

On-line Ramsey Theory for Bounded Degree Graphs*

Jane Butterfield Tracy Grauman William B. Kinnersley
Kevin G. Milans Christopher Stocker Douglas B. West

University of Illinois
Urbana IL, U.S.A.

Submitted: Nov 9, 2010; Accepted: Jun 15, 2011; Published: Jul 1, 2011

Mathematics Subject Classification: 05C55, 05C57

Abstract

When graph Ramsey theory is viewed as a game, “Painter” 2-colors the edges of a graph presented by “Builder”. Builder wins if every coloring has a monochromatic copy of a fixed graph G . In the on-line version, iteratively, Builder presents one edge and Painter must color it. Builder must keep the presented graph in a class \mathcal{H} . Builder wins the game (G, \mathcal{H}) if a monochromatic copy of G can be forced. The *on-line degree Ramsey number* $\mathring{R}_\Delta(G)$ is the least k such that Builder wins (G, \mathcal{H}) when \mathcal{H} is the class of graphs with maximum degree at most k . Our results include:

- 1) $\mathring{R}_\Delta(G) \leq 3$ if and only if G is a linear forest or each component lies inside $K_{1,3}$.
- 2) $\mathring{R}_\Delta(G) \geq \Delta(G) + t - 1$, where $t = \max_{uv \in E(G)} \min\{d(u), d(v)\}$.
- 3) $\mathring{R}_\Delta(G) \leq d_1 + d_2 - 1$ for a tree G , where d_1 and d_2 are two largest vertex degrees.
- 4) $4 \leq \mathring{R}_\Delta(C_n) \leq 5$, with $\mathring{R}_\Delta(C_n) = 4$ except for finitely many odd values of n .
- 5) $\mathring{R}_\Delta(G) \leq 6$ when $\Delta(G) \leq 2$.

The lower bounds come from strategies for Painter that color edges red whenever the red graph remains in a specified class. The upper bounds use a result showing that Builder may assume that Painter plays “consistently”.

1 Introduction

The classical problem of graph Ramsey theory specifies a target graph G and seeks a graph H such that every 2-coloring of $E(H)$ produces a monochromatic copy of G . For such H , we write $H \rightarrow G$ and say that H *arrows* G . More generally, when every s -coloring of $E(H)$ produces a monochromatic G , we write $H \xrightarrow{s} G$. Ramsey’s Theorem guarantees for every G that such a graph H exists. The *Ramsey number* $R(G; s)$ (or $R(G)$ when $s = 2$) is the minimum number of vertices in such a graph H .

*Email addresses: jbutter2@uiuc.edu, grauman2@uiuc.edu, wkinner2@uiuc.edu,
milans@uiuc.edu, stocker2@math.uiuc.edu, west@math.uiuc.edu. Research of D.B. West supported in part by the NSA under Award No. H98230-10-1-0363. Research of other authors supported by the NSF grant DMS 08-38434 “EMSW21-MCTP: Research Experience for Graduate Students”.

Given a graph parameter ρ , the ρ -Ramsey number $R_\rho(G; s)$ is $\min\{\rho(H) : H \xrightarrow{s} G\}$; this is the ordinary Ramsey number $R(G; s)$ when ρ is the number of vertices. When $\rho(G)$ is the clique number $\omega(G)$, Nešetřil and Rödl [19] proved that $R_\omega(G; s) = \omega(G)$, extending the special case for $s = 2$ by Folkman [8]. The *size Ramsey number* of G is the minimum number of edges in a graph H with $H \rightarrow G$. The first result on the size Ramsey number, by Erdős, Faudree, Rousseau, and Schelp [10], was that for $G = K_n$ it equals the obvious upper bound $\binom{R(K_n)}{2}$ (see [2, 3, 7, 24] for further results on size Ramsey number).

When $\rho(G)$ is the chromatic number $\chi(G)$, Burr, Erdős, and Lovász [5] showed that $R_\chi(G)$ equals the Ramsey number of the family of homomorphic images of G , where the Ramsey number of a family \mathcal{G} is the minimum number of vertices in a graph H such that every 2-coloring of $E(H)$ produces a monochromatic copy of some graph in \mathcal{G} . They also conjectured $\min\{R_\chi(G; s) : \chi(G) = k\} = k^s + 1$, which has been proved by Zhu [27, 28].

Burr, Erdős, and Lovász [5] also studied the *degree Ramsey number* R_Δ , where $\Delta(G)$ is the maximum vertex degree in G . Further results on R_Δ will appear in [14, 15, 18].

Another modern variation in graph Ramsey theory is an on-line or “game” version introduced by Beck [4]. We consider the 2-color case, but the game extends naturally to s colors. Two players, Builder and Painter, play a game with a target graph G . During each round, Builder presents a new edge uv to Painter (the endpoints may be vertices not yet used). Painter must color uv red or blue. Builder wins if a monochromatic copy of G arises. We say that Builder can *force* G if Builder has a strategy to win.

When Builder’s moves are unrestricted, Builder wins by playing a large complete graph (by Ramsey’s Theorem). As in parameter Ramsey theory, the problem becomes more interesting when Builder must keep the presented graph in a class \mathcal{H} . This defines the *on-line Ramsey game* (G, \mathcal{H}) . Given G and \mathcal{H} , which player has a winning strategy?

We say that (G, \mathcal{H}) is played *on* \mathcal{H} . Grytczuk, Hałuszczak, and Kierstead [12] showed that Builder wins on the class of k -colorable graphs when G is k -colorable. Also, Builder wins on the class of forests when G is a forest. For $G = K_3$, Painter wins on outerplanar graphs but Builder wins on planar 2-degenerate graphs. On planar graphs, Builder wins when G is a cycle or is a 4-cycle plus one chord (a slight extension is that Builder can force any fixed cycle plus chords at any one vertex). They conjectured that on planar graphs, Builder wins if and only if G is outerplanar. Petříčková [20] disproved this by showing that Builder can force $K_{2,3}$ on the class of planar graphs.

For any graph parameter ρ , we define the *on-line ρ -Ramsey number* \mathring{R}_ρ of G to be the least k such that Builder can force G when playing on the family $\{H : \rho(H) \leq k\}$. The main result of [12] is that $\mathring{R}_\chi(G) = \chi(G)$ for every graph.

The notation $\tilde{r}(G)$ has been used for the on-line size Ramsey number. Grytczuk, Kierstead, and Prałat [13] proved that $\tilde{r}(P_n) \leq 4n - 7$ for $n \geq 2$ (they found the exact values for $n \leq 6$; see [21, 22, 23] for additional exact results on \tilde{r}). They also proved $\tilde{r}(G) \geq \frac{1}{2}b(D - 1) + m$ when G has m edges, maximum degree D , and vertex cover number b . Using the latter, they proved that the maximum of $\tilde{r}(G)$ over trees with m edges is $\Theta(m^2)$. It is conjectured that $\tilde{r}(K_n)/R(K_n) \rightarrow 0$ as $n \rightarrow \infty$ (see Conlon [6]). Kierstead and Konjevod [16] studied an extension of on-line Ramsey games to s -uniform hypergraphs. A variant of on-line Ramsey games in which Builder is replaced with a

sequence of random edges is studied in [11].

In this paper, we study the *on-line degree Ramsey number* $\mathring{R}_\Delta(G)$. Let \mathcal{S}_k be the class of graphs with maximum degree at most k ; $\mathring{R}_\Delta(G)$ is the least k such that Builder wins (G, \mathcal{S}_k) . Since the on-line model gives Builder more power, always $\mathring{R}_\rho(G; s) \leq R_\rho(G; s)$; we will compare results on these two parameters.

Our easy Theorem 2.2 is $\mathring{R}_\Delta(G) \geq \Delta(G) - 1 + \max_{uv \in E(G)} \min\{d_G(u), d_G(v)\}$, where $d_G(v)$ is the degree of vertex v in G . This yields optimal lower bounds for various small graphs, as noted later. When G has adjacent vertices of maximum degree, the lower bound becomes $\mathring{R}_\Delta(G) \geq 2\Delta(G) - 1$, which proves sharpness of our general upper bound for a tree G in terms of the maximum degree: $\mathring{R}_\Delta(G) \leq 2\Delta(G) - 1$. This upper bound argument does not extend to the multicolor setting. Nevertheless, using other techniques, Kinnersley [17] has proved that $\mathring{R}_\Delta(G; s) \leq s\Delta(G) - (s - 1)$ when G is a tree, with equality when G has adjacent vertices of maximum degree. He also obtained results on the “non-diagonal” case where one seeks a copy of tree G_i in color i for some i .

For $s = 2$, we prove a stronger upper bound for trees: $\mathring{R}_\Delta(G) \leq d_1 + d_2 - 1$, where d_1, \dots, d_n is the nonincreasing list of the vertex degrees (Theorem 3.1). Here Builder’s strategy makes no cycles, thus yielding also the result of [12] on forests; their proof is shorter but does not give a good upper bound on the maximum degree used by Builder.

For cycles, we prove that $4 \leq \mathring{R}_\Delta(C_n) \leq 5$ for all n (Sections 4 and 5). In fact, the value is 4 for all n . Here we prove $\mathring{R}_\Delta(C_n) = 4$ when n is even or equals 3. Our methods also prove $\mathring{R}_\Delta(C_n) = 4$ for odd n between 337 and 514 or at least 689. Rolnick [25] extended our methods to complete the proof that $\mathring{R}_\Delta(C_n) = 4$ for all n , so we omit the details of our final construction for odd cycles.

We include the techniques developed for cycles because we use them to obtain upper bounds for all graphs with maximum degree 2. A major open question is whether there exists a function f such that $\mathring{R}_\Delta(G) \leq f(k)$ when $\Delta(G) \leq k$. We prove that $f(2)$ exists; in fact $f(2) \leq 6$ (Theorem 5.6). We do not know whether this bound is sharp.

Our lower bounds rely on “greedy” strategies for Painter, in which Painter makes an edge red if and only if it keeps the red graph within a specified class, such as \mathcal{S}_k or the class of linear forests (a *linear forest* is a graph whose components are all paths). Greedy Painters of both types are used to prove that $\mathring{R}_\Delta(G) \leq 3$ if and only if G is a linear forest or each component of G is a subgraph of the claw $K_{1,3}$ (Theorem 2.5). Thus when Builder wins (G_1, \mathcal{H}) and (G_2, \mathcal{H}) , it does not follow that Builder wins $(G_1 + G_2, \mathcal{H})$; for example, $\mathring{R}_\Delta(P_4) = \mathring{R}_\Delta(K_{1,3}) = 3$, but $\mathring{R}_\Delta(P_4 + K_{1,3}) > 3$.

Upper bounds require strategies for Builder. Optimal strategies for paths, stars, and triangles need only induction and the pigeonhole principle. For other graphs, we simplify Builder’s task. We prove (Theorem 2.11) that Builder may assume that Painter plays “consistently”, meaning that the color Painter assigns to an edge depends only on the components of the current edge-colored graph containing its endpoints. This reduction applies to the Ramsey game (G, \mathcal{H}) whenever \mathcal{H} is monotone (all subgraphs of graphs in \mathcal{H} lie in \mathcal{H}) and additive (disjoint unions of graphs in \mathcal{H} lie in \mathcal{H}). Hence the “Consistent Painter Theorem” may be useful for on-line Ramsey problems other than $\mathring{R}_\Delta(G)$.

2 Tools: Greedy Painter and Consistent Painter

This section presents techniques for proving bounds on $\mathring{R}_\Delta(G)$. Upper bounds arise from strategies for Builder; lower bounds from strategies for Painter. Although lower bounds are usually more difficult, we begin with a class of strategies for Painter.

Definition 2.1. Let \mathcal{F} be a family of graphs. The *greedy \mathcal{F} -Painter* colors each new edge red if the resulting red graph lies in \mathcal{F} ; otherwise, the edge is colored blue.

When $k = \Delta(G) - 1$, the greedy \mathcal{S}_k -Painter establishes a useful general lower bound.

Theorem 2.2. For every graph G , $\mathring{R}_\Delta(G) \geq \Delta(G) - 1 + \max_{uv \in E(G)} \min\{d(u), d(v)\}$.

Proof. Let $k = \Delta(G) - 1$, and let $t = \max_{uv \in E(G)} \min\{d(u), d(v)\}$. The greedy \mathcal{S}_k -Painter never makes a red G , because no vertex ever has $\Delta(G)$ incident red edges. In a blue G , some edge represents an edge $xy \in E(G)$ such that $\min\{d_G(x), d_G(y)\} = t$. Making this edge blue requires k red edges already at at least one endpoint. Within the blue G , each endpoint has at least t edges. Hence at x or y at least $k + t$ edges have been played. \square

Theorem 2.2 yields $\mathring{R}_\Delta(G) \geq 2\Delta(G) - 1$ when G has adjacent vertices of maximum degree. Furthermore, $\mathring{R}_\Delta(G) \geq d_1 + d_2 - 1$ when vertices whose degrees are the first two edges in the nonincreasing degree list d are adjacent. Thus the upper bound for trees in Theorem 3.1 is sharp. The greedy \mathcal{S}_k -Painter also easily yields the lower bound in [13] for the on-line size Ramsey number ($\tilde{r}(G) \geq \frac{1}{2}b(D - 1) + m$ when G has m edges, maximum degree D , and vertex cover number b).

Induction and the pigeonhole principle lead to optimality of the lower bound in Theorem 2.2 for paths and stars. If Builder can force a monochromatic G , then by the pigeonhole principle Builder can force t monochromatic copies of G in the same color.

Corollary 2.3. $\mathring{R}_\Delta(K_{1,m}) = m$ and $\mathring{R}_\Delta(P_k) = 3$ (for $k \geq 4$).

Proof. The lower bounds follow from Theorem 2.2; the upper use strategies for Builder.

For $K_{1,m}$, use induction on m ; trivially $\mathring{R}_\Delta(K_{1,1}) = 1$. For $m > 1$, Builder first plays on \mathcal{S}_{m-1} to force m disjoint copies of $K_{1,m-1}$ in the same color. Builder then plays a star $K_{1,m}$ whose leaves are their centers. The resulting graph is in \mathcal{S}_m , and $K_{1,m}$ is forced.

For P_k , we prove by induction on k that Builder playing on \mathcal{S}_3 can force a monochromatic path with at least k vertices such that no other edges have been played at the endpoints; the single edge has this property for $k = 2$. For $k \geq 3$, Builder first plays on \mathcal{S}_3 to force $k - 2$ such paths of the same color having at least $k - 1$ vertices each; let “red” be this color. Builder then plays a path Q with k vertices using one endpoint of each of these paths plus two new vertices as the endpoints of Q . If any edge of Q is red, then the desired path arises in red; if they are all blue, then Q becomes the desired path. \square

For comparison with degree Ramsey number, note that $R_\Delta(P_k) = 4$ for $k \geq 7$ ([26] and [1] combined), while $R_\Delta(K_{1,m})$ equals $2m - 1$ for odd m and $2m - 2$ for even m [5].

Greedy Painters also enable us to characterize the graphs G such that $\mathring{R}_\Delta(G) \leq 3$. It is trivial that $\mathring{R}_\Delta(G) \leq 1$ if and only if G is a matching. Also $\mathring{R}_\Delta(G) \leq 2$ is easily characterized.

Proposition 2.4. $\mathring{R}_\Delta(G) \leq 2$ if and only if each component of G is a subgraph of P_3 .

Proof. Builder forces any such graph by presenting enough disjoint triangles. For necessity, if $\mathring{R}_\Delta(G) \leq 2$, then $\Delta(G) \leq 2$. By Theorem 2.2, the vertices with degree 2 are nonadjacent. Hence each component of G is a subgraph of P_3 . \square

Theorem 2.5. $\mathring{R}_\Delta(G) \leq 3$ if and only if each component of G is a path or each component of G is a subgraph of the claw $K_{1,3}$.

Proof. By Corollary 2.3, Builder can force the claw or any path on \mathcal{S}_3 . By the pigeonhole principle, Builder can thus force any disjoint union of subgraphs of $K_{1,3}$. Also Builder can force a path long enough to contain any specified disjoint union of paths.

For necessity, suppose that $\mathring{R}_\Delta(G) \leq 3$, so Builder can force G on \mathcal{S}_3 . Consider a greedy \mathcal{L} -Painter, where \mathcal{L} is the family of linear forests (disjoint unions of paths). If G appears in red, then each component of G is a path.

Suppose that G appears in blue. A vertex v with degree 3 in G has three incident blue edges and hence no incident red edges, since Builder is playing on \mathcal{S}_3 . For the greedy \mathcal{L} -Painter to make these edges blue, each neighbor already has two incident red edges. Hence a blue claw must be a full component of the blue graph.

It remains to show that Builder cannot force a monochromatic graph containing both P_4 and $K_{1,3}$ in \mathcal{S}_3 . Since such a graph G has maximum degree at least 3 and has an edge with both endpoints having degree at least 2, Theorem 2.2 implies that $\mathring{R}_\Delta(G) \geq 4$. \square

Theorem 2.5 shows that Builder can force P_4 or $K_{1,3}$ in \mathcal{S}_3 but not their disjoint union. The family of graphs that Builder can force when playing on a given family is not always closed under disjoint union.

We proved these results by using the pigeonhole principle to force many monochromatic graphs. Pigeonholing also applies to 2-edge-colored graphs. When Builder presents m edges, Painter can produce at most 2^m distinguishable 2-edge-colored graphs. By presenting isomorphic copies of the graph formed by these m edges, Builder can force many copies of some single pattern. Nevertheless, when strategies become more complicated and repeated copies of larger patterns are needed, use of the pigeonholing argument becomes unwieldy. Arguments simplify if Builder can assume that Painter plays “consistently”.

Definition 2.6. A Painter strategy is *consistent* if the color Painter chooses for an edge uv depends only on the 2-edge-colored component(s) containing u and v when uv arrives.

For example, a consistent Painter always colors an isolated triangle in the same way. If there are nonisomorphic ways to order the edges of a graph (such as K_4), then a consistent Painter may produce different colorings depending the order in which the edges arrive.

Our aim is to reduce the problem of proving that Builder wins to proving that Builder wins against consistent Painters. The argument can be given for the s -color model, but we state it only for two colors, red and blue. We need several technical notions.

Definition 2.7. Given a monotone additive family \mathcal{H} , an \mathcal{H} -strategy specifies a color for each pair (H, e) such that H is a 2-edge-coloring of a graph in \mathcal{H} and e is an edge not in H (either or both endpoints of e may be new vertices). An \mathcal{H} -list is an ordering of the edges of some graph in \mathcal{H} ; every initial segment of an \mathcal{H} -list forms a graph in \mathcal{H} . For each \mathcal{H} -list E and each \mathcal{H} -strategy \mathcal{A} , let $\mathcal{A}(E)$ denote the edge-colored graph that results when Builder presents E to \mathcal{A} . An edge-colored graph F contains another such graph F' if there is an injection of $V(F')$ into $V(F)$ that preserves edges and preserves their colors.

To reduce the Builder problem to winning against consistent Painters, we will show that for every Painter strategy there is a consistent strategy that does at least as well for Painter. That is, when \mathcal{A} is an \mathcal{H} -strategy, there is a consistent \mathcal{H} -strategy \mathcal{A}' such that any 2-edge-colored graph Builder can force against \mathcal{A}' can also be forced against \mathcal{A} . A special set of 2-edge-colored graphs will enable us to produce \mathcal{A}' .

Definition 2.8. A uv -augmentation of a 2-edge-colored graph H with nonadjacent vertices u and v is obtained by adding uv to H with color red or blue. For a monotone additive family \mathcal{H} , a class \mathcal{C} of connected 2-edge-colored graphs is \mathcal{H} -coherent if it contains K_1 and satisfies the following *augmentation property*: If H is a 2-edge-colored copy of a graph H' in \mathcal{H} , and H' has nonadjacent vertices u and v such that $H' + uv$ is a connected graph in \mathcal{H} and the component(s) of H are in \mathcal{C} , then \mathcal{C} contains a uv -augmentation of H .

An \mathcal{H} -coherent class \mathcal{C} yields a consistent \mathcal{H} -strategy \mathcal{A}' as follows. When an edge uv is added to the current 2-colored graph H , \mathcal{A}' consults \mathcal{C} to find which color on uv yields a uv -augmentation in \mathcal{C} for the component(s) of H containing the endpoints of the added edge. When both colors yield uv -augmentations, \mathcal{A}' always makes the same choice.

Definition 2.9. Let \mathbf{C} be the class of connected 2-edge-colored (unlabeled) graphs; every 2-edge-coloring of a graph in \mathcal{H} is a multiset of elements of \mathbf{C} having finitely many distinguishable components, each with finite multiplicity. Given an \mathcal{H} -strategy \mathcal{A} , a 2-edge-colored graph H is \mathcal{A} -realizable if for some \mathcal{H} -list E , the outcome $\mathcal{A}(E)$ contains H . A family $\mathcal{C} \subseteq \mathbf{C}$ is \mathcal{A} -plentiful if, for every finite subset $C \subseteq \mathcal{C}$ and every positive integer n , the 2-edge-colored graph consisting of n components isomorphic to each element of C is \mathcal{A} -realizable.

In order to be \mathcal{H} -coherent for some monotone additive family \mathcal{H} , a family \mathcal{C} contained in \mathbf{C} must somehow be “large enough”. In order to be \mathcal{A} -plentiful for some \mathcal{H} -strategy \mathcal{A} , the family \mathcal{C} must somehow be “small enough”. We seek a family achieving both properties. We use Zorn’s Lemma in the following form: if every chain in a partial order P has an upper bound, then P has a maximal element.

Lemma 2.10. *If \mathcal{H} is a monotone additive family of graphs, and \mathcal{A} is an \mathcal{H} -strategy, then some family \mathcal{C} is both \mathcal{H} -coherent and \mathcal{A} -plentiful.*

Proof. Note first that $\{K_1\}$ is \mathcal{A} -plentiful. Also, if $\mathcal{C}_1, \mathcal{C}_2, \dots$ are \mathcal{A} -plentiful families with $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \dots$, then the union of these families is also \mathcal{A} -plentiful, because the definition of \mathcal{A} -plentiful requires \mathcal{A} -realizability only of repeated copies of *finite* subsets, and each finite subset appears in some \mathcal{C}_j . It follows from Zorn's Lemma that there is a maximal \mathcal{A} -plentiful family \mathcal{C} containing K_1 . We claim that \mathcal{C} is \mathcal{H} -coherent.

By construction, $K_1 \in \mathcal{C}$. For a fixed 2-edge-colored graph H in \mathcal{C} , let H' be its underlying graph in \mathcal{H} . Let u and v be nonadjacent vertices in H' with $H' + uv \in \mathcal{H}$. Let H_1 and H_2 be the possible uv -augmentations of H (using red or blue on uv). If neither lies in \mathcal{C} , then neither $\mathcal{C} \cup \{H_1\}$ nor $\mathcal{C} \cup \{H_2\}$ is \mathcal{A} -plentiful, by the maximality of \mathcal{C} . Hence there are positive integers t_1 and t_2 and finite sets $C_1, C_2 \subseteq \mathcal{C}$ such that the 2-edge-colored graphs $t_1(C_1 \cup \{H_1\})$ and $t_2(C_2 \cup \{H_2\})$ are not \mathcal{A} -realizable, where for $C \subseteq \mathcal{C}$ we use qC to denote the 2-edge-colored graph with q copies of each element of C as components.

Let $D = C_1 \cup C_2 \cup \{H\}$. Since D is a finite subset of \mathcal{C} , and \mathcal{C} is \mathcal{A} -plentiful, $2(t_1 + t_2 - 1)D$ is \mathcal{A} -realizable via some \mathcal{H} -list E . When E is presented, $\mathcal{A}(E)$ contains at least $2(t_1 + t_2 - 1)$ disjoint copies of H .

Since \mathcal{H} is additive, the list E' formed by adding to E the copies of uv in $2(t_1 + t_2 - 1)$ components isomorphic to H' is an \mathcal{H} -list; Builder may legally present these edges after E . Consider the first $t_1 + t_2 - 1$ of the added edges. Either \mathcal{A} colors at least t_1 of them red, or \mathcal{A} colors at least t_2 of them blue. In the first case, $t_1(C_1 \cup \{H_1\})$ is \mathcal{A} -realizable: we have obtained t_1 copies of H_1 , and at least $t_1 + t_2 - 1$ copies of H remain (needed if $H \in C_1$). In the second case, $t_2(C_2 \cup \{H_2\})$ is similarly \mathcal{A} -realizable. The contradiction implies that \mathcal{C} contains a uv -augmentation of H .

Essentially the same argument shows that \mathcal{C} contains a uv -augmentation of H when H consists of two 2-edge-colored components, each containing one of u and v . \square

Theorem 2.11. *If \mathcal{H} is a monotone additive family of graphs, and \mathcal{A} is an \mathcal{H} -strategy for Painter, then there is a consistent \mathcal{H} -strategy \mathcal{A}' such that for every \mathcal{H} -list E' , there is an \mathcal{H} -list E such that $\mathcal{A}(E) \supseteq \mathcal{A}'(E')$. That is, Builder can force against \mathcal{A} any monochromatic target that Builder can force against \mathcal{A}' .*

Proof. By Lemma 2.10, there is an \mathcal{H} -coherent, \mathcal{A} -plentiful family $\mathcal{C} \subseteq \mathbf{C}$. As described after Definition 2.8, from the \mathcal{H} -coherence of \mathcal{C} we define a consistent \mathcal{H} -strategy \mathcal{A}' .

When \mathcal{A}' is given a new edge uv , the definition of \mathcal{C} being \mathcal{H} -coherent implies that the uv -augmentation chosen by \mathcal{A}' for the component being formed is in \mathcal{C} . Thus every component of $\mathcal{A}'(E')$ is in \mathcal{C} . Since \mathcal{C} is \mathcal{A} -plentiful, it follows that $\mathcal{A}'(E') \subseteq \mathcal{A}(E)$ for some \mathcal{H} -list E . \square

To show that Builder wins (G, \mathcal{H}) , it now suffices to show that Builder can force a monochromatic G against any consistent \mathcal{H} -strategy for Painter. In particular, if some \mathcal{H} -list results in a particular 2-edge-colored component, then Builder can recreate another copy of that component by playing an isomorphic list of edges on a new set of vertices.

Theorem 2.11 applies to all monotone additive families, not just \mathcal{S}_k . For example, to prove sufficiency in the conjecture in [12], one need only show that Builder can force any outerplanar graph when playing on planar graphs against a consistent Painter.

3 Trees

We apply Theorem 2.11 first to prove an upper bound on $\mathring{R}_\Delta(G)$ when G is a tree. When G is a tree, our main result is that $\mathring{R}_\Delta(G) \leq d_1 + d_2 - 1$, where d_1 and d_2 are two largest entries in the list of vertex degrees. When there is an edge joining vertices of degrees d_1 and d_2 , Theorem 2.2 yields $\mathring{R}_\Delta(G) \geq d_1 - 1 + d_2$ (whether or not G is a tree). In particular, for the double-star $S_{a,b}$ (the tree with $a + b$ vertices having vertices of degrees a and b), we have $\mathring{R}_\Delta(S_{a,b}) = a + b - 1$. (In comparison, $R_\Delta(S_{a,b}) = 2 \max\{a, b\} - 2$, plus 1 when a and b are equal and odd [18]; for general trees, $R_\Delta(G) \leq 4\Delta(G) - 4$ [15].)

Theorem 3.1. *Let G be any n -vertex tree. If the vertex degrees are d_1, \dots, d_n in nonincreasing order, then $\mathring{R}_\Delta(G) \leq d_1 + d_2 - 1$. Equality holds when G has adjacent vertices of degrees d_1 and d_2 .*

Proof. Under the condition given for equality, Theorem 2.2 provides the lower bound. We prove that Builder can force (against a consistent Painter) a monochromatic rooted tree in which the root has d_1 children, all other non-leaves have $d_2 - 1$ children, and all leaves have distance more than l from the root, where $l = \text{diam}(G)$. Such a tree contains a monochromatic G . At any point in the game, let H be the graph that has been presented so far. In fact, H will be a forest.

Builder maintains candidate trees T_R and T_B with edges in red and blue, respectively. Initially, these trees consist only of their root vertices. Moreover, Builder keeps T_R and T_B in different components of H . A vertex of T_R or T_B is *satisfied* when it has the desired number of children in that tree (d_1 for the root, $d_2 - 1$ for others). Each tree has an *active vertex*, x_R or x_B respectively, which is an unsatisfied vertex of least depth. A *bad edge* is an edge incident to T_R or T_B having the color of the other tree. The active vertex becomes *dangerous* when it has $d_2 - 1$ incident bad edges.

If the two active vertices are not both dangerous, then Builder plays an edge joining a non-dangerous active vertex to a new vertex. If the new edge has the color of that tree, then it enters the tree; otherwise, it is an additional bad edge at the active vertex.

Builder plays pendant edges at active vertices until an active vertex becomes satisfied or both active vertices become dangerous. When an active vertex becomes satisfied, a new active vertex is chosen in that tree from the unsatisfied vertices of least depth.

When both active vertices are dangerous, Builder plays $x_R x_B$. Since T_R and T_B are in different components H_R and H_B , still H is a forest. If Painter colors $x_R x_B$ red, then Builder makes x_B a child of x_R in T_R . Because x_B was active and dangerous, x_B is already incident to $d_2 - 1$ red edges; its neighbors along these edges become its children in T_R , and so it is satisfied. Possibly x_R is now satisfied, in which case a new active vertex is chosen for T_R .

What we previously called T_B now lies inside the component of H containing T_R . Before continuing, Builder regenerates T_B in a new component of H . Builder plays edges on new vertices isomorphic to the list that produced H_B ; since Painter is consistent, the resulting edge-colored graph is isomorphic to H_B , yielding a new copy of T_B with new active vertex x_B . (See Figure 1, where solid edges are red and dashed edges are blue.)

The process now continues, with T_R having been augmented by the old x_B and its children and T_B having been regenerated; the new copy of x_B is dangerous. (Of course, if Painter had colored $x_R x_B$ blue, the roles of red and blue would be interchanged in Builder's response.)

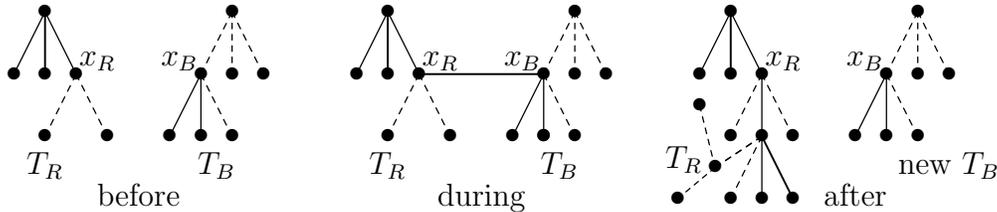


Figure 1: Strategy for trees with $d_1 = d_2 = 3$ when x_R and x_B are dangerous

At each move, the edge played by Builder is incident to both active vertices or to one active vertex and a new vertex. When a bad edge is added at an active vertex, the other endpoint is given degree 1. Hence when one active vertex y becomes a child of the active vertex in the other tree, the vertices that now become children of y in that tree are leaves. We conclude that every non-leaf vertex other than the active vertex is satisfied.

Hence when a vertex becomes active, it has degree 1 in H . While active, it receives at most $d_2 - 1$ bad edges, and it can receive only enough incident edges in the color of its tree to make it satisfied. Hence its degree remains at most $d_1 + d_2 - 1$ (it takes d_1 children to satisfy a root). After becoming satisfied, a vertex receives no more incident edges.

There is another way to become satisfied. When $x_R x_B$ is played and colored red, the $d_2 - 1$ edges that were bad at x_B now satisfy it in T_R . The tree T_B remains attached to x_B , but since x_B was not satisfied in T_B it had at most $d_1 - 1$ incident blue edges. Again its degree is at most $d_1 + d_2 - 1$, and it receives no more incident edges.

Finally, we must argue that this strategy forces a monochromatic tree (T_R or T_B) in which all vertices having distance at most l from the root are satisfied. When x_R and x_B are both dangerous, one receives a good incident edge and the other is recreated by regenerating its component. Hence eventually one of them becomes satisfied. Since x_R and x_B are unsatisfied vertices of least depth, we already have the desired monochromatic tree if one of them has distance more than l from its root. Otherwise, we increase the number of satisfied vertices having distance at most l from the root in one tree or the other. The number of possible such vertices is bounded by $2d_1 \sum_{i=1}^l d_2^{i-1}$, so eventually Builder forces the desired tree. \square

In the strategy in Theorem 3.1, H remains a forest in \mathcal{S}_k , where $k = d_1 + d_2 - 1$. We conclude that Builder can force any forest when playing on the family of forests. This statement was proved more simply in [12], but their proof did not provide a good bound on the maximum degree used. On the other hand, the proof in [12] extends easily to the s -color setting. Theorem 3.1 uses the fact that active vertices in each of two colors can be made adjacent by a single edge; our proof does not extend to the multicolor setting.

4 Weighted Graphs and Even Cycles

Strategies for Builder playing on \mathcal{S}_k often involve keeping track of how many edges have been played incident to each vertex. The argument we gave for paths (Corollary 2.3) had this flavor; to facilitate the induction we needed to maintain degree 1 at the leaves of the monochromatic path, but we could allow degree 3 at internal vertices.

Definition 4.1. A c -weighted graph is a graph G equipped with a nonnegative integer capacity function c on $V(G)$. A copy of a c -weighted graph G exists in a graph H if G embeds as a subgraph of H via an injection f such that $d_H(f(v)) \leq c(v)$ for all $v \in V(G)$.

When the capacity function is constant, say $c(v) = k$ for all $v \in V(G)$, we simply refer to the c -weighted graph G as a k -weighted graph. The statement that Builder wins (G, \mathcal{S}_k) is equivalent to the statement that Builder can force the k -weighted graph G when playing on the unrestricted family of all graphs. Vertices that acquire more than k incident edges are forbidden from the desired monochromatic copy of G when Builder is restricted to \mathcal{S}_k .

We use weighted graphs primarily to discuss $\mathring{R}_\Delta(C_n)$. When the capacity function is not constant, we list its values in some fixed order to specify a weighted graph. In particular, an (a, b, c, d) -claw is a monochromatic weighted claw with capacity a at the center and capacities b, c, d at the leaves. Similarly, an (a, b, c) -triangle is a monochromatic weighted triangle with capacities a, b, c at the vertices.

Proposition 4.2. $\mathring{R}_\Delta(C_3) = 4$.

Proof. For the lower bound, note that C_3 does not satisfy the characterization of graphs with $\mathring{R}_\Delta(G) \leq 3$ in Theorem 2.5.

For the upper bound, Builder seeks a $(4, 4, 4)$ -triangle. If Builder obtains a $(4, 2, 2, 2)$ -claw, then Builder wins by presenting a triangle on its leaves.

Builder first presents a claw with center u , winning if Painter makes it monochromatic. Otherwise, we may assume that the claw has one blue edge ux and red edges uy and uz .

Now Builder presents a claw C with center x and three new vertices as leaves. If monochromatic, C is a $(4, 2, 2, 2)$ -claw and Builder wins. Otherwise, C has a blue edge xw .

Now Builder presents uw ; note that u now has degree 4 and w has degree 2. Coloring uw blue completes a blue triangle on $\{u, x, w\}$. Otherwise, there is now a red $(4, 2, 2, 2)$ -claw with center u and leaves w, y, z . \square

Such detailed analysis of weighted graphs forced by Builder can yield the answers for other small graphs. For the graph $K_{1,3}^+$ obtained by adding one edge to $K_{1,3}$, Theorem 2.2 yields $\mathring{R}_\Delta(K_{1,3}^+) \geq 4$. Builder guarantees $\mathring{R}_\Delta(K_{1,3}^+) \leq 5$ by first forcing three disjoint monochromatic $(5, 1, 1, 1)$ -claws in the same color, then playing a triangle on the leaves of each, then playing a triangle using one vertex from each of those triangles. Using a more detailed analysis, Rolnick [25] proved that $\mathring{R}_\Delta(K_{1,3}^+) = 4$.

Similarly, for the graph C_4^+ obtained by adding one chord to a 4-cycle, Theorem 2.2 yields $\mathring{R}_\Delta(C_4^+) \geq 5$. A Builder strategy that starts by forcing a 4-weighted monochromatic copy of $K_{1,4}$ can be used to show that $\mathring{R}_\Delta(C_4^+) \leq 7$. The exact value is not known.

Next we apply the consistent Painter and weighted graphs to determine $\mathring{R}_\Delta(C_n)$ when n is even. When Builder forces a monochromatic graph against a consistent Painter, it always arises in the same color each time Builder forces it on new vertices.

Lemma 4.3. *Let F_1 and F_2 be weighted graphs Builder can force in red against a consistent Painter. For $i \in \{1, 2\}$, let u_i be a vertex of F_i with capacity c_i . Form F from $F_1 + F_2$ by adding u_1u_2 and changing the capacities of u_1 and u_2 to $c_1 + 2$ and $c_2 + 2$. If $c_1, c_2 \leq t - 2$ and n is even, then Builder can force a red F or a blue t -weighted n -cycle.*

Proof. Builder forces $n/2$ copies of both F_1 and F_2 in red. Builder then plays an n -cycle on the copies of u_1 and u_2 , alternating between copies of u_1 and copies of u_2 . If these edges are all blue, then they form a blue t -weighted n -cycle. Otherwise, a red F arises. \square

Theorem 4.4. *If n is even, then $\mathring{R}_\Delta(C_n) = 4$.*

Proof. By Theorem 2.5, $\mathring{R}_\Delta(C_n) > 3$; consider the upper bound. A consistent Painter always gives an isolated triangle the same coloring. It has a monochromatic 2-weighted P_3 , which we may assume by symmetry is red.

Let $p = n/2$. Suppose that Builder can force a red tree T , with degree 2 at the leaves and degree 4 at non-leaves, such that T has p -sets L and L' of leaves with the distance in T between leaves chosen from L and L' being $n - 1$. Builder can then play an n -cycle alternating between L and L' . If these edges are not all blue, then a red n -cycle arises. It therefore suffices to force such a red tree when Painter avoids making a blue C_n .

Let F_1 and F_2 both equal the 2-weighted P_3 , with centers u_1 and u_2 . By Lemma 4.3, regardless of p , Builder can force in red the weighted tree T_2 obtained by adding u_1u_2 and giving u_1 and u_2 capacity 4. This is the desired tree for $p = 2$ (see Figure 2).

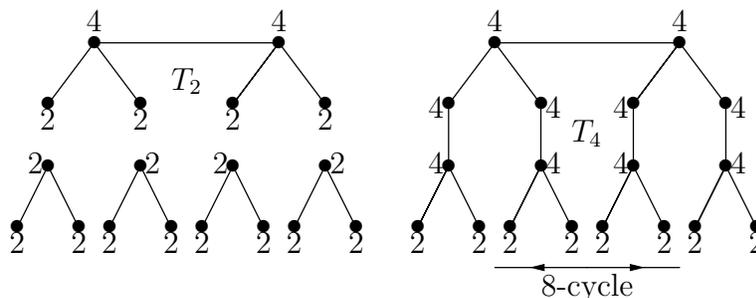


Figure 2: Trees used to force even cycles

Four more applications of Lemma 4.3 yield in red the weighted tree T_4 obtained by adding edges from the leaves of T_2 to the centers of copies of P_3 , with capacity 2 at the leaves and capacity 4 otherwise. This is the desired tree for $p = 4$ (see Figure 2).

Repeating this process doubles the number of leaves as p increases by 2, generating sufficiently many leaves to obtain L and L' in the tree T_p when p is even. Also, in T_6 there are eight vertices available on each side; as desired, the distance between leaves in the two sets is 11. Applying the same operation to expand only at L and not at L' yields leaf sets

of sizes 16 and 8 with the distance between leaves from the two sets being 7. Choosing seven leaves from each set yields T_7 . Repeating the expansion process at all leaves now generates T_p for all larger odd p .

Unfortunately, this process does not make enough leaves in each of two sets to form T_3 or T_5 . Hence we force a monochromatic C_6 or C_{10} in a different way. Returning to the isolated triangles, we consider two cases. In the figure, the solid straight edges are red.

Case 1: *Painter makes isolated triangles all red.* Builder plays four isolated triangles; these become all red, with all vertices having degree 2. Builder now uses Lemma 4.3 to force a red matching joining one triangle to vertices in the other three; the matched vertices now have degree 4. The distance between any two nonadjacent vertices of degree 2 is now 5. By playing a 6-cycle on these vertices, Builder forces a monochromatic C_6 .

Case 2: *Painter makes isolated triangles majority red.* Again Builder plays four triangles, with degree 2 at each vertex. In the figure, blue edges are bold and dashed (the dashed ones are not needed). Again Builder forces the red matching, using the center of each red P_3 . Now the distance between vertices remaining with degree 2 is 5 or 6. Instead of a 6-cycle, Builder plays only four additional edges as shown in Figure 3, using two existing blue edges to help force a monochromatic C_6 whether the four new edges are all blue or any red.

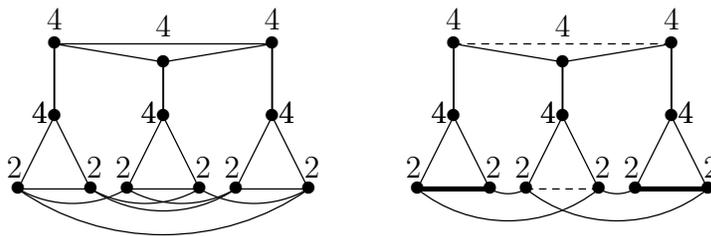


Figure 3: Forcing C_6

A similar strategy forces C_{10} , again with two cases by whether the consistent Painter makes isolated triangles monochromatic (see Figure 4). In each case, we may assume that the majority color on the triangles is red. As with C_6 , we force the desired cycle or a “tree” of triangles by repeatedly applying Lemma 4.3. The tree is deeper by one level of triangles in order to add 4 to the length of the paths that may lead to red cycles. When the isolated triangles are not monochromatic, again the leftmost and rightmost “subtrees” are too far apart in red, so we again use two known blue edges and add just eight edges joining the middle third to the outside thirds. Making any such edge red completes a red 10-cycle, and making all eight of them blue completes a blue 10-cycle. \square

In contrast, $R_\Delta(C_3) = R_\Delta(C_4) = 5$ [18]. For even cycles, we have $\mathring{R}_\Delta(C_n) = 4$, but the best known general upper bound for $R_\Delta(C_n)$ is 96 [15]. Also, $R_\Delta(C_n) \leq 3458$ for all odd n [15]. In the next section, we discuss the exact value of $\mathring{R}_\Delta(C_n)$ for odd n .

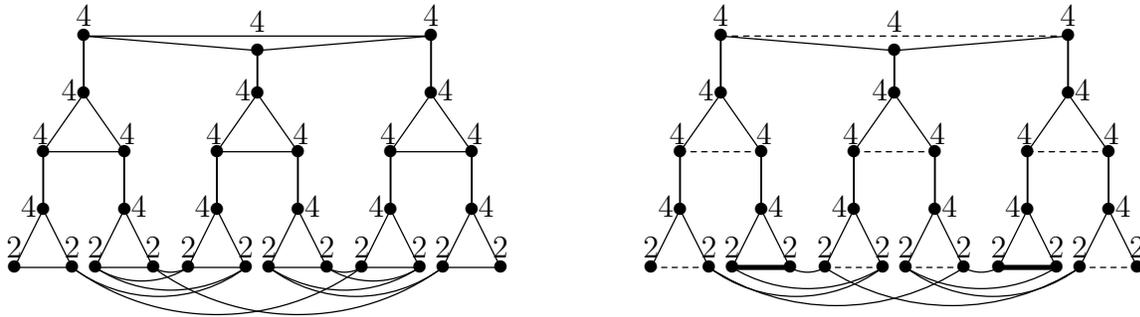


Figure 4: Forcing C_{10}

5 Odd Cycles and Graphs with Maximum Degree 2

From Theorem 2.5, it follows that $\mathring{R}_\Delta(C_n) \geq 4$ for each n . So far, we have proved that $\mathring{R}_\Delta(C_n) = 4$ when n is even or $n = 3$. Here, we prove that $\mathring{R}_\Delta(C_n) \leq 5$ for all n and discuss the techniques used to prove $\mathring{R}_\Delta(C_n) = 4$ also when n is odd.

After discussing odd cycles, we close the section by using these techniques to prove that $\mathring{R}_\Delta(G) \leq 6$ whenever $\Delta(G) \leq 2$. A graph with maximum degree 2 is a disjoint union of paths and cycles, but we have seen that the ability of Builder to force G and G' when playing on \mathcal{S}_k does not imply that Builder can force $G + G'$ on \mathcal{S}_k .

As usual, we may assume that Builder plays against a consistent Painter. The results in this section can be proved without this assumption, but with it the arguments are marginally shorter, and using the consistent Painter keeps our phrasing consistent.

We begin with two lemmas that describe how Builder can extend given strategies to force larger structures; these are similar to Lemma 4.3, where we could add any edge joining two graphs that can be forced in red. When the alternative cycle in blue is odd rather than even, we have weaker results that require either 1) using the same vertex in two copies of the same graph, or 2) restricting one of the graphs to be a single vertex. Lemma 5.1 holds also for even n , but when n is even Lemma 4.3 is much stronger.

Lemma 5.1. *Against a consistent Painter, let F be a weighted graph Builder can force in red, with $u \in V(F)$ having capacity c . Form F' from $F + F$ by adding an edge joining the two copies of u and changing the capacity at its endpoints to $c + 2$. If n is odd, then Builder can force a red F' or a blue t -weighted n -cycle, where $t = c + 2$.*

Proof. Builder forces n copies of F in red and then plays an n -cycle on the copies of u . Painter must produce a blue t -weighted n -cycle or a red F' . \square

The full generality of the next lemma, with $r \neq b$, will be used in Theorem 5.6.

Lemma 5.2. *Against a consistent Painter, let F be a weighted graph Builder can force in red, with $u \in V(F)$ having capacity c . Let F' be the weighted graph obtained from F by changing the capacity at u to $c + 2$ and adding a new vertex v with capacity 2 adjacent only to u . Let $t = \max\{c + 2, 4\}$. If b is odd, then Builder can force a red F' or a red t -weighted r -cycle or a blue t -weighted b -cycle.*

Proof. Builder forces $r(b-1)/2$ red copies of F against this Painter. Next, Builder plays an $r(b-1)$ -cycle, alternating between copies of u and new vertices. Using red on any such edge produces a red copy of F' ; otherwise, there is a blue $r(b-1)$ -cycle in which alternate vertices have degree 2. This cycle decomposes into r paths P_1, \dots, P_r , each consisting of $b-1$ consecutive edges. Since each path has even length, we may assume that the endpoints of each path have degree 2.

Next, for each path P_j , Builder plays the edge joining its two endpoints. Using blue for any such edge creates a blue b -cycle (respecting capacities). Otherwise, these edges form a red r -cycle in which every vertex has degree 4. \square

Theorem 5.3. *If n is odd, then $\mathring{R}_\Delta(C_n) \leq 5$.*

Proof. Let F be the 3-weighted path with n vertices, with vertices v_0, \dots, v_{n-1} in order, and let G be the 5-weighted n -cycle. By Corollary 2.3, Builder can force F . In fact, against a consistent Painter, Builder can force F in a particular color, say red. By repeated application of Lemma 5.2, Builder can force a monochromatic 5-weighted n -cycle or the red weighted tree F' , where F' is obtained from F as follows. By repeatedly appending single edges, grow from v_i a path of length $\min\{i, n-1-i\}$ to a new leaf x_i . Let the capacities of vertices in F' be 5 on v_2, \dots, v_{n-2} , 4 on the other non-leaves of F' , 3 on v_0 and v_{n-1} , and 2 on the other leaves of F' (see Figure 5).

Let $p = \lfloor n/2 \rfloor$; note that v_p is the central vertex of F . Builder now presents a cycle C through the n leaves of F' , in the order $x_0, x_{p+1}, x_1, x_{p+2}, x_2, \dots, x_{n-1}, x_p$. Consecutive vertices in this list are separated by distance $n-1$ in F' . Using red on any such edge produces a red 5-weighted n -cycle; otherwise, the 5-weighted n -cycle arises in blue. \square

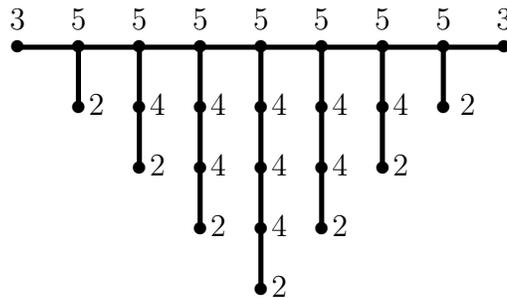


Figure 5: Forcing 5-weighted odd cycles

The idea behind Theorem 5.3 is to force a monochromatic tree with many leaves of weight 2 at the same distance from the center. Doing so was easy with weight 5 allowed along an initial path. To prove that $\mathring{R}_\Delta(C_n) = 4$ when n is odd and large, we developed a Builder strategy on \mathcal{S}_4 that iteratively combines four copies of a monochromatic tree (using Lemma 5.1) to force the next monochromatic tree. In the resulting sequence T_0, T_1, \dots of trees, the number of leaves roughly quadruples with each step, while the diameter roughly doubles. The leaves of T_r can then be further extended (by Lemma 5.1) so that all the leaves have the same distance from the center.

The last step in Theorem 5.3 can then be used to force C_n when n is odd. The trouble is that n must be at least the diameter of T_r and at most the number of leaves. Since the number of leaves grows faster, the argument succeeds for large n . Examining the details, the argument works when $n \geq 689$ and when $337 \leq n \leq 514$ to show that $C_n = 4$.

The insight by Rolnick [25] is that the leaves need not all have the same distance from the center. It suffices to use Lemma 5.1 to force a monochromatic 4-weighted tree having n leaves with weight 2 that can be ordered cyclically so that the distance in the tree between successive leaves in the order is $n - 1$. This allows him to use smaller monochromatic trees, and their diameter is not an issue. With these ideas, his general argument obtains $\mathring{R}_\Delta(C_n) = 4$ when n is odd and at least 13. He still must give ad hoc arguments (like those for $n = 6$ and $n = 10$ in Theorem 4.4) for $n \in \{5, 7, 9, 11\}$.

Our final result is that $\mathring{R}_\Delta(G) \leq 6$ when $\Delta(G) \leq 2$. We need two more lemmas.

Lemma 5.4. *Choose $p, q \in \mathbb{N}$ with q even and $2^{\lfloor p/4 \rfloor} \geq q/2 \geq 2$. If Builder can force a red 2-weighted P_3 against a consistent Painter, then Builder can force a red 4-weighted p -cycle or a blue 4-weighted q -cycle.*

Proof. Painter plays to avoid a blue q -cycle. If p is even, then Builder can use Lemma 4.3 to force a red tree of diameter $p - 1$ having degree at most 4 at non-leaves and degree at most 2 at leaves, as in Theorem 4.4. Since $p - 1$ is odd, there are isomorphic subtrees obtained by deleting the central edge. If instead p is odd, then the tree for $p - 1$ can be extended by one edge at each leaf in one of those subtrees. The number of leaves in each such subtree is $2^{\lfloor p/4 \rfloor}$. By the choice of p , there are at least $q/2$ leaves in each subtree. Builder plays a q -cycle through these leaves, alternating between the two sides. This yields a red p -cycle or a blue q -cycle, 4-weighted. \square

Lemma 5.5. *If r and b are both odd, then Builder can force a red C_r or a blue C_b on \mathcal{S}_6 against any consistent Painter.*

Proof. By symmetry, we may assume $b \leq r$. By presenting a b -cycle, Builder forces a red 2-weighted P_2 . With Painter avoiding a red C_r and blue C_b , repeated application of Lemma 5.2 forces a red P_r with degree 2 at the leaves and 4 at the non-leaves. With further applications of Lemma 5.2, red paths can be grown from the non-leaves of the red P_r to form a copy of the tree F' illustrated in Figure 5, with degree 6 in place of 5 and degree 4 in place of 3.

Now Builder plays a cycle through b leaves in F' , chosen and ordered as in Theorem 5.3 so that successive vertices are separated by distance $r - 1$ in F' . Using red on any such edge produces a red 6-weighted r -cycle; otherwise, a 4-weighted b -cycle arises in blue. \square

Theorem 5.6. *If $\Delta(G) \leq 2$, then $\mathring{R}_\Delta(G) \leq 6$.*

Proof. Fix a graph G with maximum degree 2. We may assume that every component is a cycle, since otherwise we force the graph obtained by completing each path component to a cycle. Without loss of generality, we may assume that Painter makes at least two edges red when given an isolated triangle. That is, Builder forces a 2-weighted P_3 in red.

If Builder cannot force a monochromatic G in \mathcal{S}_6 , then there must be integers r and b such that Builder cannot force a red C_r and cannot force a blue C_b . By Lemma 5.5, we may assume that r and b are not both odd.

If b is even, then Builder can force a red $(r-1)b$ -cycle or a blue b -cycle, 4-weighted, by Lemma 5.4. If the former, then by cutting the cycle into b paths of length $r-1$ and playing a cycle through their endpoints, Builder forces a red r -cycle or a blue b -cycle, 6-weighted. Call this “inscribing” a b -cycle.

Hence we may assume that b is odd and r is even. Let $p = (r-1)(2b-1) - 1$; note that $2^{\lfloor p/4 \rfloor} \geq (b-1)r/2$. By Lemma 5.4, Builder can force a red p -cycle or a blue $(b-1)r$ -cycle, 4-weighted. If the latter, then inscribing an r -cycle forces a desired cycle, 6-weighted.

Hence we may assume that a red p -cycle arises, 4-weighted. Builder cuts it into $b-1$ consecutive paths of length $r-1$ and one path of length $(r-1)b-1$. Playing a cycle on the breakpoints between paths forces a 6-weighted red r -cycle or a 6-weighted blue b -cycle or a red $(r-1)b$ cycle where two adjacent vertices have degree at most 6 and the remaining vertices have degree at most 4. Since $r \geq 4$, in this cycle Builder can inscribe a b -cycle that avoids those two high-degree vertices. A desired cycle arises, 6-weighted. \square

A short proof of the weaker bound $\dot{R}_\Delta(G) \leq 8$ when $\Delta(G) \leq 2$ avoids Lemma 5.5. For b even, argue as before. For b odd, $(b-1)r$ is even, and Builder forces a red $(r-1)(b-1)r$ -cycle or blue $(b-1)r$ -cycle, 4-weighted. Inscribing an r -cycle in the latter gives a 6-weighted desired cycle. Inscribing a $(b-1)r$ -cycle in the former gives a 6-weighted red r -cycle or blue $(b-1)r$ -cycle, and then inscribing an r -cycle yields a desired cycle, 8-weighted.

References

- [1] N. Alon, G. Ding, B. Oporowski, and D. Vertigan, Partitioning into graphs with only small components, *J. Combin. Theory Ser. B* 87 (2003), 231–243.
- [2] J. Beck, On size Ramsey number of paths, trees and cycles I, *J. Graph Theory* 7 (1983), 115–130.
- [3] J. Beck, On size Ramsey number of paths, trees and cycles II. *Mathematics of Ramsey Theory* (Springer), *Algorithms and Combin.* 5 (1990), 34–45.
- [4] J. Beck, Achievement games and the probabilistic method, in: *Combinatorics, Paul Erdős is Eighty*, vol. 1, *Bolyai Soc. Math. Stud.* (1993), 51–78.
- [5] S. Burr, P. Erdős, and L. Lovász, On graphs of Ramsey type. *Ars Combinatoria* 1 (1976), 167–190.
- [6] D. Conlon, On-line Ramsey numbers. *SIAM J. Discr. Math.* 23 (2009/10), 1954–63.
- [7] J. Donadelli, P.E. Haxell, and Y. Kohayakawa, A note on the size-Ramsey number of long subdivisions of graphs, *RAIRO-Inf. Theor. Appl.* 39 (2005), 191–206.
- [8] J. Folkman, Graphs with monochromatic complete subgraphs in every edge coloring. *SIAM J. Appl. Math.* 18 (1970), 19–24
- [9] G.A. Dirac, Some theorems on abstract graphs. *Proc. London Math. Soc.* (3) 2, (1952), 69–81.

- [10] P. Erdős, R.J. Faudree, C.C. Rousseau, and R.H. Schelp, The size Ramsey number, *Period. Math. Hungar.* 9 (1978), 145–161.
- [11] E. Friedgut, Y. Kohayakawa, V. Rödl, A. Ruciński, and P. Tetali, Ramsey games against a one-armed bandit. Special issue on Ramsey theory. *Combin. Probab. Comput.* 12 (2003), 515–545.
- [12] J.A. Grytczuk, M. Hałuszczak, and H.A. Kierstead, On-line Ramsey theory. *Electron. J. Combin.* 11 (2004), Paper #R57, 10 pp.
- [13] J.A. Grytczuk, H.A. Kierstead, and P. Prałat, On-line Ramsey numbers for paths and stars, *Discrete Math. Theor. Comp. Sci.* 10 (2008), 63–74.
- [14] P. Horn, K.G. Milans, and V. Rödl, Degree Ramsey numbers for blowups of trees, draft.
- [15] T. Jiang, K.G. Milans, and D.B. West, Degree Ramsey number of cycles is bounded, submitted.
- [16] H.A. Kierstead and G. Konjevod, Coloring number and on-line Ramsey theory for graphs and hypergraphs *Combinatorica* 29 (2009), 49–64.
- [17] W.B. Kinnersley, Multicolor on-line degree Ramsey numbers for trees and cycles, draft.
- [18] W.B. Kinnersley, K.G. Milans, and D.B. West, Degree Ramsey numbers of graphs, submitted.
- [19] J. Nešetřil and V. Rödl, Type theory of partition properties of graphs, in: *Recent advances in graph theory (Proc. Second Czechoslovak Sympos., Prague, 1974)*, (Academia, Prague, 1975), 405–412.
- [20] Š. Petříčková, *Combinatorial Game Theory*, Diploma Thesis, University of West Bohemia, Pilsen.
- [21] P. Prałat, A note on small on-line Ramsey numbers for paths and their generalization, *Australasian J. Combinatorics*, 40 (2008), 27–36.
- [22] P. Prałat, A note on off-diagonal small on-line Ramsey numbers for paths, *Ars Combinatoria*, 10pp, accepted.
- [23] P. Prałat, $R^*(3, 4) = 17$, *Electron. J. Combin.* 15 (2008), no. 1, Paper #R67, 13pp.
- [24] V. Rödl and E. Szemerédi, On size Ramsey numbers of graphs with bounded maximum degree, *Combinatorica* 20 (2000), 257–262.
- [25] D. Rolnick, On-line Ramsey theory: Building and painting graphs, one edge at a time, submitted.
- [26] C. Thomassen, Two-coloring the edges of a cubic graph such that each monochromatic component is a path of length at most 5, *J. Combin. Theory Ser. B* 75 (1999), 100–109.
- [27] X. Zhu, Chromatic Ramsey numbers. *Discrete Math.* 190 (1998), 215–222.
- [28] X. Zhu, The fractional version of Hedetniemi's conjecture is true, *European Journal of Combinatorics* 32 (2011), available online April 13, 2011.