

Online coloring a token graph

Kevin G. Milans, Michael C. Wigal

December 22, 2017

Abstract

We study a combinatorial coloring game between two players, Spoiler and Algorithm, who alternate turns. First, Spoiler places a new token at a vertex in G , and Algorithm responds by assigning a color to the new token. Algorithm must ensure that tokens on the same or adjacent vertices receive distinct colors. Spoiler must ensure that the token graph (in which two tokens are adjacent if and only if their distance in G is at most 1) has chromatic number at most w . Algorithm wants to minimize the number of colors used, and Spoiler wants to force as many colors as possible. Let $f(w, G)$ be the minimum number of colors needed in an optimal Algorithm strategy.

A graph G is *online-perfect* if $f(w, G) = w$. We give a forbidden induced subgraph characterization of the class of online-perfect graphs. When G is not online-perfect, determining $f(w, G)$ seems challenging; we establish $f(w, G)$ asymptotically for some (but not all) of the minimal graphs that are not online-perfect. The game is motivated by a natural online coloring problem on the real line which remains open.

1 Introduction

Our work is motivated by the following simple combinatorial game between two players, Spoiler (sometimes called Presenter) and Algorithm. Spoiler moves first, selecting a point $x \in \mathbb{R}$ (repetition allowed), with the restriction that for each open unit interval I , Spoiler may play in I at most w times. Algorithm responds by assigning a color to x such that selected points at distance less than 1 are assigned distinct colors. The value of the game, denoted $h(w)$, is the minimum number of colors that an optimal Algorithm strategy needs in the worst case. The greedy algorithm shows that $h(w) \leq 2w - 1$, as follows. If none of the colors in $\{1, \dots, 2w - 1\}$ is available for a selected point x , then either $(x - 1, x]$ or $[x, x + 1)$ contains at least w previously played points which, after a small shift, can be captured in an open unit interval I containing x . Spoiler is forbidden to play again in I , contradicting Spoiler's selection of x . Despite its simplicity, the greedy algorithm gives the best known upper bound on $h(w)$.

From below, Bosek, Felsner, Kloch, Krawczyk, Matecki, and Micek [3] proved that $h(w) \geq \lfloor \frac{3}{2}w \rfloor$. In their work, the problem of bounding $h(w)$ arises as a special case of the online chain partition problem for partially ordered sets. To obtain the lower bound on $h(w)$, they offer the following strategy for Spoiler, which has three stages. Let $k = \lfloor \frac{w}{2} \rfloor$. In the first stage, Spoiler plays k times at 0. Let A be the set of colors that Algorithm uses. In the second stage, Spoiler plays at most $2k$ times in the interval $(1, 2)$ so that all points with new colors (outside A) are less than all points with old colors (in A); Spoiler accomplishes this by always playing at a point between the new and old colors. Spoiler stops this stage once Algorithm uses k new colors. Let B be the set of new

colors, and let $y \in (1, 2)$ be a point between the new and old colors. Note that $(y - 1, y)$ contains 1 and therefore each color in B appears on a selected point in $(y - 1, y)$. Similarly $(y - 2, y - 1)$ contains 0 and therefore each color in A appears on a selected point in $(y - 2, y - 1)$. In the third stage, Spoiler plays $w - k$ times at $y - 1$. Algorithm must use $w - k$ additional colors outside $A \cup B$. In total, Algorithm uses $w + k$ colors, and $w + k = \lfloor \frac{3}{2}w \rfloor$. This strategy for Spoiler gives the best known lower bound on $h(w)$.

It would be interesting to know the asymptotics of $h(w)$. In attempting to make progress, we studied a generalization of this problem to graphs. An *independent set* in a graph G is a set of vertices that are pairwise nonadjacent. The *chromatic number* of G , denoted $\chi(G)$, is the minimum size of a partition of $V(G)$ into independent sets.

For our generalization, we use the *token model*, in which both the graph G and the width w are known to both Algorithm and Spoiler. Spoiler selects a vertex $u \in V(G)$ and plays a *token* x at u . In the *token graph*, tokens x and y are adjacent if and only if x and y are on the same or neighboring vertices in G . Spoiler must play in such a way that the token graph has chromatic number at most w . After Spoiler plays a token x , Algorithm must assign a color to x so that Algorithm's coloring of the token graph is *proper*, meaning that tokens on the same or adjacent vertices in G must be assigned distinct colors. The *value* of the game, denoted $f(w, G)$, is the minimum number of colors that an optimal strategy for Algorithm needs for the game in the worst case. The motivating problem arises when G is the graph $\mathbb{G}$ with $V(\mathbb{G}) = \mathbb{R}$ and $uv \in E(\mathbb{G})$ if and only if $|u - v| < 1$. With the exception of a few explicitly defined infinite graphs like $\mathbb{G}$, we generally assume that all graphs are finite.

In some cases, it is interesting to limit Spoiler so that Spoiler can play at most 1 token at each vertex. We call this the *restricted token model* and use $f^*(w, G)$ to denote the value of the restricted token game on G of width w . Using $\overline{G}$ for the graph complement of G , we note that the restricted token game on $\overline{\mathbb{G}}$ is an online clustering problem: Spoiler presents distinct points in $\mathbb{R}$ and Algorithm assigns colors so that the color classes, called *clusters*, have diameter less than 1. (Usually, clusters are allowed to have diameter at most 1, but standard perturbation and compactness arguments imply that these notions lead to equivalent games.) The trivial bounds are $\lfloor \frac{3}{2}w \rfloor \leq f^*(w, \overline{\mathbb{G}}) \leq 2w - 1$. The first nontrivial bounds are due to Epstein and van Stee [7], who proved $(\frac{8}{5} - o(1))w \leq f^*(w, \overline{\mathbb{G}}) \leq \frac{7}{4}w$. Ehmsen and Larson [6] improved the upper bound to $f^*(w, \overline{\mathbb{G}}) \leq \frac{5}{3}w$ and Kawahara and Kobayashi [10] improved the lower bound to $f^*(w, \overline{\mathbb{G}}) \geq (\frac{13}{8} - o(1))w$.

1.1 Other Models

Several models of online graph coloring have been studied. In the *classical model*, Spoiler presents the vertices of a graph in some order $v_1, \dots, v_n$ and Algorithm must decide on a color for v_i knowing only the subgraph induced by $\{v_1, \dots, v_i\}$. For an algorithm $\mathcal{A}$ and a graph family $\mathcal{G}$, the *performance ratio* of $\mathcal{A}$ on $\mathcal{G}$, denoted $\rho_{\mathcal{A}}(n, \mathcal{G})$, is the maximum, over all n -vertex graphs $G \in \mathcal{G}$ and orderings of $V(G)$, of the number of colors that $\mathcal{A}$ uses on G divided by $\chi(G)$. It is not difficult to show that even when $\mathcal{G}$ is the family of acyclic graphs, every algorithm $\mathcal{A}$ satisfies $\rho_{\mathcal{A}}(n, \mathcal{G}) \geq 1 + \lfloor \lg n \rfloor$, and so the performance ratio of an algorithm in the classical model typically depends on the number of vertices in the input graph. In 1989, for the class $\mathcal{G}$ of all graphs, Lovász, Saks, and Trotter [13] proved that there exists an algorithm $\mathcal{A}$ such that $\rho_{\mathcal{A}}(n, \mathcal{G}) = (1 + o(1))(2n / \log^* n)$, where $\log^* n$ is the least integer k such that the k th iterated logarithm function $\log^{(k)}$ satisfies $\log^{(k)}(n) < 1$. Also in 1989, M. Szegedy (unpublished) showed that every online algorithm $\mathcal{A}$

satisfies $\rho_{\mathcal{A}}(n, \mathcal{G}) \geq \Omega(n/(\log n)^2)$; for this and lower bounds on the performance of randomized algorithms, see Halldórsson and Szegedy [9]. In 1998, Kierstead [11] showed that there exists an algorithm $\mathcal{A}$ such that $\rho_{\mathcal{A}}(n, \mathcal{G}) \leq O(n \log^{(3)} n / \log^{(2)}(n))$, giving the best known performance ratio for the class of all graphs in the classical model. For an older but still useful survey on results in the classical model, see the survey paper by Kierstead [12].

In addition to the token model that we study, there are at least two other *known-graph models*. Halldórsson [8] examined a model in which a host graph G is known to both players. Spoiler presents vertices to Algorithm, promising that the presented graph H is a subgraph of G . However, if G contains many copies of H , then Algorithm does not know which of these copies is H . Consider, for example, the case $G = C_6$. If the first two vertices that Spoiler presents are nonadjacent, then the pair may be at distance 2 in G (in which case Algorithm prefers that they share a color) or at distance 3 (in which case Algorithm prefers that they be colored differently). The performance ratio of Algorithm is the number of colors that Algorithm uses divided by $\chi(G)$. Halldórsson constructed for each n an n -vertex graph G such that every algorithm has performance ratio at least $\Omega(n/(\log n)^2)$ on G , obtaining essentially the same lower bound as Halldórsson–Szegedy against a stronger Algorithm.

A second known-graph model appears in a paper by Bartal, Fiat, and Leonardi [2] and an extended abstract by the same authors [1]. Again, the host graph G is known to both Algorithm and Spoiler. In this model, Spoiler reveals the location of vertices in G . The difficulty for Algorithm is that Spoiler may stop the game at any subgraph H of G , and the performance ratio of Algorithm is the number of colors used divided by $\chi(H)$ (not $\chi(G)$). This model can also be viewed as a slight variant of the restricted token model in which Algorithm does not know w . As usual, bounds on the performance ratio are typically given in terms of n , where $n = |V(G)|$. Bartal, Fiat, and Leonardi proved that there is an algorithm that achieves performance ratio $O(n^{1/2})$ for each n -vertex graph G and that every algorithm has performance ratio $\Omega(n^{1-\log_5 4})$. The lower bound applies even to randomized algorithms.

1.2 Our Results

Since determining the asymptotics of $h(w)$, or equivalently $f(w, \mathbb{G})$, appears difficult, it is natural to study the token model in simpler settings. A graph G is *online-perfect* if $f(w, G) = w$ for each w . Our main result is a forbidden-subgraph characterization of the family of online-perfect graphs. The minimal graphs that are not online-perfect are the odd cycles of length at least 5, the 5-cycle with 1 or 2 non-crossing chords, and the bull graph. These graphs are perhaps the simplest candidates for host graphs where the token game becomes an interesting problem. Of these, we obtain the asymptotic value of $f(w, G)$ when G is an odd cycle or the 5-cycle with 2 non-crossing chords. In particular, we show that if n is odd and $n \geq 5$, then $f(w, C_n) = (\frac{n}{n-1} + o(1))w$ where $o(1)$ represents a function tending to 0 as $w \rightarrow \infty$. Also, $f(w, G) = \lceil \frac{5}{4}w \rceil$ when w is even and G is the 5-cycle with 2 non-crossing chords. The cases that G is the bull graph or that G is a 5-cycle with a single chord remain open.

2 Characterization of online-perfect graphs

Our first strategy for Algorithm shows that bipartite graphs are online-perfect. A *clique* in G is a set of vertices that are pairwise adjacent, and the *clique number* of a graph G , denoted $\omega(G)$, is

the maximum size of a clique in G . Since no independent set contains more than one vertex of a clique, we have $\chi(G) \geq \omega(G)$.

Proposition 1. *If G is bipartite, then $f(w, G) = w$.*

Proof. Let G be an (U, V) -bigraph, and let L be a list of w colors. Algorithm maintains the invariant that for each $u \in U$, the colors used for tokens on u form a prefix of L . Similarly, for each $v \in V$, the colors used for tokens on v form a suffix of L . When Spoiler plays a token x at $u \in U$, Algorithm assigns to x the first color in L not already used for a token on u . When Spoiler plays a token y at $v \in V$, Algorithm assigns to y the last color in L not already used for a token on v . We claim that Algorithm produces a proper coloring of the token graph. Indeed, if x and y are assigned the same color but are on adjacent vertices $u \in U$ and $v \in V$, then u and v together contain more than w tokens, implying that the token graph has clique number larger than w . $\square$

In a graph G , vertices u and v are *twins* if u and v have the same neighbors in $V(G) - \{u, v\}$. Given a graph G and a vertex u in G , we *clone* u by introducing a new vertex u' that is a twin of u ; in the new graph, u and u' may or may not be adjacent.

Proposition 2. *If G' is obtained from G by cloning a vertex, then $f(w, G') = f(w, G)$.*

Proof. Suppose that G' be obtained from G by cloning u to a new vertex u' . Since G is an induced subgraph of G' , we have $f(w, G') \geq f(w, G)$. To show $f(w, G') \leq f(w, G)$, we let $\mathcal{A}$ be an optimal strategy for Algorithm for G and give an Algorithm strategy $\mathcal{A}'$ for G' . The strategy depends on whether or not $uu' \in E(G)$.

First, suppose $uu' \in E(G')$. We have $\mathcal{A}'$ simulate $\mathcal{A}$ on a token game on G . If Spoiler plays a token at a vertex $v \notin \{u, u'\}$, then $\mathcal{A}'$ plays a token at v in the simulation, and $\mathcal{A}'$ gives the real token the same color as $\mathcal{A}$ gives to the simulated token. If Spoiler plays a token at u or u' , then $\mathcal{A}'$ plays a token at u in the simulation and again responds with the same color as $\mathcal{A}$. Since u and u' are adjacent in G' , the token graph in the real game is the same as the token graph in the simulation. Therefore the width of the real game and the width of the simulation are equal and we have $f(w, G') \leq f(w, G)$.

Suppose that u and u' are nonadjacent. In this case, we modify the strategy slightly so that the number of tokens at u in the simulation is equal to the maximum of the number of tokens at u and u' in the real game. If Spoiler plays a token at u or u' which would increase this maximum, then $\mathcal{A}'$ plays a token at u in the simulation and responds with the same color. If Spoiler plays a token at u when u' already has at least as many tokens, then $\mathcal{A}'$ does not play any tokens in the simulation and responds with one of the colors appearing at u' that does not appear at u . The case that Spoiler plays a token at u' when u already has at least as many tokens is similar. Again, the width of the real game and the width of the simulation are equal, and it follows that $f(w, G') \leq f(w, G)$. $\square$

For graphs G_1 and G_2 , we use $G_1 + G_2$ to denote the disjoint union of G_1 and G_2 . The *join* of G_1 and G_2 , denoted $G_1 \vee G_2$, is the graph obtained from disjoint copies of G_1 and G_2 with all vertices in the copy of G_1 adjacent to all vertices in the copy of G_2 . The following proposition, which we include for completeness, is an easy consequence of the theory of *cographs*, the graphs that do not contain an induced copy of P_4 .

Proposition 3. *A graph G is P_4 -free if and only if G is obtainable from K_1 by cloning vertices.*

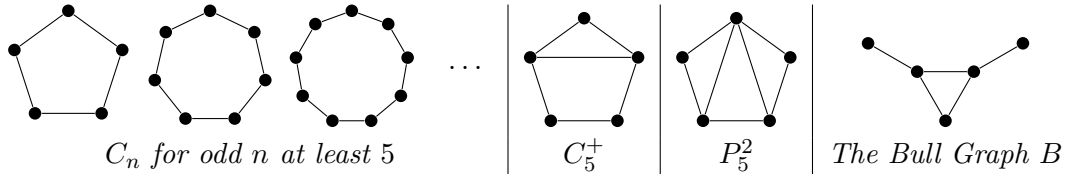
Proof. It is well known that if G is P_4 -free and $|V(G)| \geq 2$, then either $G = G_1 + G_2$ or $G = G_1 \vee G_2$ for some P_4 -free graphs G_1 and G_2 . We obtain $G_1 + G_2$ by starting with a single vertex u_1 , cloning to a nonadjacent vertex u_2 , and inductively obtaining G_1 from u_1 and G_2 from u_2 . We obtain $G_1 \vee G_2$ similarly, except that we begin with adjacent vertices u_1 and u_2 .

Conversely, cloning vertices preserves being P_4 -free. Indeed, it is easy to check that P_4 does not contain a pair of twins. If G is P_4 -free but the graph G' obtained from G by cloning $u \in V(G)$ to a twin $u' \in V(G')$ produces a copy P of P_4 , then u and u' are both in P . Since u and u' are twins in G' , they are also twins in P , a contradiction. $\square$

When G is an edge-transitive graph, we use G^- to denote the graph obtained from G by removing an edge. Similarly, when the complement of G is edge-transitive, we use G^+ to denote the graph obtained from G by adding an edge. The *distance k -power* of G , denoted G^k , is the graph on $V(G)$ with $uv \in E(G^k)$ if and only if the distance between u and v in G is at most k .

Theorem 4. *Let G be a graph. The following are equivalent.*

- (1) G is online-perfect.
- (2) $f(2, G) = 2$.
- (3) G does not contain any of the following as an induced subgraph:



- (4) G is obtainable from a bipartite graph by cloning vertices.

Proof. It is obvious that (1) implies (2). To show that (2) implies (3), we give strategies for Spoiler that force 3 colors in a game of width 2. Let u and v be vertices in a graph G that has induced uv -paths of both parities. Spoiler plays a token x at u and a token y at v . If Algorithm assigns the same color to x and y , then Spoiler forces two more colors by playing a token at each of the internal vertices on an induced uv -path of odd length. If Algorithm assigns distinct colors to x and y , then Spoiler forces a third color by playing a token at each of the internal vertices on an induced uv -path of even length. In both cases, the token graph is a path, and so the game has width 2. Except for the bull graph B , each graph in (3) has vertices u and v and induced uv -paths of both parities. A different Spoiler strategy is needed for B . Let $v_1, \dots, v_5$ be the vertices of B along its spanning path. Spoiler first plays a token x at v_1 and a token y at v_5 . If Algorithm assigns the same color to x and y , then Spoiler plays a token at v_2 and another at v_4 . This forces two new colors, and the token graph is P_4 . Otherwise, Algorithm assigns distinct colors to x and y . Spoiler plays a token z at v_3 ; without loss of generality, we may assume that x and z have distinct colors. Spoiler forces a third color by playing a token at v_2 . With one token on each vertex in $\{v_1, v_2, v_3, v_5\}$, the components of the token graph are P_3 and P_1 , and again the token graph is bipartite. It follows that if $f(2, G) = 2$, then G does not contain an induced copy of any of the graphs listed in (3).

We show that (3) implies (4) by contradiction; let G be a minimum counter-example. If G contains twins u and u' , then $G - u'$ is a smaller graph avoiding induced copies of all graphs in

(3), implying that $G - u'$ is obtainable from a bipartite graph by cloning. We could then obtain G from $G - u'$ by cloning u . It follows that G does not contain twins. We consider two cases. First, suppose that $\text{diam}(G) \leq 2$. It must be that G contains an induced copy P of P_4 , or else Proposition 3 implies that G is obtainable by cloning from the bipartite graph K_1 . Let $u_1, \dots, u_4$ be the vertices of P . Since P is induced and $\text{diam}(G) \leq 2$, it follows that u_1 and u_4 have a common neighbor v that is not on P . Hence v completes a 5-cycle with P in which all chords are incident to v , and so $V(P) \cup \{v\}$ induces one of $\{C_5, C_5^+, P_5^2\}$, all of which are forbidden.

We may assume that $\text{diam}(G) \geq 3$. Among all vertices in G that are at distance 3 from some vertex, select a vertex u , favoring vertices whose neighborhoods are independent sets. For $k \geq 0$, let G_k be the subgraph of G induced by vertices at distance k from u . We claim that if H is a component of G_k and $v_1 \in V(G_{k-1})$, then v_1 is adjacent to all vertices in H or none of them. Indeed, if this is not the case, then let k be the least integer where this fails. Since v_1 has a neighbor and a non-neighbor in H and H is connected, there exists $v_0 w_0 \in E(H)$ such that $v_1 v_0 \in E(G)$ but $v_1 w_0 \notin E(G)$. Let $w_1 \in V(G_{k-1})$ be a neighbor of w_0 , and let t be the least integer such that G has paths $v_0 v_1 \dots v_t x$ and $w_0 w_1 \dots w_t x$ with $v_j, w_j \in V(G_{k-j})$ and $x \in V(G_{k-(t+1)})$. By the minimality of t , we have that $v_i w_{i+1}, w_i v_{i+1} \notin E(G)$ for $1 \leq j < t$. By the minimality of k , we have that $v_i w_i \notin E(G)$ for $1 \leq i < t$. It follows that $v_0 v_1 \dots v_t x w_t w_{t-1} \dots w_0$ is a $(2t+3)$ -cycle C in which the chords are a subset of $\{v_0 w_1, v_t w_t\}$. If $t \geq 2$, then either C is an induced odd cycle of length at least 7, or either chord completes an induced copy of the bull. If $t = 1$, then C is a 5-cycle in which all chords are incident to w_1 , and so $V(C)$ induces a graph in $\{C_5, C_5^+, P_5^2\}$.

Next, we claim that if H is a component of G_k with $k \geq 2$ and $x \in V(G_{k+1})$, then x is adjacent to all vertices in H or none of them. Otherwise, we again obtain $vw \in E(H)$ such that $xv \in E(G)$ but $xw \notin E(G)$. Let $y \in G_{k-1}$ be a neighbor of v ; also $yw \in E(H)$. Let $z \in V(G_{k-2})$ be a neighbor of y , and observe that $\{v, w, x, y, z\}$ induces a copy of the bull.

We claim that G_k has no edges for $k \geq 2$. Let H be a component of G_k , and let $x \in V(G_{k-1})$ contain H in its neighborhood. Note that H is P_4 -free, or else an induced copy of P_4 in H together with x yields an induced copy of P_5^2 . It follows from Proposition 3 that either $H = K_1$ or H contains a pair of twins. Since $k \geq 2$, a pair of twins in H would also be a pair of twins in G , and so $H = K_1$ as claimed.

Note that G_1 also has no edges. Indeed, if z is a vertex at maximum distance k from u , then $k \geq 3$ and z is an isolated vertex in G_k . It follows that all neighbors of z are isolated vertices in G_{k-1} and therefore $N(z)$ is an independent set. If $N(u)$ were not independent, then we would prefer z to u in our initial selection of u . It follows that $N(u)$ is an independent set, and $V(G_1) = N(u)$. Since each G_k has no edges, it follows that G is a bipartite graph, contradicting that G is a counter-example. Therefore (3) implies (4).

It remains to show that (4) implies (1). By Proposition 1, every bipartite graph is online-perfect. By Proposition 2, cloning vertices preserves online-perfection. $\square$

A graph G is *perfect* if $\chi(H) = \omega(H)$ for each induced subgraph H of G . The Strong Perfect Graph Theorem of Chudnovsky, Robertson, Seymour, and Thomas [5] states that G is perfect if and only if G does not contain an odd hole or an odd antihole. A *hole* is an induced cycle of length at least 4, and an *antihole* is an induced subgraph whose complement is a hole. Note that a set of 5 consecutive vertices on a cycle of length larger than 5 induces a copy of P_5 , and $\overline{P_5} = C_5^+$. It follows that an antihole on more than 5 vertices contains an induced copy of C_5^+ . By (3), odd holes and odd antiholes are forbidden in online-perfect graphs, and so every online-perfect graph is perfect.

It also follows from (4) and Proposition 3 that G is online-perfect if and only if G is obtainable from a bipartite graph by replacing each vertex with a copy of a P_4 -free graph.

3 Minimal graphs that are not online-perfect

When G is not online-perfect, it contains an induced copy of one of the obstructions from Theorem 4. It is natural to study the value of the online-coloring game for these graphs; although these are the simplest graphs that are not online-perfect, we are only able to establish the asymptotics for the odd cycles and P_5^2 . We begin with a strategy for Algorithm that generalizes Proposition 1.

A (p, q) -coloring of a graph G is an assignment ϕ that maps each vertex in G to a set of q colors from a universe of p colors such that $\phi(u)$ and $\phi(v)$ are disjoint when $uv \in E(G)$. Usually, we use $[p]$ for the universe of colors and write $\phi: V(G) \rightarrow \binom{[p]}{q}$, where $\binom{[p]}{q}$ denotes the family of subsets of $[p]$ of size q . The *fractional chromatic number* of G , denoted $\chi_f(G)$, is the infimum of p/q over all pairs (p, q) such that G has a (p, q) -coloring. There is an equivalent alternative formulation of the fractional chromatic number as the optimum value of a linear program involving the independent set polytope of G , and it follows that each graph G has a (p, q) -coloring such that $\chi_f(G) = p/q$.

Proposition 5 (Fractional Coloring Strategy). *If G has a (p, q) -coloring, then $f(w, G) \leq p \left\lceil \frac{w}{2q} \right\rceil$. Consequently, $f(w, G) \leq (\frac{\chi_f(G)}{2} + o(1))w$.*

Proof. Let $\phi: V(G) \rightarrow \binom{[p]}{q}$ be a (p, q) -coloring, where $\chi_f(G) = p/q$. For a list L , let L^r denote the list obtained by reversing L . Also, let $L_1 \cdot L_2$ be the concatenation of lists L_1 and L_2 , and when I is a set of integer indices, let $\prod_{i \in I} L_i$ denote the concatenation of lists where entries in L_i come before L_j if and only if $i < j$.

Let $t = \lceil w/(2q) \rceil$ and let $S_1, \dots, S_p$ be disjoint lists of t colors. For each vertex u , the *forward list* at u , denoted F_u , is $\prod_{j \in A} S_j$ and the *reverse list* at u , denoted R_u , is $\prod_{j \in B} S_j^r$, where $A = \phi(u)$ and $B = [p] - \phi(u)$. Let $L_u = F_u \cdot R_u$. Algorithm maintains the invariant that, for each vertex u , the tokens played at u are colored with a prefix of L_u . When Spoiler plays a token x at u , Algorithm assigns to x the first color in L_u that is not already used for a token at u . Provided that Algorithm produces a proper coloring of the token graph, we have $f(w, G) \leq tp = \left\lceil \frac{w}{2q} \right\rceil p < (\frac{w}{2q} + 1)p = (\frac{p}{2q} + \frac{p}{w})w = (\chi_f(G)/2 + o(1))w$.

Suppose for a contradiction that a token x is played at u but its designated color α already appears on a token at a neighbor v . Since $\phi(u)$ and $\phi(v)$ are disjoint, the colors in F_u appear in R_v in reverse order, and colors in F_v appear in R_u in reverse order. We claim that all colors in $F_u \cup F_v$ appear on tokens at u or v . Note that all colors preceding α in $F_u \cdot R_u$ appear on tokens at u . If α is in F_u , then colors in F_u following α appear before α in R_v . By the prefix invariant at v , all colors preceding α in $F_v \cdot R_v$ appear on tokens at v ; these colors include F_v as well as those in F_u following α . Otherwise, $\alpha \in R_u$, and so every color in F_u appears on tokens at u . If $\alpha \in R_v$, then every color in F_v appears on tokens at v . If $\alpha \in F_v$, then colors in F_v preceding α appear at v and colors in F_v following α appear before α in R_u and therefore are used on tokens at u .

In all cases, since $|F_u| = |F_v| = qt$, it follows that at least $2qt$ tokens other than x have been played at u and v . These tokens form a clique in the token graph, and so $w > 2qt$, contradicting our choice of t . $\square$

For a graph G and a set of vertices $S \subseteq G$, we use $G[S]$ to denote the subgraph of G induced by S .

Proposition 6 (Spoiler Strategy). *Let $U \subseteq V(G)$ where $U = \{u_1, \dots, u_t\}$, and suppose that*

- (1) *there are vertices x and y such that u_1xyu_t is a path and $G[\{u_1, x, y, u_t\}]$ is bipartite, and*
- (2) *there is a common neighbor z_i of u_i and u_{i+1} such that $G[U \cup \{z_i\}]$ is bipartite for $1 \leq i < t$.*

If w is even, then $f(w, G) \geq (1 + \frac{1}{2t})w$.

Proof. With $k = w/2$, Spoiler plays k tokens at u_1 and u_t . Let S_i be the set of colors at u_i . If $|S_1 \cap S_t| \geq k/t$ then Spoiler plays k tokens at each of $\{x, y\}$. This forces $2k + k/t$ colors. Since the subgraph of G induced by $\{u_1, x, y, u_t\}$ is bipartite, the chromatic number of the token graph is at most $2k$.

Otherwise $|S_1 \cap S_t| < k/t$ and $|S_1 - S_t| > k - k/t$. Spoiler plays k tokens at each u_i for $1 < i < t$. Since $S_1 - S_t$ is contained in $\bigcup_{i < t} S_i - S_{i+1}$, it follows that for some i , we have $|S_i - S_{i+1}| \geq \frac{|S_1 - S_t|}{(t-1)} > \frac{k}{t}$. Spoiler plays k tokens at z_i . Algorithm uses disjoint sets of k colors at z_i and u_{i+1} , plus at least k/t additional colors at u_i ; this forces at least $2k + k/t$ colors in total. Since the subgraph of G induced by $U \cup \{z_i\}$ is bipartite, the token graph has chromatic number at most $2k$ in this case also. $\square$

Corollary 7 (Odd Holes). *Let n be odd and at least 5. We have $\frac{n}{n-1} \cdot 2 \lfloor \frac{w}{2} \rfloor \leq f(w, C_n) \leq n \lceil \frac{w}{n-1} \rceil$ and $f(w, C_n) = \frac{n}{n-1}w$ when $n-1$ divides w . Therefore $f(w, C_n) = (\frac{n}{n-1} + o(1))w$, where the $o(1)$ term tends to 0 for each fixed n as $w \rightarrow \infty$.*

Proof. Let $w' = 2 \lfloor \frac{w}{2} \rfloor$, so that w' is the largest even integer at most w . With $t = (n-1)/2$ and $u_1, \dots, u_t$ spaced along C_n so that u_i and u_{i+1} are at distance 2 for $1 \leq i < t$, the Spoiler Strategy gives $f(w, C_n) \geq f(w', C_n) \geq (1 + \frac{1}{2t})w' = \frac{n}{n-1} \cdot 2 \lfloor \frac{w}{2} \rfloor$. Since C_n has an $(n, (n-1)/2)$ -coloring, the Algorithm Strategy gives $f(w, C_n) \leq n \lceil \frac{w}{n-1} \rceil$. When n is odd and $n-1$ divides w , the bounds reduce to $f(w, C_n) = \frac{n}{n-1}w$. $\square$

Corollary 8. *We have $\frac{5}{2} \lfloor \frac{w}{2} \rfloor \leq f(w, C_5^+) \leq 3 \lceil \frac{w}{2} \rceil$.*

Proof. Let $w' = 2 \lfloor \frac{w}{2} \rfloor$, so that w' is the largest even integer at most w . Let u_1 and u_2 be two nonadjacent vertices of degree two belonging to C_5^+ . The Spoiler Strategy gives $f(w', C_5^+) \geq f(w', C_5^+) \geq \frac{5}{4}w' = \frac{5}{2} \lfloor \frac{w}{2} \rfloor$. Since C_5^+ has a $(3, 1)$ -coloring, the Algorithm Strategy gives $f(w, C_5^+) \leq 3 \lceil \frac{w}{2} \rceil$. $\square$

Corollary 9. *We have $\frac{7}{3} \lfloor \frac{w}{2} \rfloor \leq f(w, B) \leq 3 \lceil \frac{w}{2} \rceil$.*

Proof. Let $w' = 2 \lfloor \frac{w}{2} \rfloor$, so that w' is the largest even integer at most w . In the bull B , let u_1 and u_3 be vertices of degree 1 and let u_2 be the vertex of degree 2. The Spoiler Strategy gives $f(w, B) \geq f(w', B) \geq \frac{7}{6}w' = \frac{7}{3} \lfloor \frac{w}{2} \rfloor$. Since B has a $(3, 1)$ -coloring, the Algorithm Strategy gives $f(w, B) \leq 3 \lceil \frac{w}{2} \rceil$. $\square$

4 Asymptotics for P_5^2

Applying the argument of Corollary 8 to P_5^2 shows that $f(w, P_5^2)$ is bounded by $(1 - o(1))\frac{5}{4}w$ from below and $(1 + o(1))\frac{3}{2}w$ from above. In this section, we improve Algorithm's strategy to obtain $f(w, P_5^2)$ exactly when w is even.

Theorem 10. *We have $\frac{5}{2} \lfloor \frac{w}{2} \rfloor \leq f(w, P_5^2) \leq \frac{5w+2}{4}$ for each w . Consequently, when w is even, we have $f(w, P_5^2) = \lceil \frac{5}{4}w \rceil$.*

Proof. The lower bound follows the same argument as in Corollary 8. For the upper bound, we give a strategy for Algorithm. It is convenient to label the vertices of P_5^2 according to the structure of the complementary graph. The complement of P_5^2 is the union of an isolated vertex v_0 and a path $v_1v_2v_3v_4$. Each color class that Algorithm uses appears on vertices forming a nonempty independent set in P_5^2 ; these are the 5 singletons plus the pairs v_1v_2 , v_2v_3 , and v_3v_4 . For each nonempty set $I \subseteq \{0, \dots, 4\}$, we let y_I be the number of colors assigned by algorithm that appear on vertices in $\{v_i : i \in I\}$ and no other vertex. We view the y_I as variables whose values change throughout the game, and we define the vector $y = (y_0, y_1, y_{12}, y_2, y_{23}, y_3, y_{34}, y_4)$.

We are now able to describe Algorithm's strategy. If Spoiler plays a token at v_0 , then Algorithm responds by assigning the token a new color. (Since v_0 is dominating in P_5^2 , Algorithm has no choice here.) The pairs v_1v_2 and v_3v_4 are *greedy*; that is, if Spoiler plays a token at a vertex in $\{v_1, \dots, v_4\}$, then Algorithm first attempts to extend a color class which appears only on its greedy mate to accommodate the new token. Otherwise, Algorithm assigns the token a new color unless this would result in $\min\{y_2, y_3\} > (w+1)/4$. In this case, Spoiler has played a token at $v_i \in \{v_2, v_3\}$; let j index the other vertex in $\{v_2, v_3\}$. Algorithm assigns the new token at v_i a color which appears only at v_j .

Observe that Algorithm maintains the following invariants. Since v_1v_2 and v_3v_4 are greedy pairs, we have $\min\{y_1, y_2\} = \min\{y_3, y_4\} = 0$. We also have $\min\{y_2, y_3\} \leq (w+1)/4$. Also, since $v_0v_1v_3$, $v_0v_1v_4$, and $v_0v_2v_4$ are triangles in P_5^2 , the width condition implies that each of these triples has at most w tokens. This gives the following three bounds, which we present as a matrix inequality.

$$\begin{array}{c} y_0 \quad y_1 \quad y_{12} \quad y_2 \quad y_{23} \quad y_3 \quad y_{34} \quad y_4 \\ \begin{array}{l} v_0v_1v_3 \\ v_0v_1v_4 \\ v_0v_2v_4 \end{array} \begin{pmatrix} 1 & 1 & 1 & & 1 & 1 & 1 & \\ 1 & 1 & 1 & & & & 1 & 1 \\ 1 & & 1 & 1 & 1 & & 1 & 1 \end{pmatrix} y \leq \begin{pmatrix} w \\ w \\ w \end{pmatrix} \end{array}$$

Suppose that the game is played to its conclusion. We bound the total number of colors S used by Algorithm by examining 4 cases, according to which variables in the pairs $\{y_1, y_2\}$ and $\{y_3, y_4\}$ are zero. Let S_1 , S_2 , and S_3 be the sums of the terms on the left hand sides of the inequalities in rows 1, 2, and 3, respectively. If $y_2 = y_4 = 0$, then we have $S = S_1 + y_2 + y_4 \leq w$. Similarly, if $y_1 = y_3 = 0$, then $S = S_3 + y_1 + y_3 \leq w$. If $y_1 = y_4 = 0$, then we have $S = S_1 + y_2 = S_3 + y_3$ and hence $S \leq w + \min\{y_2, y_3\} \leq (5w+1)/4$.

Otherwise, $y_2 = y_3 = 0$. If also $y_{23} = 0$, then $S = S_2 + y_2 + y_{23} + y_3 \leq w$. Hence we may assume that $y_{23} > 0$, implying that at some point Algorithm extends a color which appears on only one of v_2, v_3 so that it appears on both. Consider the last such time that Algorithm increases y_{23} . Before the corresponding token is played, we have $\max\{y_2, y_3\} > (w+1)/4$ and $\min\{y_2, y_3\} + 1 > (w+1)/4$. After this token is played and algorithm assigns a color increasing y_{23} , we have that $\min\{y_2, y_3\} > (w+1)/4 - 1$ and so $\min\{y_2, y_3\} \geq (w+2)/4 - 1$. However, as

this is the last time that y_{23} is increased and yet $y_2 = y_3 = 0$ at the end of the game, it must be that y_{12} and y_{34} are each incremented at least $(w+2)/4 - 1$ times. Therefore, on termination, we have $y_{12} + y_{34} \geq (w+2)/2 - 2$. With respect to the values at termination, we compute

$$\begin{aligned}
3S &= (S_1 + y_2 + y_4) + (S_2 + y_2 + y_{23} + y_3) + (S_3 + y_1 + y_3) \\
&\leq 3w + y_1 + y_4 + y_{23} \\
&= 3w + S - (y_0 + y_{12} + y_{34}) \\
&\leq 3w + S - \frac{w-2}{2}
\end{aligned}$$

and it follows that $S \leq (5w+2)/4$. □

5 Open Problems

There are many open problems. Perhaps the two most compelling problems are to determine the asymptotics of $f(w, \mathbb{G})$ and $f^*(w, \overline{\mathbb{G}})$. There are natural extensions of both problems to higher dimensions, where $\mathbb{G}_{d,p}$ is the graph on $\mathbb{R}^d$ with $uv \in E(\mathbb{G}_{d,p})$ if and only if the p -norm distance between u and v is less than 1. Chan and Zarrabi-Zadeh [4] noted that if $f^*(w, \overline{\mathbb{G}}) \leq cw$ for some constant c , then $f^*(w, \overline{\mathbb{G}_{d,\infty}}) \leq c2^{d-1}w$. We are not aware of any other work on the higher-dimensional analogues.

It would be nice to obtain $f(w, G)$ asymptotically for the remaining minimally non-online-perfect graphs, when $G = C_5^+$ or G is the bull graph. We believe the following may be approachable.

Conjecture 11. $f(w, C_5^+) = (\frac{5}{4} + o(1))w$.

By constructing large graphs that simulate the flexibility of Spoiler in the classical model, one can show that for each k , there exists a graph G_k such that $f(2, G_k) \geq k$. However, the following question remains unresolved.

Question 1. *Is there a constant c such that $f(w, G) \leq (c + o(1))w$ for each (finite) graph G ?*

We know only that c must be at least $5/4$.

Acknowledgments

We thank Csaba Biro for preliminary discussions on the value of the game on powers of infinite paths.

References

- [1] Yair Bartal, Amos Fiat, and Stefano Leonardi. Lower bounds for on-line graph problems with application to on-line circuit and optical routing. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 531–540, New York, NY, USA, 1996. ACM.

- [2] Yair Bartal, Amos Fiat, and Stefano Leonardi. Lower bounds for on-line graph problems with application to on-line circuit and optical routing. *SIAM Journal on Computing*, 36(2):354–393, 2006.
- [3] Bartłomiej Bosek, Stefan Felsner, Kamil Kłoch, Tomasz Krawczyk, Grzegorz Matecki, and Piotr Micek. On-line chain partitions of orders: a survey. *Order*, 29(1):49–73, 2012.
- [4] Timothy M. Chan and Hamid Zarrabi-Zadeh. A randomized algorithm for online unit clustering. *Theory of Computing Systems*, 45(3):486–496, Oct 2009.
- [5] Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Ann. of Math. (2)*, 164(1):51–229, 2006.
- [6] Martin R. Ehmsen and Kim S. Larsen. Better bounds on online unit clustering. *Theoretical Computer Science*, 500(Supplement C):1 – 24, 2013.
- [7] Leah Epstein and Rob Van Stee. On the online unit clustering problem. *ACM Trans. Algorithms*, 7(1):7:1–7:18, December 2010.
- [8] Magnús M. Halldórsson. Online coloring known graphs. *Electron. J. Combin.*, 7:Research Paper 7, 9, 2000.
- [9] Magnus M. Halldórsson and Mario Szegedy. Lower bounds for on-line graph coloring. *Theoretical Computer Science*, 130(1):163 – 174, 1994.
- [10] Jun Kawahara and Koji M. Kobayashi. An improved lower bound for one-dimensional online unit clustering. *Theoretical Computer Science*, 600(Supplement C):171 – 173, 2015.
- [11] H. A. Kierstead. On-line coloring k -colorable graphs. *Israel Journal of Mathematics*, 105(1):93–104, Dec 1998.
- [12] Hal A. Kierstead. Coloring graphs on-line. In Amos Fiat and Gerhard J. Woeginger, editors, *Online Algorithms: The State of the Art*, pages 281–305. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.
- [13] László Lovász, Michael Saks, and W. T. Trotter. An on-line graph coloring algorithm with sublinear performance ratio. *Discrete Math.*, 75(1-3):319–325, 1989.