An On-line Competitive Algorithm for Coloring P_8-free Bipartite Graphs

Piotr Micek[1] and Veit Wiechert[2]

[1] Theoretical Computer Science Department, Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland,
piotr.micek@tcs.uj.edu.pl
[2] Technische Universität Berlin, Berlin, Germany,
wiechert@math.tu-berlin.de

Abstract. The existence of an on-line competitive algorithm for coloring bipartite graphs remains a tantalizing open problem. So far there are only partial positive results for bipartite graphs with certain small forbidden graphs as induced subgraphs, in particular for P_7-free bipartite graphs. We propose a new on-line competitive coloring algorithm for P_8-free bipartite graphs. Our proof technique improves the result, and shortens the proof, for P_7-free bipartite graphs.

1 Introduction

A *proper coloring* of a graph is an assignment of colors to its vertices such that adjacent vertices receive distinct colors. It is easy to devise an (linear time) algorithm 2-coloring bipartite graphs. Now, imagine that an algorithm receives vertices of a graph to be colored one by one knowing only the adjacency status of the vertex to vertices already colored. The color of a vertex must be fixed before the algorithm sees the next vertices and it cannot be changed afterwards. This kind of algorithm is called an *on-line* coloring algorithm.

Formally, an *on-line graph* (G, π) is a graph G with a permutation π of its vertices. An *on-line coloring algorithm* A takes an on-line graph (G, π), say $\pi = (v_1, \ldots, v_n)$ as an input. It produces a proper coloring of the vertices of G where the color of a vertex v_i, for $i = 1, \ldots, n$, depends only on the subgraph of G induced by $v_1, \ldots, v_i$. It is convenient to imagine that consecutive vertices along π are revealed by some adaptive (malicious) adversary and the coloring process is a game between that adversary and an on-line algorithm.

Still, it is an easy exercise that if an adversary presents a bipartite graph and all the time the graph presented so far is connected then there is an on-line

V. Wiechert is supported by the Deutsche Forschungsgemeinschaft within the research training group 'Methods for Discrete Structures' (GRK 1408).
P. Micek is supported by Polish National Science Center UMO-2011/03/D/ST6/01370.

algorithm 2-coloring these graphs. But if an adversary can present a bipartite graph without any additional constraints then (s)he can trick out any on-line algorithm to use an arbitrary number of colors!

Indeed, there is a strategy for adversary forcing any on-line algorithm to use at least $\lfloor \log n \rfloor + 1$ colors on a forest of size n. On the other hand, the First-Fit algorithm (that is an on-line algorithm coloring each incoming vertex with the least admissible natural number) uses at most $\lfloor \log n \rfloor + 1$ colors on forests of size n. When the game is played on bipartite graphs, an adversary can easily trick out First-Fit and force $\lceil \frac{n}{2} \rceil$ colors on a bipartite graph of size n. Lovász, Saks and Trotter [12] proposed a simple on-line algorithm (in fact as an exercise; see also [8]) using at most $2 \log n + 1$ colors on bipartite graphs of size n. This is best possible up to an additive constant as Gutowski et al. [4] showed that there is a strategy for adversary forcing any on-line algorithm to use at least $2 \log n - 10$ colors on a bipartite graph of size n.

For an on-line algorithm A by $A(G, \pi)$ we mean the number of colors that A uses against an adversary presenting graph G with presentation order π.

An on-line coloring algorithm A is *competitive* on a class of graphs $\mathcal{G}$ if there is a function f such that for every $G \in \mathcal{G}$ and permutation π of vertices of G we have $A(G, \pi) \leqslant f(\chi(G))$. As we have discussed, there is no competitive coloring algorithm for forests. But there are reasonable classes of graphs admitting competitive algorithms, e.g., interval graphs can be colored on-line with at most $3\chi - 2$ (where χ is the chromatic number of the presented graph; see [11]) and cocomparability graphs can be colored on-line with a number of colors bounded by a tower function in terms of χ (see [9]). Also classes of graphs defined in terms of forbidden induced subgraphs were investigated in this context. For example, P_4-free graphs (also known as cographs) are colored by First-Fit optimally, i.e. with χ colors, since any maximal independent set meets all maximal cliques in a P_4-free graph. Also P_5-free graphs can be colored on-line with $O(4^{\chi})$ colors (see [10]). And to complete the picture there is no competitive algorithm for P_6-free graphs as Gyárfás and Lehel [6] showed a strategy for adversary forcing any on-line algorithm to use an arbitrary number of colors on bipartite P_6-free graphs.

Confronted with so many negative results, it is not surprising that Gyárfás, Király and Lehel [5] introduced a relaxed version of competitiveness of an on-line algorithm. The idea is to measure the efficiency of an on-line algorithm compared to the best on-line algorithm for a given input (instead of the chromatic number). Hence, the *on-line chromatic number* of a graph G is defined as

$$\chi_*(G) = \inf_A \max_\pi A(G, \pi),$$

where the infimum is taken over all on-line algorithms A and the maximum is taken over all permutation π of vertices of G. An on-line algorithm A is *on-line competitive* for a class of graphs $\mathcal{G}$, if there is a function f such that for every $G \in \mathcal{G}$ and permutation π of vertices of G we have $A(G, \pi) \leqslant f(\chi_*(G))$.

Why are on-line competitive algorithms interesting? Imagine that you design an algorithm and the input graph is not known in advance. If your algorithm is on-line competitive then you have an insurance that whenever your algorithm uses many colors on some graph G with presentation order π then any other on-line algorithm may be also forced to use many colors on the same graph G with some presentation order π' (and it includes also those on-line algorithms which are designed only for this single graph G!). The idea of comparing the outputs of two on-line algorithms directly (not via the optimal off-line result) is present in the literature. We refer the reader to [1], where a number of measures are discussed in the context of on-line bin packing problems. In particular, the relative worst case ratio, introduced there, is closely related to our setting for on-line colorings.

It may be true that there is an on-line competitive algorithm for all graphs. This is open as well for the class of all bipartite graphs. To the best of the authors knowledge, there is no promissing approach for the negative answer for these questions. However, there are some partial positive results. Gyárfás and Lehel [7] have shown that First-Fit is on-line competitive for forests and it is even optimal in the sense that if First-Fit uses k colors on G then the on-line chromatic number of G is k as well. They also have shown [5] that First-Fit is competitive (with an exponential bounding function) for graphs of girth at least 5. Finally, Broersma, Capponi and Paulusma [2] proposed an on-line coloring algorithm for P_7-free bipartite graphs using at most $8\chi_* + 8$ colors on graphs with on-line chromatic number χ_*.

The contribution of this paper is the following theorem.

Theorem 1. *There is an on-line competitive algorithm that properly colors P_8-free bipartite graphs. Moreover, there are on-line coloring algorithms for bipartite graphs using at most*

(i) $4\chi_* - 2$ *colors on P_7-free graphs,*

(ii) $3(\chi_* + 1)^2$ *colors on P_8-free graphs,*

where χ_* is the on-line chromatic number of the presented graph.

We wish to point out that we can improve the given bounds on the absolute values. But since this would need a more involved analysis and on the other hand the improvement would be small, we decided to present the weaker results. Furthermore, we can use our techniques to show that there is an on-line competitive algorithm for coloring P_9-free bipartite graphs. This result is not presented in this paper due to the page limitation.

2 Forcing structure

In this section we introduce a family of bipartite graphs without long induced paths (P_6-free) and with arbitrarily large on-line chromatic number. All the on-line algorithms we are going to study have the property that whenever they use many colors on a graph G then G has a *large* graph from our family as an induced subgraph and therefore G has a large on-line chromatic number, as desired.

A connected bipartite graph G has a unique partition of vertices into two independent sets. We call these partition sets the *sides* of G. A vertex v in a

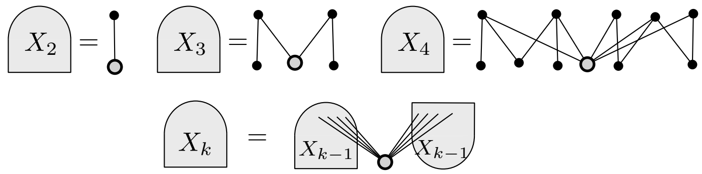

Fig. 1. Family of bipartite graphs

bipartite graph G is *universal* to a subgraph C of G if v is adjacent to all vertices of C in one of the sides of G.

Consider a family of connected bipartite graphs $\{X_k\}_{k\geqslant 1}$ defined recursively as follows. Each X_k has a distinguished vertex called the *root*. The side of X_k containing the root of X_k, we call the *root side* of X_k, while the other side we call the *non-root side*. X_1 is a single vertex being the root. X_2 is a single edge with one of its vertices being the root. X_k, for $k \geqslant 3$, is a graph formed by two disjoint copies of X_{k-1}, say X^1_{k-1} and X^2_{k-1}, with no edge between the copies, and one extra vertex v adjacent to all vertices on the root side of X^1_{k-1} and all vertices on the non-root side of X^2_{k-1}. The vertex v is the root of X_k. Note that for each k, the root of X_k is adjacent to the whole non-root side of X_k, i.e., the root of X_k is universal in X_k. See Figure 1 for a schematic drawing of the definition of X_k.

A family of P_6-free bipartite graphs with arbitrarily large on-line chromatic number was first presented in [6]. The family $\{X_k\}_{k\geqslant 1}$ was already studied in [3], in particular Claim 2 is proved there. Due to page limitation we omit the proof here. We encourage the reader to verify that X_k is P_6-free for $k \geqslant 1$.

Claim 2 *If G contains X_k as an induced subgraph, then $\chi_*(G) \geqslant k$.*

3 P_7-free bipartite graphs

In this section we present an on-line algorithm using at most $4\chi_* - 2$ colors on P_7-free bipartite graphs with on-line chromatic number χ_*. The algorithm itself, see Algorithm 1, is taken from [3, 2] where it is called *BicolorMax* and proved that it uses at most $2\chi_* - 1$ colors on P_6-free bipartite graphs and at most $8\chi_* + 8$ colors on P_7-free bipartite graphs with on-line chromatic number χ_*. Thus, we improve the bounding function for the P_7-free case and yet we present a much simpler proof.

Algorithm 1 uses two disjoint pallettes of colors, $\{a_n\}_{n\geqslant 1}$ and $\{b_n\}_{n\geqslant 1}$. In the following whenever the algorithm fixes a color of a vertex v we are going to refer to it by color(v). Also for any set of vertices X we denote color$(X) = \{\text{color}(x) \mid x \in X\}$. We say that v has *color index i* if color$(v) \in \{a_i, b_i\}$.

Suppose an adversary presents a new vertex v. Let $G_i[v]$ be the subgraph on the vertices presented so far that have a color from $\{a_1, \ldots, a_i, b_1, \ldots, b_i\}$ and with one extra vertex, namely v, which is uncolored yet. Now, $C_i[v]$ denotes the connected component of $G_i[v]$ containing vertex v. Furthermore, let $C_i(v)$ be the graph $C_i[v]$ without vertex v. The graph $C_i(v)$ is not necessarily connected and in fact, as we will show, whenever v has color index $k \geqslant 2$, $C_i(v)$ contains at least 2 connected components for $1 \leqslant i < k$. Note that by these definitions it already follows that if $w \in C_i(v)$ then $C_j[w] \subseteq C_i(v)$ for all $j \leqslant i$ since w is presented before v and has a smaller color index than v. We say that a color c is *mixed* in a connected induced subgraph C of G if c is used on vertices on both sides of C.

Now we are ready for a description of Algorithm 1.

Algorithm 1: On-line competitive for P_7-free bipartite graphs

```
an adversary introduces a new vertex v
m ← max {i ⩾ 1 | a_i is mixed in C_i[v]} + 1          // max {} := 0
let I_1, I_2 be the sides of C_m[v] such that v ∈ I_1
if a_m ∈ color(I_2) then color(v) = b_m
else color(v) = a_m
```

It is not hard to see that Algorithm 1 colors bipartite graphs properly. We leave it as an exercise for the reader.

Claim 3 *Algorithm 1 gives a proper coloring of on-line bipartite graphs.*

The following claim is already proven in [3, 2], for the sake of completeness we added it here. It is quite essential for the other proofs.

Claim 4 *Suppose an adversary presents a bipartite graph G to Algorithm 1. Let $v \in G$ and let x, y be two vertices from opposite sides of $C_i[v]$ both colored with a_i. Then x and y lie in different connected components of $C_i(v)$.*

Proof. Let v, x and y be like in the statement of the claim. We are going to prove that at any moment after the introduction of x and y, x and y lie in different connected components of the subgraph spanned by vertices colored with $a_1, b_1, \ldots, a_i, b_i$.

Say x is presented before y. First note that $x \notin C_i[y]$ as otherwise x had to be on the opposite side to y (because it is on the opposite side at the time v is presented) and therefore y would receive color b_i. Now consider any vertex w presented after y and suppose the statement is true before w is introduced. If $x \notin C_i[w]$ or $y \notin C_i[w]$ then whatever color is used for w this vertex does not merge the components of x and y in the subgraph spanned by vertices presented so far and colored with $a_1, b_1, \ldots, a_i, b_i$. Otherwise $x, y \in C_i[w]$. This means that color a_i is mixed in $C_i[w]$ and therefore w receives a color with an index at least $i + 1$. Thus, the subgraph spanned by the vertices of the colors $a_1, b_1, \ldots, a_i, b_i$ stays the same and x and y remain in different connected components of this graph.

Since all vertices in $C_i(v)$ are colored with $a_1, b_1, \ldots, a_i, b_i$, we conclude that x and y lie in different components of $C_i(v)$. □

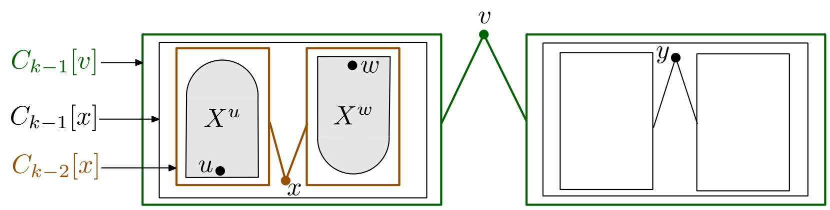

Fig. 2. Schematic drawing of some defined components. The color classes of G belong to the top and bottom side of the boxes. Vertex v has color index k. Vertices x and y certify that color a_{k-1} is mixed in $C_{k-1}[v]$. The component $C_{k-1}[v]$ consists of both green boxes and vertex v, which is merging these.

As a consequence of Claim 4 it holds that if v has color index $k \geqslant 2$ then $C_i(v)$ is disconnected for all $1 \leqslant i < k$. This is simply because there are vertices x and y certifying that color a_i is mixed in $C_i[v]$. It also means that if we forget about the vertices presented so far that are not in $C_i[v]$, vertex v is merging independent connected subgraphs of G which are intuitively large in the case $i = k-1$ as they contain vertices (x and y) with a high color index. See Figure 2 for a better understanding.

Claim 5 *Suppose an adversary presents a P_7-free bipartite graph to Algorithm 1. Let v be a vertex with color index k and $1 \leqslant i < k$. Then v is universal to all but possibly one component of $C_i(v)$.*

Proof. Suppose to the contrary that there are 2 components C_1 and C_2 in $C_i(v)$ to which v is not universal. Then there are vertices $v_1 \in C_1$ and $v_2 \in C_2$ that both have distance at least 3 from v in $C_i[v]$. It follows that a shortest path connecting v_1 and v in $C_i[v]$, combined with a shortest path connecting v and v_2 in $C_i[v]$ results in an induced path of length at least 7, a contradiction. □

Theorem 6. *Algorithm 1 uses at most $4\chi_* - 2$ colors on P_7-free bipartite graphs with an on-line chromatic number χ_*.*

Proof. Let G be a P_7-free bipartite graph that is presented by an adversary to Algorithm 1. Suppose color a_{2k} for some $k \geqslant 1$ is used on a vertex v of G. We prove by induction on k that $C_{2k-1}[v]$ contains X_{k+1} as an induced subgraph such that v corresponds to the root of X_{k+1}.

If $k = 1$ then v is colored with a_2 and v must have neighbors in $C_1[v]$. We embed X_2 being a single edge onto v and its neighbor. So suppose $k \geqslant 2$ and v is colored with a_{2k}. Since color a_{2k-1} is mixed in $C_{2k-1}[v]$ there are vertices x and y of color a_{2k-1} lying on opposite sides of $C_{2k-1}[v]$. By Claim 4 vertices x and y lie in different components of $C_{2k-1}(v)$, say C^x and C^y. If we forget about the color index of v in Figure 2 then C^x corresponds to the left and C^y to the right green box. Using Claim 5 we conclude that v must be universal to at least one of C^x and C^y. We can assume that this is true for C^x. Since the color index of x is smaller than of v, we have that $C_{2k-2}[x] \subseteq C^x$. Now consider vertices u and w in $C_{2k-2}[x]$ certifying that color a_{2k-2} is mixed in $C_{2k-2}[x]$. Let C^u and C^w be the components of $C_{2k-2}(x)$ containing u and v, respectively, which are distinct by Claim 4 (see left and right brown box in Figure 2). Observe that $C_{2k-3}[u]$ and $C_{2k-3}[w]$ are subgraphs of C^u and C^w, respectively. By the induction hypothesis there are induced copies X^u and X^w of X_k in $C_{2k-3}[u]$ and $C_{2k-3}[w]$, respectively, such the roots correspond to u and v (see Figure 2). Since $X^u \subseteq C^u$ and $X^w \subseteq C^w$, there is no edge between the copies and v is universal to both them. Using the fact that u and w appear on opposite sides of $C_{2k-1}[v]$ we conclude that v together with X^u and X^w form an X_{k+1} with v being the root of it. This completes the induction.

Let now $k \geqslant 1$ be maximal such that Algorithm 1 used the color a_{2k} on a vertex of G. It might be that also the colors b_{2k}, a_{2k+1} and b_{2k+1} are used, but not any a_j or b_j for $j \geqslant 2k+2$. Thus, Algorithm 1 used at most $4k+2$ colors. On the other hand, G contains an X_{k+1} and by Claim 2 it follows that $\chi_*(G) \geqslant k+1$. We conclude that Algorithm 1 used at most $4k+2 \leqslant 4\chi_*(G)-2$ colors. □

4 P_8-free bipartite graphs

Inspired by Algorithm 1 and an argument from the previous section we present a new on-line algorithm for bipartite graphs and refer to it by Algorithm 2. In this section we prove that this algorithm is on-line competitive for P_8-free bipartite graphs.

Algorithm 2 uses three disjoint pallettes of colors, $\{a_n\}_{n\geqslant 1}$, $\{b_n\}_{n\geqslant 1}$ and $\{c_n\}_{n\geqslant 1}$. Similar to the case of P_7-freeness we make the following definitions. Whenever the algorithm fixes a color of a vertex v we are going to refer to it by color(v). Also for any set of vertices X we denote $\text{color}(X) = \{\text{color}(x) \mid x \in X\}$. We say that a vertex v has *color index* i, if $\text{color}(v) \in \{a_i, b_i, c_i\}$.

Now, suppose an adversary presents a new vertex v of a bipartite graph G. Then let $G_i[v]$ be the subgraph spanned by the vertices presented so far and colored with a color from $\{a_1, \ldots, a_i, b_1, \ldots, b_i, c_1, \ldots, c_i\}$ and vertex v, which is uncolored yet. With $C_i[v]$ we denote the connected component of $G_i[v]$ containing v. For convenience put $C_0[v] = \{v\}$. Furthermore, let $C_i(v)$ be the graph $C_i[v]$ without vertex v. For a vertex x in $C_i(v)$ it will be convenient to denote by $C_i^x(v)$ the connected component of $C_i(v)$ that contains x. We say that a color c is *mixed* in a connected subgraph C of G if c is used on vertices on both sides of C.

Again, a proof for the proper coloring we leave as a fair exercise.

Claim 7 *Algorithm 2 gives a proper coloring of on-line bipartite graphs.*

To prove the following claim, it is enough to follow the lines of the proof for Claim 4.

Algorithm 2: On-line competitive for P_8-free bipartite graphs

an adversary introduces a new vertex v
$m \leftarrow \max\{i \geqslant 1 \mid a_i \text{ is mixed in } C_i[v]\} + 1$ // $\max\{\} := 0$
let I_1, I_2 be the sides of $C_m[v]$ such that $v \in I_1$
if $a_m \in \text{color}(I_2)$ **then** $\text{color}(v) = b_m$
else if $c_m \in \text{color}(I_2)$ **then** $\text{color}(v) = a_m$
else if $\exists u \in I_1 \cup I_2$ *and* $\exists u' \in I_2$ *such that* u *has color index* $j \geqslant m - \lfloor\sqrt{m-1}\rfloor$ *and* u' *is universal to* $C_{j-1}[u]$ **then** $\text{color}(v) = c_m$
else $\text{color}(v) = a_m$

Claim 8 *Suppose an adversary presents a bipartite graph G to Algorithm 2. Let $v \in G$ and let x, y be two vertices from opposite sides of $C_i[v]$ both colored with a_i. Then x and y lie in different connected components of $C_i(v)$.*

Now, whenever a vertex v has color index $k \geqslant 2$, let v_1 and v_2 be the first introduced vertices in $C_{k-1}(v)$ that certify that color a_{k-1} is mixed in $C_{k-1}(v)$. By Claim 8 it follows that $C^{v_1}_{k-1}(v)$ and $C^{v_2}_{k-1}(v)$ are distinct (and in particular no edge is between them). In the following we will refer to v_1 and v_2 as the *children* of v.

In the contrast to the P_7-free case, observe that v does not have to be universal to at least one of the components $C^{v_1}_{k-1}(v)$ and $C^{v_2}_{k-1}(v)$ if G is only P_8-free. However, as the next claim shows, we can expect a vertex on the other side of v in $C_{k-1}(v)$ that is universal to some component in this case. This component will be only slightly smaller than $C_{k-1}(v)$ whenever we use Claim 9 in our final proof. Also note that Algorithm 2 is inspired by this observation.

Claim 9 *Suppose an adversary presents a P_8-free bipartite graph G to Algorithm 2. Let x be a vertex with color index $i \geqslant 2$. Suppose that vertex $y \in C_{i-1}(x)$, with color index j, lies on the other side of x in G and y is not adjacent to x. Then one of the following holds:*

(i) *x has a child x' such that x is universal to $C_{i-2}[x']$, or*
(ii) *x has a neighbor in $C^{y}_{i-1}(x)$ that is universal to $C_{j-1}[y]$.*

Proof. We can assume that y has color index $j \geqslant 2$, as otherwise $C_{j-1}[y] = C_0[y] = \{y\}$ and vacuosly any neighbor of x is universal to $C_{j-1}[y]$ (as the side it must be adjacent to is empty). Let x_1 and x_2 be the children of x. By Claim 8 the components $C^{x_1}_{i-1}(x)$ and $C^{x_2}_{i-1}(x)$ are distinct. Vertex y is contained in at most one of them, say $y \notin C^{x_1}_{i-1}(x)$, and therefore $C^{x_1}_{i-1}(x)$ and $C^{y}_{i-1}(x)$ are distinct. In order to prove the claim suppose that (i) is not satisfied. Then x is not universal to $C_{i-2}[x_1]$ and in particular not to $C^{x_1}_{i-1}(x)$. It follows that there is an induced path of length 4 ending at x in $C^{x_1}_{i-1}[x]$. As G is P_8-free we conclude that $C^{y}_{i-1}[x]$ does not contain an induced path of length 5 with one endpoint in x. With this observation in mind we will prove (ii) now.

First, let us consider the case that x has a neighbor z in $C_{j-1}[y] \subseteq C^{y}_{i-1}(x)$ (see Figure 3 for this case). Since x and y are not adjacent we have $y \neq z$.

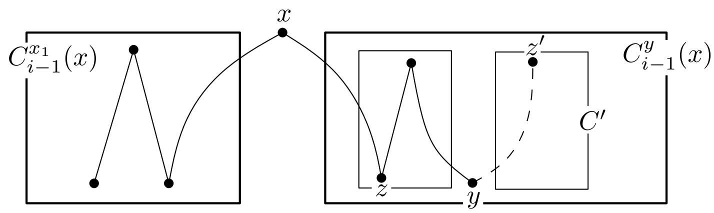

Fig. 3. Claim 9: Situation in which x has a neighbor z in $C_{j-1}[y]$.

As $C^z_{j-1}[y]$ is connected, there is an induced path P connecting x and y that has only vertices of $C^z_{j-1}(y)$ as inner vertices. Clearly, P has even length at least 4. Now the color index of y, namely $j \geqslant 2$, assures the existence of a mixed pair in $C_{j-1}[y]$ and with Claim 8 it follows that $C_{j-1}(y)$ has at least two connected components. In particular, there is a component C' of $C_{j-1}(y)$ other than $C^z_{j-1}(y)$. Clearly, y has a neighbor z' in C', which we use to prolong P at y. Since there is no edge between $C^z_{j-1}(y)$ and C', vertex z' is not adjacent to the inner vertices of P. And as G is bipartite z' cannot be adjacent to x. We conclude the existence of an induced path of length 5 in $C^y_{i-1}[x]$ with x and z' being its endpoints, a contradiction.

Second, we consider the case that x has no neighbor in $C_{j-1}[y]$. By our assumptions a shortest path connecting x and y in $C^y_{i-1}[x]$ must have length exactly 4. Let $P = (x, r, s, y)$ be such a path. We claim that vertex r is universal to $C_{j-1}[y]$. Suppose to contrary that there is a vertex s' in $C_{j-1}[y]$ which is on the other side of y and which is not adjacent to r. Let $Q = (y, s_1, r_1, \ldots, s_{\ell-1}, r_{\ell-1}, s_\ell = s')$ be a shortest path connecting y and s' in $C_{j-1}[y]$. For convenience put $s_0 = s$. Now we choose the minimal $m \geqslant 0$ such that r is adjacent to s_m but not to s_{m+1}. Such an m exists since r is adjacent to $s_0 = s$ but not to s_ℓ. If $m = 0$ then the path (x, r, s, y, s_1) is an induced path of length 5 and if $m > 0$ then the path $(x, r, s_m, r_m, s_{m+1})$ has length 5 and is induced unless x and r_m are adjacent. But the latter is not possible since x has no neighbor in $C_{j-1}[y]$. Thus, in both cases we get a contradiction and we conclude that r is universal to $C_{j-1}[y]$. □

In the following we write $v \to_i w$ for $v, w \in G$, if there is a sequence $v = x_1, \ldots, x_j = w$ with $j \leqslant i$ and $x_{\ell+1}$ is a child of x_ℓ, for all $\ell \in \{1, \ldots, j-1\}$. Moreover, we define $S_i(v) = \{w \mid v \to_i w\}$.

We make some immediate observations concerning this definition. Let $v \in G$ be a vertex with color index $k \geqslant 2$. Then all vertices in $S_i(v)$ have color index at least $k-i+1$. Furthermore, each vertex in $S_i(v)$ is connected to v by a path in G and all vertices in the path, except v, have color index at most $k-1$. This proves that $S_i(v) \subseteq C_{k-1}[v]$, for all $i \geqslant 1$. Note also that if v_1 and v_2 are the children of v then we have $S_i(v) = \{v\} \cup S_{i-1}(v_1) \cup S_{i-1}(v_2)$ and $S_{i-1}(v_1) \subseteq C^{v_1}_{k-1}(v)$, $S_{i-1}(v_2) \subseteq C^{v_2}_{k-1}(v)$. By Claim 8, we get that $C^{v_1}_{k-1}(v)$ and $C^{v_2}_{k-1}(v)$ are distinct. In particular, $S_{i-1}(v_1)$ and $S_{i-1}(v_2)$ are disjoint and there is no edge between them.

For a vertex $v \in G$, $S_i(v)$ is *complete* in G if for every $u, w \in S_i(v)$ such that $u \to_i w$ and u, w lying on opposite sides of G, we have u and w being adjacent in G. Note that v is a universal vertex in $S_i(v)$, provided $S_i(v)$ is complete.

Claim 10 *Suppose an adversary presents a bipartite graph G to Algorithm 2. Let $v \in G$ be a vertex with color index k and let $k \geqslant i \geqslant 1$. If $S_i(v)$ is complete then $S_i(v)$ contains an induced copy of X_i in G with v being the root of the copy.*

Proof. We prove the claim by induction on i. For $i = 1$ we work with $S_1(v)$ and X_1 being graphs with one vertex only, so the statement is trivial. For $i \geqslant 2$, let v_1 and v_2 be the children of v. Recall that $S_i(v) = \{v\} \cup S_{i-1}(v_1) \cup S_{i-1}(v_2)$. Since $S_i(v)$ is complete it also follows that $S_{i-1}(v_1)$ and $S_{i-1}(v_2)$ are complete. So by the induction hypothesis there are induced disjoint copies X^1_{i-1}, X^2_{i-1} of X_{i-1} in $S_{i-1}(v_1)$ and $S_{i-1}(v_2)$, respectively, and rooted in v_1, v_2, respectively. Recall that $S_{i-1}(v_1)$ and $S_{i-1}(v_1)$ are disjoint and there is no edge between them. Thus, the copies X^1_{i-1} and X^2_{i-1} of X_{i-1} are disjoint and there is no edge between them, as well. Since $S_i(v)$ is complete v is universal to both of the copies, and since v_1 and v_2 lie on opposite sides in G we get that the vertices of $X^1_{i-1} \cup X^2_{i-1} \cup \{v\}$ induce a copy of X_i in G. □

Claim 11 *Suppose an adversary presents a P_8-free bipartite graph G to Algorithm 2 and suppose vertex v is colored with a_k. Then $C_k[v]$ contains an induced copy of $X_{\lfloor\sqrt{k}\rfloor}$ such that its root lies on the same side as v in G.*

Proof. The claim is easy to verify for $1 \leqslant k \leqslant 8$. So suppose that $k \geqslant 9$. Consider the set $S_{\lfloor\sqrt{k}\rfloor}(v)$. If it is complete then by Claim 10 we get an induced copy of $X_{\lfloor\sqrt{k}\rfloor}$ with a root mapped to v. So we are done.

From now on we assume that $S_{\lfloor\sqrt{k}\rfloor}(v)$ is not complete. Let (I_1, I_2) be the bipartition of $C_k[v]$ such that $v \in I_1$. First, we will prove that there are vertices $z, z' \in C_k[v]$ such that $z' \in I_1$, z has color index $\ell \geqslant k - \lfloor\sqrt{k-1}\rfloor$ and z' is universal to $C_{\ell-1}[z]$. To do so we consider the reason why Algorithm 2 colors v with a_k instead of b_k or c_k.

The first possible reason is that the second if-condition of the algorithm is satisfied, that is, there is a vertex $u \in I_2$ colored with c_k. Now u can only receive color c_k if there are vertices $w, w' \in C_k[u]$ such that w' is on the other side of u in $C_k[u]$, w has color index $j \geqslant k - \lfloor\sqrt{k-1}\rfloor$ and w' is universal to $C_{j-1}[w]$. Since $C_k(u) \subseteq C_k(v)$ and $u \in I_2$ we have $w' \in I_1$. Therefore, w and w' prove the existence of vertices that we are looking for in this case.

The second reason for coloring v with a_k is that Algorithm 2 reaches its last line. In particular this means, that there is no vertex of color a_k or c_k in I_2. Now we are going to make use of the fact that $S_{\lfloor\sqrt{k}\rfloor}(v)$ is not complete. There are vertices $x, y \in S_{\lfloor\sqrt{k}\rfloor}(v) \subseteq C_k[v]$ such that $x \to_{\lfloor\sqrt{k}\rfloor} y$, vertices x and y lie on different sides of $C_k[v]$ and are not adjacent. Let i and j be the color indices of x and y, respectively. Note that $k \geqslant i > j \geqslant k - \lfloor\sqrt{k}\rfloor + 1$. By Claim 9 vertex x has a child x' such that x is universal to $C_{i-2}[x']$ or x has a neighbor $r \in C^y_{i-1}(x)$ that is universal to $C_{j-1}[y]$. In the first case let w' be x and w be x', and in the second case we set w' to be r and w to be y. Then, in both cases it holds that w' is universal to $C_{\ell-1}[w]$, where ℓ is the color index of w. Furthermore we have

$$\ell \geqslant \min\{i-1, j\} \geqslant k - \lfloor\sqrt{k}\rfloor + 1 \geqslant k - \lfloor\sqrt{k-1}\rfloor$$

for all $k \geqslant 1$. Since $w' \in C_k[v]$ we have $w' \in I_1$ or $w' \in I_2$. However, the latter is not possible as otherwise w and w' would fulfill the conditions of the third if-statement in Algorithm 2, which contradicts the fact that Algorithm 2 reached the last line for v. We conclude that $w' \in I_1$, which completes the proof of our subclaim.

So suppose we have vertices z and z' like in our subclaim. Let z_1 and z_2 be the children of z (they exist as $\ell \geqslant 2$). Both vertices received color $a_{\ell-1}$ and are on different sides of G. By the induction hypothesis $C_{\ell-1}[z_1]$ and $C_{\ell-1}[z_2]$ contain a copy of $X_{\lfloor\sqrt{\ell-1}\rfloor}$ such that the roots are on the same side as z_1 and z_2, respectively. Since there is no edge between $C_{\ell-1}[z_1]$ and $C_{\ell-1}[z_2]$ and both are contained in $C_\ell[z]$, it follows that z' together with the copies of $X_{\lfloor\sqrt{\ell-1}\rfloor}$ induce a copy of $X_{\lfloor\sqrt{\ell-1}\rfloor+1}$ that has z' as its root. Since $C_\ell[z]$ is contained in $C_k[v]$ and z' is on the same side as v and since

$$\lfloor\sqrt{\ell-1}\rfloor + 1 \geqslant \left\lfloor\sqrt{k - \lfloor\sqrt{k-1}\rfloor - 1}\right\rfloor + 1 \geqslant \lfloor\sqrt{k}\rfloor,$$

for all $k \geqslant 1$, this completes the proof. □

Now we are able to prove our main theorem.

Theorem 12. *On each on-line P_8-free bipartite graph (G, π), Algorithm 2 uses at most $3(\chi_*(G)+1)^2$ colors.*

Proof. Let k be the highest color index on the vertices of G. By the definition of Algorithm 2 the color a_k appears on some vertex of G. Using Claim 11 it follows that G contains $X_{\lfloor\sqrt{k}\rfloor}$. Now, by Claim 2, $\chi_*(G) \geqslant \lfloor\sqrt{k}\rfloor$ and therefore Algorithm 2 uses at most $3k \leqslant 3(\lfloor\sqrt{k}\rfloor + 1)^2 \leqslant 3(\chi_*(G)+1)^2$ colors. □

References

1. Joan Boyar and Lene M. Favrholdt. The relative worst order ratio for online algorithms. *ACM Trans. Algorithms*, 3(2):Art. 22, 24, 2007.
2. H. J. Broersma, A. Capponi, and D. Paulusma. A new algorithm for on-line coloring bipartite graphs. *SIAM Journal on Discrete Mathematics*, 22(1):72–91, February 2008.
3. H.J. Broersma, A. Capponi, and D. Paulusma. On-line coloring of h-free bipartite graphs. In Tiziana Calamoneri, Irene Finocchi, and GiuseppeF. Italiano, editors, *Algorithms and Complexity*, volume 3998 of *Lecture Notes in Computer Science*, pages 284–295. Springer Berlin Heidelberg, 2006.
4. Grzegorz Gutowski, Jakub Kozik, and Piotr Micek. Lower bounds for on-line graph colorings. submitted.

5. A. Gyárfás, Z. Király, and J. Lehel. On-line competitive coloring algorithms. Technical report TR-9703-1, available online at http://www.cs.elte.hu/tr97/tr9703-1.ps, 1997.
6. A. Gyárfás and J. Lehel. On-line and first fit colorings of graphs. *Journal of Graph Theory*, 12(2):217–227, 1988.
7. András Gyárfás and Jenő Lehel. First fit and on-line chromatic number of families of graphs. *Ars Combin.*, 29(C):168–176, 1990. Twelfth British Combinatorial Conference (Norwich, 1989).
8. H. A. Kierstead. Recursive and on-line graph coloring. In *Handbook of recursive mathematics, Vol. 2*, volume 139 of *Stud. Logic Found. Math.*, pages 1233–1269. North-Holland, Amsterdam, 1998.
9. Henry A. Kierstead, Stephen G. Penrice, and William T. Trotter. On-line coloring and recursive graph theory. *SIAM Journal on Discrete Mathematics*, 7(1):72–89, 1994.
10. Henry A. Kierstead, Stephen G. Penrice, and William T. Trotter. On-line and first-fit coloring of graphs that do not induce P_5. *SIAM Journal on Discrete Mathematics*, 8(4):485–498, 1995.
11. Henry A. Kierstead and William T. Trotter. An extremal problem in recursive combinatorics. In *Proceedings of the Twelfth Southeastern Conference on Combinatorics, Graph Theory and Computing, Vol. II*, volume 33 of *Congressus Numerantium*, pages 143–153, 1981.
12. László Lovász, Michael E. Saks, and William T. Trotter. An on-line graph coloring algorithm with sublinear performance ratio. *Discrete Mathematics*, 75(1-3):319–325, 1989.