Discrete Mathematics and Theoretical Computer Science DMTCS vol. **15:2**, 2013, 195–214

On-line Ranking of Split Graphs

Piotr Borowiecki[†] and Dariusz Dereniowski[‡]

Dep. of Algorithms and System Modeling, Faculty of Electronics, Telecommunications and Informatics,
Gdańsk University of Technology, Poland

received 21st May 2009, revised 20th May 2012, 27th Feb. 2013, 29th July 2013, accepted 9th Aug. 2013.

A vertex ranking of a graph G is an assignment of positive integers (colors) to the vertices of G such that each path connecting two vertices of the same color contains a vertex of a higher color. Our main goal is to find a vertex ranking using as few colors as possible. Considering on-line algorithms for vertex ranking of split graphs, we prove that the worst case ratio of the number of colors used by any on-line ranking algorithm and the number of colors used in an optimal off-line solution may be arbitrarily large. This negative result motivates us to investigate semi on-line algorithms, where a split graph is presented on-line but its clique number is given in advance. We prove that there does not exist a $(2-\varepsilon)$-competitive semi on-line algorithm of this type. Finally, a 2-competitive semi on-line algorithm is given.

Keywords: graph coloring, graph ranking, on-line algorithm, greedy algorithm, semi on-line algorithm, advice complexity, split graph

---

1 Introduction

In this paper we consider the vertex ranking problem for simple, finite, undirected graphs $G = (V, E)$ with the vertex set V, edge set E and *order* $n = |V(G)|$. A *coloring* of a graph $G = (V, E)$ is a function $c : V(G) \to \mathbb{Z}^+$ such that for each $uv \in E(G)$ we have $c(u) \neq c(v)$. Moreover, if each path between two vertices of the same color q contains a vertex of color greater than q, then c is a *ranking* of G. A ranking that uses k colors is called a k*-ranking* and the smallest k for which there exists a k-ranking of G, denoted by $\chi_r(G)$, is the *ranking number* of G. See also [9] for a survey on graph rankings.

In what follows we use the standard graph theoretical notation and terminology. In particular, given $U \subseteq V(G)$, $G[U]$ denotes the subgraph of G *induced* by U with vertex set U and edge set $E(G[U]) = \{uv \in E(G) \,|\, u, v \in U\}$. If H is an induced subgraph of G, it is customary to write $H \leq G$, while $H \simeq G$ is used when H is isomorphic to G. The complement $\overline{G}$ of a graph G is the graph with $V(\overline{G}) = V(G)$ and $E(\overline{G}) = \{uv \,|\, uv \notin E(G)\}$. For $v \in V(G)$, $N_G(v) = \{u \in V(G) \,|\, uv \in E(G)\}$ is the *neighborhood* of v in G, while $N_G[v] = N_G(v) \cup \{v\}$ is called the *closed neighborhood* of v in G. A set of vertices I,

---

[†]This project has been partially supported by Narodowe Centrum Nauki under contract DEC-2011/02/A/ST6/00201

[‡]Partially supported by the Foundation for Polish Science (FNP) and by Narodowe Centrum Nauki under contract DEC-2011/02/A/ST6/00201

$I \subseteq V(G)$, is *independent* in G, if no two vertices of I are adjacent. The cardinality of a maximum independent set in G is called the *independence number* of a graph G, denoted by $\alpha(G)$. We use the symbol N_r, $r \geq 0$, to denote the edgeless graph of order r. A subgraph isomorphic to a complete graph K_r is called a *clique*. The order of the largest clique in G is called the *clique number* of G, denoted by $\omega(G)$. We use $H \cup G$ for the union of vertex disjoint graphs H and G, while $2H$ denotes the graph consisting of two vertex disjoint copies of H. For any other notions undefined here we refer the reader to [1, 19, 23].

The on-line version of the ranking problem is frequently viewed as a game between two players, *Presenter* (the adversary) and *Painter* (the coloring algorithm). In contrast to the off-line approach, Painter does not know the structure of the graph $G = (V, E)$ that has to be colored. Presenter starts the game and reveals subsequent vertices of G in some order $(v_1, \ldots, v_n)$ unknown to Painter. More precisely, each vertex v_i is presented together with the edges $E_i = \{v_i v_j \in E(G) \,|\, j < i\}$. Consequently, after presentation of v_i, Painter knows only the structure of the subgraph G_i induced by $\{v_1, \ldots, v_i\}$. Painter has to irrevocably assign a permissible color $c(v_i)$ to the vertex v_i before v_{i+1} is presented. The aim of Painter is to use as few colors as possible while Presenter aims at finding an ordering of the vertices forcing Painter to use as many colors as possible. The game can also be viewed as a sequence $(P_1, A_1, \ldots, P_n, A_n)$ of alternate moves $P_i = (v_i, E_i)$ of Presenter and moves A_i of the ranking algorithm. Defining a pair $R_i = (P_i, A_i)$ as the i*-th round* we will also view the game as a sequence of n rounds $(R_1, \ldots, R_n)$. In what follows we use $\chi_{\mathsf{A}}(G, \pi)$ to denote the number of colors used by the coloring algorithm A for the ordering π of the vertex set, and $\chi_{\mathsf{A}}(G)$ for the maximum number of colors that may be required by A to color G, i.e., $\chi_{\mathsf{A}}(G) = \max_\pi \chi_{\mathsf{A}}(G, \pi)$. The *on-line ranking number* of a graph G is the minimum $\chi_{\mathsf{A}}(G)$ taken over all on-line ranking algorithms A. We say that an on-line ranking algorithm is c*-competitive* if there exists a constant b such that $\chi_{\mathsf{A}}(G) \leq c \cdot \chi_r(G) + b$. We say for brevity that an algorithm is *constant competitive* if it is c-competitive for some constant c. We refer the reader interested in a more extensive description of a classical on-line coloring to [3, 18].

Several results concerning on-line vertex ranking are known. Schiermeyer *et al.* [20] characterized graphs with the on-line vertex ranking number equal to 3 and proved that the *greedy algorithm*, which always assigns the smallest permissible color to the incoming vertex, produces a $(3 \log_2 n)$-ranking for a path P_n. Note that $\chi_r(P_n) = 1 + \lfloor \log_2 n \rfloor$. On-line rankings of paths and cycles were also considered by Bruoth and Horňák [6], who proved an even better bound of $2\lfloor \log_2 n \rfloor + 1$ for both classes of graphs. This, in particular, implies that the ratio of the on-line and the off-line ranking numbers for paths is bounded by 2. The lower bound of $1.619 \log_2 n - 1$ for the on-line ranking number of paths was given by the same authors in [7]. It was also proved by Semanišin and Soták [22] that there exists an infinite sequence of graphs for which the ratio of the on-line and the off-line ranking numbers can be arbitrarily large. Ghoshal *et al.* [14] introduced the problem of finding a *minimal ranking* which is defined as a ranking with the property that decreasing the color assigned to any vertex results in a function that is not a ranking. The *arank number* of a graph G is the largest k such that there exists a minimal k-ranking of G. As pointed out by Isaak *et al.* [16] there is a connection between on-line rankings and minimal rankings. In particular, the arank number equals the number of colors assigned in the worst case by the greedy ranking algorithm. So, the number of colors assigned by the greedy on-line algorithm is at most the arank number of the input graph. Ranking is also closely related to parity coloring (see, e.g., [4, 8]) and conflict-free coloring [12]. A *parity coloring* is a coloring with the property that every path contains some color an odd number of times, while in a *conflict-free coloring* every path has to use some color exactly once.

In this paper we investigate the on-line version of the ranking problem for split graphs. A graph G is called a *split graph* if there exists a bipartition (C, I) of its vertex set such that I is an independent set and C induces a clique. Such a partition is called a *split bipartition*. Notice that a split bipartition does not have to be unique but it always satisfies one of the three conditions given in the following theorem.

Theorem 1.1 (P. L. Hammer, B. Simeone, [15]) *Let the vertex set of a split graph G be partitioned into an independent set I and a set C inducing a clique. Then, exactly one of the following conditions holds:*

(a) $|I| = \alpha(G)$ *and* $|C| = \omega(G)$,

(b) $|I| = \alpha(G)$ *and* $|C| = \omega(G) - 1$,

(c) $|I| = \alpha(G) - 1$ *and* $|C| = \omega(G)$.

We say that G is a *split graph of type* $\mathcal{A}$ if $V(G)$ has a split bipartition satisfying (a), while G is a *split graph of type* $\mathcal{B}$ if its vertex set can be partitioned such that (b) or (c) holds. Note that the complement $\overline{G}$ of a split graph G is also a split graph. The class of split graphs is also known as the intersection of chordal and co-chordal graphs [13].

Whenever G is a split graph, then $\chi_r(G) = n - \alpha(G) + 1$ (see [9]). Combining this fact with Theorem 1.1 we obtain

Theorem 1.2 *For any split graph G it holds that*

$$\chi_r(G) = \begin{cases} \omega(G) + 1 & \textit{if G is of type } \mathcal{A}, \\ \omega(G) & \textit{if G is of type } \mathcal{B}. \end{cases}$$

In Section 2 we prove that the worst case ratio of the number of colors used by any on-line ranking algorithm and the number of colors used in an optimal off-line solution may be arbitrarily large. This negative result motivates us to strengthen an on-line algorithm by providing it with some additional information about the graph to be colored. This kind of approach is usually called *semi on-line* and has been widely studied, e.g., for various scheduling problems (see, e.g., [11, 17, 21]). In general a semi on-line algorithm may be given the values of some invariants of a graph in advance, i.e., before Presenter reveals the first vertex of a graph. The choice of the appropriate invariants is a separate and challenging problem, e.g., a semi on-line ranking algorithm for complete m-partite graphs $K_{k_1,\ldots,k_m}$ presented in [22] knows in advance the values of $k_1, \ldots, k_m$ that uniquely describe the structure of the graph. In Section 3 we prove that for any $\varepsilon > 0$, there does not exist a semi on-line ranking algorithm that is $(2-\varepsilon)$-competitive even if it is given the clique number of a split graph. The remaining parts of the paper focus on the statement and analysis of a 2-competitive semi on-line ranking algorithm that knows the clique number of a split graph in advance. In particular, in Section 4, prior to the formal description of the algorithm we introduce the concept of *split tripartitions* that are used for on-line classification of vertices. The idea is useful both for the formulation of the algorithm in Section 5 as well as for its analysis in Sections 6 and 7. The analysis is broken into two parts. First, in Section 6, we observe that though there are infinitely many possible Presenter's moves that may occur during the presentation of a split graph, it is possible to distinguish a finite number of *types* of moves. This fact is crucial, as it allows to represent all possible sequences of Presenter's moves as the directed walks of a well-structured digraph D, called the *state transition digraph*. The state transition digraph is defined so that its walks reflect certain aspects of the evolution of graph structure during the game. It is important to point out that the directed walk corresponding to a particular game is independent of the ranking algorithm. Hence, a similar approach may also be used for the analysis of on-line algorithms for other combinatorial problems on split graphs.

From the results proved in this paper it follows that for split graphs there does not exist a constant competitive on-line ranking algorithm that could guarantee the usage of an interval of colors containing color 1. Consequently, minimization of the largest color differs from the minimization of the number of colors which is our main interest. We remark on the former criterion in Section 8.

2 On-line Ranking

The main purpose of this section is to describe Presenter's strategy from which it follows that there does not exist a constant competitive on-line ranking algorithm for split graphs. This serves as the main reason to consider semi on-line algorithms.

Theorem 2.1 *For any on-line ranking algorithm* A *and any integer* $p > 0$*, there exists a split graph* G *such that* $\chi_{\mathsf{A}}(G)/\chi_r(G) \geq p$*.*

Proof: Let A be an arbitrary on-line ranking algorithm. If A assigns color 1 to v_1, then Presenter fixes v_1 to be the center of a star S_n (which is a split graph) and continues the presentation of the remaining vertices, thus forcing A to color them with pairwise different colors. This yields $\chi_{\mathsf{A}}(S_n) = n$, while $\chi_r(S_n) = 2$. Therefore, let us assume without loss of generality that A used color $k > 1$ for the first vertex. Also note that since we describe a strategy of Presenter, there is no loss of generality in assuming that $v_1v_2 \in E(G)$. Presenter maintains a bipartition (C, I) of the vertex set, i.e., as soon as the revealed vertex gets colored, Presenter assigns it either to the independent set I or to the set C inducing a clique. The only exception is the decision on the assignment of the first vertex which is postponed until the second vertex is colored. Then, a vertex with the smaller color is assigned to C, while the other one is assigned to I. From now on, for $i > 2$, each vertex v_i revealed by Presenter is joined to all vertices currently in C. If A uses for v_i a color smaller than k, then v_i is assigned to C. Otherwise, it is assigned to I. The strategy of assigning vertices with the smallest colors to C ensures that any A is forced to introduce a new color whenever a subsequent vertex is revealed (notice that C induces a clique, while reusing a color of any vertex from I spoils ranking). Consequently, $\chi_{\mathsf{A}}(G) = n$. On the other hand, since A started with color k, the number of vertices assigned to C is at most k. This also bounds the clique number of a graph used by Presenter, i.e., $\omega(G) \leq k + 1$, and since Presenter constructs a split graph of type $\mathcal{B}$, we have $\chi_r(G) \leq k + 1$. Hence, the difference $\chi_{\mathsf{A}}(G) - \chi_r(G)$ as well as $\chi_{\mathsf{A}}(G)/\chi_r(G)$ may be arbitrarily large. □

3 Lower Bounds for Semi On-line Ranking

In this section we argue that the knowledge of an upper bound on the clique number of a graph is a prerequisite for any semi on-line ranking algorithm to be constant competitive in the class of split graphs. Moreover, in the first theorem we reveal that even when the clique number of a split graph is given in advance, the algorithm has to satisfy some additional conditions; for if not, it is not constant competitive.

Theorem 3.1 *Let* A *be a semi on-line ranking algorithm that is given* ω *and let* k *be one of the colors that* A *used for the first two vertices. If* $k < \omega + 1$*, then for any* $p > 0$*, there exists a split graph* G *such that* $\omega(G) \leq \omega$ *and* $\chi_{\mathsf{A}}(G)/\chi_r(G) \geq p$*.*

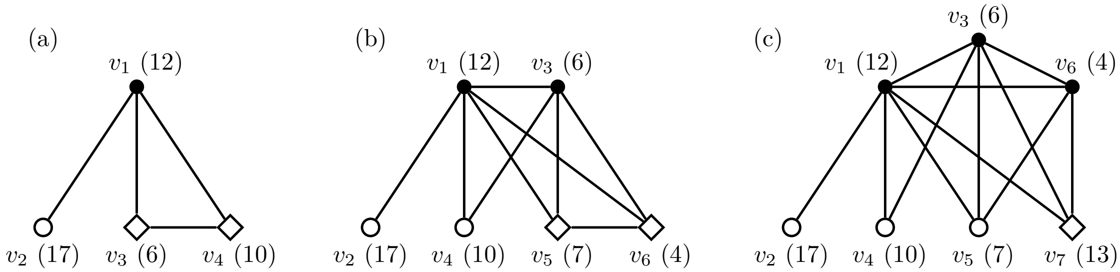

Fig. 1: An example for Theorem 3.2. Numbers in braces denote sample colors. Black vertices belong to C, white to I, while diamonds represent vertices before Presenter's decision on their assignment.

Proof: We use the strategy of Presenter from Theorem 2.1 and prove that its slight modification is also successful against any A that is given ω but colors v_1 or v_2 with color $k < \omega + 1$. The case of $k = 1$ is trivial. Without loss of generality we may assume that v_1 was assigned color $k = \omega$ and v_2 some color greater than k. According to the strategy, v_1 gets assigned to C, while v_2 is assigned to I. The game continues and since the number of colors smaller than k is limited, either A loses because it may be forced to use in I arbitrarily many colors greater than k (they all have to be pairwise different because of ranking) or it is faced with the set C that consists of the vertices colored $2, \ldots, k$. As long as $|C| < \omega$, the algorithm is not allowed to use color 1 for v_i, since it would be immediately assigned to C and used by Presenter to force an arbitrary number of new colors by presenting a star with the central vertex v_i. On the other hand, there are no more permissible colors smaller than k and hence using new colors greater than k for v_i and for arbitrarily many subsequent vertices is unavoidable. □

It is worth pointing out that Presenter's strategies used in the preceding theorems do not work if for both vertices v_1, v_2 the algorithm uses colors greater than the clique number. Since Presenter adds to C one of these vertices, at most $\omega - 1$ colors smaller than $\omega + 1$ may further be required to color all vertices in C. Therefore, the algorithm can save color 1 until ω colors are already present in C and it can safely use it for any subsequent vertex which will undoubtedly belong to I. In the next theorem we prove that for any $\varepsilon > 0$, there does not exist a semi on-line algorithm A that is $(2-\varepsilon)$-competitive, even when A is given the clique number of a split graph G and colors v_1, v_2 using colors greater than $\omega(G)$.

Theorem 3.2 *Let* A *be a semi on-line ranking algorithm that is given the clique number of a graph. Then, there exists a split graph* G *such that* $\chi_{\mathsf{A}}(G) \geq 2\chi_r(G) - 1$.

Proof: Let k be the color used by A for the first vertex. By Theorem 3.1, $k \geq \omega + 1$. As in the previous proofs, Presenter maintains a bipartition (C, I) of the vertex set. The presentation starts with two adjacent vertices. Then, each vertex v_i, $i > 2$, revealed by Presenter is adjacent to all vertices currently in C and for even values of i it is also adjacent to v_{i-1} (see Figure 1(a) – (b)). After the presentation of a vertex v_i, when i is even, the colors of v_i and v_{i-1} are compared and the vertex with the lower color is added to C while the one with the higher color is added to I. It follows directly from the construction that no color, which has already been assigned to a vertex in C, can be reused for the currently presented vertex v_i. To see that the same holds for colors used in I, observe that for any vertex $v \in I$, there exists $u \in C$ that is adjacent to v and v_i, and such that $c(u) < c(v)$. Hence, because of ranking, we get $\chi_{\mathsf{A}}(G_i) = i$.

On the other hand, for every second round the clique number increases, i.e., $\omega(G_i) = i/2 + 1$ when i is even, and $\omega(G_i) = (i + 1)/2$ otherwise. Since G_i is a split graph of type $\mathcal{B}$, $\chi_r(G_i) = \omega(G_i)$ and after an odd round we have $\chi_{\mathsf{A}}(G_i) = 2\chi_r(G_i) - 1$ (see Figure 1(c)). $\Box$

A natural question of the existence of a 2-competitive semi on-line ranking algorithm will be answered in the affirmative in Section 5. However, before we can formulate our algorithm, we need to provide some structural properties of split graphs. These properties are essential for the statement and analysis of the algorithm.

4 Split Tripartitions

As we will see in the next section, an important feature of our semi on-line ranking algorithm is its ability to maintain a specific classification of the vertices revealed by Presenter. In order to formally describe the classification process we have to define the split tripartition, which in turn relies on the notion of a *simplicial vertex*, i.e., a vertex whose closed neighborhood induces a clique. Since the class of split graphs is equal to the intersection of chordal and co-chordal graphs each vertex of a split graph G is simplicial in G or its complement $\overline{G}$. The set of vertices that are simplicial as well in G as in $\overline{G}$ is of special importance for the above-mentioned classification.

Definition 4.1 *The* split tripartition *of $V(G)$ is its partition (C, S, I) such that each vertex in C is not simplicial in G, and each vertex in I is not simplicial in $\overline{G}$, while S consists of all vertices that are simplicial both in G and in $\overline{G}$.*

Clearly, the vertex set of every split graph has a unique split tripartition. From the above definition there follow simple but important properties of split tripartitions. We state them here for a further reference.

Property 4.1 *If G is a split graph and (C, S, I) is the split tripartition of $V(G)$, then*

(a) *$G[S]$ is either a clique or an edgeless graph,*

(b) *each vertex in S is adjacent to all vertices in C and has no neighbors in I.*

In the remaining part of this section we analyze selected relations between vertices of two split graphs G and H such that G is obtained from H by adding a vertex v (possibly with some incident edges). The properties will be further used in the analysis of a single round of the on-line ranking game in Section 5 as well as for the definition of types of Presenter's moves introduced in Section 6.

Property 4.2 *Let G be a split graph and let $H \leq G$ such that $V(G) \setminus V(H) = \{v\}$. If (C, S, I) and (C', S', I') are the split tripartitions of $V(H)$ and of $V(G)$, respectively, then $C \subseteq C'$ and $I \subseteq I'$.*

In the following two lemmas we distinguish important configurations in the neighborhood of the above-mentioned vertex v.

Lemma 4.1 *There does not exist a split graph G that contains an induced subgraph H, $V(G) \setminus V(H) = \{v\}$ such that for the split tripartition (C, S, I) of $V(H)$ it holds that $N_G(v) \cap I \neq \emptyset$ and $C \nsubseteq N_G(v)$.*

Proof: Suppose for a contradiction that there exists a split graph G with the given properties and let (C', S', I') be the split tripartition of $V(G)$. Let $x \in I$ and $y \in C$ be selected in such a way that $xv \in E(G)$ and $yv \notin E(G)$. Observe that whenever $xy \in E(G)$, then x is not simplicial in G and hence

$x \in C'$. However, by Property 4.2, it holds that $x \in I'$, a contradiction. On the other hand if $xy \notin E(G)$, then we get a contradiction by an analogous argument applied to $\overline{G}$. $\square$

Lemma 4.2 *Let H be an induced subgraph of a split graph G such that $V(G) \setminus V(H) = \{v\}$ and let (C, S, I) be the split tripartition of $V(H)$.*

(a) *If $H[S] \simeq N_p$, for some $p \geq 0$, and $C \nsubseteq N_G(v)$, then $|S \cap N_G(v)| \leq 1$.*

(b) *If $H[S] \simeq K_p$, for some $p \geq 2$, and $N_G(v) \cap I \neq \emptyset$, then $|S \cap N_G(v)| \geq |S| - 1$.*

Proof: (a) Let (C', S', I') be the split tripartition of $V(G)$. Assume, on the contrary, that $|S \cap N_G(v)| > 1$ and let $x, y \in S$ be the neighbors of v in G. Since x and y are not adjacent, the vertex v is not simplicial in G, i.e., $v \in C'$. Let $u \in C \setminus N_G(v)$. Since by Property 4.2 it holds that $C \subseteq C'$, the vertex u belongs to C'. Hence, C' contains both u and v that are not adjacent, a contradiction. Part (b) follows directly from (a), by examining the complement graph. $\square$

5 Semi On-line 2-competitive Algorithm

In order to close the algorithmic gap we define a semi on-line algorithm `Split Ranking` (SR for short), which for any split graph G, given its clique number ω, uses at most $2\chi_r(G)$ colors. Algorithm SR processes every vertex in three stages: *presentation*, *classification* and *coloring*. We start with an informal description of these three stages of SR.

The main purpose of the presentation stage is to update the structure of the graph. The new vertex v_i and the edges $E_i = \{v_i v_j \in E(G) \mid j < i\}$ revealed by Presenter are added to the graph G_{i-1}, which results in the graph G_i. Revealing a subsequent vertex together with some edges provides a new piece of structural information on the target graph G.

Then, the classification stage begins. Its purpose is the computation of the split tripartition (C_i, S_i, I_i) of $V(G_i)$. The set S_i can be seen as a buffer that holds vertices that cannot be placed in C_i or I_i. A vertex v is kept in the buffer until it becomes clear that in some round v can be moved to one of the two other sets of the split tripartition, i.e., when v is no longer simplicial either in G or $\overline{G}$. Recall that, by Property 4.2, in contrast to the assignment of vertices to S_i, any assignment to C_i or I_i is permanent in the sense that $C_i \subseteq C_j$ and $I_i \subseteq I_j$ for each $j > i$. Though not important for the formulation of our algorithm, this fact is crucial for its analysis.

The coloring stage relies on two variables lc and hc with the names being abbreviations for lower and higher color, respectively. Their initial values are equal to $2\omega + 2$ and are assigned in the initialization stage of SR, that is, prior to the presentation of v_1. In the coloring stage, SR uses color 1 for every vertex v_i assigned to I_i. Otherwise, the color for v_i is selected greedily with respect to the interval $[lc, hc]$. More precisely, SR uses color lc, provided that this results in a ranking of G_i. If this is not possible, SR checks color $lc - 1$ and whenever the color is permissible, lc decreases by 1 and its new value is used as the color for v_i. Otherwise, i.e., if both colors lc and $lc - 1$ violate the definition of ranking, then v_i is colored with the color obtained after incrementation of hc by 1. At the end of the coloring stage it holds that $\{c(v_1), \ldots, c(v_i)\} = \{1\} \cup \{lc, \ldots, hc\}$ or $\{c(v_1), \ldots, c(v_i)\} = \{lc, \ldots, hc\}$. See Algorithm 1 for the pseudocode of the algorithm SR.

Algorithm 1 Split ranking SR (ω);

Begin
$V(G_0) \leftarrow \emptyset$, $E(G_0) \leftarrow \emptyset$, $C \leftarrow \emptyset$, $S \leftarrow \emptyset$, $I \leftarrow \emptyset$; *// initialization*
$i \leftarrow 0$;
$lc \leftarrow 2\omega + 2$;
$hc \leftarrow lc$;
repeat
  $i \leftarrow i + 1$; *// presentation*
  read (v_i, E_i);
  $V(G_i) \leftarrow V(G_{i-1}) \cup \{v_i\}$;
  $E(G_i) \leftarrow E(G_{i-1}) \cup E_i$;

  compute the split tripartition (C_i, S_i, I_i) of $V(G_i)$; *// classification*

  if $v_i \in I_i$ **then** *// coloring*
    $c(v_i) \leftarrow 1$;
  else if lc is permissible for v_i **then**
    $c(v_i) \leftarrow lc$;
  else if $lc - 1$ is permissible for v_i **then**
    $lc \leftarrow lc - 1$;
    $c(v_i) \leftarrow lc$;
  else
    $hc \leftarrow hc + 1$;
    $c(v_i) \leftarrow hc$;

until end of input;
End.

Now, let us consider an example of the execution of SR presented in Figure 2. Since the first round always results in the assignment of v_1 to S_1, we depict the remaining rounds $R_2, \dots, R_8$. The description of a particular round R_i, $i \in \{2, \dots, 8\}$, includes the structure of G_i and the split tripartition $(C_{i-1}, S_{i-1}, I_{i-1})$ of $V(G_{i-1})$. We use arrows to depict the classification of the appropriate vertices in S_{i-1} and the assignment of v_i to C_i, S_i or I_i. Since SR receives the clique number in advance (in this case $\omega = 4$), the initialization of the variables results in $hc = lc = 10$. For each round, the values of lc and hc at the end of the round are also given. Note that lc decreases in rounds R_2, R_4 and R_7. The only round in which SR uses color 1 is R_5 (v_5 is classified to I_5). The value of hc increases in R_8. Since v_8 is adjacent to v_4 and v_6 (both with color 8) v_8 must receive a color greater than 8, for otherwise we would not have a ranking. Consequently, neither lc nor $lc - 1$ is permissible for v_8. Therefore, SR increases the value of hc and assigns it to v_8.

6 Presenter's Moves

In order to analyze the performance of our algorithm we introduce the notion of the state transition digraph, which formally represents all possible states of the on-line ranking game. The vertices of the state transition digraph are called *states* while arcs are called *transitions*. For any transition (s, s'), the states

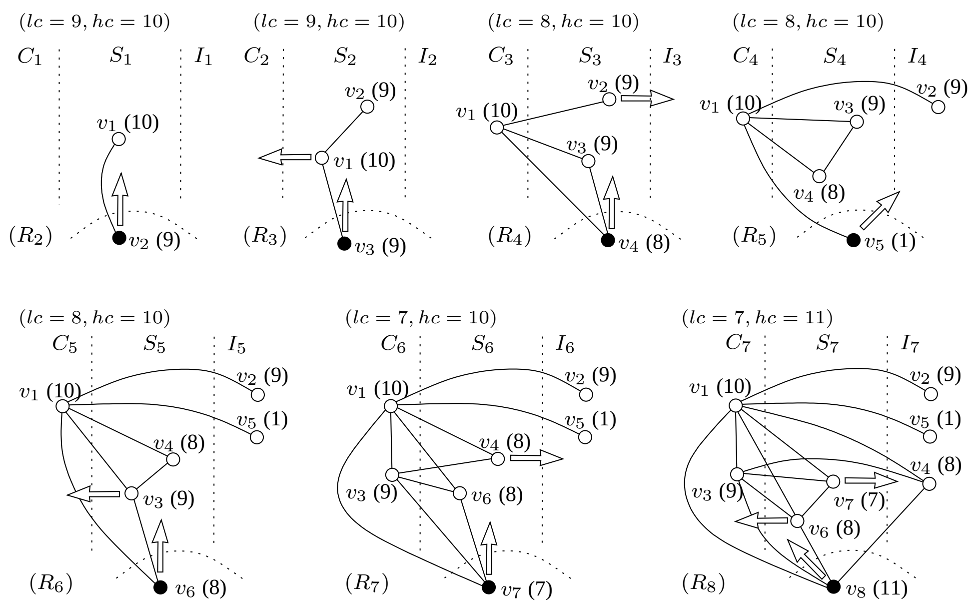

Fig. 2: An example of the execution of the algorithm SR (rounds $R_2, \ldots, R_8$). Numbers in braces denote the colors assigned by SR.

s and s' are called *pre-state* and *post-state*, respectively. In what follows each move of Presenter will be thought of as an event that causes a transition from the current state to the appropriate post-state.

More formally, let $\mathcal{G}$ be a family of graphs that are complete or edgeless, and let $D = (W, T)$ be an infinite digraph with the vertex set W and arc set T. Moreover, let $\varphi : \mathcal{G} \to W(D)$ be a bijection. We say that the digraph D is the *state transition digraph* if for any $H, H' \in \mathcal{G}$ we have $(\varphi(H), \varphi(H')) \in T(D)$ if and only if there exist a split graph G_{i-1} such that $G_{i-1}[S_{i-1}] \simeq H$ and Presenter's move $P_i = (v_i, E_i)$ resulting in G_i for which $G_i[S_i] \simeq H'$. Observe that, for a particular game, the sequence of graphs $(G_0, G_1, \ldots, G_i)$, $G_0 \simeq N_0$, resulting from Presenter's moves $(P_1, P_2, \ldots, P_i)$ corresponds to the directed walk $(\varphi(G_0[S_0]), \varphi(G_1[S_1]), \ldots, \varphi(G_i[S_i]))$ in D. Also note that, by the definition, for any transition (s, s') there may exist non-isomorphic split graphs G_{k-1}, G_{l-1} such that $G_{k-1}[S_{k-1}] \simeq G_{l-1}[S_{l-1}] \simeq \varphi^{-1}(s)$ and $G_k[S_k] \simeq G_l[S_l] \simeq \varphi^{-1}(s')$. In what follows we slightly simplify our notation by writing $G[S_i]$ instead of $G_i[S_i]$ and $N(v_i)$ in place of $N_{G_i}(v_i)$. Since φ is a bijection, there will be also no ambiguity whenever we use H instead of $\varphi(H)$, as a label for a vertex of D, e.g., K_2 instead of $\varphi(K_2)$, or when we write "state $G[S_i]$" in place of "state $\varphi(G[S_i])$".

Though the number of states is infinite, we distinguish a finite number of types of Presenter's moves. This considerably simplifies the analysis of the coloring stage. The type of a move $P_i = (v_i, E_i)$ depends solely on the pre-state $G[S_{i-1}]$ and the adjacency of v_i to the vertices in the sets C_{i-1}, S_{i-1} and I_{i-1} of the split tripartition $(C_{i-1}, S_{i-1}, I_{i-1})$ of $V(G_{i-1})$. Similarly, the post-state resulting from P_i is uniquely determined by the pre-state and the adjacency conditions, and it can be easily calculated. Therefore, the

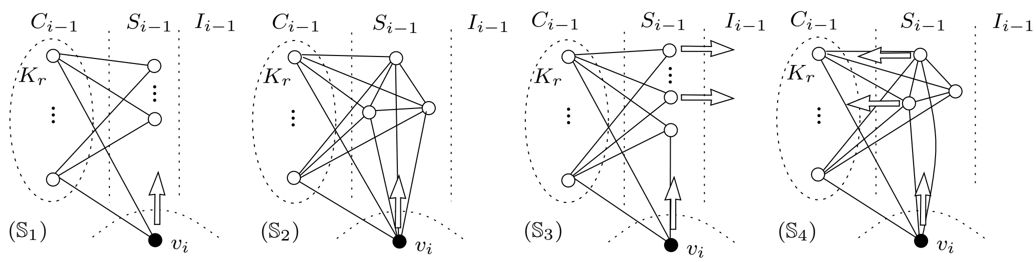

Fig. 3: The types of moves for which v_i is classified to S_i

post-state is not required in move's type definition, but we explicitly give a description of each post-state to make the list of types more useful for further analysis. All types of moves are also presented in Figures 3, 4 and 5. In each figure the structure of G_{i-1} is reflected by distinguishing its split tripartition while v_i is shown separately, together with all edges in E_i. The arrows, when present, show the result of the classification, i.e., they indicate whether v_i or some vertices from S_{i-1} belong to C_i, S_i or I_i (recall that by Property 4.2, $C_{i-1} \subseteq C_i$ and $I_{i-1} \subseteq I_i$). To make the figures and further analysis more clear, we omit all parts of G_{i-1} that are irrelevant for the classification and we select symbols for the types of moves so that they reflect the classification of v_i, i.e., the symbols $\mathbb{S}$, $\mathbb{C}$ and $\mathbb{I}$ are used when v_i is classified to S_i, C_i or I_i, respectively.

The list of the types of moves was organized in such a way that it allows for a clear proof of its completeness in Lemma 6.1 and so that it reflects all possible migrations of vertices from S_{i-1} to C_i or I_i. In the next section we classify rounds of the on-line ranking game as won, tie or lost by SR. As we will see later, the result of each round strongly depends on the type of Presenter's move. Some types that seem to be almost the same may differ significantly. For instance, when Presenter makes a move of type $\mathbb{I}_5$, the vertex v_i is adjacent to all vertices in S_{i-1}, while in a move of type $\mathbb{I}_4$ the set S_{i-1} contains a vertex that is not adjacent to v_i (see Figure 5). In the next section we argue that this seemingly insignificant difference causes that SR always wins if Presenter makes a move of type $\mathbb{I}_5$, and it looses or eventually ties in the latter case. Also, the results of the rounds with moves of "similar" types $\mathbb{I}_1$ and $\mathbb{S}_1$ may differ, e.g., because of reusing a color from previous rounds, when moves of type $\mathbb{S}_1$ form some specific sequences (we address this issue in Lemma 7.2). There are also some types of moves, e.g., $\mathbb{I}_3, \mathbb{I}_6$ and $\mathbb{C}_1, \mathbb{C}_5$ that we distinguish mainly for the completeness and clearness of the argument.

Types of Presenter's moves:

$\mathbb{S}_1$ Pre-state: $G[S_{i-1}] \simeq N_p, p \geq 0$ ($\mathbb{S}_1^\circ$ when $p = 0$, $\mathbb{S}_1^\bullet$ when $p > 0$).
Adjacency conditions: $N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \subseteq N(v_i)$ and $N(v_i) \cap S_{i-1} = \emptyset$.
Post-state: $G[S_i] \simeq N_{p+1}$.

$\mathbb{S}_2$ Pre-state: $G[S_{i-1}] \simeq K_p, p \geq 2$.
Adjacency conditions: $N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \subseteq N(v_i)$ and $S_{i-1} \subseteq N(v_i)$.
Post-state: $G[S_i] \simeq K_{p+1}$.

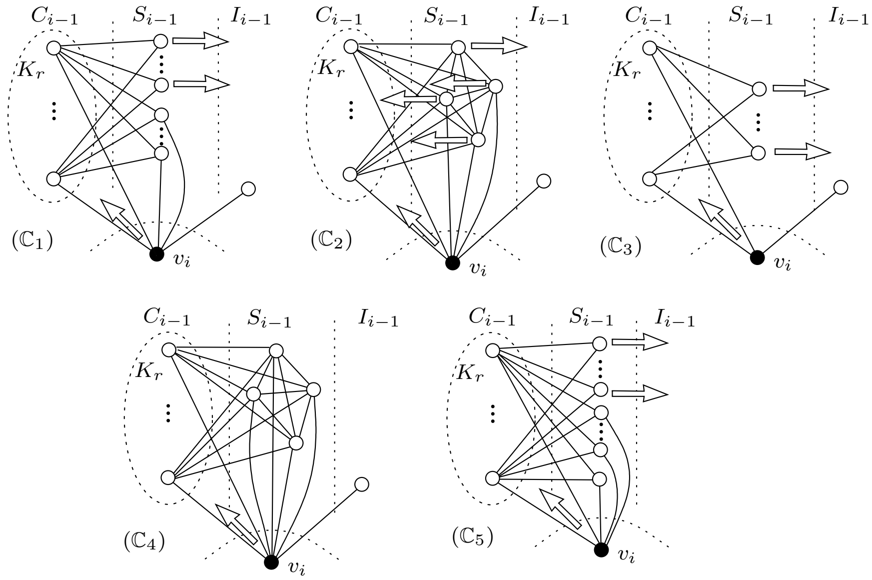

Fig. 4: The types of moves for which v_i is classified to C_i

$\mathbb{S}_3$ Pre-state: $G[S_{i-1}] \simeq N_p, p \geq 1$.
Adjacency conditions: $N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \subseteq N(v_i)$ and $|N(v_i) \cap S_{i-1}| = 1$.
Post-state: $G[S_i] \simeq K_2$.

$\mathbb{S}_4$ Pre-state: $G[S_{i-1}] \simeq K_p, p \geq 2$.
Adjacency conditions: $N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \subseteq N(v_i)$ and $|N(v_i) \cap S_{i-1}| = p - 1$.
Post-state: $G[S_i] \simeq N_2$.

$\mathbb{C}_1$ Pre-state: $G[S_{i-1}] \simeq N_p, p \geq 1$.
Adjacency conditions: $N(v_i) \cap I_{i-1} \neq \emptyset$ and $C_{i-1} \subseteq N(v_i)$ and $t = |N(v_i) \cap S_{i-1}| \geq 1$.
Post-state: $G[S_i] \simeq N_t$.

$\mathbb{C}_2$ Pre-state: $G[S_{i-1}] \simeq K_p, p \geq 2$.
Adjacency conditions: $N(v_i) \cap I_{i-1} \neq \emptyset$ and $C_{i-1} \subseteq N(v_i)$ and $|N(v_i) \cap S_{i-1}| = p - 1$.
Post-state: $G[S_i] \simeq N_0$.

$\mathbb{C}_3$ Pre-state: $G[S_{i-1}] \simeq N_p, p \geq 0$ ($\mathbb{C}_3^{\circ}$ when $p = 0$, $\mathbb{C}_3^{\bullet}$ when $p > 0$).
Adjacency conditions: $N(v_i) \cap I_{i-1} \neq \emptyset$ and $C_{i-1} \subseteq N(v_i)$ and $N(v_i) \cap S_{i-1} = \emptyset$.
Post-state: $G[S_i] \simeq N_0$.

$\mathbb{C}_4$ Pre-state: $G[S_{i-1}] \simeq K_p, p \geq 2$.
Adjacency conditions: $N(v_i) \cap I_{i-1} \neq \emptyset$ and $C_{i-1} \subseteq N(v_i)$ and $S_{i-1} \subseteq N(v_i)$.
Post-state = pre-state.

$\mathbb{C}_5$ Pre-state: $G[S_{i-1}] \simeq N_p, p \geq 2$.
Adjacency conditions: $N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \subseteq N(v_i)$ and $t = |N(v_i) \cap S_{i-1}| \geq 2$.
Post-state: $G[S_i] \simeq N_t$.

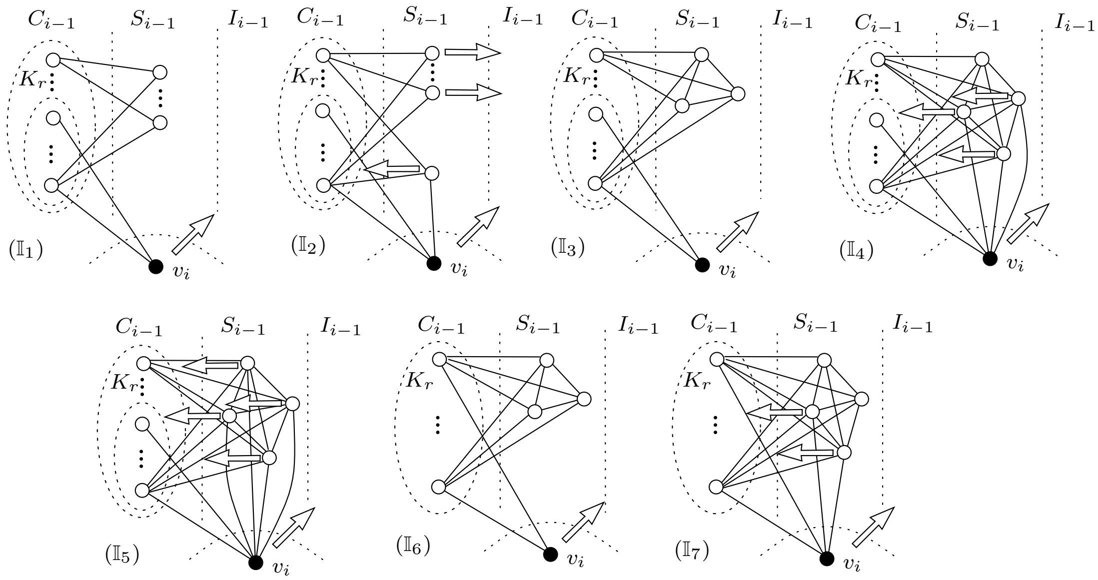

Fig. 5: The types of moves for which v_i is classified to I_i

$\mathbb{I}_1$ Pre-state: $G[S_{i-1}] \simeq N_p, p \geq 0$.
Adjacency conditions: $N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \nsubseteq N(v_i)$ and $N(v_i) \cap S_{i-1} = \emptyset$.
Post-state = pre-state.

$\mathbb{I}_2$ Pre-state: $G[S_{i-1}] \simeq N_p, p \geq 1$.
Adjacency conditions: $N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \nsubseteq N(v_i)$ and $|N(v_i) \cap S_{i-1}| = 1$.
Post-state: $G[S_i] \simeq N_0$.

$\mathbb{I}_3$ Pre-state: $G[S_{i-1}] \simeq K_p, p \geq 2$.
Adjacency conditions: $N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \nsubseteq N(v_i)$ and $N(v_i) \cap S_{i-1} = \emptyset$.
Post-state = pre-state.

$\mathbb{I}_4$ Pre-state: $G[S_{i-1}] \simeq K_p, p \geq 2$.
Adjacency conditions: $N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \nsubseteq N(v_i)$ and $1 \leq t = |N(v_i) \cap S_{i-1}| \leq p-1$.
Post-state: $G[S_i] \simeq K_{p-t}$.

$\mathbb{I}_5$ Pre-state: $G[S_{i-1}] \simeq K_p, p \geq 2$.
Adjacency conditions: $N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \nsubseteq N(v_i)$ and $S_{i-1} \subseteq N(v_i)$.
Post-state: $G[S_i] \simeq N_0$.

$\mathbb{I}_6$ Pre-state: $G[S_{i-1}] \simeq K_p, p \geq 2$.
Adjacency conditions: $N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \subseteq N(v_i)$ and $N(v_i) \cap S_{i-1} = \emptyset$.
Post-state = pre-state.

$\mathbb{I}_7$ Pre-state: $G[S_{i-1}] \simeq K_p, p \geq 3$.
Adjacency conditions: $N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \subseteq N(v_i)$ and $1 \leq t = |N(v_i) \cap S_{i-1}| \leq p-2$.
Post-state: $G[S_i] \simeq K_{p-t}$.

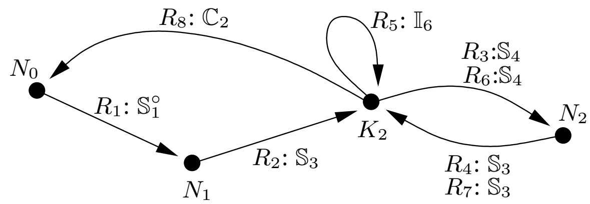

Fig. 6: The subdigraph of the state transition digraph D containing the directed walk, which represents an execution of the algorithm SR for the instance presented in Figure 2

In Figure 6 we use the types of moves to continue the analysis of the game presented in Figure 2. In particular, we depict a digraph whose arcs represent all transitions in the rounds $R_1, \ldots, R_8$ of the game. Each arc is labeled with the rounds in which the corresponding transitions occur and with the types of the appropriate Presenter's moves.

Now, we argue that no case is missing and that the types of moves cover pairwise disjoint cases, i.e., for the given split tripartition of $V(G_{i-1})$ every move satisfies the adjacency conditions of exactly one type.

Lemma 6.1 *For any split graph G_{i-1}, every Presenter's move $P_i = (v_i, E_i)$ is of exactly one of the types $\mathbb{I}_1, \ldots, \mathbb{I}_7, \mathbb{S}_1, \ldots, \mathbb{S}_4, \mathbb{C}_1, \ldots, \mathbb{C}_5$.*

Proof: In order to take into account all possible moves of Presenter we consider two complementary cases for each set of the split tripartition $(C_{i-1}, S_{i-1}, I_{i-1})$ of $V(G_{i-1})$. Namely, for C_{i-1} we have either $C_{i-1} \subseteq N(v_i)$ or $C_{i-1} \nsubseteq N(v_i)$, for I_{i-1} it holds either $N(v_i) \cap I_{i-1} = \emptyset$ or $N(v_i) \cap I_{i-1} \neq \emptyset$, while for S_{i-1} by Property 4.1(a) we have either $G[S_{i-1}] \simeq N_p$, $p \geq 0$ or $G[S_{i-1}] \simeq K_p$, $p \geq 2$. Hence, the proof falls into eight natural cases, where by Lemma 4.1 two of them, for which $N(v_i) \cap I_{i-1} \neq \emptyset$ and simultaneously $C_{i-1} \nsubseteq N(v_i)$ are not possible, so it is enough to consider the remaining six cases.

Case 1. [$N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \subseteq N(v_i)$ and $G[S_{i-1}] \simeq N_p$, $p \geq 0$]
Observe that the only types of moves to consider are $\mathbb{S}_1^\circ, \mathbb{S}_1^\bullet, \mathbb{S}_3$ and $\mathbb{C}_5$. If $p = 0$, then we clearly have a move of type $\mathbb{S}_1^\circ$. For $p = 1$, if $S_{i-1} \cap N(v_i) = \emptyset$, then the move is of type $\mathbb{S}_1^\bullet$ and of type $\mathbb{S}_3$, when $|S_{i-1} \cap N(v_i)| = 1$. Similarly, for $p \geq 2$, the move is of type $\mathbb{S}_3$ or $\mathbb{S}_1^\bullet$, when v_i has one neighbor or has no neighbors in S_{i-1}, respectively, while for $|S_{i-1} \cap N(v_i)| \geq 2$, the move is of type $\mathbb{C}_5$.

Case 2. [$N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \subseteq N(v_i)$ and $G[S_{i-1}] \simeq K_p$, $p \geq 2$]
The only types of moves that can be taken into account are $\mathbb{S}_2, \mathbb{S}_4, \mathbb{I}_6$ and $\mathbb{I}_7$. If $S_{i-1} \cap N(v_i) = \emptyset$, then we clearly have a move of type $\mathbb{I}_6$. If $|S_{i-1} \cap N(v_i)| = p - 1$, then we have a move of type $\mathbb{S}_4$. If $S_{i-1} \subseteq N(v_i)$, then the move is of type $\mathbb{S}_2$. It remains to consider the case when $1 \leq |S_{i-1} \cap N(v_i)| \leq p - 2$, which directly corresponds to a move of type $\mathbb{I}_7$.

Case 3. [$N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \nsubseteq N(v_i)$ and $G[S_{i-1}] \simeq N_p$, $p \geq 0$]
It will be enough to consider $\mathbb{I}_1$ and $\mathbb{I}_2$. If $p = 0$, then obviously v_i has no neighbor in S_{i-1}. Hence, the move is of type $\mathbb{I}_1$. By Lemma 4.2(a) we have that $|S_{i-1} \cap N(v_i)| \leq 1$. Therefore, for $p > 0$, the move is of type $\mathbb{I}_1$, when $S_{i-1} \cap N(v_i) = \emptyset$ and of type $\mathbb{I}_2$, when $|S_{i-1} \cap N(v_i)| = 1$.

Case 4. [$N(v_i) \cap I_{i-1} = \emptyset$ and $C_{i-1} \nsubseteq N(v_i)$ and $G[S_{i-1}] \simeq K_p$, $p \geq 2$]
The only types of moves to consider are $\mathbb{I}_3, \mathbb{I}_4$ and $\mathbb{I}_5$. If $|S_{i-1} \cap N(v_i)| = 0$, then the move is of type $\mathbb{I}_3$,

post-state / pre-state	N_0	$N_1 \simeq K_1$	N_2	N_3	$\cdots$	K_2	K_3	K_4	$\cdots$
N_0	$\mathbb{I}_1, \mathbb{C}_3^\circ$	$\mathbb{S}_1^\circ$	–	–		–	–	–	
$N_1 \simeq K_1$	$\mathbb{I}_2, \mathbb{C}_3^\bullet$	$\mathbb{I}_1, \mathbb{C}_4$	$\mathbb{S}_1^\bullet$	–		$\mathbb{S}_3$	–	–	
N_2	$\mathbb{I}_2, \mathbb{C}_3^\bullet$	$\mathbb{C}_1$	$\mathbb{I}_1, \mathbb{C}_1, \mathbb{C}_5$	$\mathbb{S}_1^\bullet$		$\mathbb{S}_3$	–	–	
N_3	$\mathbb{I}_2, \mathbb{C}_3^\bullet$	$\mathbb{C}_1$	$\mathbb{C}_1, \mathbb{C}_5$	$\mathbb{I}_1, \mathbb{C}_1, \mathbb{C}_5$		$\mathbb{S}_3$	–	–	
$\cdots$					$\cdots$				
K_2	$\mathbb{I}_5, \mathbb{C}_2$	$\mathbb{I}_4$	$\mathbb{S}_4$	–		$\mathbb{I}_3, \mathbb{I}_6, \mathbb{C}_4$	$\mathbb{S}_2$	–	
K_3	$\mathbb{I}_5, \mathbb{C}_2$	$\mathbb{I}_4$	$\mathbb{S}_4$	–		$\mathbb{I}_4, \mathbb{I}_7$	$\mathbb{I}_3, \mathbb{I}_6, \mathbb{C}_4$	$\mathbb{S}_2$	
K_4	$\mathbb{I}_5, \mathbb{C}_2$	$\mathbb{I}_4$	$\mathbb{S}_4$	–		$\mathbb{I}_4, \mathbb{I}_7$	$\mathbb{I}_4, \mathbb{I}_7$	$\mathbb{I}_3, \mathbb{I}_6, \mathbb{C}_4$	
$\cdots$					$\cdots$				$\cdots$

Tab. 1: Selected states and the appropriate types of Presenter's moves

and of type $\mathbb{I}_5$, when $|S_{i-1} \cap N(v_i)| = |S_{i-1}| = p$. If none of these conditions is satisfied, then the move is of type $\mathbb{I}_4$.

Case 5. $[N(v_i) \cap I_{i-1} \neq \emptyset$ and $C_{i-1} \subseteq N(v_i)$ and $G[S_{i-1}] \simeq N_p, p \geq 0]$
Note that the only types of moves that can be taken into account are $\mathbb{C}_1, \mathbb{C}_3^\circ$ and $\mathbb{C}_3^\bullet$. If $p = 0$, then we have a move of type $\mathbb{C}_3^\circ$. For $p > 0$, the move is of type $\mathbb{C}_3^\bullet$, when $N(v_i) \cap S_{i-1} = \emptyset$, and of type $\mathbb{C}_1$, when $|S_{i-1} \cap N(v_i)| \geq 1$.

Case 6. $[N(v_i) \cap I_{i-1} \neq \emptyset$ and $C_{i-1} \subseteq N(v_i)$ and $G[S_{i-1}] \simeq K_p, p \geq 2]$
It is enough to consider $\mathbb{C}_2$ and $\mathbb{C}_4$. By Lemma 4.2(b) we have that $|S_{i-1} \cap N(v_i)| \geq p - 1$. Hence, for any $p \geq 2$, we have a move of type $\mathbb{C}_2$, when $|S_{i-1} \cap N(v_i)| = p - 1$, while for $|S_{i-1} \cap N(v_i)| = p$ the move is of type $\mathbb{C}_4$. □

Though there are infinitely many possible states, the number of types of moves is finite. A closer analysis of the state transition digraph's structure reveals some interesting properties that are crucial when proving 2-competitiveness of SR. See Table 1 for a collection of representative states and types of moves that correspond to the appropriate transitions.

7 Analysis of the Coloring Stage

Let $\pi = (v_1, \ldots, v_n)$ be a fixed ordering of the vertices of a split graph G and let $r_i = \chi_{\mathsf{SR}}(G_i, \pi)/\chi_r(G_i)$ for each $i \in \{1, \ldots, n\}$. In order to prove the main result we have to analyze all possible sequences (r_n). Obviously $r_i \geq 1$ and $\chi_{\mathsf{SR}}(G_i, \pi) - \chi_{\mathsf{SR}}(G_{i-1}, \pi) \in \{0, 1\}$ for each ordering π and for each $i > 1$. Concerning the denominator of r_i observe that it may differ from that of r_{i-1} when the move P_i results in the increase of the clique number but also when it changes the type of a split graph. A Presenter's move that changes the type from $\mathcal{A}$ to $\mathcal{B}$ ($\mathcal{B}$ to $\mathcal{A}$) is called an $\mathbb{AB}$-*move* ($\mathbb{BA}$-*move*, respectively). If the type is not changed the move is called an $\mathbb{X}$-*move*. Since a $\mathbb{BA}$-move P_i cannot increase the clique number, we have $\omega(G_i) = \omega(G_{i-1})$ and consequently

$$\chi_r(G_i) \stackrel{\mathcal{A}}{=} \omega(G_i) + 1 = \omega(G_{i-1}) + 1 \stackrel{\mathcal{B}}{=} \chi_r(G_{i-1}) + 1. \tag{1}$$

Analogously, since every $\mathbb{AB}$-move increases the clique number, i.e., $\omega(G_i) = \omega(G_{i-1}) + 1$, we have

$$\chi_r(G_i) \stackrel{\mathcal{B}}{=} \omega(G_i) = \omega(G_{i-1}) + 1 \stackrel{\mathcal{A}}{=} \chi_r(G_{i-1}). \tag{2}$$

For $\mathbb{X}$-moves the types of G_i and G_{i-1} are the same. Therefore, an increase of the clique number implies an increase of the ranking number. It finally follows that $\chi_r(G_i) - \chi_r(G_{i-1}) \in \{0, 1\}$ for each $i > 1$. The above-mentioned properties let us classify the rounds of our game. We use names which represent the viewpoint of the algorithm. A round R_i is said to be:

(a) a *tie round*, if $\chi_r(G_i) = \chi_r(G_{i-1})$ and $\chi_{\mathsf{SR}}(G_i, \pi) = \chi_{\mathsf{SR}}(G_{i-1}, \pi)$,

(b) a *lost round*, if $\chi_r(G_i) = \chi_r(G_{i-1})$ and $\chi_{\mathsf{SR}}(G_i, \pi) = \chi_{\mathsf{SR}}(G_{i-1}, \pi) + 1$,

(c) a *won round*, if $\chi_r(G_i) = \chi_r(G_{i-1}) + 1$ and $\chi_{\mathsf{SR}}(G_i, \pi) \leq \chi_{\mathsf{SR}}(G_{i-1}, \pi) + 1$.

Consequently, for a tie round $r_i = r_{i-1}$, while for a lost one $r_i > r_{i-1}$. Note that without loss of generality we may assume that whenever i is large enough, $\chi_{\mathsf{SR}}(G_i, \pi) > \chi_r(G_i)$, i.e., $r_i > 1$ (for otherwise SR would be optimal). Thus, for a won round we have that $r_i < r_{i-1}$, except for the several rounds at the very beginning of the game when $r_i = 1$.

The type of a round $R_i = (P_i, A_i)$ depends both on Presenter's move P_i and the algorithm's response A_i. Let us analyze their interactions in a more detailed manner, in particular, we focus on distinguishing won, lost and tie rounds.

Observe that a graph G is of type $\mathcal{A}$ if and only if S of the tripartition (C, S, I) of $V(G)$ is empty, which follows from the uniqueness of the split bipartition of $V(G)$ for graphs of this type. Thus, the necessary condition for the move P_i to be a $\mathbb{BA}$-move is its post-state to be N_0. The possible types of such moves are: $\mathbb{I}_1, \mathbb{I}_2, \mathbb{I}_5, \mathbb{C}_2$ and $\mathbb{C}_3$. Note that a move of type $\mathbb{I}_1$ does not change the type of a split graph, while no move of type $\mathbb{C}_3^\circ$ may occur when G_{i-1} is of type $\mathcal{B}$. Following these observations, we conclude that the only $\mathbb{BA}$-moves are those of type $\mathbb{C}_2$, $\mathbb{C}_3^\bullet$, $\mathbb{I}_2$ and $\mathbb{I}_5$. By Equation (1) and by the definition of a won round, whenever Presenter makes one of these moves, the round is won.

Let us now analyze the $\mathbb{AB}$-moves. Since, as noted before, a split graph is of type $\mathcal{A}$ if and only if the set S of the split tripartition of its vertex set is empty, the pre-state of each $\mathbb{AB}$-move is N_0. Hence, the only candidates for $\mathbb{AB}$-moves are the moves of type $\mathbb{S}_1^\circ, \mathbb{I}_1$ and $\mathbb{C}_3^\circ$. However, if G_{i-1} is of type $\mathcal{A}$, then each move of type $\mathbb{I}_1$ or $\mathbb{C}_3^\circ$ results in G_i also of type $\mathcal{A}$. This implies that the only $\mathbb{AB}$-move is a move of type $\mathbb{S}_1^\circ$. In order to avoid tedious analysis of the algorithm's responses we assume that whenever Presenter makes a move of type $\mathbb{S}_1^\circ$, the algorithm uses a new color, and since by Equation (2), $\chi_r(G_i) = \chi_r(G_{i-1})$ the round is lost. The only exception is $R_1 = (P_1, A_1)$ with P_1 of type $\mathbb{S}_1^\circ$ which is a won round.

The remaining moves are $\mathbb{X}$-moves. Recall that since $\mathbb{X}$-moves do not change the type of a split graph, the ranking number increases only if the clique number does, i.e., when $\omega(G_i) > \omega(G_{i-1})$. It is not hard to see that the only $\mathbb{X}$-moves that increase the clique number are the moves of type $\mathbb{S}_2, \mathbb{S}_3, \mathbb{C}_1, \mathbb{C}_3^\circ, \mathbb{C}_4$ and $\mathbb{C}_5$. Hence, the corresponding rounds are won. If Presenter makes a move of type $\mathbb{I}_1, \mathbb{I}_3, \mathbb{I}_4, \mathbb{I}_6$ or $\mathbb{I}_7$, then SR obviously reuses color 1 and since none of these moves increases the clique number, we have tie rounds, with the only exception, when the algorithm uses color 1 for the first time. The round is lost then, but only once per game. To simplify the analysis, all rounds in which Presenter makes a move of type $\mathbb{S}_4$ are assumed to be lost, while for any move of type $\mathbb{S}_1^\bullet$ the round may be lost or tie (see the detailed analysis in the proof of Lemma 7.2).

Before we continue the analysis let us summarize the above observations in the form of the following lemma.

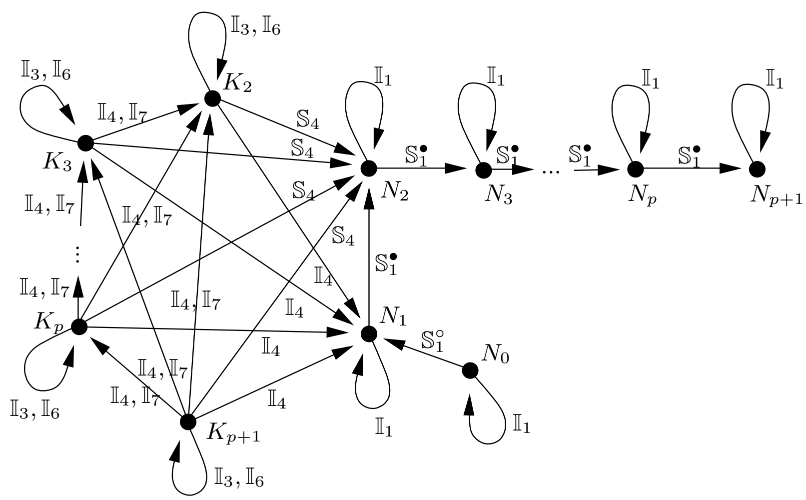

Fig. 7: The subdigraph D' of the state transition digraph D

Lemma 7.1 *Each round $R_i = (P_i, A_i)$ with P_i of type:*

(a) $\mathbb{S}_1^\circ$*, when $i = 1$, or* $\mathbb{I}_2, \mathbb{I}_5, \mathbb{S}_2, \mathbb{S}_3, \mathbb{C}_1, \mathbb{C}_2, \mathbb{C}_3^\circ, \mathbb{C}_3^\bullet, \mathbb{C}_4, \mathbb{C}_5$*, is won,*

(b) $\mathbb{I}_1, \mathbb{I}_3, \mathbb{I}_4, \mathbb{I}_6, \mathbb{I}_7$*, when there exists v_j, $j < i$, such that $c(v_j) = 1$, is tie,*

(c) $\mathbb{I}_1, \mathbb{I}_3, \mathbb{I}_4, \mathbb{I}_6, \mathbb{I}_7$*, when there does not exist v_j, $j < i$, such that $c(v_j) = 1$, is lost.*

The above-mentioned properties of moves of various types allow us to prove the following result.

Lemma 7.2 *Let R_t, R_q, $t < q$, be two lost rounds such that neither of them uses color 1 for the first time. Then, there exists i, $t < i < q$, such that R_i is won. Moreover, the first lost round is preceded by at least two won rounds.*

Proof: Let D' be the subdigraph of the state transition digraph D, containing only these arcs of D that correspond to the rounds with the moves of types *not* mentioned in Lemma 7.1(a) (see Figure 7). It is crucial to observe that D' is acyclic (except for the loops). Consequently, any directed walk in D that represents transitions which occur during the game and contains no arc corresponding to a won round is a directed path in D'. Therefore, it remains to argue that there does not exist a directed path in D' containing two arcs that correspond to some lost rounds.

Looking for such directed paths, observe that whenever Presenter makes some number of moves of type $\mathbb{S}_1$ in a row, i.e.,

$$(\mathbb{S}_1^\circ, \mathbb{S}_1^\bullet, \mathbb{S}_1^\bullet, \ldots) \tag{3}$$

or

$$(\mathbb{S}_1^\bullet, \mathbb{S}_1^\bullet, \mathbb{S}_1^\bullet \ldots)\,, \tag{4}$$

the thesis follows if the color used by SR in response to the first move of the above sequences is further reused as a response to all the subsequent moves of type $\mathbb{S}_1^\bullet$, which results in rounds that are not lost. The same holds for sequences starting with $\mathbb{S}_4$, i.e.,

$$(\mathbb{S}_4, \mathbb{S}_1^\bullet, \mathbb{S}_1^\bullet, \ldots)\,. \tag{5}$$

In fact any sequence (4) or (5) may be preceded by a sequence composed of the moves of types $\mathbb{I}_3, \mathbb{I}_4, \mathbb{I}_6$ and $\mathbb{I}_7$. Similarly, there may be the rounds when Presenter makes a move of type $\mathbb{I}_1$ between the moves in sequences (3)-(5). However, all loops of D' correspond to the rounds that are tie or eventually one of them is lost, when color 1 is used for the first time. Therefore, we do not have to take them into account in this proof.

We continue with the assumption that the value of the variable lc is greater than 1 during the entire execution of the algorithm. We will justify this assumption later on. First we argue that for any execution of SR, the color used in round R_{i-1}, corresponding to the first move in any of the sequences (3)-(5), can be reused in the subsequent rounds of these sequences. Observe that, as a result of round R_{i-1}, we have that $v_{i-1} \in S_{i-1}$ and S_{i-1} is independent (see the post-states of $\mathbb{S}_1^\circ, \mathbb{S}_1^\bullet$ and $\mathbb{S}_4$). Next, in round R_i with a move of type $\mathbb{S}_1^\bullet$, the vertex v_i is classified to S_i and by the definition of $\mathbb{S}_1^\bullet$ it holds that S_i is also independent. Moreover, $v_{i-1} \in S_i$. Since in the rounds with the moves of types $\mathbb{S}_1, \ldots, \mathbb{S}_4$ the algorithm uses neither color 1 nor color hc, it follows that $c(v_{i-1})$ is the smallest color assigned to the vertices in $C_{i-1} \cup S_{i-1}$. Therefore, by Property 4.1(b), $c(v_i) = c(v_{i-1})$ does not violate the definition of ranking, and consequently, $c(v_{i-1})$ is permissible also for v_i.

It remains to prove that during every execution of the algorithm, the value of lc decreases at most 2ω times, which justifies our earlier assumption that lc is always greater than 1. Obviously, in every game round R_1 is won. It is also not hard to see that rounds R_i, $i > 1$, are tie as long as G_i is edgeless. Moreover, any round R_i such that $|E(G_i)| \geq 1$ and $E(G_{i-1}) = \emptyset$ is won. Thus, in every game the first lost round is always preceded by at least two won rounds. As argued earlier, each sequence (3)-(5) is preceded by a won round, and contains at most one move corresponding to a lost round. By the definition of a won round, $\chi_r(G_i) = \chi_r(G_{i-1}) + 1$. Hence, there are $p \leq \chi_r(G) \leq \omega(G) + 1$ won rounds and at most $p - 1 \leq \omega(G)$ rounds that are lost. Consequently, there are at most $2\omega(G) + 1$ different values of lc used during the entire execution of SR and hence at most $2\omega(G)$ moves may result in the decrease of lc. This implies that $lc > 1$ in each iteration of the main loop of SR. □

Theorem 7.1 *For any split graph G we have $\chi_{\mathsf{SR}}(G) \leq 2\chi_r(G)$.*

Proof: Let π be any permutation of the vertices of G and let R_i be the round in which the color of v_i has been set, $i \in \{1, \ldots, n\}$. We are going to prove by induction on i that for each $i \in \{1, \ldots, n\}$

(i) if there is no round R_j, $j \leq i$, in which SR uses color 1, then $r_i < 2$,

(ii) if there exists a round R_j, $j \leq i$, in which SR uses color 1, then $r_i \leq 2$.

First observe that G_1 has a single vertex, which means that $\chi_{\mathsf{SR}}(G_1, \pi) = \chi_r(G_1) = 1$ and the hypothesis holds. Assume now that (i) and (ii) hold for each $i' \in \{1, \ldots, i\}$ and we will prove the claim for G_{i+1}, $i < n$. The proof falls naturally into the three cases:

Case 1. R_{i+1} is a tie round.
By the definition, $\chi_r(G_{i+1}) = \chi_r(G_i)$ and $\chi_{\mathsf{SR}}(G_{i+1}, \pi) = \chi_{\mathsf{SR}}(G_i, \pi)$. So, $r_{i+1} = r_i$ and the induction hypothesis completes the proof in this case.

Case 2. R_{i+1} is a won round.
We have that $\chi_r(G_{i+1}) = \chi_r(G_i) + 1$ and $\chi_{\mathsf{SR}}(G_{i+1}, \pi) \leq \chi_{\mathsf{SR}}(G_i, \pi) + 1$. So, by (ii) (and analogously by (i))

$$r_{i+1} \leq \frac{\chi_{\mathsf{SR}}(G_i, \pi) + 1}{\chi_r(G_i) + 1} \leq \frac{2\chi_r(G_i) + 1}{\chi_r(G_i) + 1} < 2,$$

and the hypothesis follows.

Case 3. R_{i+1} is a lost round.

Subcase 3.1 Let us first analyze the situation when R_{i+1} is the round in which color 1 was used for the first time. By assumption (i) we have $r_i < 2$, which implies $\chi_{\mathsf{SR}}(G_i, \pi) \leq 2\chi_r(G_i) - 1$. Hence,

$$r_{i+1} = \frac{\chi_{\mathsf{SR}}(G_{i+1}, \pi)}{\chi_r(G_{i+1})} = \frac{\chi_{\mathsf{SR}}(G_i, \pi) + 1}{\chi_r(G_i)} \leq \frac{(2\chi_r(G_i) - 1) + 1}{\chi_r(G_i)} = 2,$$

where the second equality follows directly from the definition of a lost round.

Subcase 3.2 Now assume that SR used for v_{i+1} some color larger than 1 and that color 1 has not been used in rounds $R_1, \ldots, R_i$. Additionally, let $i' = 0$ if there was no lost round preceding R_{i+1} or let $i' = \max\{j \in \{1, \ldots, i\} \,|\, R_j \text{ is lost}\}$ otherwise. Assume that there were k won rounds in $\{R_{i'+1}, \ldots, R_i\}$. By Lemma 7.2 we have that $k \geq 1$ when $i' > 0$ and that $k \geq 2$ when $i' = 0$. By the definition of a won round, $\chi_r(G_i) = \chi_r(G_{i'}) + k$. Notice that all rounds in $\{R_{i'+1}, \ldots, R_i\}$ are either won or tie, which implies that $\chi_{\mathsf{SR}}(G_i, \pi) \leq \chi_{\mathsf{SR}}(G_{i'}, \pi) + k$. Then,

$$r_{i+1} = \frac{\chi_{\mathsf{SR}}(G_{i+1}, \pi)}{\chi_r(G_{i+1})} = \frac{\chi_{\mathsf{SR}}(G_i, \pi) + 1}{\chi_r(G_i)} \leq \frac{\chi_{\mathsf{SR}}(G_{i'}, \pi) + k + 1}{\chi_r(G_{i'}) + k}.$$

We need to prove that $r_{i+1} < 2$. If $i' = 0$, then $\chi_{\mathsf{SR}}(G_{i'}, \pi) = 0$ and $\chi_r(G_{i'}) = 0$ and the thesis follows. Otherwise, by (i) of the induction hypothesis,

$$r_{i+1} < \frac{2\chi_r(G_{i'}) + k + 1}{\chi_r(G_{i'}) + k} = \frac{2(\chi_r(G_{i'}) + k) + 1 - k}{\chi_r(G_{i'}) + k} \leq 2,$$

which proves (i).

Subcase 3.3 If for v_{i+1} the algorithm used a color larger than 1, but color 1 has already been used in one of the preceding rounds, then it is enough to notice that the first usage of color 1 always results in a lost round. The rest of the proof is analogous to Subcase 3.2 and we get $r_{i+1} \leq 2$.

This completes the analysis of all possible cases, and since π was chosen arbitrarily, $\chi_{\mathsf{SR}}(G) \leq 2\chi_r(G)$ follows. $\square$

8 Conclusions

The semi on-line algorithm SR presented in this paper uses at most $2\chi_r(G)$ colors for any split graph G. Recall that the optimization goal used in this work, i.e., minimization of the number of pairwise different colors assigned by the algorithm, is different from the maximum color criterion. We now point out that SR is constant competitive for the latter criterion as well. By construction, SR assigns color 1 to a vertex v_i that belongs to I_i, and one can prove that if $v_i \in S_i$, then $c(v_i)$ equals the value of lc at the end of round R_i. Moreover, if a move corresponding to some round R_i is of one of the types $\mathbb{C}_1, \ldots, \mathbb{C}_5$, then $|C_i| > |C_{i-1}|$. Therefore, since $|C_n| \leq \omega(G)$ there are at most $\omega(G)$ rounds with the moves of types $\mathbb{C}_1, \ldots, \mathbb{C}_5$, which implies that hc increases at most $\omega(G)$ times. Consequently, the maximum color used by SR is $3\omega(G)+2$, which means that SR is 3-competitive in the sense of the second optimization criterion.

Although for minimization of the number of colors we have proved that SR is 2-competitive provided that $\omega(G)$ is given, it is worth pointing out that it suffices when SR only knows an upper bound on $\omega(G)$. In fact, we have proved that the algorithm works correctly when lc and hc are set to at least $2\omega(G)+2$ in the initialization stage. Therefore, every algorithm that can find an upper bound k on $\omega(G)$ and calls $\mathsf{SR}(k)$ is also 2-competitive. In particular, there exists a 2-competitive algorithm that is given in advance the order n of a graph G.

Observe that a similar reasoning cannot be applied to semi on-line algorithms that know n, whenever we minimize the maximum color. We can argue, however, that there exists a $(3\sqrt{n}+2)$-competitive algorithm B of such type. To this end we first extend SR (let SR' denote the corresponding new algorithm) in such a way that whenever lc reaches the value of 1 in round i, then in the subsequent rounds SR' uses pairwise different colors that are greater than or equal to the value of hc in round i. Let B proceed by calling $\mathsf{SR}'(\sqrt{n})$. If $\omega(G) < \sqrt{n}$, then as argued in the proof of Lemma 7.2, the value of lc is greater than 1 during the entire execution of SR. Therefore, the largest color used by B is bounded by $3\sqrt{n}+2$ and the claim trivially follows. Otherwise, $\chi_r(G) \geq \sqrt{n}$ and SR' produces a vertex ranking such that for any graph G, the maximum color never exceeds n. Consequently, SR' is $\sqrt{n}$-competitive in this case. On the other hand, $3\sqrt{n}+2$ constitutes an asymptotically tight bound, i.e., no semi on-line algorithm A, that is given n in advance, can be $(\sqrt{n}-\varepsilon)$-competitive for the maximum color minimization. Indeed, on the contrary, suppose that such an algorithm exists and consider two natural cases. If the first color used by A is less than $\sqrt{n}$, then by an easy generalization of Theorem 3.1, there exists Presenter's strategy that forces A to use n colors for a graph G with $\chi_r(G) \leq \sqrt{n}$. On the other hand, if the first color used is at least $\sqrt{n}$, then Presenter fixes G to be N_n (note that $\chi_r(N_n) = 1$). Clearly, in both cases $\chi_{\mathsf{A}}(G)/\chi_r(G) \geq \sqrt{n}$, which contradicts our assumption.

Since the lower bound given in Theorem 3.2 holds for both criteria (every vertex has a different color), an interesting direction for further research is a construction of a semi on-line ranking algorithm which is 2-competitive with respect to the minimization of the maximum color.

We finish with the remark that Dobrev *et al.* [10] introduced a formal framework that allows to classify on-line problems according to how much information (advice bits) about the future input parts is needed for solving them optimally or with a specific competitive ratio (see also [2, 5] for other results). From the results proved in this paper it also follows that at least $\lfloor \log_2 \omega \rfloor + 1$ bits of advice are required for any on-line ranking algorithm to be constant competitive on split graphs and that this number of bits is sufficient to achieve a constant competitive ratio for both optimization criteria.

Acknowledgements

We would like to acknowledge the anonymous referees for their valuable comments that permitted us to significantly improve the readability of the paper.

References

[1] J. Bang-Jensen, G. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer (2009)

[2] P. Bianchi, H.-J. Böckenhauer, J. Hromkovič and L. Keller, Online coloring of bipartite graphs with and without advice, *Proc. of the 18th Annual International Conference on Computing and Combinatorics* (COCOON 2012), LNCS 7434 (2012) 519–530

[3] P. Borowiecki, On-line coloring of graphs, in: M. Kubale (ed.): *Graph Colorings*, *Contemporary Mathematics* 352, American Mathematical Society (2004) 21–33

[4] P. Borowiecki, K. Budajová, S. Jendrol' and S. Krajči, Parity vertex colouring of graphs, *Discussiones Mathematicae Graph Theory* 31 (2011) 183–195

[5] H.-J. Böckenhauer, D. Komm, Ra. Královič, Ri. Královič and T. Mömke, On the advice complexity of online problems, *Proc. of the 20th International Symposium on Algorithms and Computation* (ISAAC 2009), LNCS 5878 (2009) 331–340

[6] E. Bruoth, M. Horňák, On-line ranking number for cycles and paths, *Discussiones Mathematicae Graph Theory* 19 (1999) 175–197

[7] E. Bruoth, M. Horňák, A lower bound for on-line ranking number of a path, *Discrete Mathematics* 307 (11–12) (2007) 1347–1355

[8] D.P. Bunde, K. Milans, D.B. West and H. Wu, Parity and strong parity edge-colorings of graphs, *Congressus Numerantium* 187 (2007) 193–213

[9] D. Dereniowski, Rank coloring of graphs, in: M. Kubale (ed.): *Graph Colorings*, *Contemporary Mathematics* 352, American Mathematical Society (2004) 79–93

[10] S. Dobrev, R. Královič and D. Pardubská, How much information about the future is needed?, *Proc. of the 34th Conference on Current Trends in Theory and Practice of Computer Science* (SOFSEM 2008), LNCS 4910 (2008) 247–258

[11] T. Ebenlendr, J. Sgall, Semi-Online Preemptive Scheduling: One Algorithm for All Variants, *Theory of Computing Systems* 48 (3) 577–613

[12] G. Even, Z. Lotker, D. Ron and S. Smorodinsky, Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks, *SIAM Journal on Computing* 33 (2003) 94–136

[13] S. Földes, P.L. Hammer, Split Graphs, *Congressus Numerantium* 19 (1977) 311–315

[14] J. Ghoshal, R. Laskar and D. Pillone, Minimal rankings, *Networks* 28 (1) (1996) 45–53

[15] P.L. Hammer, B. Simeone, The splittance of a graph, *Combinatorica* 1 (3) (1981) 275–284

[16] G. Isaak, R. Jamison and D. Narayan, Greedy rankings and arank numbers, *Information Processing Letters* 109 (2009) 825–827

[17] H. Kellerer, V. Kotov, M. Grazia Speranza and Zs.Tuza, Semi on-line algorithms for the partition problem, *Operations Research Letters* 21 (1997) 235–242

[18] H.A. Kierstead, Coloring Graphs On-line, in: A. Fiat and G.J. Woeginger (eds): *Online Algorithms. The State of the Art*, LNCS 1442 (1998) 281–305

[19] M. Kubale (ed.), *Graph Colorings*, *Contemporary Mathematics* 352, American Mathematical Society (2004)

[20] I. Schiermeyer, Zs. Tuza and M. Voigt, On-line rankings of graphs, *Discrete Mathematics* 212 (1–2) (2000) 141–147

[21] S. Seiden, J. Sgall and G. Woeginger, Semi-online scheduling with decreasing job sizes, *Operations Research Letters* 27 (5) (2000) 215–221

[22] G. Semanišin, R. Soták, A note on on-line ranking number of graphs, *Czechoslovak Mathematical Journal* 56 (2) (2006) 591–599

[23] D.B. West, *Introduction to Graph Theory*, Prentice Hall (2001)