]$. Since $k - 1 \leq \lfloor n/2 \rfloor$, and $U_1 \cap U_2 = \emptyset$, we can find $A \supseteq U_1$ and $B \supseteq U_2$ with $A \cap B = \emptyset$ and $|A| = \lceil n/2 \rceil$ and $|B| = \lfloor n/2 \rfloor$.

Consider the edges between A and B , denoted by $E[A, B]$. Since $|U_i| \leq k - 1$, by convexity we must have

$$\left| \binom{U_i}{2} \cap E[A, B] \right| \leq \left\lfloor \frac{(k-1)^2}{4} \right\rfloor,$$

for every i . However, the edges induced by U_3, \dots, U_6 must cover all edges between A and B , so we must have

$$4 \left\lfloor \frac{(k-1)^2}{4} \right\rfloor \geq \left\lfloor \frac{n^2}{4} \right\rfloor,$$

contradicting the definition of $k = k(n)$ for any value of $n \pmod{4}$. □

Hence, Builder will work to find six subsets of $V(K_n)$, each being contained within a connected component of some color class, wherein each pair of vertices are contained in one of these sets. The following lemma presents the primary tool in Builder's strategy.

Lemma 12. *Let $\chi: E(K_n) \rightarrow [3]$ and suppose that Builder has queried some edges giving rise to graphs C_1, C_2, C_3 where C_i is the graph formed by the exposed edges of color i .*

Suppose that $V_1 \subseteq V(C_1)$ and $V_2 \subseteq V(C_2)$ are subsets of connected components of C_1 and C_2 , respectively. There exists a procedure $\text{COMPEXTEND}(\chi, V_1, V_2)$ which returns a tuple (X_1, X_2, X_3) of subsets of $V(K_n)$ using at most $2|V_1| + 2|V_2|$ additional queries. If C_i^ denotes the graph formed by the exposed edges of color i after calling $\text{COMPEXTEND}(\chi, V_1, V_2)$, then (X_1, X_2, X_3) satisfies:*

1. $X_1, X_2, X_3 \subseteq V_1 \cup V_2$,
2. $X_1 \supseteq V_1$ and $X_2 \supseteq V_2$,
3. for all $i \in [3]$, X_i is a subset of some connected component of C_i^* , and
4. one of the following:
 - (a) $X_1 = V_1 \cup V_2$,
 - (b) $X_2 = V_1 \cup V_2$, or
 - (c) $X_3 \supseteq (V_1 \setminus X_2) \cup (V_2 \setminus X_1)$.

Proof. We first define COMPEXTEND and then prove the claimed properties.

```

1: procedure  $\text{COMPEXTEND}(\chi, V_1, V_2)$ 
2:    $X_1 \leftarrow V_1$ 
3:    $X_2 \leftarrow V_2$ 
4:    $X_3 \leftarrow \emptyset$ 
5:   loop
6:     if there is  $u \in X_1 \setminus (X_2 \cup X_3)$  and  $v \in (X_2 \cap X_3) \setminus X_1$  then
7:       Query the edge  $uv$ 
8:        $X_{\chi(uv)} \leftarrow X_{\chi(uv)} \cup \{u, v\}$ 
9:     else if there is  $u \in X_2 \setminus (X_1 \cup X_3)$  and  $v \in (X_1 \cap X_3) \setminus X_2$  then
10:      Query the edge  $uv$ 
11:       $X_{\chi(uv)} \leftarrow X_{\chi(uv)} \cup \{u, v\}$ 
12:     else if  $X_3 \subseteq X_1 \cap X_2$  and there is  $u \in X_1 \setminus X_2$  and there is  $v \in X_2 \setminus X_1$  then
13:      Query the edge  $uv$ 

```

```

14:         if  $\chi(uv) \in \{1, 2\}$  then
15:              $X_{\chi(uv)} \leftarrow X_{\chi(uv)} \cup \{u, v\}$ 
16:         else
17:              $X_3 \leftarrow \{u, v\}$ 
18:         end if
19:     else
20:         return  $(X_1, X_2, X_3)$ 
21:     end if
22: end loop
23: end procedure

```

Items (1) and (2) are straightforward to check. To verify Item (3), notice that in Lines 8, 11 and 15, if we extended X_i to include some vertex u , then this is because there is some $v \in X_i$ with $\chi(uv) = i$. In particular, since V_1, V_2 were subsets of connected components of C_1, C_2 , respectively, we have that at every stage, X_i is a subset of a connected component in the currently exposed edges of color i .

Item (4) follows from the observation that this is the only way that the procedure will ever break the loop and return (X_1, X_2, X_3) . Thus, to verify Item (4), we simply must show that the procedure does indeed terminate. We do so by showing that the procedure terminates after at most $2|V_1| + 2|V_2| + 1$ loops, implying also that it queried at most $2|V_1| + 2|V_2|$ additional edges.

For an integer r , let X_i^r denote the value of X_i in $\text{COMPEXTEND}(\chi, V_1, V_2)$ before running through the loop for the $(r + 1)$ st time, so $X_1^0 = V_1$, $X_2^0 = V_2$ and $X_3^0 = \emptyset$. Notice that we always have $X_1^r \subseteq X_1^{r+1}$ and $X_2^r \subseteq X_2^{r+1}$ and $X_1^r \cup X_2^r = V_1 \cup V_2 \supseteq X_3^r$.

Set $a(r) := 2|X_1^r| + 2|X_2^r| + |X_3^r \setminus (X_1^r \cap X_2^r)|$. Certainly $a(0) = 2|V_1| + 2|V_2|$ and for any r ,

$$\begin{aligned}
 a(r) &= 2|X_1^r \cup X_2^r| + 2|X_1^r \cap X_2^r| + |X_3^r \setminus (X_1^r \cap X_2^r)| \\
 &= 2|X_1^r \cup X_2^r| + |X_1^r \cap X_2^r| + |X_3^r \cup (X_1^r \cap X_2^r)| \\
 &\leq 3|X_1^r \cup X_2^r| + |X_1^r \cap X_2^r| \leq 4|V_1 \cup V_2| \leq 4|V_1| + 4|V_2|.
 \end{aligned}$$

We claim that if $\text{COMPEXTEND}(\chi, V_1, V_2)$ has *not* output (X_1, X_2, X_3) after the $(r + 1)$ st loop (that is, the procedure will run through the loop for an $(r + 2)$ nd time), then $a(r + 1) \geq a(r) + 1$, thus implying the claim. We break into the following cases depending on how loop $r + 1$ terminates:

1. If the procedure reaches Line 8, then one of the following occurs:

- (a) $X_3^{r+1} = X_3^r \cup \{u\}$. Here, since $u \notin X_2^r \cup X_3^r$, we have $|X_3^{r+1} \setminus (X_1^{r+1} \cap X_2^{r+1})| = |X_3^r \setminus (X_1^r \cap X_2^r)| + 1$, so $a(r + 1) = a(r) + 1$.
- (b) $|X_i^{r+1}| \geq |X_i^r| + 1$ for some $i \in \{1, 2\}$. Thus, even though $|X_3^{r+1} \setminus (X_1^{r+1} \cap X_2^{r+1})|$ may decrease by 1, we still have $a(r + 1) \geq a(r) + 2 - 1 = a(r) + 1$.

2. If the procedure reaches Line 11, then $a(r + 1) \geq a(r) + 1$ by a symmetric argument.

3. If the procedure reaches Line 15, then we have $|X_i^{r+1}| \geq |X_i^r| + 1$ for some $i \in \{1, 2\}$ and also $|X_3^{r+1} \setminus (X_1^{r+1} \cap X_2^{r+1})| = |X_3^r \setminus (X_1^r \cap X_2^r)| = 0$, so $a(r+1) = a(r) + 2$.
4. Finally, if the procedure reaches Line 17, then $X_i^{r+1} = X_i^r$ for $i \in \{1, 2\}$ and $X_3^{r+1} = \{u, v\}$. Since $u, v \notin X_1^r \cap X_2^r$, we have $|X_3^{r+1} \setminus (X_1^{r+1} \cap X_2^{r+1})| = 2$ while $|X_3^r \setminus (X_1^r \cap X_2^r)| = 0$, so again $a(r+1) = a(r) + 2$. \square

We now have all of the necessary tools describe Builder's full strategy and prove Theorem 3'.

Proof of Theorem 3'. Let $\chi: E(K_n) \rightarrow \{r, g, b\}$ be a 3-coloring. Builder begins by choosing $v \in V(K_n)$ arbitrarily and queries all $n - 1$ edges incident to v . Let $R = \{u \in V(K_n) \setminus \{v\} : \chi(vu) = r\}$ and define G and B analogously for colors g and b . Note that R, G, B are subsets of the vertices of some connected components in the currently exposed edges in colors r, g, b , respectively. Now, Builder uses $\text{COMPEXTEND}(\chi, R, G)$, $\text{COMPEXTEND}(\chi, G, B)$ and $\text{COMPEXTEND}(\chi, B, R)$ (with the appropriate relabeling of the colors) to find tuples (R_1, G_2, B_3) , (G_1, B_2, R_3) and (B_1, R_2, G_3) , respectively, as in Lemma 12. This requires at most

$$(2|R| + 2|G|) + (2|R| + 2|B|) + (2|B| + 2|G|) = 4(|R| + |B| + |G|) = 4(n - 1)$$

additional queries, thus bringing the total number of queries to at most $5(n - 1)$.

We claim that Builder has located a monochromatic tree on at least $k(n)$ vertices. Suppose that C_r, C_g, C_b are the graphs formed by the exposed edges in colors r, g, b , respectively. If $R_1 \cup R_2 \cup R_3$ is a subset of a connected component of C_r , set $R_1^* = R_1 \cup R_2 \cup R_3 \cup \{v\}$ and $R_2^* = \emptyset$, and otherwise set $R_1^* = R_1 \cup R_2 \cup \{v\}$ and $R_2^* = R_3$. Define G_i^* and B_i^* analogously for $i \in \{1, 2\}$.

In any case, by Lemma 12, we know that for each $i \in \{1, 2\}$, R_i^* is a subset of the vertices of a connected component of C_r , G_i^* is a subset of the vertices of a connected component of C_g , and B_i^* is a subset of a connected component of C_b , thus we need only show that at least one of these sets has size at least $k(n)$. We do this by appealing to Lemma 11.

By definition, $R_1^* \cap R_2^* = G_1^* \cap G_2^* = B_1^* \cap B_2^* = \emptyset$, so we need only show that

$$E(K_n) = \binom{R_1^*}{2} \cup \binom{R_2^*}{2} \cup \binom{G_1^*}{2} \cup \binom{G_2^*}{2} \cup \binom{B_1^*}{2} \cup \binom{B_2^*}{2},$$

i.e. every pair of vertices of K_n are contained together in one of these six sets. Let $x, y \in V(K_n)$ be two distinct vertices. If, say, $x = v$, then $y \in R \cup B \cup G$, and so x, y are contained together in one of R_1^*, G_1^*, B_1^* . If we have $x, y \in R$, then certainly $x, y \in R_1^*$, and similarly if $x, y \in G$ or $x, y \in B$.

Thus, suppose that, without loss of generality, $x \in R, y \in G$. Certainly $x \in R_1 \subseteq R_1^*$ and $y \in G_2 \subseteq G_1^*$, so suppose that $x \notin G_1^*$ and $y \notin R_1^*$. Then by Item 4 in Lemma 12, we must have $B_3 \supseteq (R \setminus G_2) \cup (G \setminus R_1)$, so $x, y \in B_3$. Hence, either $x, y \in B_1^*$ or $x, y \in B_2^*$. \square

Painter’s strategy. Painter’s strategy here is very similar to her strategy in the proof of Theorem 1’.

Proof of Theorem 3’. Since $\tilde{R}_3(\mathcal{T}_{k(n)+1}; n) \geq \tilde{R}'_3(\mathcal{T}_{k(n)+1}; n)$, it suffices to give only a strategy for the cornering game.

Let χ be a 3-coloring of $E(K_n)$ formed by starting with a proper edge-coloring of K_4 using three colors, “blowing up” each vertex into a cluster of size roughly $n/4$, and then coloring the edges within the clusters arbitrarily. Formally speaking, start with a partition $[n] = V_1 \sqcup V_2 \sqcup V_3 \sqcup V_4$ with $|V_i| \in \{\lfloor n/4 \rfloor, \lceil n/4 \rceil\}$ for all i , and let χ be the 3-coloring of $E(K_n)$ given by

- $\chi(e) = r$ if $e \in E[V_1, V_2] \cup E[V_3, V_4]$,
- $\chi(e) = b$ if $e \in E[V_1, V_3] \cup E[V_2, V_4]$,
- $\chi(e) = g$ if $e \in E[V_1, V_4] \cup E[V_2, V_3]$, and
- $\chi(e)$ is arbitrary otherwise.

It is straightforward to verify that χ does not contain a monochromatic tree on $k(n)+1$ vertices.

For an edge e and color $c \in \{r, b, g\}$, let $\chi_{e,c}$ denote the coloring where e gets color c and every other edge is colored as in χ . Notice that if e is any edge not completely contained in some V_i , then $\chi_{e,c}$ actually contains a spanning tree in color c whenever $c \neq \chi(e)$.

When Builder queries an edge e , Painter colors e as in χ , unless e is the last unexposed edge which is not completely contained in one of the V_i ’s. In this situation, Painter gives e either color c_1 or c_2 where $c_1, c_2 \neq \chi(e)$.

Thus, Painter’s coloring will always contain a monochromatic tree on $k(n)+1$ vertices (in fact, it will always contain a monochromatic spanning tree), but Builder must query every edge not completely contained in some V_i to determine which color contains said tree. As such, Builder must query at least $\sum_{i \neq j \in [4]} |V_i||V_j| \geq 6 \lfloor \frac{n}{4} \rfloor^2$ edges to do so. \square

4 Remarks

We begin by showing that \tilde{R}' is indeed a generalization of \tilde{R} .

Lemma 13. Fix $t \geq 2$ and let $\mathcal{G}_1, \dots, \mathcal{G}_t$ be families of graphs. If $n \geq R(\mathcal{G}_1, \dots, \mathcal{G}_t)$, then $\tilde{R}'(\mathcal{G}_1, \dots, \mathcal{G}_t; n) = \tilde{R}(\mathcal{G}_1, \dots, \mathcal{G}_t; n)$.

Proof. We need only show the inequality $\tilde{R}(\mathcal{G}_1, \dots, \mathcal{G}_t; n) \leq \tilde{R}'(\mathcal{G}_1, \dots, \mathcal{G}_t; n)$. By the assumption on n , any strategy for Painter in the locating game is also a strategy for Painter in the cornering game.

Suppose that Builder has exposed some collection of edges, giving rise to color classes C_1, \dots, C_t , and that Builder has won the cornering game by determining, without loss

of generality, that color 1 must contain a copy of some $G_1 \in \mathcal{G}_1$. Consider the coloring $\chi: E(K_n) \rightarrow [t]$ defined by $\chi(e) = i$ if $e \in C_i$ and $\chi(e) = t$ otherwise. Since χ could be Painter's hidden coloring and $C_1 = \chi^{-1}(1)$, it must be the case that C_1 contains some $G_1 \in \mathcal{G}_1$. Therefore, Builder has additionally won the locating game. \square

We showed in Theorem 1' that $\tilde{R}_t(rK_2; n) \leq O(n \log t)$ whenever $n \geq (t+1)r - t + 1$. While we have already remarked that the precise bound in Theorem 1' is not necessarily tight (Remark 9), we wonder if it is tight up to a constant:

Question 14. Fix $t, r \geq 2$ and set $n = (t+1)r - t + 1$. Is $\tilde{R}_t(rK_2; n) \geq \Omega(n \log t)$?

Notice that, by Lemma 13, it does not matter whether we consider \tilde{R} or \tilde{R}' in this question.

We additionally do not know the correct answer even in the case of only 2 colors. In this situation, since $n = 3r - 1 \not\equiv 0 \pmod{3}$, we have $\tilde{R}_2(rK_2; n) \leq n - 1$ (Remark 10).

Question 15. For $r \geq 1$ and $n = 3r - 1$, is it the case that $\tilde{R}_2(rK_2; n) = n - 1$?

Turning now to trees, we showed in Theorem 3' that $\tilde{R}_3(\mathcal{T}_{k(n)}; n) \leq 5(n - 1)$. We wonder if this result can be improved:

Question 16. What is the smallest c such that in any 3-coloring of $E(K_n)$, Builder can locate a monochromatic tree on $k(n)$ vertices by querying at most $(c + o(1))n$ edges?

Again, thanks to Lemma 13, it does not matter whether we consider \tilde{R} or \tilde{R}' here. Theorem 3' shows that $c \leq 5$; furthermore, certainly $c \geq 3/2$ since Painter can simply color the first $k(n) - 1$ edges red, the next $k(n) - 1$ edges blue, and the next $k(n) - 1$ edges green.

Theorems 2 and 3 were extended by Gyárfás [8] to show that any t -coloring of $E(K_n)$ must contain a monochromatic tree on at least $\frac{n}{t-1}$ vertices. Füredi [5] showed that this bound can be improved slightly in the case where an affine plane of order $t - 1$ does not exist. These suggest the natural question:

Question 17. What is the least number of queries necessary for Builder to locate a monochromatic tree on at least $\frac{n}{t-1}$ vertices in a t -coloring of $E(K_n)$?

We suspect that $\tilde{R}_t(\mathcal{T}_{n/(t-1)}; n) \leq O_t(n)$, which was the case for both 2 and 3 colors.

Acknowledgments

We are grateful to the anonymous referee, whose comments helped significantly improve the presentation of our results.

References

- [1] J. Beck. Achievement games and the probabilistic method. In *Combinatorics, Paul Erdős is eighty, Vol. 1*, Bolyai Soc. Math. Stud., pages 51–78. János Bolyai Math. Soc., Budapest, 1993.
- [2] E. Cockayne and P. J. Lorimer. The Ramsey number for stripes. *Journal of the Australian Mathematical Society*, 19(2):252–256, 1975.
- [3] D. Conlon. On-line Ramsey numbers. *SIAM J. Discrete Math.*, 23(4):1954–1963, 2009/10.
- [4] D. Conlon, J. Fox, A. Grinshpun, and X. He. Online Ramsey numbers and the subgraph query problem. In *Building Bridges II*, pages 159–194. Springer, 2019.
- [5] Z. Füredi. Covering the complete graph by partitions. *Discrete Mathematics*, 75(1-3):217–226, 1989.
- [6] L. Gerencsér and A. Gyárfás. On Ramsey-type problems. *Ann. Univ. Sci. Budapest. Eötvös Sect. Math.*, 10:167–170, 1967.
- [7] D. Gonzalez, X. He, and H. Zheng. An upper bound for the restricted online Ramsey number. *Discrete Mathematics*, 342(9):2564–2569, 2019.
- [8] A. Gyárfás. Partíciófedések és lefogóhalmazok hipergráfokban. *Tanulmányok-MTA Számítástechn. Automat. Kutató Int. Budapest*, (71):62, 1977.
- [9] A. Kurek and A. Ruciński. Two variants of the size Ramsey number. *Discuss. Math. Graph Theory*, 25(1-2):141–149, 2005.
- [10] P. Micek and W. Pegden. Personal communication.