Institute of Computer Science
Faculty of Mathematics and Computer Science
Jagiellonian University

ON-LINE GRAPH COLORING

Iwona Cieślik

PhD Thesis
Supervisor: Prof. Paweł M. Idziak

Kraków 2006

I would like to express my deepest gratitude
to my supervisor Prof. Paweł M.Idziak
for his care and patience,
for his time devoted to my scientific development,
for his invaluable help in my research work
during writing this thesis.

Contents

Chapter 1

Introduction

The traditional design and analysis of algorithms assumes that the entire input is available when the algorithm starts. In other words, problems are presented together with a complete knowledge of the entire input. In many cases this assumption does not correspond to the real world situation. There are algorithmic problems in which the input is only partially available because some input data will arrive in the future and is not accessible at present. Examples of such problems include: real-time systems, routing in communication networks, scheduling tasks on servers, paging in a virtual memory, dynamic storage allocation, etc. We call these problems *on-line* problems. The algorithms for these problems make decisions and give answers without having a complete knowledge of the possible development of the situation. Thus *on-line algorithms* are a special class of algorithms in which we impose some restriction on the way of obtaining (and therefore processing) the input data. Input is not known at the beginning, and it is delivered during the performance of an algorithm. Additionally, the decision made by an algorithm can not by changed in its further performance. Since the knowledge of the entire input is not available, the action may seem to be correct at the moment, but it may turn out to be wrong (or at least not optimal) later. Thus the on-line algorithms often produce results that are worse than these given by traditional (off-line) algorithms. On-line algorithms are approximative in the sense that they do not guarantee the optimal solution.

On-line problems may be viewed as two-person games. We call the players of the game *the algorithm* and *the spoiler*. At each step the spoiler presents (i.e., uncovers) a new part of the data. The algorithm, in turn, extends his existing solution to include the new part of the structure. No previously made decisions can be changed at any step. The aim of the algorithm is to find a result as close as possible to the result of an optimal (off-line) algorithm. The aim of the spoiler is to force the algorithm to make wrong decisions and find solution that are far from the optimal ones.

The main data structures used to describe relationships occurring in the real world are graphs. Many practical problems can be reduced to the problem of graph coloring (sometimes with some additional constraints). The problem of graph coloring using the smallest possible number of colors is NP-hard. The corresponding decision problem (i.e., is there a coloring which uses at most k colors?) is NP-complete [20]. This problem remains NP-complete even for planar graphs of degree at most 4 [13] (although it's trivial for planar graphs for $k \neq 3$ due to the four colors theorem). This means that finding an optimal solution of graph coloring is computationally hard. For this reason, it is worthwhile to search for approximating algorithms, which are trying to find the solution as close as possible to the optimal ones, and return the output quickly. The on-line algorithms may serve as examples of such algorithms.

In 1966 R.Graham [14] considered the problem of scheduling tasks on identical machines. A variant of this problem, in which the performance of the task is not allowed to be interrupted can be interpreted as a question of coloring of an interval graph. An equivalent scenario of dynamic storage allocation due to M.Chrobak and M.Ślusarek [8] became a motivation for introducing the notion of the on-line coloring of interval graphs and, more generally, various families of graphs.

We would like to compare the results of on-line and off-line algorithms. A useful measure for such a comparison are the *competitive functions*. They characterize the quality of the on-line algorithm by comparing its results to the results of the optimal off-line algorithms. An on-line algorithm $\mathcal{A}$ is *competitive* if there exists a function f such that on each input x the (numerical) result $\mathcal{A}(x)$ obtained by $\mathcal{A}$ is bounded above by $f(opt(x))$, where $opt(x)$ is an optimal (off-line) solution. In this case we say that f is a *competitive function* for the algorithm $\mathcal{A}$. It is known that the on-line coloring problem for arbitrary graphs is hard, that is, there is no on-line coloring algorithm with a competitive function (It follows from the fact that there is no competitive algorithm for trees [2, 17]). In this thesis we try to isolate and describe families of graphs for which there exist competitive on-line coloring algorithms. The results show some relationships: the power of on-line coloring depends, to some extent, on the absence of certain structures in the input graphs. These structures, induced subgraphs, allow the spoiler to force additional colors. On the other hand if a graph does not contain such forcing subgraphs, the spoiler has his hands tied and he is able to cheat the on-line algorithms only in a very restricted way. Determining such forcing structures is one of the goals of this thesis.

1.1 Basic definitions and notions

A *graph* $\mathbb{G}$ is a pair $\mathbb{G} = (G, E_{\mathbb{G}})$ where $E_{\mathbb{G}}$ is some set of the two element subset of G. The set of all two element subset of a set G we denote by $\binom{G}{2}$ (thus $E_{\mathbb{G}} \subseteq \binom{G}{2}$). The elements of G are called *vertices* of $\mathbb{G}$ and the elements of $E_{\mathbb{G}}$ are called *edges*. The graphs we consider in this dissertation are finite. For graph terminology, which is not introduced in the following, we refer the reader to [5] and [11].

For two graphs $\mathbb{G} = (G, E_{\mathbb{G}})$ and $\mathbb{H} = (H, E_{\mathbb{H}})$ if $H \subseteq G$ and $E_{\mathbb{H}} \subseteq E_{\mathbb{G}}$, then $\mathbb{H}$ is a *subgraph* of $\mathbb{G}$, denoted by $\mathbb{H} \subseteq \mathbb{G}$. In such a case we say that $\mathbb{G}$ *contains* $\mathbb{H}$. If $\mathbb{H} \subseteq \mathbb{G}$ and $E_{\mathbb{H}}$ contains all the edges $\{x, y\} \in E_{\mathbb{G}}$ with $x, y \in H$ then $\mathbb{H}$ is an *induced subgraph* of $\mathbb{G}$. In this case we say that H induces $\mathbb{H}$ in $\mathbb{G}$ and we write $\mathbb{H} = \mathbb{G}|_H$. On the other hand, if a graph $\mathbb{G}$ does not contain a graph $\mathbb{H}$ as an induced subgraph we say that $\mathbb{G}$ is $\mathbb{H}$-free. We write $\emptyset$ for the null graph $(\emptyset, \emptyset)$. If a graph $\mathbb{G}$ contains one vertex x only (and therefore no edges) we write $\mathbb{G} = \{x\}$. The *complement* $\overline{\mathbb{G}}$ of $\mathbb{G}$ is a graph $(G, \binom{G}{2} \setminus E_{\mathbb{G}})$.

A vertex x is *incident* with an edge e if $x \in e$. An edge $\{x, y\}$ is usually written as xy (or yx). Two vertices incident with an edge are its *ends*. We say that these vertices are *adjacent* or *neighbors*. If all the vertices of $\mathbb{G}$ are pairwise adjacent (i.e. $E_{\mathbb{G}} = \binom{G}{2}$) then $\mathbb{G}$ is *complete* and it is called a *clique*. Two graphs $\mathbb{G}$ and $\mathbb{H}$ are said to be isomorphic if there is a bijection between the vertices of the graphs such that two vertices are adjacent in $\mathbb{G}$ if and only if their corresponding vertices are adjacent in $\mathbb{H}$. Any graph isomorphic to a clique with s vertices is denoted by $\mathbb{K}_s$.

If H is a set of vertices, we write $\mathbb{G} \setminus H$ for $\mathbb{G}|_{G \setminus H}$. In other words, $\mathbb{G} \setminus H$ is obtained by deleting from $\mathbb{G}$ all the vertices in $H \cap G$ and their incident edges. The *degree*, $deg_{\mathbb{G}}(x)$, of a vertex x in $\mathbb{G}$ is the number of all edges incident to x. The degree, $\triangle(\mathbb{G})$, of a graph $\mathbb{G}$ is the largest degree of all its vertices. The set of all neighbors of a vertex x in $\mathbb{G}$ is called an *adjacency set* or a *neighborhood* of x and is denoted by $N_{\mathbb{G}}(x)$. Moreover, $N_{\mathbb{G}}(X)$ denotes the neighborhood of $X \subseteq G$, that is, the set of vertices which are adjacent to at least one vertex in X. Pairwise, non-adjacent vertices are called *independent*. More generally, a set of vertices of $\mathbb{G}$ is independent if no two of its elements are adjacent in $\mathbb{G}$. Any graph consisting of t independent vertices is said to be an *independent set* and denoted by $\mathbb{I}_t$.

A *k-element path* is a graph consisting of exactly k vertices $x_1, \ldots, x_k$ and exactly $k-1$ edges x_1x_2, $x_2x_3, \ldots, x_{k-1}x_k$. Any graph isomorphic to a path with k vertices is denoted by $\mathbb{P}_k$. If for any two vertices x, y in $\mathbb{G}$, there is a path in $\mathbb{G}$ from x to y, then $\mathbb{G}$ is said to be *connected*; otherwise, $\mathbb{G}$ is *disconnected* (the null graph $\emptyset$ and graphs containing one vertex are connected). A *connected component* of $\mathbb{G}$ is a maximal (with respect to inclusion) connected subgraph of $\mathbb{G}$. Thus, any disconnected graph can be divided into at least two connected components.

A *k-element cycle* is a graph consisting of exactly k vertices $x_1, \ldots, x_k$ and exactly k edges x_1x_2, $x_2x_3, \ldots, x_{k-1}x_k$ and x_1x_k. Any graph isomorphic to a cycle with k vertices is denoted by $\mathbb{C}_k$. The length of a cycle is the number k of its edges (or equivalently of its vertices). An odd cycle is a cycle of odd length, similarly, an even cycle is a cycle of even length.

A *forest* is a graph without cycles. A connected forest is called a *tree*. The vertices of degree 1 in a tree are its *leaves*. Sometimes, one vertex of the tree is distinguished and called the *root*. A rooted tree is a tree with a distinguished root. Trees form a subfamily of *bipartite graphs*. A graph is bipartite if there exists a partition of its vertices into 2 classes such that the ends of each edge are in different classes. In other words, a graph is bipartite if and only if it does not contain any odd cycle as a subgraph. A bipartite graph in which every two vertices from different classes are adjacent is called *complete*. Any graph isomorphic to a complete bipartite graph whose classes contain s and t vertices, respectively, is denoted by $\mathbb{K}_{s,t}$.

An edge which joins two vertices of a cycle but is not itself an edge of the cycle is a *chord* of that cycle. Thus, an induced cycle in $\mathbb{G}$ is one that has no chords. A graph in which every cycle of length at least k has at least one chord is called k-chordal. In other words, k-chordal graphs do not contain induced cycles of size bigger than $k-1$. The 4-chordal graphs are called, for short, chordal graphs.

The customary way to picture a finite graph is to draw a dot for each vertex and join the dots by a line if the corresponding vertices form an edge. If we can draw a graph on a plane in such a way that no two edges meet in a point other than a common end, we say that this graph is *planar*.

A *vertex-coloring* (or *coloring*, for short) of a graph $\mathbb{G} = (G, E_{\mathbb{G}})$ is a function $c : G \to \mathbb{N}_+$ which assigns to each vertex a positive integer (called a color of a vertex) in such a way that if two vertices are adjacent the assigned numbers are different. The *coloring problem* is a task of finding a coloring with the smallest possible number of colors. Such a number, for a graph $\mathbb{G}$, is called the *chromatic number* of $\mathbb{G}$ and denoted by $\chi(\mathbb{G})$. If $\chi(\mathbb{G}) \leqslant k$, we say that $\mathbb{G}$ is k-colorable. Note that the coloring of a graph with k colors leads to a partition of the vertex set of $\mathbb{G}$ into k independent sets. For example, the 2-colorable graphs are exactly the bipartite graphs.

The maximal size of cliques in a graph $\mathbb{G}$, denoted by $\omega(\mathbb{G})$, is the trivial lower bound for the chromatic number of $\mathbb{G}$. A graph $\mathbb{G}$ is *perfect* if every induced subgraph $\mathbb{H} \subseteq \mathbb{G}$ has the chromatic number equal to the size of the maximal clique in $\mathbb{G}$ (i.e. $\chi(\mathbb{H}) = \omega(\mathbb{H})$), thus $\omega(\mathbb{H})$ colors always suffices to color the vertices of $\mathbb{H}$. In the early 1960's C.Berge observed [3, 4] that several classical families of graphs, for example: trees, bipartite graphs, chordal graphs are perfect. In [15] authors showed that a complement of a perfect graph is also perfect. Recently M.Chudnovsky, N.Robertson, P.D.Seymour and R.Thomas [9] proved that a graph $\mathbb{G}$ is perfect if and only if neither $\mathbb{G}$ nor $\overline{\mathbb{G}}$ contains an odd cycle of length at least 5 as an induced subgraph. Perfect graphs are interesting from an algorithmic point of view. One of the reasons is the fact that determining the chromatic number for perfect graph is solvable in polynomial time [16].

An on-line version of the coloring problem may be viewed as a game between the spoiler and the on-line coloring algorithm. In each step of the game the spoiler presents one vertex of a graph together with edges connecting it to the previously presented vertices. The graph $\mathbb{G}$ together with the order of presentation of its vertices is called a *presentation* of $\mathbb{G}$. As each vertex is presented, the on-line coloring algorithm assigns a color (a positive integer) to it. This color cannot be changed afterwards. The goal of the algorithm is to find graph coloring with the smallest possible number of colors. The goal of the spoiler is to force the algorithm to use as many colors as possible. The most familiar on-line coloring algorithms are so called *greedy* algorithms, which try to assign – to the current vertex – a color already in use as long as this is possible. A simple, but important, example of on-line greedy algorithm is the algorithm using the *First-Fit strategy*. This algorithm assigns to the new vertex x the least possible color (number) not already assigned to any vertex adjacent to x.

Similarly, we define an on-line *clique covering* problem. The clique covering problem is a task of finding the partition of the vertices into the smallest possible number of cliques. The off-line solution of this problem, that is, the smallest number of cliques needed to cover the graph is denoted by $v(\mathbb{G})$. The on-line clique covering algorithm assigns positive integers to the presented vertices in such a way that vertices with the same number form a clique. The assigned cliques cannot be changed afterwards. It is easy to see that an optimal coloring of a graph $\mathbb{G}$ gives an optimal clique covering of $\overline{\mathbb{G}}$ (the complement of $\mathbb{G}$). Unfortunately, when analyzing some special family of graphs, the complements of some of them may not fall into this family. Thus, the above relation between coloring of $\mathbb{G}$ and clique covering of $\overline{\mathbb{G}}$ often does not help in coloring and clique covering problems for particular families of graphs. An often used on-line clique covering algorithm is the algorithm using the First Fit strategy. It is defined in the similar greedy way to the coloring algorithm with that strategy. From all the numbers which can be used for a current vertex, the clique covering algorithm using the First Fit strategy chooses the least one. Whenever it is clear from the context, which on-line algorithm is considered (coloring or clique covering) we denote an on-line algorithm using the First Fit strategy by $\mathcal{FF}$.

The quality of an on-line algorithm is measured by a function which compares its results to the optimal (off-line) results. An on-line graph algorithm $\mathcal{A}$ is *competitive* for the graph family $\mathcal{F}$ if there is a function f, called a *competitive function*, such that for any graph $\mathbb{G} \in \mathcal{F}$ and for any its presentation

$$\mathcal{A}(\mathbb{G}) \leqslant f(opt(\mathbb{G})),$$

where $\mathcal{A}(\mathbb{G})$ is the (numerical) result obtained by $\mathcal{A}$ for the fixed presentation of $\mathbb{G} \in \mathcal{F}$ and $opt(\mathbb{G})$ is an optimal (off-line) solution. Actually, we require that the displayed inequality holds for all but finitely many graphs $\mathbb{G} \in \mathcal{F}$. If a competitive function for the algorithm $\mathcal{A}$ is linear on the family $\mathcal{F}$, we define a *competitive ratio* to be the leading coefficient of the function f. In other words, if there exist a and c such that $\mathcal{A}(\mathbb{G}) \leq c \cdot opt(\mathbb{G}) + a$ for all but finitely many graphs $\mathbb{G} \in \mathcal{F}$, the number c is called a *competitive ratio* of the algorithm $\mathcal{A}$. For on-line coloring algorithm $\mathcal{A}$, $\mathcal{A}(\mathbb{G})$ is denoted by $\chi_{\mathcal{A}}(\mathbb{G})$, while for on-line clique covering algorithm $\mathcal{A}$, $\mathcal{A}(\mathbb{G})$ is denoted by $v_{\mathcal{A}}(\mathbb{G})$.

The on-line graph coloring has been widely studied for various families of graphs [2, 7, 17, 18, 19, 21, 22, 25]. There are some families of graphs for which there exist on-line competitive coloring algorithms, and there are others, for which no such algorithms exist. For example, there are on-line competitive algorithms for the following families of graphs: $\mathbb{P}_4$-free graphs [17], $\mathbb{P}_5$-free graphs [18, 22], $2\mathbb{K}_2$-free graphs [17], complements of bipartite and chordal graphs [17], interval graphs [21], circular arc graphs [25], split graphs [17]. On the other hand, there is no competitive on-line algorithm for some simple extensions of the above families: for trees [2, 17] or for $\mathbb{P}_6$-free graphs [17]. In particular, there is no competitive coloring algorithm for all graphs.

We also define the competitiveness of an on-line problem. A function f is *a competitive function for an on-line problem* for graph family $\mathcal{F}$ if

- there exists a competitive algorithm $\mathcal{A}$ solving this problem with a competitive function f

and

- for each on-line algorithm $\mathcal{B}$ solving this problem, its competitive function g is worse than the function f, i.e., $f(opt(\mathbb{G})) \leqslant g(opt(\mathbb{G}))$ for all but finitely many graphs $\mathbb{G} \in \mathcal{F}$.

In our proofs the second condition often reduces to construct a strategy for the spoiler such that for each on-line algorithm $\mathbb{B}$ it produces infinitely many on-line graphs $\mathbb{G} \in \mathcal{F}$ (i.e., graphs together with their presentations) on which $\mathcal{B}$ returns $\mathcal{B}(\mathbb{G}) \geqslant f(opt(\mathbb{G}))$.

Note that in an off-line problem a graph is presented on input as a set of vertices and a set of edges. The order of vertices and edges is not important. In an on-line problem an input graph is always delivered as a sequence of vertices and edges. The result returned by an on-line algorithm depends heavily on the order in which the input data are presented. Therefore, all graphs considered in this thesis are *on-line graphs*, that is, graphs together with a presentation. In this setting a graph with two different presentations is, in fact, considered as two different on-line graphs. For simplicity we will omit the words "together with some presentation" for on-line graphs as long as it is clear from the context whether we consider off-line or on-line graphs.

Since the result produced by an on-line algorithm depends on a graph presentation, it is worthwhile to consider the behavior of the on-line algorithms with restrictions imposed on the graph presentation. The main restriction we consider called a *presentation in a connected way*, is that the input graph is connected at each step of an on-line game. More precisely, the graph $\mathbb{G} = (\{x_1, \ldots, x_k\}, E)$ is presented in a connected way, whenever the vertices of $\mathbb{G}$ are presented in the order $x_1, \ldots, x_k$ then for every $i = 1, \ldots, k$ the subgraph induced by $\{x_1, \ldots, x_i\}$ is connected. In such a case it is more difficult to force mistakes and the on-line algorithms can sometimes perform essentially better. In Chapter 2 we will see that there is no competitive function for on-line coloring of trees and, more generally, of bipartite graphs. On the other hand, Proposition 2.2.1 says that if $\mathbb{G}$ is bipartite and presented in a connected way then an natural on-line coloring algorithm using the First Fit strategy colors it optimally, that is, $\chi_{\mathcal{FF}}(\mathbb{G}) = \chi(\mathbb{G})$.

1.2 List of notations used throughout the paper

- G = set of vertices of a graph $\mathbb{G}$.
- $E_{\mathbb{G}}$ = set of edges of a graph $\mathbb{G}$.
- $x_1 E_{\mathbb{G}}\, x_2 \equiv$ there exists an edge between vertices x_1 and x_2 in a graph $\mathbb{G}$.
- $N_{\mathbb{G}}(x)$ = adjacency set of a vertex x, that is, $N_{\mathbb{G}}(x) = \{y \in G : x E_{\mathbb{G}}\, y\}$.
- $N_{\mathbb{G}}(X) = \bigcup_{x \in X} N_{\mathbb{G}}(x)$.
- $|V|$ = size of a set V.
- $\omega(\mathbb{G})$ = size of the maximal clique in $\mathbb{G}$.
- $deg_{\mathbb{G}}(x)$ = degree of a vertex x, i.e., the number of edges incident to x in a graph $\mathbb{G}$.
- $\triangle(\mathbb{G})$ = degree of a graph $\mathbb{G}$, i.e., the largest degree of all vertices of $\mathbb{G}$.
- $\chi(\mathbb{G})$ = chromatic number of a graph $\mathbb{G}$, i.e., the minimal number of colors needed to color $\mathbb{G}$.
- $\chi_{\mathcal{A}}(\mathbb{G})$ = number of colors used by an on-line coloring algorithm $\mathcal{A}$ to color $\mathbb{G}$.
- $v(\mathbb{G})$ = minimal number of cliques needed to cover a graph $\mathbb{G}$.
- $v_{\mathcal{A}}(\mathbb{G})$ = number of cliques used by an on-line clique covering algorithm $\mathcal{A}$ to cover a graph $\mathbb{G}$.
- $c_{\mathcal{A}}(x)$ = color used by an on-line algorithm $\mathcal{A}$ to color x.
- $cl_{\mathcal{A}}(x)$ = clique used by an on-line algorithm $\mathcal{A}$ to cover x.
- $C_{\mathcal{A}}(\mathbb{G}, V)$ = set of colors used by an on-line algorithm $\mathcal{A}$ coloring a graph $\mathbb{G}$ to color vertices in $V \subseteq G$.
- $Cl_{\mathcal{A}}(\mathbb{G}, V)$ = set of cliques used by an on-line algorithm $\mathcal{A}$ covering a graph $\mathbb{G}$ to cover vertices in $V \subseteq G$.
- $C_{\mathcal{A}}(\mathbb{G}, \mathbb{H})$ = set of colors used by an on-line algorithm $\mathcal{A}$ coloring a graph $\mathbb{G}$ to color $\mathbb{H} \subseteq \mathbb{G}$.

- $Cl_{\mathcal{A}}(\mathbb{G}, \mathbb{H})$ = set of cliques used by an on-line algorithm $\mathcal{A}$ covering a graph $\mathbb{G}$ to cover $\mathbb{H} \subseteq \mathbb{G}$.

While it is clear from the context which graph is colored (covered) we simply use $C_{\mathcal{A}}(V)$, $Cl_{\mathcal{A}}(V)$, $C_{\mathcal{A}}(\mathbb{H})$ and $Cl_{\mathcal{A}}(\mathbb{H})$.

- $C_{\mathcal{A}}(\mathcal{F}) = \bigcup_{\mathbb{G} \in \mathcal{F}} C_{\mathcal{A}}(\mathbb{G})$ = set of colors used by an on-line algorithm $\mathcal{A}$ to color graphs from a family $\mathcal{F}$.

- $Cl_{\mathcal{A}}(\mathcal{F}) = \bigcup_{\mathbb{G} \in \mathcal{F}} Cl_{\mathcal{A}}(\mathbb{G})$ = set of cliques used by an on-line algorithm $\mathcal{A}$ to cover graphs from a family $\mathcal{F}$.

While it is clear from the context which optimal coloring (or clique covering) is considered we also use the following notations:

- $c(x)$ = color of a vertex x in an optimal coloring.

- $cl(x)$ = clique covering a vertex x in an optimal clique covering.

- $C(V)$ = set of colors used to color vertices in V in an optimal coloring.

- $Cl(V)$ = set of cliques used to cover vertices in V in an optimal clique covering.

- $C(\mathbb{G})$ = set of colors used to color $\mathbb{G}$ in an optimal coloring.

- $Cl(\mathbb{G})$ = set of cliques used to cover $\mathbb{G}$ in an optimal clique covering.

1.3 Results

In this thesis we analyze the competitiveness of coloring and clique covering problems for various families of graphs. We mostly analyze the case when some structures are forbidden in the input graphs. One of the methods most often used to force many colors while coloring or covering an on-line graph is to form a special configuration. These configurations are called *forcing structures* or *forcing subgraphs*. A graph is forcing if it necessarily appears when forcing the on-line algorithms to use arbitrarily many colors or cliques. That is, graphs which do not contain forcing subgraphs as induced subgraphs cheat the on-line algorithms in some restricted way only. More precisely, a graph $\mathbb{H}$ is forcing for coloring (for clique covering) if and only if there is a competitive on-line coloring (clique covering) algorithm for the family of $\mathbb{H}$-free graphs. In this thesis we are concerned with the competitiveness on the following graphs: $\mathbb{K}_s$-free graphs, $\mathbb{K}_{s,t}$-free graphs, $\mathbb{C}_4$-free graphs, $\mathbb{P}_5$-free graphs and on perfect and k-chordal graphs. We consider such graphs in attempt to make a classification of their competitive functions and also to find some forcing structures for on-line coloring and clique covering problem. The results are presented in Chapters 2-5. The last chapter, Chapter 6, contains some concluding remarks.

In Chapter 2 we consider perfect and k-chordal graphs. Note that, due to [2, 17], the families such as trees, bipartite graphs and chordal graphs can be arbitrarily cheated by the spoiler during the on-line coloring. We analyze the case when these graphs are presented in a connected way. This additional assumption allows bipartite graphs to be colored on-line optimally. It does not help with coloring of chordal graphs and, as a result, of perfect and k-chordal graphs. The $\mathcal{FF}$ on-line coloring algorithm is shown to be optimal for chordal graphs if the input graph is presented in a perfect elimination order (defined in Chapter 2) and the on-line $\mathcal{FF}$ clique covering algorithm becomes optimal for chordal graphs if the input chordal graph is presented in an inverse perfect elimination order (defined in Chapter 2). In this chapter we also present lower bounds for a competitive ratio of clique covering for bipartite and chordal graph (presented in a connected way) which are $\frac{3}{2}$ and 2, respectively. These numbers have been already proven to be the upper bounds (A.Gyárfás and J.Lehel [17]). On the other hand, we show that there is no competitive on-line clique covering algorithm for perfect and k-chordal graphs (presented in a connected way).

In Chapter 3 we prove that there is no competitive on-line coloring algorithm for $\mathbb{K}_s$-free graphs ($s \geqslant 3$) presented in a connected way. The same result follows for s-colorable graphs ($s \geqslant 3$) and planar graphs presented in a connected way. We also show that there are on-line clique covering algorithms for these families with a competitive ratios: $\frac{s}{2}$ for $\mathbb{K}_s$-free graphs and $\frac{s+1}{2}$ for s-colorable graphs. As a corollary we are able to put a competitive ratio of clique covering for planar graphs between 2 and $\frac{5}{2}$. The results implies that $\mathbb{K}_s$ is a forcing structure for clique covering.

Chapter 4 deals with $\mathbb{K}_{s,t}$-free graphs. A $\mathbb{K}_{1,t}$ graph is a star with t branches (or equivalently a $(t+1)$–element tree with t leaves). It is shown to be a forcing structure for on-line coloring. If $\mathbb{G}$ does not contain $\mathbb{K}_{1,t}$ as an induced subgraph, the spoiler is able to force at most $(t-1)\cdot\chi(\mathbb{G})-t+2$ colors (a weaker upper bound for this competitive ratio was announced by [7]). This function is also a lower bound (as shown in Chapter 4). This means that a competitive ratio for the best on-line coloring algorithm on $\mathbb{K}_{1,t}$-free graphs is $t-1$. On the other hand, $\mathbb{K}_{1,t}$ is not a forcing structure for the on-line clique covering problem and graphs $\mathbb{K}_{s,t}$ for $s \geqslant 2$ and $t \geqslant 3$ are forcing structures neither for coloring nor for clique covering. This means that there are graphs without these induced subgraphs, on which the spoiler can arbitrarily cheat. There are no on-line competitive algorithms for this family.

The remaining case of $\mathbb{K}_{2,2}$-free graphs is discussed in Chapter 4 as well. The family of $\mathbb{K}_{2,2}$-free graphs contains all trees, so that there is no competitive coloring algorithm for it. The problem of on-line clique covering for this family is still open. A.Gyárfás and J.Lehel [17] have shown an exponential upper bound $2^{opt(\mathbb{G})}-1$ for a competitive function for clique covering of $\mathbb{K}_{2,2}$-free graphs and no non-trivial lower bound was known. We show that a lower bound for a competitive function for this problem is at least quadratic: $\lfloor\frac{opt(\mathbb{G})\cdot(opt(\mathbb{G})+4)}{4}\rfloor$, therefore there is no on-line clique covering algorithm with a competitive ratio for $\mathbb{K}_{2,2}$-free graphs. Moreover, a competitive function for the best on-line greedy clique covering algorithm is at least $\binom{opt(\mathbb{G})+1}{2}$. There is still a huge gap between quadratic lower bound and exponential upper bound for a competitive function for this problem.

Chapter 5 analyzes the competitiveness of the problems for $\mathbb{P}_k$-free graphs. These graphs have been studied by A.Gyárfás, J.Lehel [17, 18] and H.A.Kierstead, S.G.Penrice, W.T.Trotter [22]. A.Gyárfás and J.Lehel have shown that the on-line $\mathcal{FF}$ coloring algorithm is optimal for $\mathbb{P}_4$-free graphs and that there is no on-line competitive coloring algorithm for $\mathbb{P}_6$-free graphs [18]. The remaining case of $\mathbb{P}_5$-free graphs is investigated in this chapter. Again, only an exponential upper bound was known [22]. We show the quadratic lower bound $\binom{opt(\mathbb{G})+1}{2}$ for this competitive function. To attack the remaining gap we analyze a special case of $\mathbb{P}_5$-free graphs: $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graphs and $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs. For these families the competitive functions for on-line coloring problem are quadratic, while the competitive ratios for on-line clique covering problem are 2. Additionally, we prove that there is a competitive ratio for on-line coloring and clique covering problems for $(\mathbb{C}_4, 2\mathbb{K}_2)$-free graphs and it is equal to 1.

The statements of our results are summarized in several tables. The tables are located at the beginning of each chapter. In each table the sign of infinity indicates the fact that there is no competitive function in the considered situation while $opt(\mathbb{G})$ denotes the optimum (off-line) solution in an appropriate case.

Chapter 2

Perfect and k-chordal graphs

2.1 Bipartite and chordal graphs

The on-line algorithms for bipartite and chordal graphs were analyzed by D.R.Bean [2] and A.Gyárfás and J.Lehel [17]. We start with some remarks that follow from their results. First, we repeat their proof showing that there is no on-line coloring algorithm with a competitive function for trees. We need their construction as we will often refer and/or modify it.

Theorem 2.1.1 (A.Gyárfás, J.Lehel [17]) *For every on-line coloring algorithm $\mathcal{A}$ and for every positive integer n there exists a tree $\mathbb{T}_n$ such that $\chi_{\mathcal{A}}(\mathbb{T}_n) \geqslant n$.*

Proof. For an arbitrary on-line coloring algorithm $\mathcal{A}$ we are looking for a tree $\mathbb{T}_n$ which forces $\mathcal{A}$ to use at least n colors. To construct $\mathbb{T}_n$ we induct on n. Let $\mathbb{T}_1$ be a single vertex. Assume that for $i = 1, \ldots, n-1$ there is a strategy for the spoiler which forces $\mathcal{A}$ to use at least i distinct colors on $\mathbb{T}_i$. In order to build $\mathbb{T}_n$ the spoiler first constructs the trees $\mathbb{T}_1, \ldots, \mathbb{T}_{n-1}$. Next, the spoiler chooses roots for $\mathbb{T}_1, \ldots, \mathbb{T}_{n-1}$ such that the number of root's colors is $n-1$. The tree $\mathbb{T}_n$ is formed by adding a new vertex x, joined to all the roots of $\mathbb{T}_1, \ldots, \mathbb{T}_{n-1}$. The algorithm $\mathcal{A}$ is forced to color it with a color different from the ones assigned to these roots. Thus $\chi_{\mathcal{A}}(\mathbb{T}_n) \geqslant n$. □

Obviously, $\chi(\mathbb{T}_n) \leqslant 2$. On the other hand, we are able to force any on-line coloring algorithm to use at least n colors. If there was an on-line coloring algorithm $\mathcal{A}$ with a competitive function $f : \mathbb{N} \to \mathbb{R}$ then

$$n \leqslant \chi_{\mathcal{A}}(\mathbb{T}_n) \leqslant f(\chi(\mathbb{T}_n)) \leqslant f(2)$$

for all n. It shows the following.

Corollary 2.1.2 *There is no on-line coloring algorithm with a competitive function for forests.* □

Graph $\mathbb{G}$	**Presentation method**	**Coloring**	**Cliques covering**
Tree	-	∞ [2, 17], 2.1.2	$\frac{3}{2} \cdot opt(\mathbb{G})$ 2.1.7 and [17]
Tree	connected	$opt(\mathbb{G})$ 2.2.1	$\frac{3}{2} \cdot opt(\mathbb{G})$ 2.2.4 and [17]
Bipartite	-	∞ [2, 17], 2.1.3	$\frac{3}{2} \cdot opt(\mathbb{G})$ 2.1.7 and [17]
Bipartite	connected	$opt(\mathbb{G})$ 2.2.1	$\frac{3}{2} \cdot opt(\mathbb{G})$ 2.2.4 and [17]
Chordal	-	∞ [2, 17], 2.1.4	$2 \cdot opt(\mathbb{G}) - 1$ [17], 2.1.8 and 2.1.9
Chordal	connected	∞ 2.2.3	$2 \cdot opt(\mathbb{G}) - 1$ 2.2.5 and [17]
Chordal	PEO order	∞ [2, 17], 2.3.1	$opt(\mathbb{G})$ 2.3.2
Chordal	IPEO order	$opt(\mathbb{G})$ 2.3.4	$2 \cdot opt(\mathbb{G}) - 1$ 2.3.3 and [17]
k-chordal for $k \geq 5$	connected	∞ 2.4.1	∞ 2.4.4 based on [17]
Perfect	connected	∞ 2.6.1	∞ 2.6.2 based on [17]

Table 2.1: The competitive functions for on-line problems

Since forests are cycle-free, it is important to keep in mind that cycles are not the forcing structures for the coloring problem. Forests are simultaneously bipartite and chordal, so that we immediately get the following two corollaries.

Corollary 2.1.3 *There is no on-line coloring algorithm with a competitive function for bipartite graphs.* □

Corollary 2.1.4 *There is no on-line coloring algorithm with a competitive function for chordal graphs.* □

Except coloring, A.Gyárfás and J.Lehel [17] considered on-line clique covering problem for chordal and bipartite graphs. In particular, they analyzed the covering algorithm using the First Fit strategy (the $\mathcal{FF}$ algorithm). Recall that this algorithm works by assigning the least possible positive integer as a clique to the current vertex presented by the spoiler.

Theorem 2.1.5 (A.Gyárfás, J.Lehel [17]) *The competitive function of the on-line clique covering algorithm using the First Fit strategy is exactly* $\frac{3}{2} \cdot v(\mathbb{G})$ *on bipartite graphs.* □

We show that, in fact, none of the on-line algorithms for bipartite graphs (or even trees) has better competitiveness.

Proposition 2.1.6 *There is no on-line clique covering algorithm* $\mathcal{A}$ *for bipartite graphs with a competitive ratio better than* $\frac{3}{2}$*. Actually, for every on-line clique covering algorithm* $\mathcal{A}$ *there exists an infinite family* $\mathcal{F}$ *of trees with* $v_{\mathcal{A}}(\mathbb{T}) \geqslant \frac{3}{2} \cdot v(\mathbb{T})$*, whenever* $\mathbb{T} \in \mathcal{F}$*.*

Proof. At the beginning of his construction, the spoiler presents a path with k vertices $x_1, \ldots, x_k$. The algorithm $\mathcal{A}$ covers this path with some number of cliques with at most 2 vertices in each clique. Next, the spoiler adds new vertices, forcing $\mathcal{A}$ to create new cliques. For each i the spoiler runs as follows (see Figure 2.1):

1. If $cl_{\mathcal{A}}(x_i) = cl_{\mathcal{A}}(x_{i+1})$, then the spoiler adds a new vertex y_i with an edge $y_i x_i$ and a new vertex y_{i+1} with an edge $y_{i+1} x_{i+1}$. The algorithm $\mathcal{A}$ is forced to assign them two new cliques.

2. If $cl_{\mathcal{A}}(x_{i-1}) \neq cl_{\mathcal{A}}(x_i)$ and $cl_{\mathcal{A}}(x_i) \neq cl_{\mathcal{A}}(x_{i+1})$ the spoiler adds a vertex y_i with an edge $y_i x_i$. There are two cases:

   (a) the new vertex is assigned to a new clique, i.e., $cl_{\mathcal{A}}(y_i) \neq cl_{\mathcal{A}}(x_i)$.

   (b) the new vertex is assigned to an existing clique, i.e., $cl_{\mathcal{A}}(y_i) = cl_{\mathcal{A}}(x_i)$. In this case the spoiler adds two additional vertices: z_i and u_i with edges $z_i x_i$, $u_i y_i$. They are assigned to new cliques.

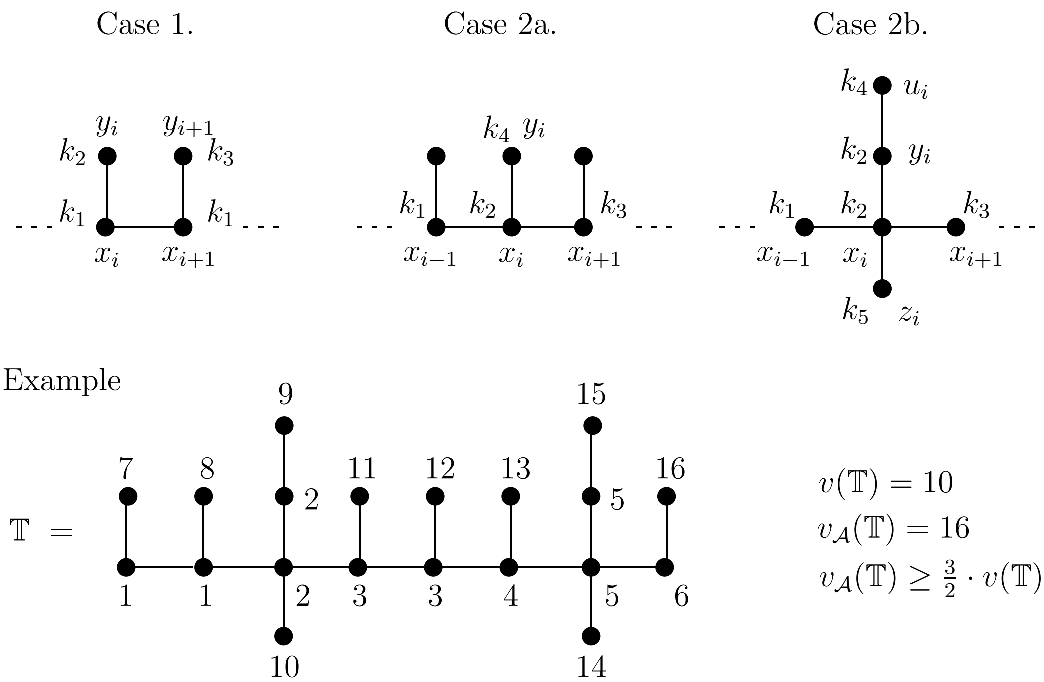


Figure 2.1: Construction of a forcing tree.

The spoiler proceeds along the path $x_1, \ldots, x_k$ adding one vertex (in case 1 and 2a) to each x_i and 3 vertices in case 2b. Define

$a =$ number of i's such that x_i was treated as described in case 2a,

$b =$ number of i's such that x_i was treated as described in case 2b,

$c =$ number of i's such that x_i was treated as described in case 1.

It is easy to see that in the unique optimal covering one needs:

- 1 clique (of the form $\{x_i, y_i\}$) for i in cases 1 and 2a,
- 2 cliques (of the form $\{u_i, y_i\}$ and $\{x_i, z_i\}$) for i in case 2b.

Therefore, $v(\mathbb{T}) = k + b = c + a + 2b$. On the other hand, $\mathcal{A}$ was forced to use $v_{\mathcal{A}}(\mathbb{T}) = \frac{3}{2}c + 2a + 3b$ cliques so that $v_{\mathcal{A}}(\mathbb{T}) \geqslant \frac{3}{2} \cdot v(\mathbb{T})$. □

Combining Theorem 2.1.5 with Proposition 2.1.6 we get

Corollary 2.1.7 *The best on-line clique covering algorithm for bipartite graphs (as well as for trees) is the $\mathcal{FF}$ algorithm with the competitive function $\frac{3}{2} \cdot v(\mathbb{G})$, where $\mathbb{G}$ is an input graph.* □

The next important result from [17] describes competitiveness of the on-line $\mathcal{FF}$ clique covering algorithm for chordal graphs. Recall that the $\mathcal{FF}$ algorithm is greedy. Moreover, from all cliques (positive integers) which can be used to cover the current vertex presented by the spoiler, $\mathcal{FF}$ chooses the least one.

Theorem 2.1.8 (A.Gyárfás, J.Lehel [17]) *The on-line clique covering algorithm using the First Fit strategy for chordal graphs has a competitive ratio at most* 2. *Actually,*

$$v_{\mathcal{FF}}(\mathbb{G}) \leqslant 2 \cdot v(\mathbb{G}) - 1$$

for every chordal graph $\mathbb{G}$. □

We improve this result by showing that, in fact, the $\mathcal{FF}$ algorithm is the best on-line clique covering algorithm for chordal graphs.

Theorem 2.1.9 *For every on-line clique covering algorithm* $\mathcal{A}$ *and for every positive integer* v *there exists a chordal graph* $\mathbb{G}$ *with* $v(\mathbb{G}) = v$ *such that*

$$v_{\mathcal{A}}(\mathbb{G}) \geqslant 2 \cdot v(\mathbb{G}) - 1.$$

Proof. We describe a way the spoiler builds a chordal graph $\mathbb{G}_v$ with $v(\mathbb{G}_v) = v$ forcing the algorithm $\mathcal{A}$ to use $2 \cdot v - 1$ cliques. To construct $\mathbb{G}_v$ we induct on v. Let $\mathbb{G}_1$ be a single vertex. Assume that for $i = 1, \ldots, v-1$ there is a strategy for the spoiler to force $\mathcal{A}$ to use at least $2 \cdot i - 1$ cliques to cover $\mathbb{G}_i$. In order to build $\mathbb{G}_v$ the spoiler presents two joined vertices x_1 and x_2. There are two cases:

1. $\mathcal{A}$ covers both vertices x_1, x_2 with the same clique.
   - If v is even, the spoiler builds two copies $\mathbb{A}_1$, $\mathbb{A}_2$ of $\mathbb{G}_{\frac{v}{2}}$ such that each vertex of $\mathbb{A}_i$ is joined to x_i (see Figure 2.2). The algorithm $\mathcal{A}$ covers these two copies of $\mathbb{G}_{\frac{v}{2}}$ with at least $2 \cdot (2 \cdot \frac{v}{2} - 1) = 2v - 2$ cliques, which, together with the clique covering x_1 and x_2, gives $2 \cdot v - 1$ cliques used.
   - If v is odd, the spoiler builds graphs $\mathbb{A}_1 = \mathbb{G}_{\frac{v+1}{2}}$ and $\mathbb{A}_2 = \mathbb{G}_{\frac{v-1}{2}}$ such that each vertex of $\mathbb{A}_i$ is joined to x_i (see Figure 2.2). The algorithm $\mathcal{A}$ covers these graphs with at least $2 \cdot \frac{v+1}{2} - 1 + 2 \cdot \frac{v-1}{2} - 1 = 2v - 2$ cliques, which, together with the clique covering x_1 and x_2, gives $2 \cdot v - 1$ cliques used.

   The resulting graph $\mathbb{G}_v$ is covered by $\mathcal{A}$ with at least $2 \cdot v - 1$ cliques. However, it is easy to show that it can be covered with v cliques: Obviously, each of the $\mathbb{A}_i$'s can be covered with

$$v(\mathbb{A}_1) = \begin{cases} \frac{v}{2}, & \text{if } v \text{ is even} \\ \frac{v+1}{2}, & \text{if } v \text{ is odd} \end{cases}$$

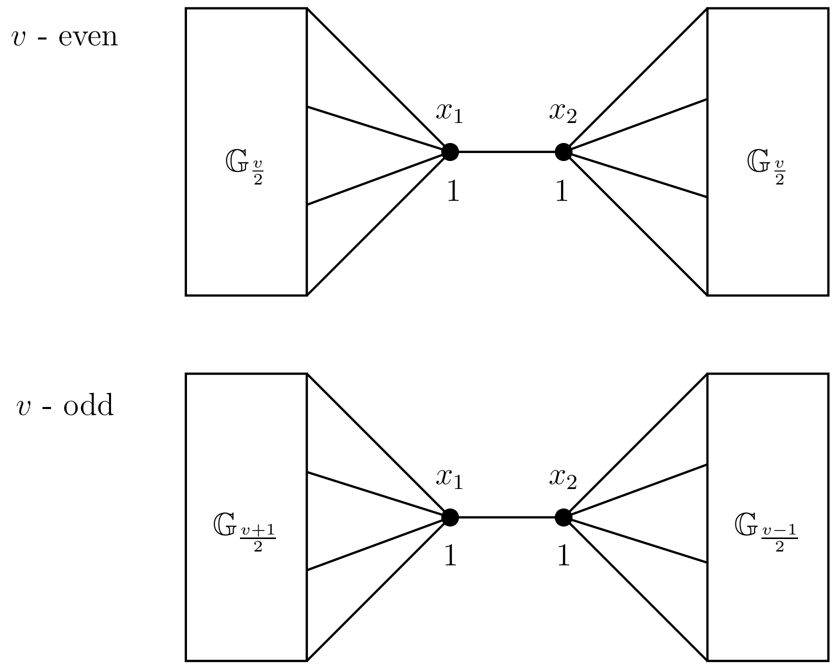


Figure 2.2: Construction of $\mathbb{G}_v$: case 1.

$$v(\mathbb{A}_2) = \begin{cases} \frac{v}{2}, & \text{if } v \text{ is even} \\ \frac{v-1}{2}, & \text{if } v \text{ is odd} \end{cases}$$

cliques. Since the vertex x_i is joined to all vertices of $\mathbb{A}_i$, it can be added to one of cliques of $\mathbb{A}_i$. Therefore $v(\mathbb{G}_v) = v(\mathbb{A}_1) + v(\mathbb{A}_2) = v$.

2. The vertices x_1, x_2 are covered with two different cliques.

   Then the spoiler presents a vertex x_3 joined to both x_1 and x_2. There are two cases possible (see Figure 2.3):

   2a. $cl_{\mathcal{A}}(x_3) \neq cl_{\mathcal{A}}(x_1)$ and $cl_{\mathcal{A}}(x_3) \neq cl_{\mathcal{A}}(x_2)$.
   Then the spoiler builds a graph $\mathbb{G}_{v-1}$ such that x_3 is the first vertex of it. The algorithm $\mathcal{A}$ covers $\mathbb{G}_{v-1}$ with $2 \cdot (v-1) - 1 = 2v - 3$ cliques, which, together with $\{x_1\}$ and $\{x_2\}$, gives $2 \cdot v - 1$ cliques used. However, the resulting graph can be covered with v cliques: $v - 1$ for $\mathbb{G}_{v-1}$ and one for $\{x_1, x_2\}$.

   2b. $cl_{\mathcal{A}}(x_3) = cl_{\mathcal{A}}(x_1)$ or $cl_{\mathcal{A}}(x_3) = cl_{\mathcal{A}}(x_2)$.
   By symmetry we consider only the case $cl_{\mathcal{A}}(x_3) = cl_{\mathcal{A}}(x_1)$.
   In this case, the spoiler presents a vertex x_4 joined to x_2 and x_3. If it is covered by a new clique, the spoiler behaves similarly to the case

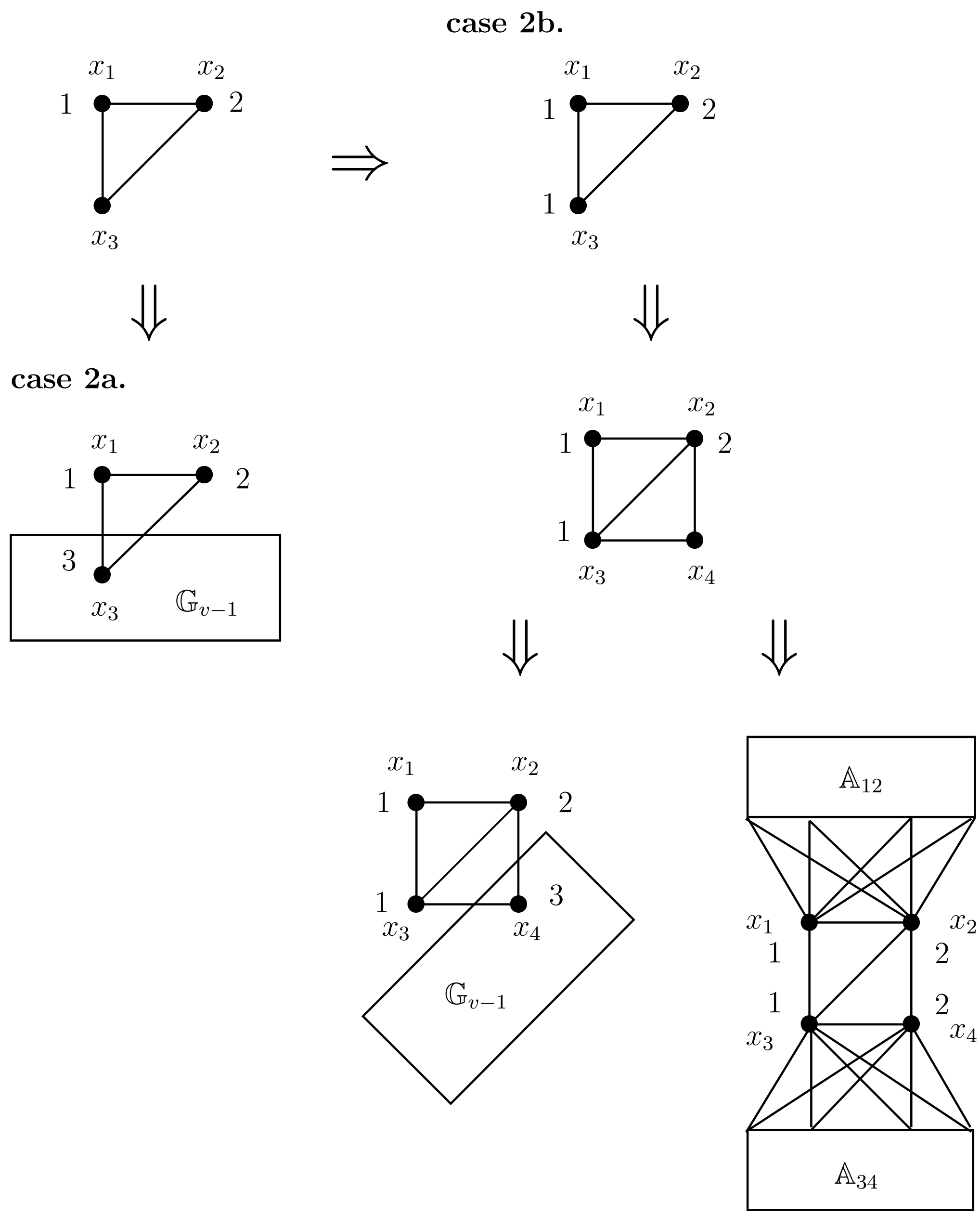


Figure 2.3: Construction of $\mathbb{G}_v$: case 2.

2a; building a graph $\mathbb{G}_{v-1}$ starting with x_4. If x_4 is covered with the clique $cl_{\mathcal{A}}(x_2)$, the spoiler builds two graphs in a way modelled after case 1: two graphs $\mathbb{A}_{12}$, $\mathbb{A}_{34}$, each of the form $\mathbb{G}_t$:

$$\mathbb{A}_{12} = \begin{cases} \mathbb{G}_{\frac{v}{2}}, & \text{if } v \text{ is even} \\ \mathbb{G}_{\frac{v+1}{2}}, & \text{if } v \text{ is odd} \end{cases}$$

$$\mathbb{A}_{34} = \begin{cases} \mathbb{G}_{\frac{v}{2}}, & \text{if } v \text{ is even} \\ \mathbb{G}_{\frac{v-1}{2}}, & \text{if } v \text{ is odd} \end{cases}$$

are built such that each vertex of $\mathbb{A}_{ij}$ is joined to both x_i and x_j. As in case 1 the algorithm $\mathcal{A}$ is forced to use the cliques $\{x_1, x_3\}$, $\{x_2, x_4\}$ and $2v - 2$ new cliques on $\mathbb{A}_{12} \cup \mathbb{A}_{34}$, so that $v_{\mathcal{A}}(\mathbb{G}_v) = 2v$. On the other hand, $\mathbb{A}_{12} \cup \mathbb{A}_{34}$ can be covered off-line with v cliques and $\{x_i, x_j\}$ can be easily added to one of the cliques used for $\mathbb{A}_{ij}$.

Inducting on v one can easily see that each of the $\mathbb{G}_v$'s is chordal. □

Combining Theorems 2.1.8 and 2.1.9 we get

Corollary 2.1.10 *The best on-line clique covering algorithm for chordal graphs is the $\mathcal{FF}$ algorithm with the competitive function $2 \cdot v(\mathbb{G}) - 1$, where $\mathbb{G}$ is an input chordal graph.* □

Summarizing our results we see that cycles are the forcing structures for clique covering problem while they are not forcing structures for coloring.

2.2 Bipartite and chordal graphs presented in a connected way

In section 2.1 we showed that every on-line coloring algorithm for bipartite and chordal graphs can be forced to use arbitrarily many colors. Now, we add an assumption that the graphs are presented in a connected way. This means that an input graph is connected at each step of the construction. More precisely, the graph $\mathbb{G} = (\{x_1, \ldots, x_k\}, E)$ is presented in a connected way, whenever the vertices are presented in the order $x_1, \ldots, x_k$ and for every $i = 1, \ldots, k$ the subgraph induced by $\{x_1, \ldots, x_i\}$ is connected. In this section we observe that in this "connected" setting, the algorithm using the First Fit strategy colors optimally the class of bipartite graphs. This is a big contrast with a nonexistence of any competitive algorithm in general (not necessary "connected") setting. Later on we show that this additional restriction for graph presentation does not help at all in on-line coloring and clique covering of chordal graphs.

Proposition 2.2.1 *If a bipartite graph $\mathbb{G}$ is presented in a connected way, then the on-line coloring algorithm using the First Fit strategy colors it optimally, that is*

$$\chi_{\mathcal{FF}}(\mathbb{G}) = \chi(\mathbb{G}).$$

Proof. Let $\mathbb{G} = (V_1 \cup V_2, E)$ be a bipartite graph presented in a connected way in the order $x_1, x_2, \ldots, x_k$. Since there is an edge between the first two vertices x_1 and x_2, they have to be in different sets V_1, V_2. Without loss of generality we may assume that $x_1 \in V_1$ and $x_2 \in V_2$. The $\mathcal{FF}$ algorithm colors x_1 with color 1 and x_2 with color 2.

We prove by induction that the $\mathcal{FF}$ algorithm assigns color 1 to every vertex in V_1 and the color 2 to every vertex from V_2. This means that $\mathcal{FF}$ colors $\mathbb{G}$ optimally. Suppose that this is true for every vertex v_i where $i < p$ and that the vertex x_p is on input. Surely, there exists $j \in \{1, \ldots, p-1\}$ such that $x_p E x_j$. If $c_{\mathcal{FF}}(x_j) = 1$ then $x_j \in V_1$, thus $x_p \in V_2$. The $\mathcal{FF}$ algorithm assigns to x_p color 2. Otherwise, $x_p \in V_1$ and $\mathcal{FF}$ assigns to it color 1. □

In section 2.1 we proved that there is no competitive coloring algorithm for chordal graphs. This was established by considering trees (Corollary 2.1.4). These trees had not been presented in a connected way. Actually, by Proposition 2.2.1, trees (as well as all bipartite graphs) are too simple to force additional colors in a "connected" setting. Two next theorems show that in fact chordal graphs are much more complicated even if presented in a connected way.

Theorem 2.2.2 *The on-line coloring algorithm using the First Fit strategy for chordal graphs presented in a connected way does not have a competitive function.*

Proof. Let $n \geqslant 3$. We describe a way the spoiler constructs a 3-colorable chordal graph $\mathbb{G}_n$, colored by the $\mathcal{FF}$ algorithm with at least n colors. To construct $\mathbb{G}_n$ we induct on n. In order to build $\mathbb{G}_3$, the spoiler presents vertices x_1, x_2, x_3, x_4 with edges x_2x_1, x_3x_2, x_4x_2, x_4x_3 as pictured on Figure 2.4. The $\mathcal{FF}$ algorithm colors it with 3 colors.

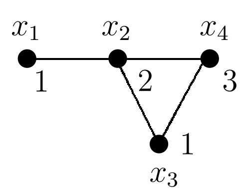


Figure 2.4: The graph $\mathbb{G}_3$.

Suppose that for $i = 1, \ldots, n-1$ there is a strategy for the spoiler which forces $\mathcal{FF}$ to use i colors on $\mathbb{G}_i$. Moreover, suppose that every graph $\mathbb{G}_i$ has a special induced path called its proper path. We say that a path induced by $\{x_1, x_2, \ldots, x_i\}$ is proper in $\mathbb{G}_i$, if $c_{\mathcal{FF}}(x_1) < \ldots < c_{\mathcal{FF}}(x_i)$. The proper path in $\mathbb{G}_3$ consists of vertices x_1, x_2, x_4.

The construction of $\mathbb{G}_n$ is pictured on Figure 2.5. First, the spoiler builds $\mathbb{G}_{n-1}$ with its proper path induced by $\{x_1, \ldots, x_{n-1}\}$. Next, he adds a new vertex y_1 with an edge y_1x_{n-1}. The $\mathcal{FF}$ algorithm colors it with the first color, namely

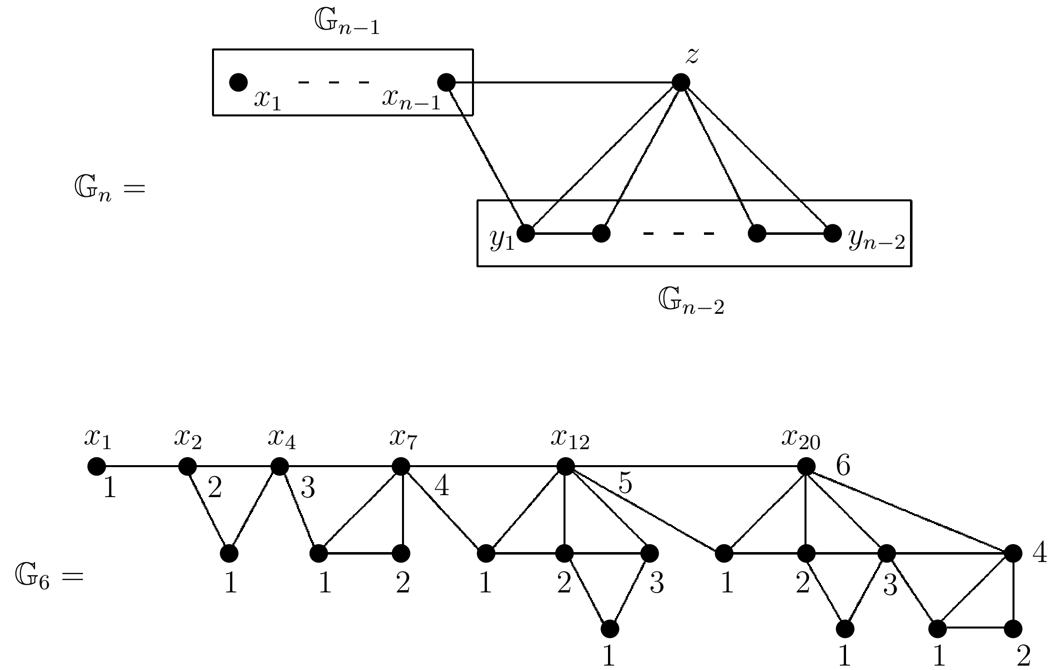


Figure 2.5: Construction of $\mathbb{G}_n$.

$c_{\mathcal{FF}}(x_1)$. Then the spoiler starts with the vertex y_1 and constructs $\mathbb{G}_{n-2}$ with the path induced by $\{y_1, \ldots, y_{n-2}\}$ as its proper path. At the end, he adds a vertex z with edges zx_{n-1} and zy_i for every $i = 1, \ldots, n-2$. Since vertices $y_1, \ldots, y_{n-2}, x_{n-1}$ have different colors, $\mathcal{FF}$ is forced to color z with a new color. The path induced by $\{x_1, \ldots, x_{n-1}, z\}$ is proper in the graph $\mathbb{G}_n$. In particular, $\chi_{\mathcal{FF}}(\mathbb{G}_n) \geqslant n$.

Clearly $\mathbb{G}_n$ is chordal and the maximal clique in it consists of 3 vertices. Since chordal graphs are perfect $\chi(\mathbb{G}_n) = 3$. □

The technique presented in the proof of Theorem 2.2.2 works for all greedy algorithms. We generalize it to make it work against every possible algorithm.

Theorem 2.2.3 *None of the on-line coloring algorithms for chordal graphs presented in a connected way has a competitive function.*

Proof. Let $\mathcal{A}$ be an on-line coloring algorithm. We claim that for every n there exists a 3-colorable chordal graph $\mathbb{G}_n$ presented in a connected way such that

$$\chi_{\mathcal{A}}(\mathbb{G}_n) \geqslant n.$$

We prove this by generalizing the technique used in the proof of Theorem 2.2.2. First, we change a bit the definition of a *proper path*. The path induced

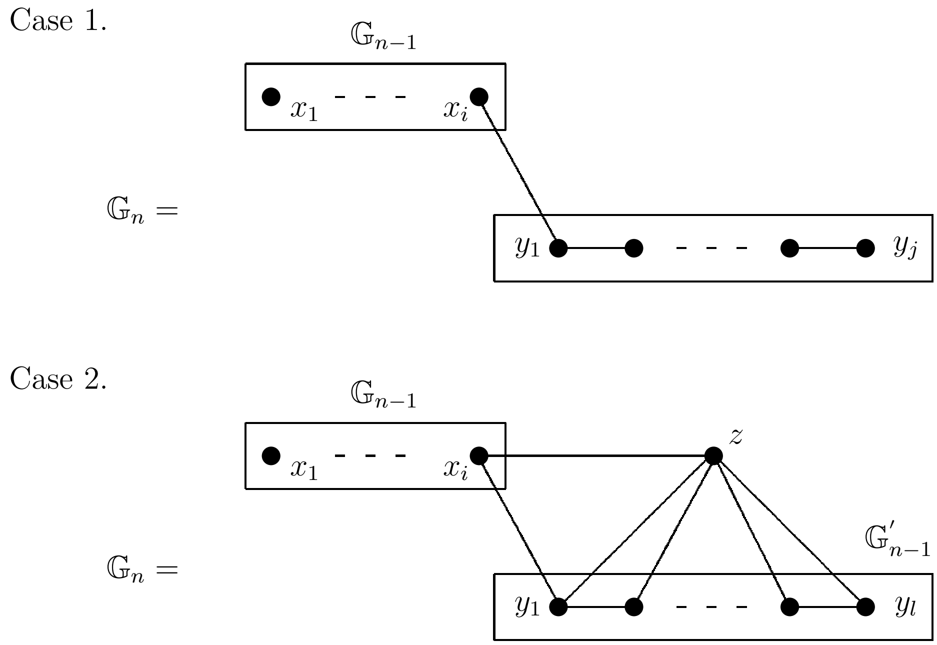


Figure 2.6: Construction of $\mathbb{G}_n$

by $\{x_1, \ldots, x_i\}$ is proper in $\mathbb{G}_n$, if x_1 is the first presented vertex, x_i is the last presented vertex and $|C_{\mathcal{A}}(\{x_1, \ldots, x_i\})| = n$.

To construct $\mathbb{G}_n$ we induct on n. The graph $\mathbb{G}_3$ consists of vertices x_1, x_2, x_3, x_4 with edges x_2x_1, x_3x_2, x_4x_2, x_4x_3. The algorithm $\mathcal{A}$ colors them with at least 3 colors, a proper path of $\mathbb{G}_3$ is induced either by $\{x_1, x_2, x_4\}$ or by $\{x_1, x_2, x_3\}$. Suppose that for $i = 1, \ldots, n-1$ there is a strategy for the spoiler which forces $\mathcal{A}$ to use i colors on $\mathbb{G}_i$. Moreover, suppose that every $\mathbb{G}_i$ has a proper path.

The spoiler builds the graph $\mathbb{G}_n$ as follows. First, he builds $\mathbb{G}_{n-1}$ with its proper path induced by $\{x_1, \ldots, x_i\}$. Then, he adds a new vertex y_1 with an edge y_1x_i. If $c_{\mathcal{A}}(y_1) \notin C_{\mathcal{A}}(\mathbb{G}_{n-1})$, then $\mathcal{A}$ has already used n colors. The construction is done. The path induced by $\{x_1, \ldots, x_i, y_1\}$ is proper in $\mathbb{G}_n$. Otherwise, i.e., if $c_{\mathcal{A}}(y_1) \in C_{\mathcal{A}}(\mathbb{G}_{n-1})$, then the spoiler starts with the vertex y_1 and constructs a copy of the graph $\mathbb{G}_{n-1}$, called $\mathbb{G}'_{n-1}$. The algorithm $\mathcal{A}$ colors it with at least $n-1$ colors. Let $y_1, \ldots, y_l$ be a proper path of $\mathbb{G}'_{n-1}$. There are two cases:

1. During the construction of $\mathbb{G}'_{n-1}$, $\mathcal{A}$ assigns a new color (i.e., a color not belonging to $C_{\mathcal{A}}(\mathbb{G}_{n-1})$) to some vertex y_j. Then this finishes the construction. The spoiler has forced $\mathcal{A}$ to use n colors. The path induced by $\{x_1, \ldots, x_i, y_1, \ldots, y_j\}$ is proper in $\mathbb{G}_n$.

2. $C_{\mathcal{A}}(\{y_1, \ldots, y_l\}) \subseteq C_{\mathcal{A}}(\mathbb{G}_{n-1})$. The spoiler finishes the construction by adding a vertex z with edges zx_i and zy_j for every $j = 1, \ldots, l$. Since

$|C_{\mathcal{A}}(\{y_1, \ldots, y_l\})| = n - 1$, $\mathcal{A}$ is forced to color z with a new color. The path induced by $\{x_1, \ldots, x_i, z\}$ is proper in $\mathbb{G}_n$.

In both cases, $\mathbb{G}_n$ is chordal and $\omega(\mathbb{G}_n) = 3$. Since chordal graphs are perfect, $\mathbb{G}$ is 3-colorable while $\chi_{\mathcal{A}}(\mathbb{G}_n) = n$, as claimed. □

A connected way of presentation of a graph improves on-line coloring of bipartite graphs, but it does not help at all in coloring of chordal graphs. We proceed to check if the assumption that graph is presented in a connected way helps in on-line clique covering of bipartite or chordal graphs and observe that the answer is negative. Indeed, from Theorem 2.1.5 we know that if $\mathbb{G}$ is bipartite, then the on-line $\mathcal{FF}$ clique covering algorithm has the competitive function $\frac{3}{2} \cdot v(\mathbb{G})$. On the other hand, in the proof of Proposition 2.1.6 the spoiler constructs trees $\mathbb{T}$ for which $v_{\mathcal{A}}(\mathbb{T}) \geqslant \frac{3}{2} \cdot v(\mathbb{T})$, where $\mathcal{A}$ is an arbitrary on-line clique covering algorithm. Since these trees were presented in a connected way, we immediately get:

Corollary 2.2.4 *The best on-line clique covering algorithm for bipartite graphs (as well as for trees) presented in a connected way is the $\mathcal{FF}$ algorithm with the competitive function $\frac{3}{2} \cdot v(\mathbb{G})$, where $\mathbb{G}$ is an input graph.* □

A similar conclusion holds for chordal graphs. Corollary 2.1.10 shows that the on-line $\mathcal{FF}$ clique covering algorithm for chordal graphs has the competitive function $2 \cdot v(\mathbb{G}) - 1$. Moreover, all the graphs constructed in the proof of Theorem 2.1.9 were presented in a connected way. Those graphs make $2 \cdot v(\mathbb{G}) - 1$ to be the lower bound for clique covering. Thus we have

Corollary 2.2.5 *The best on-line clique covering algorithm for chordal graphs presented in a connected way is the $\mathcal{FF}$ algorithm with the competitive function $2 \cdot v(\mathbb{G}) - 1$, where $\mathbb{G}$ is an input graph.* □

2.3 Chordal graph presented in PEO and IPEO order

Chordal graphs are also called perfect elimination graphs. This comes from the fact that a graph is chordal if it has a *perfect elimination order* (PEO for short) [23]. A vertex ordering $x_1, x_2, \ldots, x_n$ of a graph is PEO if for each vertex x_i ($i < n$), its neighbors in the set $\{x_{i+1}, \ldots, x_n\}$ induce a clique. We consider a question whether a presentation of chordal graphs in a perfect elimination order helps in the on-line coloring or in the on-line clique covering. Note that a chordal graph can admit more than one PEOs. For example, any linear order of a clique is PEO. The importance of PEO for chordal graphs comes from the linear-time algorithm that checks whether an input graph $\mathbb{G}$ is chordal. If $\mathbb{G}$ is chordal this algorithm (see, e.g., [24]) returns the positive answer together with one of the PEO admitted by $\mathbb{G}$.

Theorem 2.1.1 proves that the spoiler can force an arbitrary on-line coloring algorithm for trees to use arbitrarily many colors. For an on-line coloring algorithm $\mathcal{A}$ the proof of Theorem 2.1.1 shows the construction of trees $\mathbb{T}_n$ which force $\mathcal{A}$ to use n colors. Since the vertices of $\mathbb{T}_n$ are presented in PEO, we obtain the following.

Corollary 2.3.1 *None of the on-line coloring algorithm for chordal graphs presented in a perfect elimination order has a competitive function.* □

Although the perfect elimination order does not help in the on-line coloring for chordal graphs, it does help in clique covering.

Theorem 2.3.2 *If vertices of a chordal graph $\mathbb{G}$ are presented in a perfect elimination order, then the on-line clique covering algorithm using the First Fit strategy covers it optimally, that is,*

$$v_{\mathcal{FF}}(\mathbb{G}) = v(\mathbb{G}).$$

Proof. First, we fix some notation:

- $x_1, \ldots, x_n$ = perfect elimination order of vertices of the chordal graph $\mathbb{G}$.
- $\mathbb{A}_1, \ldots, \mathbb{A}_v$ = optimal clique covering of $\mathbb{G}$ with $v = v(\mathbb{G})$. Without loss of generality we may assume that the numbering of cliques is consistent with perfect elimination order, i.e., for all $1 \leqslant i < j \leqslant v$ there exists $x_k \in A_i$ such that if $x_l \in A_j$ then $l > k$.
- $N^+_{\mathbb{G}}(x_i) = \{\ x_j :\ j > i \text{ and } x_i E_{\mathbb{G}} x_j\}$. The perfect elimination order ensures us that the set $N^+_{\mathbb{G}}(x_i)$ forms a clique in $\mathbb{G}$.

It is convenient to assign colors to cliques created by $\mathcal{FF}$ and talk about colors instead of cliques. In other words, we analyze a special on-line coloring of $\mathbb{G}$, with the special requirement that the set of vertices with the same color forms a clique.

We say that a color c (a clique with color c) is *initialized* in a clique $\mathbb{A} \in \{\mathbb{A}_1, \ldots, \mathbb{A}_v\}$ if there exists $x_i \in A$ such that

$$cl_{\mathcal{FF}}(x_i) = c \quad \text{and} \quad cl_{\mathcal{FF}}(x_j) \neq c \quad \text{for every } j < i.$$

Claim In each clique $\mathbb{A}_i$ (for $i = 1, \ldots, v$) at most one color is initialized.

Before proving this claim we note that Theorem 2.3.2 is straightforward consequence of it. Indeed, suppose our claim holds. Since each color used by $\mathcal{FF}$ has been initialized in some clique $\mathbb{A}_i$, $v_{\mathcal{FF}}(\mathbb{G}) \leqslant v(\mathbb{G})$, as required.

Proof of the claim. Suppose that there exists a clique $\mathbb{A}_p$ in which two colors have been initialized, i.e., $\exists c_1 \neq c_2$ such that

$$\exists x_t \in A_p :\ cl_{\mathcal{FF}}(x_t) = c_1 \quad \text{and} \quad cl_{\mathcal{FF}}(x_i) \neq c_1 \quad \text{for every } i < t$$

$$\exists x_s \in A_p : \ cl_{\mathcal{FF}}(x_s) = c_2 \ \text{ and } \quad cl_{\mathcal{FF}}(x_i) \neq c_2 \quad \text{ for every } i < s$$

Without loss of generality we may assume that $t < s$. Note that this gives $c_1 < c_2$. Let $B = \{x : \ cl_{\mathcal{FF}}(x) = c_1\}$. Obviously $B \subseteq N^+_{\mathbb{G}}(x_t)$. Since both x_t and x_s are in the same clique $\mathbb{A}_p$, $x_s \in N^+_{\mathbb{G}}(x_t)$.

Consider the moment x_s was presented by the spoiler and colored (with c_2) by $\mathcal{FF}$. At this moment c_2 was used for the first time by $\mathcal{FF}$. The necessity for using the new color means (for $\mathcal{FF}$) that x_s is not connected to at least one vertex in the set B. This means that neither $B \cup \{x_s\}$ nor the the bigger set $N^+_{\mathbb{G}}(x_t)$ forms a clique. This contradicts the fact that in perfect elimination order each set of the form $N^+_{\mathbb{G}}(x_i)$ is a clique. □

By an *inverse perfect elimination order* ($IPEO$ for short) we mean an order reverse of PEO, i.e., an order $x_1, x_2, \ldots, x_n$ of vertices such that for each vertex x_i ($i > 1$), its neighbors in the set $\{x_1, x_2, \ldots, x_{i-1}\}$ induce a clique. Consequently, if a graph is presented on-line in $IPEO$ then a new vertex, added to $\mathbb{G}$, is joined only to vertices forming a clique. If vertices of a graph are presented in $IPEO$ then the results we have for an on-line coloring and clique covering are, in a sense, dual to the ones we have for PEO presentation.

Proposition 2.3.3 *There is no on-line clique covering algorithm for chordal graphs presented in an inverse elimination order with a competitive ratio better than* 2*. Moreover, for every on-line clique covering algorithm $\mathcal{A}$ and for every positive integer v there exists a chordal graph $\mathbb{G}$ with $v(\mathbb{G}) = v$ presented in an inverse perfect elimination order and such that*

$$v_{\mathcal{A}}(\mathbb{G}) \geqslant 2 \cdot v(\mathbb{G}) - 1.$$

Proof. The proof of Theorem 2.1.9 presents a strategy of the spoiler to construct chordal graphs $\mathbb{G}$ with $v_{\mathcal{A}}(\mathbb{G}) \geqslant 2 \cdot v(\mathbb{G}) - 1$. It is easy to verify that these graphs were presented in $IPEO$. □

Proposition 2.3.3, together with Theorem 2.1.8, proves that the inverse perfect elimination order does not help at all in an on-line clique covering of chordal graphs. On the other hand, $IPEO$ makes it possible to get a competitive algorithm for an on-line coloring, actually, an optimal algorithm. This stays in a huge contrast with the general setting (without $IPEO$ assumed) in which no such algorithm exists.

Proposition 2.3.4 *If a chordal graph $\mathbb{G}$ is presented in an inverse elimination order then the on-line coloring algorithm using the First Fit strategy colors it optimally, that is,*

$$\chi_{\mathcal{FF}}(\mathbb{G}) = \chi(\mathbb{G}).$$

Proof. Let $\mathbb{G}_n$ denote the subgraph of $\mathbb{G}$ induced by first n vertices. We induct on n to show that $\chi_{\mathcal{FF}}(\mathbb{G}_n) = \chi(\mathbb{G}_n)$. Chordal graphs are perfect, therefore $\chi(\mathbb{G}_n) = \omega(\mathbb{G}_n)$. Let x_{n+1} be on input. The $\mathcal{FF}$ algorithm colors it with k. There are two cases:

a. $k \leqslant \omega(\mathbb{G}_n)$
Then the number of colors used on $\mathbb{G}_{n+1}$ does not increase and we have $\chi_{\mathcal{FF}}(\mathbb{G}_{n+1}) = \chi(\mathbb{G}_{n+1})$.

b. $k = \omega(\mathbb{G}_n) + 1$
$IPEO$ guarantees that the set $N^-_{\mathbb{G}_{n+1}}(x_{n+1}) = \{x_j : j < n+1 \text{ and } x_j E_{\mathbb{G}} x_{n+1}\}$ induces a clique. Since x_{n+1} got a new color (not appearing in $C_{\mathcal{FF}}(\mathbb{G}_n)$), we know that x_{n+1} has $\omega(\mathbb{G}_n)$ neighbors, i.e., the clique $N^-_{\mathbb{G}_{n+1}}(x_{n+1}) \cup \{x_{n+1}\}$ has size $\omega(\mathbb{G}_n) + 1 = k$. This gives that $\chi_{\mathcal{FF}}(\mathbb{G}_{n+1}) = k = \omega(\mathbb{G}_n) + 1 = \omega(\mathbb{G}_{n+1}) = \chi(\mathbb{G}_{n+1})$. □

2.4 k-chordal graphs

Let $k \geqslant 4$. We say that a graph is k-chordal if each cycle of length at least k has at least one chord, that is, an edge joining two nonconsecutive vertices of the cycle. According to this definition chordal graphs are 4-chordal. Note that immediately from Theorem 2.2.3 we get the following.

Corollary 2.4.1 *Let $k \geqslant 4$. There is no on-line coloring algorithm with a competitive function for k-chordal graphs presented in a connected way.*

Next, we will see that relaxing 4-chordality slightly, i.e., to 5-chordality destroys all the competitive on-line clique covering algorithms. We start with an easy lemma.

Lemma 2.4.2

1. $\overline{\mathbb{C}}_5 = \mathbb{C}_5$.

2. *If $k > 5$ then $\mathbb{C}_4$ is an induced subgraph of $\overline{\mathbb{C}}_k$.*

Proof.

1. It is easy to see on Figure 2.7.

2. Let $k > 5$ and $\mathbb{C}_k$ be a cycle $x_1, x_2, \ldots, x_k, x_1$. Then vertices x_1, x_2, x_4, x_5 induce a cycle in $\overline{\mathbb{C}}_k$ (see Figure 2.7). □

Theorem 2.4.3 *There is no on-line clique covering algorithm with a competitive function for 5-chordal graphs presented in a connected way.*

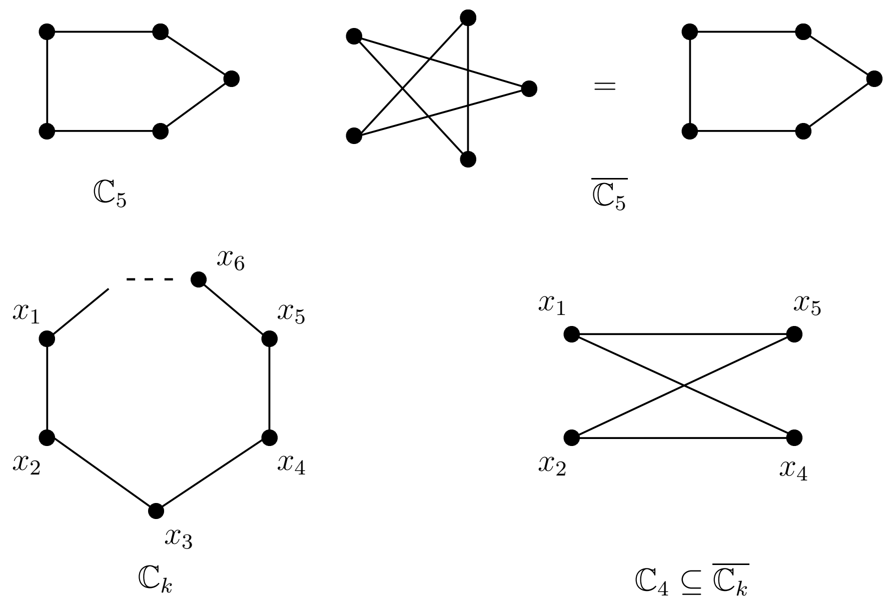


Figure 2.7: $\mathbb{C}_k$ and its complement of $\overline{\mathbb{C}}_k$

Proof. First, note that clique covering of a graph $\mathbb{G}$ is the same as coloring its complement $\overline{\mathbb{G}}$. The same holds for on-line version of these problems as long as the vertices of $\mathbb{G}$ and $\overline{\mathbb{G}}$ are presented in the very same order. Thus, in view of the proof of Theorem 2.1.1, it suffices to show that the complement $\overline{\mathbb{T}}_n$ of the tree $\mathbb{T}_n$ that had been constructed by the spoiler is

- 5-chordal
- and presented in a connected way.

The second claim can be easily satisfied by inspecting the construction of $\mathbb{T}_{n+1}$ from $\mathbb{T}_1, \ldots, \mathbb{T}_n$, as described in the proof of Theorem 2.1.1. To see the first claim, i.e., that $\overline{\mathbb{T}}_n$ is 5-chordal, we refer to Lemma 2.4.2. Indeed, if $\mathbb{G}$ (i.e., one of the $\overline{\mathbb{T}}_n$'s) contains an induced cycle $\mathbb{C}_k$ (with $k \geqslant 5$) then $\overline{\mathbb{G}}$ contains $\mathbb{C}_4$ (if $k > 5$) or $\mathbb{C}_5$ (if $k = 5$). This would give that the tree $\mathbb{T}_n$ contains a cycle. □

Since 5-chordal graphs are k-chordal for $k \geqslant 5$, we immediately get:

Corollary 2.4.4 *Let $k \geqslant 5$. There is no on-line clique covering algorithm with a competitive function for k-chordal graphs presented in a connected way.* □

2.5 $\mathbb{C}_k$-free graphs

Now we restrict ourselves to classes forbidding cycles of a given size. Our results in this setting are summarized in Table 2.2. Since chordal graphs are $\mathbb{C}_{k\geqslant 4}$-free, in view of Theorem 2.2.3 we have

Graph $\mathbb{G}$	Presentation method	Coloring	Cliques covering
$\mathbb{C}_3$-free $= \mathbb{K}_3$-free	connected	∞ 3.1.1 based on [17]	$\frac{3}{2}$ 3.2.9
$\mathbb{K}_{2,2}$-free $\mathbb{C}_4$-free	connected	∞ 2.5.1	$\lfloor \frac{opt(\mathbb{G})\cdot(opt(\mathbb{G})+4)}{4} \rfloor \leq \ldots \leq 2^{opt(\mathbb{G})} - 1$ 4.3.6 and [17]
			for greedy algorithms $\binom{opt(\mathbb{G})+1}{2} \leq \ldots \leq 2^{opt(\mathbb{G})} - 1$ 4.3.4 and [17]
$\mathbb{C}_k$-free for $k \geq 5$	connected	∞ 2.5.1	∞ 2.5.2 based on [17]

Table 2.2: Possible competitive functions for on-line problems

Corollary 2.5.1 *Let $k \geqslant 4$. There is no on-line coloring algorithm with a competitive function for $\mathbb{C}_k$-free graphs presented in a connected way.* □

Similarly for $k \geqslant 5$, k-chordal graphs are $\mathbb{C}_k$-free, thus Corollary 2.4.4 gives

Corollary 2.5.2 *Let $k \geqslant 5$. There is no on-line clique covering algorithm with a competitive function for $\mathbb{C}_k$-free presented in a connected way.* □

The remaining cases are $k = 3$ for coloring and $k = 3, 4$ for clique covering. However, $\mathbb{C}_3 = \mathbb{K}_3$ and $\mathbb{C}_4 = \mathbb{K}_{2,2}$. Both these situation, i.e., $\mathbb{K}_3$-free and $\mathbb{K}_{2,2}$-free graphs will be treated in some generality in Chapters 3 and 4, respectively.

2.6 Perfect graphs

The inequality $\chi(\mathbb{G}) \geqslant \omega(\mathbb{G})$ holds for every graph $\mathbb{G}$. A graph $\mathbb{G}$ is *perfect* if for every induced subgraph $\mathbb{H} \subseteq \mathbb{G}$ the chromatic number $\chi(\mathbb{H})$ is equal to $\omega(\mathbb{H})$. Perfect graphs are interesting from an algorithmic point of view. While determining the chromatic number of a graph is NP-complete, the same problem is solvable in polynomial time for perfect graphs [15]. We apply the results from the previous sections to perfect graphs. Since chordal graphs are perfect Theorem 2.2.3 immediately gives

Corollary 2.6.1 *There is no on-line coloring algorithm with a competitive function for perfect graphs presented in a connected way.* □

Proposition 2.6.2 *There is no on-line clique covering algorithm with a competitive function for perfect graphs presented in a connected way.*

Proof. We argue as in the proof of Theorem 2.4.3, that is, we use the complements $\overline{\mathbb{T}}_n$ of the trees $\mathbb{T}_n$ constructed in the proof of Theorem 2.1.1. We have already seen that $\overline{\mathbb{T}}_n$ were presented in a connected way. Since a complement of a perfect graph (in particular, of a tree) is perfect we are done. □

2.7 Conclusions

There is no competitive on-line coloring algorithm for such basic family of graphs as trees (Corollary 2.1.2, [2, 17]). This implies that there is no on-line coloring competitive algorithm for families containing all trees, for example, for bipartite graphs, k-chordal graphs, perfect graphs, planar graphs, and obviously for family of all graphs. On the other hand, in this chapter we proved that bipartite graphs (especially trees) are colored optimally when they are presented on-line in a connected way. This additionally restriction for graph presentation does not help at all in on-line coloring of chordal graphs (and also k-chordal graphs, planar graphs, perfect graphs).

In this chapter we also analyzed the on-line clique covering problem for bipartite and for chordal graphs. It turns out that there are competitive algorithms for these problems with small competitive ratios: $\frac{3}{2}$ (Corollary 2.1.7) and 2 (Corollary 2.1.10), respectively. However, there are no competitive algorithms for k-chordal graphs (Corollary 2.4.4) and perfect graphs (Proposition 2.6.2). The results for clique covering do not change if graphs are presented in a connected way.

Chapter 3

$\mathbb{K}_s$-free graphs

3.1 On-line coloring of $\mathbb{K}_s$-free graphs

A graph $\mathbb{G}$ is $\mathbb{K}_s$-free if every clique contained in $\mathbb{G}$ consists of at most $s-1$ vertices. Trees are $\mathbb{K}_3$-free, and therefore, each on-line coloring algorithm for $\mathbb{K}_s$-free graphs ($s \geqslant 3$) can be forced to use arbitrarily many colors (see Theorem 2.1.1). This means that there is no on-line coloring algorithm with a competitive function for $\mathbb{K}_s$-free graphs. Therefore, $\mathbb{K}_s$ is not a forcing structure for on-line coloring, as the presence of the clique $\mathbb{K}_s$ is not necessary to force arbitrarily many colors.

Bipartite graphs are $\mathbb{K}_3$-free. The $\mathcal{FF}$ algorithm colors bipartite graphs presented in a connected way optimally. There is a natural question whether a connected presentation helps in coloring $\mathbb{K}_s$-free graphs. In this section we show that such restriction does not help at all. Thus, as above, $\mathbb{K}_s$ is not a forcing structure for coloring problem even if an input graph is presented in a connected way. The results are summarized in Table 3.1.

Proposition 3.1.1 *Let $s \geqslant 3$. There is no on-line coloring algorithm with a competitive function for $\mathbb{K}_s$-free graphs presented in a connected way.*

Proof. The family of $\mathbb{K}_s$-free graphs ($s \geqslant 3$) contains trees. From the construction presented in the proof of Theorem 2.1.1 we know that no on-line coloring algorithm for trees has a competitive function. This construction, for every on-line coloring algorithm $\mathcal{A}$ and for every positive integer n produces trees $\mathbb{T}_n$ forcing $\mathcal{A}$ to use n colors. However, the trees $\mathbb{T}_n$ are not presented in a connected way. To prove Proposition 3.1.1 we modify this construction to show that for every on-line coloring algorithm $\mathcal{A}$ and for every positive integer n there exists a 3-colorable $\mathbb{K}_3$-free graph $\mathbb{G}_n$ presented in a connected way such that $\chi_{\mathcal{A}}(\mathbb{G}_n) \geqslant n$.

First, the spoiler presents $2k+1$ vertices, where k is large enough, as pictured on Figure 3.1. The algorithm $\mathcal{A}$ colors them somehow. In the following, levels 0 and 1 are used only to make the constructed graph connected. Actually, in the

Graph $\mathbb{G}$	**Presentation method**	**Coloring**	**Clique covering**
$\mathbb{K}_s$-free for $s \geq 3$	-	∞ [2, 17], 3.1.1	$\frac{s}{2}$ 3.2.1, 3.2.7
$\mathbb{K}_s$-free for $s \geq 3$	connected	∞ 3.1.1 based on [17]	$\frac{s}{2}$ 3.2.9
s-colorable for $s \geq 3$	-	∞ 3.3.1	$\frac{s+1}{2}$ 3.3.2
s-colorable for $s \geq 3$	connected	∞ 3.3.1	$\frac{s+1}{2}$ 3.3.2
Planar	-	∞ 3.4.1	$2 \leq \ldots \leq \frac{5}{2}$ 3.4.2
			for greedy algorithms $\frac{5}{2}$ 3.4.3
Planar	connected	∞ 3.4.1	$2 \leq \ldots \leq \frac{5}{2}$ 3.4.2
			for greedy algorithms $\frac{5}{2}$ 3.4.3

Table 3.1: Possible competitive ratios for on-line problems

proof of this theorem even level 1 is not needed, as the unique point of level 0 connects the graph. However, we allow the spoiler to create level 1 as in some further proofs it will play an essential role.

Securing connectivity, the spoiler "forgets" about vertices from level 0 and

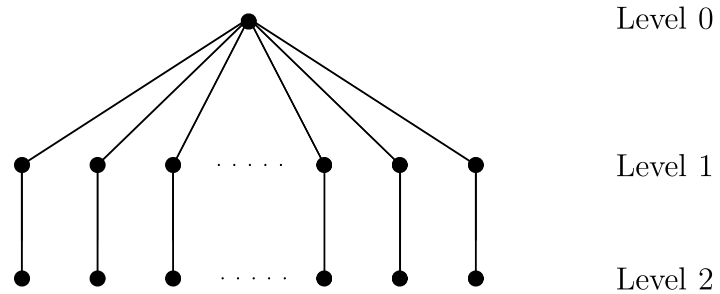


Figure 3.1: $\mathbb{G}_n$ at the beginning of the construction.

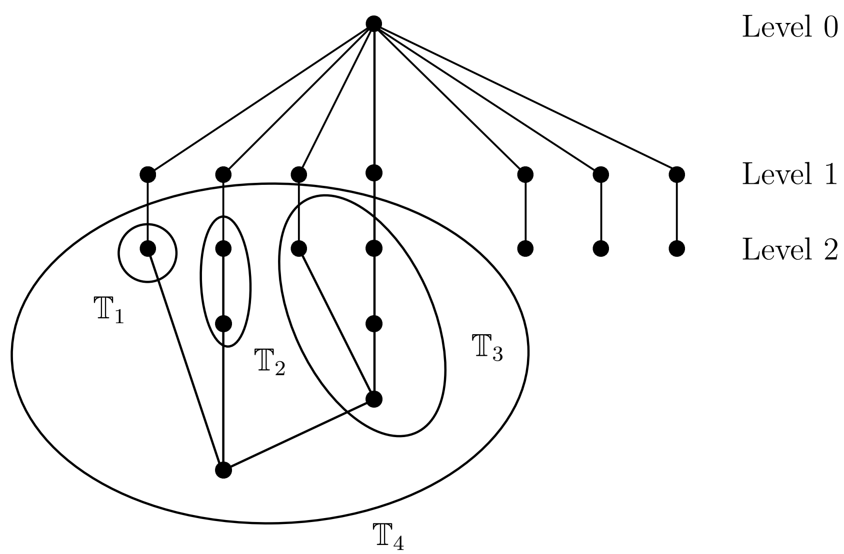


Figure 3.2: The graph $\mathbb{G}_4$.

1 and proceeds to build the tree $\mathbb{T}_n$ with leaves on level 2 (see Figure 3.2). This forces any algorithms to use at least n colors on the connected graph $\mathbb{G}_n$. Obviously, each $\mathbb{G}_n$ is $\mathbb{K}_3$-free. To see that $\mathbb{G}_n$ is 3-colorable, note that $\mathbb{G}_n$ without the only point on level 0 is a tree and therefore 2-colorable. □

3.2 On-line clique covering of $\mathbb{K}_s$-free graphs

The graphs $\mathbb{K}_s$ for $s \geqslant 3$ are not forcing structures for the coloring problem, but we will see here that they are forcing structures for the clique covering problem. In this section we show that the competitive ratio of the best on-line clique covering algorithm for $\mathbb{K}_s$-free graphs is $\frac{s}{2}$. We start our analysis with greedy algorithms.

Theorem 3.2.1 *Let $s \geqslant 3$. For every on-line greedy clique covering algorithm $\mathcal{A}$ and for every $\mathbb{K}_s$-free graph $\mathbb{G}$*

$$v_{\mathcal{A}}(\mathbb{G}) \leqslant \frac{s}{2} \cdot v(\mathbb{G}).$$

Proof. Suppose that $\mathbb{G}$ is $\mathbb{K}_s$-free and $v = v(\mathbb{G})$. Let $\mathbb{O}_1, \mathbb{O}_2, \ldots, \mathbb{O}_v$ be an optimal clique covering of $\mathbb{G}$. Note that $|O_i| \leqslant s-1$ as $\mathbb{G}$ is $\mathbb{K}_s$-free. Assume that the algorithm $\mathcal{A}$ returns m cliques, i.e., $m = v_{\mathcal{A}}(\mathbb{G})$. We show that $m \leqslant \frac{s}{2} \cdot v$.

We say that a clique $\mathbb{A}$ produced by $\mathcal{A}$ is *constrained* if there exist at least two off-line cliques $\mathbb{O}_k$ and $\mathbb{O}_l$ containing at least one vertex belonging to the clique $\mathbb{A}$, that is,

$$\exists k \neq l: \ \mathbb{A} \cap \mathbb{O}_k \neq \emptyset \neq \mathbb{A} \cap \mathbb{O}_l.$$

Inspecting the result produced by $\mathcal{A}$, i.e., the cliques $\mathbb{A}_1, \ldots, \mathbb{A}_m$ we define

- k_c = number of vertices in the join of all constrained $\mathbb{A}_i$'s,
- k_u = number of vertices in the join of all unconstrained $\mathbb{A}_i$'s,
- u = number of unconstrained $\mathbb{A}_i$'s.

Obviously, $u \leqslant k_u$. Since $|O_i| \leqslant s-1$ for each i, there are at most $(s-1) \cdot v$ vertices of $\mathbb{G}$. Therefore $k_c \leqslant (s-1) \cdot v - k_u$. Each constrained clique $\mathbb{A}_i$ consists of at least two vertices, so that there are at most $\frac{k_c}{2}$ constrained $\mathbb{A}_i$'s. These easy observations allow us to estimate the possible number of all cliques returned by $\mathcal{A}$:

$$\begin{aligned} m = v_{\mathcal{A}}(\mathbb{G}) &\leqslant u + \frac{k_c}{2} \leqslant u + \frac{(s-1) \cdot v - k_u}{2} = u + \frac{(s-1) \cdot v}{2} - \frac{k_u}{2} \\ &\leqslant \frac{(s-1) \cdot v}{2} + u - \frac{u}{2} = \frac{(s-1) \cdot v}{2} + \frac{u}{2}\ . \end{aligned}$$

Since $\mathcal{A}$ is greedy, each off-line clique $\mathbb{O}_i$ contains at most one unconstrained clique, so that $u \leqslant v$, and therefore

$$m \leqslant \frac{(s-1) \cdot v}{2} + \frac{v}{2} = \frac{s}{2} \cdot v = \frac{s}{2} \cdot v(\mathbb{G}). \qquad \square$$

Next, we show that the obtained upper bound for a competitive function is tight when restricted to greedy algorithms.

Theorem 3.2.2 *Let $s \geqslant 3$. For every on-line greedy clique covering algorithm $\mathcal{A}$ and for every even integer v there exists a $\mathbb{K}_s$-free graph $\mathbb{G}$ with $v(\mathbb{G}) = v$ such that*

$$v_{\mathcal{A}}(\mathbb{G}) \geqslant \frac{s}{2} \cdot v(\mathbb{G}).$$

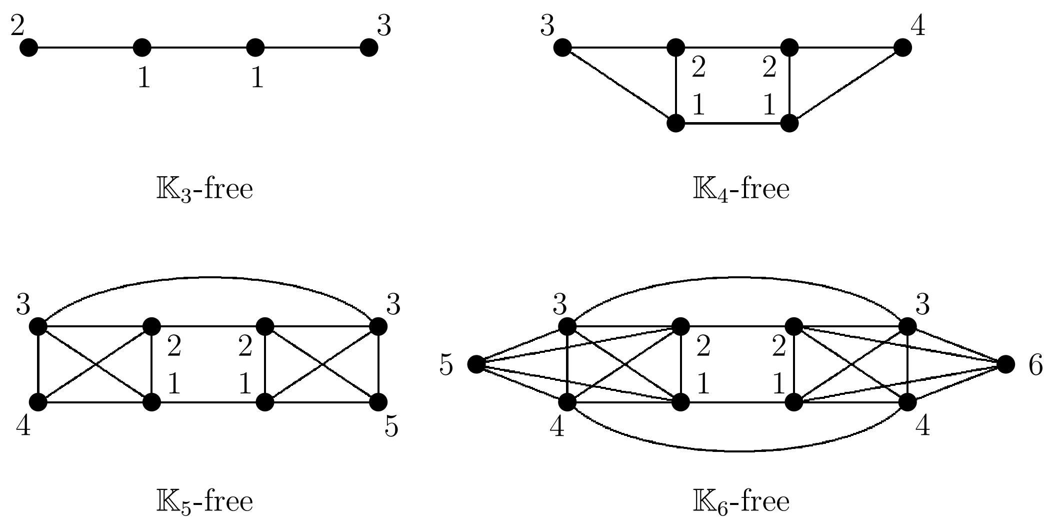


Figure 3.3: Examples

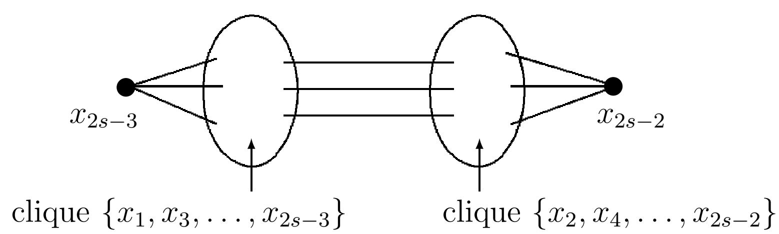


Figure 3.4: Construction of a $\mathbb{K}_s$-free graph

Proof. To prove our theorem we present a strategy for the spoiler to construct a $\mathbb{K}_s$-free graphs $\mathbb{G}$ forcing $\mathcal{A}$ to use $\frac{s}{2} \cdot v(\mathbb{G})$ cliques. Before we describe the general case, we picture the final graphs for $s = 3, 4, 5, 6$ on Figure 3.3.

In $\mathbb{K}_s$-free setting, the spoiler first presents $2 \cdot (s-2)$ vertices in the order $x_1, x_2, \ldots, x_{2(s-2)}$ such that the sets of vertices $\{x_i : i \text{ is odd}\}$ and $\{x_i : i \text{ is even}\}$ form cliques and for every $i = 1, 3, \ldots, 2s-5$ there exists an edge between x_i and x_{i+1}.

The graph $\mathbb{G}$, consisting of presented $2(s-2)$ vertices, can be covered optimally with two cliques, while the greedy algorithm $\mathcal{A}$ returns $(s-2)$ cliques of the form $\{x_i, x_{i+1}\}$. At the end, the spoiler adds a new vertex x_{2s-3} to one clique in the optimal covering of $\mathbb{G}$ and a second new vertex x_{2s-2} to the other one (see Figure 3.4). The algorithm $\mathcal{A}$ is forced to assign them to two new cliques, so that $v_{\mathcal{A}}(\mathbb{G}) = s$. This, together with $v(\mathbb{G}) = 2$, gives $v_{\mathcal{A}}(\mathbb{G}) = \frac{s}{2} \cdot v(\mathbb{G})$. It is easy to notice that the graph $\mathbb{G}$ is $\mathbb{K}_s$-free.

In order to construct a connected $\mathbb{K}_s$-free graphs with the greater number of cliques in an optimal covering, the spoiler adds a new vertex x to the obtained graph $\mathbb{G}$ and joins it to an arbitrary vertex from 2-vertex clique, determined by $\mathcal{A}$, of the form $\{x_i, x_{i+1}\}$. He starts with this vertex x and constructs the next

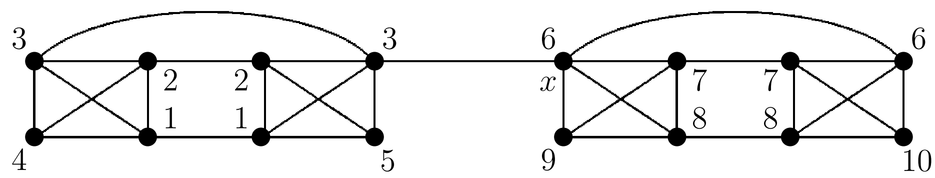


Figure 3.5: $\mathbb{K}_5$-free graph $\mathbb{G}$ with $v(\mathbb{G}) = 4$.

graph isomorphic to $\mathbb{G}$ (see Figure 3.5). □

Corollary 3.2.3 *Let $s \geqslant 3$. Each on-line clique covering greedy algorithm for $\mathbb{K}_s$-free graphs has a competitive ratio $\frac{s}{2}$.* □

Since the graphs constructed in the proof of Theorem 3.2.2 are presented in a connected way, we immediately get:

Corollary 3.2.4 *Let $s \geqslant 3$. Each on-line clique covering greedy algorithm for $\mathbb{K}_s$-free graphs presented in a connected way has a competitive ratio $\frac{s}{2}$.* □

After analysing on-line greedy clique covering algorithms, we are interested in the general situation, i.e., in algorithms that are not necessarily greedy: Is there an on-line clique covering algorithm for $\mathbb{K}_s$-free graphs with a competitive ratio lower than $\frac{s}{2}$? The answer is easy and negative for $s = 3$ and $s = 4$.

Consider the case $s = 3$. From Proposition 2.1.6 we know that for every on-line clique covering algorithm $\mathcal{A}$ there is an infinite family of trees $\mathbb{T}$ presented in a connected way such that $v_{\mathcal{A}}(\mathbb{T}) \geqslant \frac{3}{2} \cdot v(\mathbb{T})$. Since trees are $\mathbb{K}_3$-free, this, together with Theorem 3.2.1, gives that the competitive ratio for $\mathbb{K}_3$-free graphs is $\frac{3}{2}$.

In the next proposition we consider the case $s = 4$ and show that there is no on-line clique covering algorithm for $\mathbb{K}_4$-free graphs with a competitive ratio better than 2. This means that there is no non-greedy algorithm which is better than the greedy ones.

Proposition 3.2.5 *There is no on-line clique covering algorithm with a competitive ratio lower than 2 for $\mathbb{K}_4$-free graphs presented in a connected way.*

Proof. Let $\mathcal{A}$ be an arbitrary on-line clique covering algorithm. We show that there exists an infinite family of $\mathbb{K}_4$-free graphs $\mathbb{G}$ with arbitrarily large $v(\mathbb{G})$ for which $v_{\mathcal{A}}(\mathbb{G}) \geqslant 2 \cdot v(\mathbb{G})$.

In the first step, the spoiler presents a cycle consisting of four vertices: x_1, x_2, x_3, x_4. The algorithm $\mathcal{A}$ returns two, three or four cliques as presented by three cases on Figure 3.6.

In the second step, the spoiler adds vertices x_5, x_6 and joins them to the cycle as pictured on Figure 3.6. Now, if $\mathcal{A}$ uses only three cliques, the spoiler joins

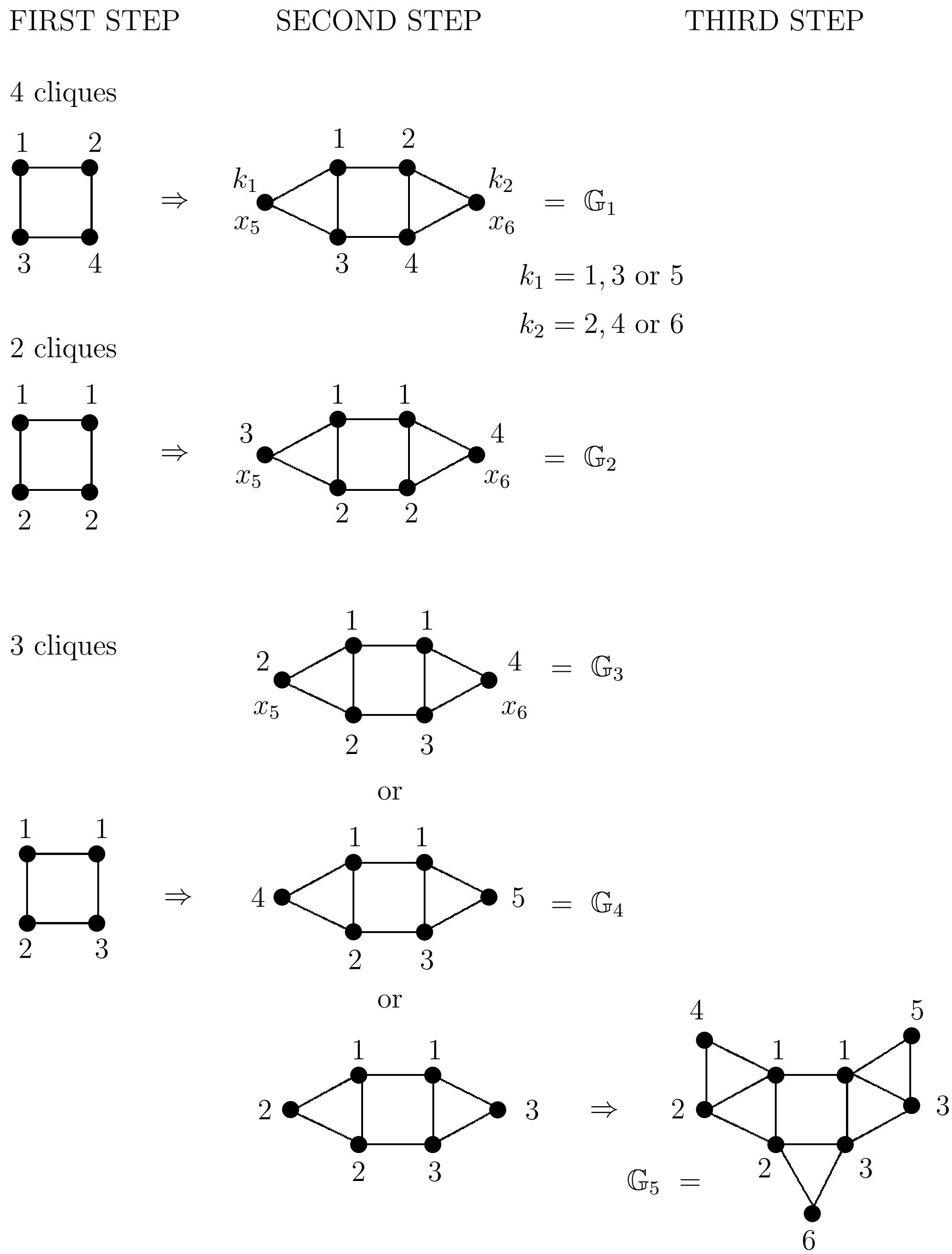


Figure 3.6: Construction of $\mathbb{K}_4$-free graphs

additional three vertices (see third step on Figure 3.6). In this way he obtains a graph of the form $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_3$, $\mathbb{G}_4$ or $\mathbb{G}_5$ for which $v_{\mathcal{A}}(\mathbb{G}) \geqslant 2 \cdot v(\mathbb{G})$.

In order to construct connected graphs with the greater number of cliques in an optimal covering the spoiler adds a new vertex x and joins it to an arbitrary vertex of 2-vertex clique determined by $\mathcal{A}$ (in case 1 to an arbitrary vertex). He starts with this vertex x and constructs the next graph of the form $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_3$, $\mathbb{G}_4$ lub $\mathbb{G}_5$ depending on the algorithm behavior. □

Summing up we have:

Corollary 3.2.6 *The best on-line clique covering algorithms for $\mathbb{K}_4$-free graphs presented in a connected way have a competitive ratio* 2. □

Now, we show that there is no on-line clique covering algorithm for $\mathbb{K}_s$-free graphs with a competitive ratio lower than $\frac{s}{2}$.

Theorem 3.2.7 *Let $s \geqslant 5$. For every on-line clique covering algorithm $\mathcal{A}$ and for every positive integer v there exists a $\mathbb{K}_s$-free graph $\mathbb{G}$ with $v(\mathbb{G}) = v$ such that*

$$v_{\mathcal{A}}(\mathbb{G}) \geqslant \frac{s}{2} \cdot v(\mathbb{G}) - \left(\frac{2s-3}{4}\right)^2 .$$

Proof. We describe a way the spoiler builds a $\mathbb{K}_s$-free graph $\mathbb{G}$ with $v(\mathbb{G}) = v$ forcing $\mathcal{A}$ to use $\frac{s}{2} \cdot v - \left(\frac{2s-3}{4}\right)^2$ cliques to cover it. For simplification we talk about colors instead of cliques returned by $\mathcal{A}$. In other words, we analyze a special on-line coloring of $\mathbb{G}$ with the requirement that the set of vertices of the same color forms a clique.

The basic tool used by the spoiler to force a new color is a construction called a join of two cliques: for two disjoint cliques $\mathbb{K}$, $\mathbb{L}$ he forms a new graph, with the help of a new point $x \notin K \cup L$, with the universe $K \cup \{x\} \cup L$ and additional edges that connect x with all vertices in $K \cup L$.

The spoiler starts with v one-element cliques $\mathbb{O}_1, \ldots, \mathbb{O}_v$. In each step, he keeps for himself an updated optimal clique covering $\mathbb{O}_1, \ldots, \mathbb{O}_v$ of the graph presented up to this step. Such a step (and update) consists of a join of some $\mathbb{O}_i$ and $\mathbb{O}_j$ with possible additional point depending on the respond of the algorithm $\mathcal{A}$. It is described by procedure *JOIN(i,j)*. In this procedure we use a concept of a tied clique. A clique $\mathbb{O}$ is *tied* if every color of its vertices appears outside $\mathbb{O}$. In other words, a clique $\mathbb{O}$ is tied if for each vertex $x \in O$ there exists $y \in G \setminus O$ with $cl_{\mathcal{A}}(y) = cl_{\mathcal{A}}(x)$.

procedure *JOIN*(i, j);
begin
if $|O_i| > |O_j|$ **then begin** *JOIN*(j, i); exit; **end**;
{the spoiler adds a new vertex x}
$\mathbb{G}' := (G \cup \{x\}\,,\ E_{\mathbb{G}} \cup \{xy :\ y \in O_i \cup O_j\})$;

WAIT for a response of $\mathcal{A}$ to color (cover) x

if $cl_{\mathcal{A}}(x) \in Cl_{\mathcal{A}}(\mathbb{O}_i)$ **then** $\mathbb{O}_j := \mathbb{O}_j \cup \{x\}$
else $\mathbb{O}_i := \mathbb{O}_i \cup \{x\}$;

if $\mathbb{O}_i$ is tied **then**
begin {the spoiler adds a new vertex z_1}
6: $\mathbb{G}' := (G' \cup \{z_1\}\,,\ E_{\mathbb{G}'} \cup \{z_1 y :\ y \in O_i\});$
7: $\mathbb{O}_i := \mathbb{O}_i \cup \{z_1\};$
end
else if $\mathbb{O}_j$ is tied **then**
begin {the spoiler adds a new vertex z_2}
8: $\mathbb{G}' := (G' \cup \{z_2\}\,,\ E_{\mathbb{G}'} \cup \{z_2 y :\ y \in O_j\});$
9: $\mathbb{O}_j := \mathbb{O}_j \cup \{z_2\};$
end;
10: $\mathbb{G} := \mathbb{G}';$
end;

There are three possible kinds of joins performed by the procedure above: wasteful join, classical join and binding join:

1. A join is *wasteful* if $\mathcal{A}$ colors a new vertex x (line 3) with a new color, i.e., $cl_{\mathcal{A}}(x) \notin Cl_{\mathcal{A}}(\mathbb{G} \setminus \{x\})$, although it could color x with some old one.

2. A join is *classical* if $\mathcal{A}$ colors a new vertex x (line 3) with a used color and this action makes one of the joined cliques tied. In other words, a join is *classical* if one of lines 6 or 8 from procedure *JOIN* is performed. Performing this line, the spoiler forces $\mathcal{A}$ to use a new color.

3. A join is *binding* if $\mathcal{A}$ colors a new vertex x (line 3) with a used color, but the cliques $\mathbb{O}_i$, $\mathbb{O}_j$ do not become tied. In other words, a join is *binding* if $cl_{\mathcal{A}}(x) \in Cl_{\mathcal{A}}(\mathbb{G} \setminus \{x\})$ and neither line 6 nor line 8 from procedure *JOIN* is performed. This is the only case in which *JOIN(i,j)* actually does not force $\mathcal{A}$ to introduce a new color.

Note that joins do not increase the number of cliques in an optimal (off-line) covering, which (after first v totally disconnected vertices) is always v. To force the algorithm $\mathcal{A}$ to use $\frac{s}{2} \cdot v(\mathbb{G}) - \left(\frac{2s-3}{4}\right)^2$ colors (cliques) the spoiler builds $\mathbb{G}$ with $v = v(\mathbb{G})$ by the following strategy:

1. First, the spoiler presents v separated vertices.

2. Next, the spoiler performs procedure *JOIN* as many times as, among $\mathbb{O}_i$'s, there exist two cliques, say $\mathbb{O}_i$ and $\mathbb{O}_j$, of size at most $s-2$ and such that *JOIN(i,j)* has not yet been performed.

Note that in order to keep the constructed graph $\mathbb{K}_s$-free, one can apply *JOIN* only to cliques that have at most $s-2$ vertices.

An important features of the spoiler's strategy are that each clique $\mathbb{O}_i$ does not contain two vertices with the same color and that the algorithm $\mathcal{A}$ may use one color at most twice. Another invariant kept by the spoiler is that none of the $\mathbb{O}_i$'s is tied. Indeed, at the beginning $\mathbb{O}_1, \ldots, \mathbb{O}_v$ are one element separated cliques, thus they are not tied. If the clique $\mathbb{O}_i$ becomes tied during the join performance, the spoiler adds to it new vertex (line 6 or 8). This vertex is colored by a new color, therefore clique $\mathbb{O}_i$ becomes not tied.

Now we count number of colors used by $\mathcal{A}$ on the resulting graph $\mathbb{G}$. In order to do this we split the cliques $\mathbb{O}_i$'s into two categories: a clique $\mathbb{O}_i$ is *complete* if $|O_i| = s-1$ and *uncomplete* otherwise. Let k and l be the numbers of complete and uncomplete cliques, respectively. Obviously $k + l = v$. Denote by $\mathbb{A}$ the induced subgraph of $\mathbb{G}$ on the vertices in all complete cliques, and by $\mathbb{B}$ the subgraph induced on the remaining vertices. As $\mathcal{A}$ use one color at most twice and none of the $\mathbb{O}_i$'s is tied, the number of colors used by $\mathcal{A}$ on $\mathbb{A}$ is at least

$$\frac{(s-2)\cdot k}{2} + k = \frac{s \cdot k}{2}.$$

Now, note that *JOIN*(i,j) may be binding if there exists two vertices $x_1, x_2 \in O_i$ (or $x_1, x_2 \in O_j$ analogously) such that $cl_{\mathcal{A}}(x_1) \neq cl_{\mathcal{A}}(x_2)$ and $cl_{\mathcal{A}}(x_1), cl_{\mathcal{A}}(x_2) \notin Cl_{\mathcal{A}}(\mathbb{G} \setminus O_i)$. In other words, if the set (of so called $\mathbb{O}_i$-private colors) $P = Cl_{\mathcal{A}}(\mathbb{O}_i) \setminus Cl_{\mathcal{A}}(\mathbb{G} \setminus O_i)$ has size at least 2. The new vertex x introduced by *JOIN*(i,j) (line 2) is colored by one of the $\mathbb{O}_i$-private colors and it is added to the clique $\mathbb{O}_j$. At this moment the color $cl_{\mathcal{A}}(x)$ is not longer private for $\mathbb{O}_i$. On the other hand, the existence of two mentioned vertices x_1, x_2 with $\mathbb{O}_i$-private colors is an effect of performance of some wasteful join *JOIN*(i,l), which introduced new vertex with a new color to the clique $\mathbb{O}_i$. This means that the number of all performed binding joins does not exceed the number of all wasteful joins.

Define:

- p_1 = number of classical joins between cliques in $\mathbb{B}$,
- p_2 = number of wasteful joins between cliques in $\mathbb{B}$,
- p_3 = number of binding joins between cliques in $\mathbb{B}$,
- L = number of all joins between cliques in $\mathbb{B}$.
- q = number of wasteful joins between cliques in $\mathbb{A}$ and in $\mathbb{B}$ such that the new vertex added during the join (line 2) was joined to the clique in $\mathbb{B}$.

Note that since each two cliques in $\mathbb{B}$ are small enough to be joined, they had been already joined. In particular, we get $L = \binom{l}{2}$. Since $p_3 \leqslant p_2 + q$, the number of new colors forced by the joins between cliques in $\mathbb{B}$ is

$$p_1 + p_2 + q \geqslant \frac{2p_1 + 2p_2 + q}{2} \geqslant \frac{2p_1 + 2p_2 + p_3 - p_2}{2} \geqslant \frac{L}{2} = \frac{l(l-1)}{4}.$$

This, together with the first l colors, gives that the number of colors used by $\mathcal{A}$ on $\mathbb{B}$ is equal $l + \frac{l(l-1)}{4}$. These observations allows us to estimate the possible number of all colors returned by $\mathcal{A}$:

$$\begin{aligned} v_{\mathcal{A}}(\mathbb{G}) &\geqslant \frac{s \cdot k}{2} + l + \frac{l(l-1)}{4} = \frac{s \cdot (v-l)}{2} + \frac{l \cdot (l+3)}{4} \\ &= \frac{s \cdot v}{2} + \frac{l}{2} \cdot \left(-s + \frac{(l+3)}{2}\right) = \frac{s \cdot v}{2} + \frac{l}{4} \cdot (l + 3 - 2s). \end{aligned}$$

Since the minimum value of $f(l) = \frac{l}{4} \cdot (l + 3 - 2s)$ is $-\frac{(2s-3)^2}{16}$, we finally get

$$v_{\mathcal{A}}(\mathbb{G}) \geqslant \frac{s}{2} \cdot v - \left(\frac{2s-3}{4}\right)^2. \qquad \square$$

The graph $\mathbb{G}$ constructed in the proof of Theorem 3.2.7 was not presented in a connected way. In the next theorem we improve spoiler's presentation technique to make it connected.

Theorem 3.2.8 *Let $s \geqslant 5$. For every on-line clique covering algorithm $\mathcal{A}$ and for every positive integer v there exists a $\mathbb{K}_s$-free graph $\mathbb{G}$ with $v(\mathbb{G}) = v$ presented in a connected way and such that*

$$v_{\mathcal{A}}(\mathbb{G}) \geqslant \frac{s}{2} \cdot v(\mathbb{G}) - \left(\frac{2s-3}{4}\right)^2.$$

Proof. To show this lower bound, we describe a way the spoiler builds a $\mathbb{K}_s$-free graph $\mathbb{G}$ presented in a connected way. The resulting graph $\mathbb{G}$ is actually the same as the one described in the proof of Theorem 3.2.7, but the spoiler presents its vertices in a different order. To keep connectivity, we omit point 1 in the spoiler's strategy and modify procedure *JOIN*, so that the *MODIFIED_JOIN* produces new $\mathbb{O}_i$'s in the off-line clique covering, when necessary. Again, for simplification we talk about colors instead of cliques returned by $\mathcal{A}$.

Let $\mathbb{O}_1, \ldots, \mathbb{O}_v$ denote cliques which are supposed to form an optimal clique covering of $\mathbb{G}$ at the end of the construction. Initially $\mathbb{O}_1 = \{x_0\}$ and $\mathbb{O}_2 = \ldots = \mathbb{O}_v = \emptyset$. To force the algorithm $\mathcal{A}$ to use $\frac{s}{2} \cdot v(\mathbb{G}) - \left(\frac{2s-3}{4}\right)^2$ colors (cliques) the spoiler performs procedure *MODIFIED_JOIN* as many times as, among $\mathbb{O}_i$'s, there exist two cliques, say $\mathbb{O}_i$ and $\mathbb{O}_j$, of size at most $s-2$ and such that *MODIFIED_JOIN(i,j)* has not yet been performed. In this procedure we use a concept of a tied clique introduced in the proof of Theorem 3.2.7. Recall, that a clique $\mathbb{O}$ is tied if each color of its vertices appears outside $\mathbb{O}$, that is, for each vertex $x \in O$ there exists $y \in G \setminus O$ with $cl_{\mathcal{A}}(y) = cl_{\mathcal{A}}(x)$. Procedure *MODIFIED_JOIN* is defined as follows.

procedure *MODIFIED_JOIN(i, j)*;
begin
if $|O_i| \neq \emptyset$ **and** $|O_j| \neq \emptyset$ **then begin** *JOIN(i, j)*; exit; **end**;
if $|O_i| > |O_j|$ **then begin** *MODIFIED_JOIN(j, i)*; exit; **end**;

{if $|O_i| = \emptyset$ then the spoiler adds two new vertices x_1 and x_2};
3: $\mathbb{G}' := (G \cup \{x_1, x_2\}\,,\ E_{\mathbb{G}} \cup \{x_1x_2\} \cup \{x_1y:\ y \in O_j\})$;
4: $\mathbb{O}_i := \{x_2\}$;

5: WAIT for a response of $\mathcal{A}$ to color x_1 and x_2

6: **if** $cl_{\mathcal{A}}(x_1) = cl_{\mathcal{A}}(x_2)$ **then** $\mathbb{O}_j := \mathbb{O}_j \cup \{x_1\}$
7: **else** $\mathbb{O}_i := \mathbb{O}_i \cup \{x_1\}$;

if $\mathbb{O}_i$ is tied **then**
begin {the spoiler adds a new vertex z_1}
8: $\mathbb{G}' := (G' \cup \{z_1\}\,,\ E_{\mathbb{G}'} \cup \{z_1y:\ y \in O_i\})$;
9: $\mathbb{O}_i := \mathbb{O}_i \cup \{z_1\}$;
end
else if $\mathbb{O}_j$ is tied **then**
begin {the spoiler adds a new vertex z_2}
10: $\mathbb{G}' := (G' \cup \{z_2\}\,,\ E_{\mathbb{G}'} \cup \{z_2y:\ y \in O_j\})$;
11: $\mathbb{O}_j := \mathbb{O}_j \cup \{z_2\}$;
end;
12: $\mathbb{G} := \mathbb{G}'$;
end;

Similarly to the proof of Theorem 3.2.7, there are three possible kinds of joins, called wasteful join, classical join and binding join. They are defined in the same way as the ones in the proof of Theorem 3.2.7 except for the situation when join performance increases the number of cliques in an optimal covering. Then we use the following.

1. A join is *wasteful* if $\mathcal{A}$ colors new vertices x_1 and x_2 (line 5) with two new colors. Note that, in this case, lines 8 and 10 from procedure *MODIFIED_JOIN* are not performed.

2. A join is *classical* if one of the joined cliques becomes tied, i.e., one of lines 8 or 10 from procedure *MODIFIED_JOIN* is performed. Performing this join, the spoiler forces $\mathcal{A}$ to use two new colors.

3. A join is *binding* if $\mathcal{A}$ colors a first vertex x_1 (line 5) with a used color, but the clique $\mathbb{O}_j$ does not set tied. In other words, a join is binding if $cl_{\mathcal{A}}(x_1) \in cl_{\mathcal{A}}(\mathbb{G} \setminus \{x_1, x_2\})$ and lines 8 and 10 from procedure *MODIFIED_JOIN* are not performed. This is the only case in which *MODIFIED_JOIN* forces $\mathcal{A}$ to use only one additional color.

As in the proof of Theorem 3.2.7 the spoiler performs *MODIFIED_JOIN* many times. In order to keep the graph presented in a connected way, each *MODIFIED_JOIN(i, j)* is allowed to be applied to two cliques $\mathbb{O}_i$ and $\mathbb{O}_j$ such that at least one of them is non-empty. This can be done, for example, if the spoiler performs first $v-1$ joins in the order determined by the following loop.

$$\textbf{for } i := 1, \ldots, v-1 \textbf{ do } \textit{MODIFIED\_JOIN}(i, i+1);$$

Next, the spoiler proceeds as in the proof of Theorem 3.2.7.

After executing this new strategy, the spoiler obtains the graph $\mathbb{G}$ with $v(\mathbb{G}) = v$. It was presented in a connected way. One way to estimate the number of colors used by $\mathcal{A}$ is to repeat the argument from the proof of Theorem 3.2.7. In the strategy analysis we use new definitions of wasteful, classical, binding joins instead of the ones from the proof of Theorem 3.2.7. The other way is to note that the resulting graph in both proofs are in fact the same (isomorphic). Call by $\mathcal{S}_0$ and $\mathcal{A}_0$ the spoiler and the algorithm from the proof of Theorem 3.2.7, and by $\mathcal{S}$ and $\mathcal{A}$ the ones in this proof. A careful inspection of procedure *MODIFIED_JOIN* gives that for each game played by the algorithm $\mathcal{A}$ with the spoiler $\mathcal{S}$ there is a way for $\mathcal{A}_0$ to play with the spoiler $\mathcal{S}_0$ such that the graph built by $\mathcal{S}$ (which is the same as the one built by $\mathcal{S}_0$) is colored in the same way as the one built by $\mathcal{S}_0$. Therefore, as in the proof of Theorem 3.2.7, we have

$$v_{\mathcal{A}}(\mathbb{G}) \geqslant \frac{s}{2} \cdot v - \left(\frac{2s-3}{4}\right)^2. \qquad \square$$

Corollary 3.2.9 *Let $s \geqslant 3$. The best on-line clique covering algorithm for $\mathbb{K}_s$-free graphs presented in a connected way has a competitive ratio $\frac{s}{2}$.* □

3.3 s-colorable graphs

In this section we are interested in on-line coloring and clique covering algorithms with competitive functions for s-colorable graphs for given s.

The 2-colorable graphs are exactly bipartite graphs. They are analyzed in Chapter 2. From Corollary 2.1.3 we know that for bipartite graphs none of the on-line coloring algorithm has a competitive function. Consequently this result holds also for s-colorable graphs for $s \geqslant 3$.

If a bipartite graph is presented in a connected way, the on-line $\mathcal{FF}$ coloring algorithm colors it optimally. On the other hand, from what we have already established, it easy to see that for 3-colorable graphs, a connected way of presentation does not help at all. Indeed, the graphs constructed in the proof of Theorem 2.2.3 are 3-colorable. Since they are presented in a connected way and force each on-line coloring algorithm $\mathcal{A}$ to use arbitrarily many colors, we immediately get

Corollary 3.3.1 *Let $s \geqslant 3$. There is no on-line coloring algorithm with a competitive function for s-colorable graphs presented in a connected way.* □

The s-colorable graphs are obviously $\mathbb{K}_{s+1}$-free. Therefore, from Theorem 3.2.1, we get $\frac{s+1}{2} \cdot v(\mathbb{G})$ to be an upper bound for a competitive function for clique covering of s-colorable graphs. The graphs constructed in the proof of the lower bound for a competitive function for $\mathbb{K}_s$-free graphs (Theorem 3.2.8) are $(s-1)$–colorable. Thus, they are good enough to be forcing graphs for on-line clique covering algorithms for s-colorable graphs, so that we get the following.

Corollary 3.3.2 *Let $s \geqslant 4$. A competitive ratio for clique covering of s-colorable graphs presented in a connected way is $\frac{s+1}{2}$. More precisely, a competitive function for this problem lies between $\frac{s+1}{2} \cdot v(\mathbb{G}) - \left(\frac{2s-1}{4}\right)^2$ and $\frac{s+1}{2} \cdot v(\mathbb{G})$, where $\mathbb{G}$ is an input graph.* □

At the moment we are unable to narrow the gap described in Corollary 3.3.2. Moreover, for $s = 3$ the gap does not exists, namely, the competitive function for clique covering 3-colorable graphs is exactly $2 \cdot v(\mathbb{G})$. It follows from the proof of Proposition 3.2.5. The graphs presented in this proof are 3-colorable and force every on-line clique covering algorithm to use $2 \cdot v(\mathbb{G})$ cliques.

Corollary 3.3.3 *The best on-line clique covering algorithms for 3-colorable graphs presented in a connected way have the competitive function $2 \cdot v(\mathbb{G})$, where $\mathbb{G}$ is an input graph.* □

The gap described in Corollary 3.3.2 disappears for greedy algorithms. The $\mathbb{K}_s$-free graphs presented in the proof of Theorem 3.2.2 are $(s-1)$–colorable and force on-line greedy algorithms to use $\frac{s}{2} \cdot v(\mathbb{G})$ cliques. Therefore, we have

Corollary 3.3.4 *Let $s \geq 2$. The on-line greedy clique covering algorithms for s-colorable graphs presented in a connected way have the competitive function $\frac{s+1}{2} \cdot v(\mathbb{G})$, where $\mathbb{G}$ is an input graph.* □

3.4 Planar graphs

A graph $\mathbb{G}$ is *planar* if it can be drawn in the plane with its edges intersecting at their vertices only, in other words, without edge crossing. Such a drawing of a planar graph is called a plane drawing or a *plane graph*. The family of planar graphs contains, for example, trees, cycles, some cliques: $\mathbb{K}_1$, $\mathbb{K}_2$, $\mathbb{K}_3$, $\mathbb{K}_4$ (but not $\mathbb{K}_5$), stars $\mathbb{K}_{1,t}$, bipartite graphs $\mathbb{K}_{2,t}$ (but not $\mathbb{K}_{3,3}$). A plane graph divides the plane into several regions called faces. In other words, a *face* in a plane graph is a subset of the plane bounded by a cycle without diagonal paths. An unbounded region of the plane graph is called the *outer face*.

Planar graphs are $\mathbb{K}_5$-free, therefore the results established in sections 3.2 and 3.1 apply. In particular, they give upper bound for clique covering. On the other hand, they do not give anything for on-line coloring. In fact, the graphs constructed in the proofs of Theorem 2.2.3 are planar so that we have the following

Corollary 3.4.1 *None of the on-line coloring algorithms for planar graphs presented in a connected way has a competitive function.* □

Returning to the on-line clique covering problem for planar graphs we know (Theorems 3.2.7 and 3.2.8) that $\frac{5}{2}$ is an upper bound for a competitive ratio. However, the graphs used to prove $\frac{5}{2}$ to be also the lower bound for $\mathbb{K}_5$-free graphs (Theorem 3.2.2) are not planar. Thus one can expect a better behavior in this case. In particular, using the $\mathbb{K}_4$-free, planar graphs from the proof of Proposition 3.2.5 we get the lower bound in the following corollary.

Corollary 3.4.2 *A competitive ratio for on-line clique covering of planar graphs presented in a connected way is at least* 2 *and at most* $\frac{5}{2}$. □

Determining the exact competitive function for this problem is still open. The following theorem solves it only for greedy algorithms.

Theorem 3.4.3 *Each on-line greedy clique covering algorithm for planar graphs presented in a connected way has the competitive function* $\lfloor \frac{5}{2} \cdot v(\mathbb{G}) \rfloor$*, where* $\mathbb{G}$ *is an input graph.*

Proof. Note that since the number of cliques in a covering is always integer, the upper bound provided for $\mathbb{K}_5$-free graphs by Theorem 3.2.1 is in fact $\lfloor \frac{5}{2} \cdot v(\mathbb{G}) \rfloor$.

Let $\mathcal{A}$ be an on-line greedy clique covering algorithm. In this proof, for every positive integer $v \geqslant 4$, we show a way the spoiler constructs a planar graph $\mathbb{G}$ with $v(\mathbb{G}) = v$ presented in a connected way and forcing $\mathcal{A}$ to return $\lfloor \frac{5}{2} \cdot v \rfloor$ cliques. Moreover, the graph $\mathbb{G}$ will be presented together with its plane drawing. During the construction the spoiler keeps, for himself, and updates an optimal clique covering $\mathbb{O}_1, \ldots, \mathbb{O}_v$.

In the first step, the spoiler presents a cycle consisting of $2v$ vertices. The algorithm $\mathcal{A}$ covers it with v cliques: $\mathbb{A}_1, \ldots, \mathbb{A}_v$ with $|A_i|$=2. After this step,

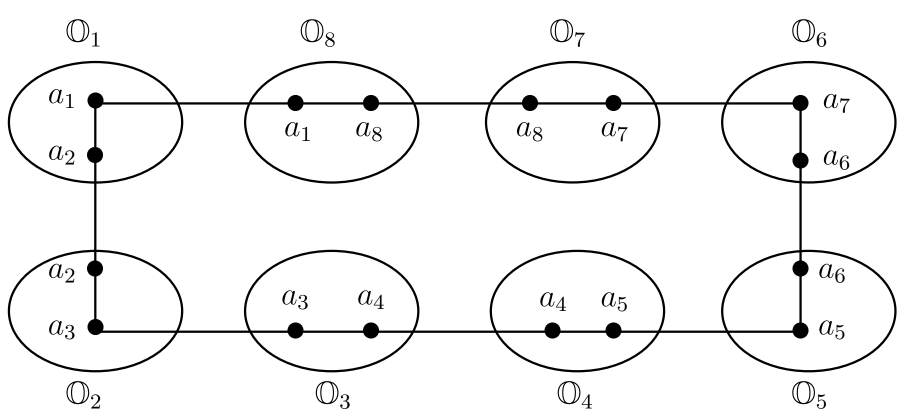


Figure 3.7: The first step of spoiler's strategy for $v = 8$. $a_i \in Cl_{\mathcal{A}}(\mathbb{A}_i)$.

the spoiler decides that each clique $\mathbb{O}_i$ consists of two vertices (see Figure 3.7) such that

$$cl_{\mathcal{A}}(x) \in cl_{\mathcal{A}}(\mathbb{G} \setminus O_i) \quad \text{whenever } x \in O_i. \tag{3.1}$$

The next steps of spoiler's strategy connect some $\mathbb{O}_i$ and $\mathbb{O}_j$ in the way described by procedure *BIJOIN(i,j)*.

Procedure *BIJOIN*(i, j);
begin
$\{$the spoiler adds new vertices a, $b\}$
1: $\mathbb{G} := (\ G \cup \{a, b\}\ ,\ E_{\mathbb{G}} \cup \{ab\} \cup \{ax : x \in O_i\} \cup\ \{by : y \in O_j\}\ )$;
2: $\mathbb{O}_i := \mathbb{O}_i \cup \{a\}$;
3: $\mathbb{O}_j := \mathbb{O}_j \cup \{b\}$;
end;

The spoiler actually presents the graph $\mathbb{G}$ together with its drawing on the plane by deciding how the vertices added by procedure *BIJOIN* should be drawn. He has a choice of two kinds of joins, called outside join and inside join. We say that a join is *outside* if vertices added during the join performance are drawn in the outer face of $\mathbb{G}$. In the other case, a join is *inside*. During the construction, the spoiler joins cliques $\mathbb{O}_1, \ldots, \mathbb{O}_v$ according to the following rules (see Figure 3.8):

- If $v \pmod 4 = 0$, the spoiler performs outside joins between cliques $\mathbb{O}_i$ and $\mathbb{O}_{i+2}$ for $i = 1, 5, 9, v-3$ and inside joins between cliques $\mathbb{O}_i$ and $\mathbb{O}_{i+2}$ for $i = 2, 6, 10, v-2$. Summing up, the spoiler performs $\frac{v}{2}$ joins.

- If $v \pmod 4 = 1$, the spoiler ignores $\mathbb{O}_v$ and joins cliques as previously. In other words, he performs outside joins between $\mathbb{O}_i$ and $\mathbb{O}_{i+2}$ for $i = 1, 5, 9, v-4$ and inside joins between $\mathbb{O}_i$ and $\mathbb{O}_{i+2}$ for $i = 2, 6, 10, v-3$. Summing up, the spoiler performs $\frac{v-1}{2}$ joins.

- If $v \pmod 4 = 2$, the spoiler joins cliques as in the first case except the cliques $\mathbb{O}_{v-4}$, $\mathbb{O}_{v-2}$, $\mathbb{O}_{v-1}$ and $\mathbb{O}_v$. At the end, he joins $\mathbb{O}_{v-4}$ with $\mathbb{O}_{v-1}$ by an inside join and $\mathbb{O}_{v-2}$ with $\mathbb{O}_v$ by an outside join. Summing up, the spoiler performs $\frac{v}{2}$ joins.

- If $v \pmod 4 = 3$, the spoiler performs joins as in the first case, but he ignores the clique $\mathbb{O}_{v-1}$. In other words, he performs outside joins between $\mathbb{O}_i$ and $\mathbb{O}_{i+2}$ for $i = 1, 5, 9, v-2$ and inside joins between cliques $\mathbb{O}_i$ and $\mathbb{O}_{i+2}$ for $i = 2, 6, 10, v-5$. Summing up, the spoiler performs $\frac{v-1}{2}$ joins.

Note that the spoiler can perform his joins, keeping the graph plane. Moreover, every join forces $\mathcal{A}$ to use new clique to cover added vertices (vertices of the type a, b in line 1), while the number of cliques in an optimal covering is not changed. Summing up, in all 4 cases the spoiler performs $\lfloor \frac{v}{2} \rfloor$ joins and forces

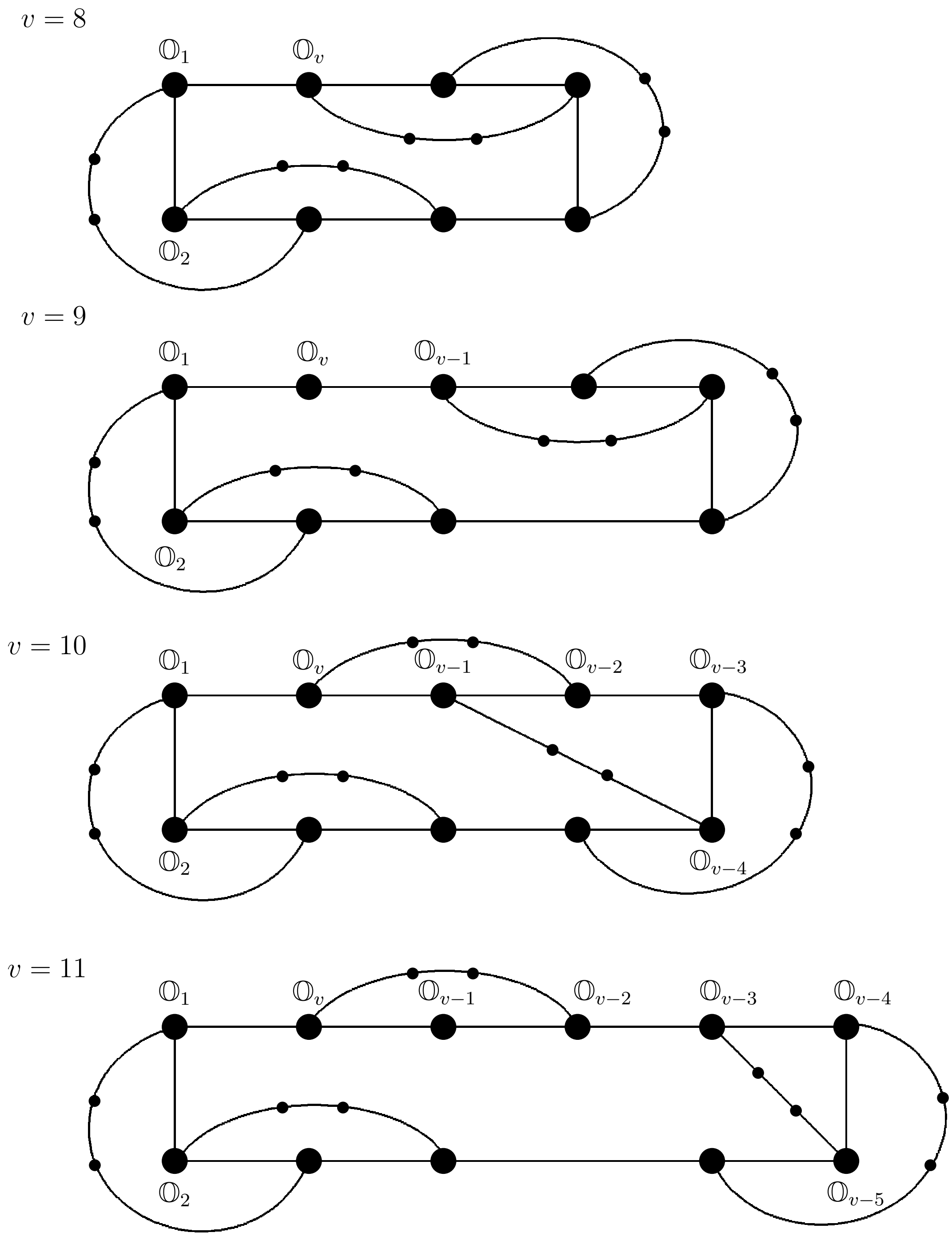


Figure 3.8: Joining cliques for $v = 8, 9, 10, 11$.

$\mathcal{A}$ to use $\lfloor \frac{v}{2} \rfloor$ additional cliques. All $\mathbb{O}_i$'s for $v \pmod 2 = 0$ and all $\mathbb{O}_i$'s but one for $v \pmod 2 = 1$ consist of 3 elements. Moreover, vertices of $\mathbb{O}_i$ are covered by $\mathcal{A}$ with $|O_i|$ different cliques and (3.1) holds. Up to this point the algorithm $\mathcal{A}$ was forced to use $\lfloor \frac{3}{2} \cdot v \rfloor$ cliques.

In the last step, the spoiler extends each clique $\mathbb{O}_i$ with one new vertex. More precisely, for each $i = 1, \ldots, v$, he adds a new vertex to $\mathbb{O}_i$ and joins it to all vertices of $\mathbb{O}_i$. Such vertices force $\mathcal{A}$ to use new cliques. Since, before this step, $\mathbb{O}_i$'s were of size at most 3, the spoiler can draw the new vertices in such a way that the resulting graph is still plane. The number of all cliques returned by $\mathcal{A}$ is $v_A(\mathbb{G}) = v + \lfloor \frac{v}{2} \rfloor + v = \lfloor \frac{5}{2} \cdot v \rfloor$. □

3.5 Conclusions

This chapter points out that there is no competitive coloring algorithm for $\mathbb{K}_s$-free graphs ($s \geqslant 3$) and, as a result, for s-colorable graphs and planar graphs even though these graphs are presented in a connected way. On the other hand, there are on-line clique covering algorithms for these families with the competitive ratios: $\frac{s}{2}$ for $\mathbb{K}_s$-free graphs and consequently $\frac{s+1}{2}$ for s-colorable graphs. This means that $\mathbb{K}_s$ is a forcing structure for clique covering. However, at the moment we are unable to calculate exactly the competitive functions for these families. As we have seen (Theorems 3.2.1 and 3.2.7, Corollary 3.3.2) they are between $\frac{s}{2} \cdot v(\mathbb{G}) - \left(\frac{2s-3}{4}\right)^2$ and $\frac{s}{2} \cdot v(\mathbb{G})$ for $\mathbb{K}_s$-free graphs and between $\frac{s+1}{2} \cdot v(\mathbb{G}) - \left(\frac{2s-1}{4}\right)^2$ and $\frac{s+1}{2} \cdot v(\mathbb{G})$ for s-colorable graphs. Moreover, the gaps disappear for greedy algorithms (Theorem 3.2.2, Corollary 3.3.4) so that these function are $\frac{s}{2} \cdot v(\mathbb{G})$ and $\frac{s+1}{2} \cdot v(\mathbb{G})$, respectively.

Problem 1 *Is there an on-line clique covering algorithm for $\mathbb{K}_s$-free graphs with a competitive function better than $\frac{s}{2} \cdot v(\mathbb{G})$?*

Problem 2 *Is there an on-line clique covering algorithm for s-colorable graphs with a competitive function better than $\frac{s+1}{2} \cdot v(\mathbb{G})$?*

In this chapter, we have also looked for a competitive ratio of clique covering for planar graphs. We have seen that it is between 2 and $\frac{5}{2}$. This leads to the following.

Problem 3 *Is there an on-line clique covering algorithm for planar graphs presented in a connected way with a competitive ratio lower than $\frac{5}{2}$?*

Chapter 4

$\mathbb{K}_{s,t}$-free graphs

One of the most often used method to force many colors while coloring a graph is to form special configurations. In particular, in the proof of Theorem 2.1.1, we have seen that $\mathbb{K}_{1,t}$ is such a forcing configuration, when the unique vertex connected to all other t vertices of $\mathbb{K}_{1,t}$ arrives. This chapter is devoted to analyze the role and forcing power of $\mathbb{K}_{1,t}$ as well as complete bipartite graphs $\mathbb{K}_{s,t}$. The results are summarized in Table 4.1.

4.1 On-line coloring of $\mathbb{K}_{s,t}$-free graphs

In this section we check for which s and t the graphs $\mathbb{K}_{s,t}$ are forcing structures for the coloring problem. The on-line algorithm using the First Fit strategy for these graphs has been analyzed by A.Capponi and C.Pilotto [7]. They proved that if $\mathbb{G}$ is $\mathbb{K}_{1,t}$-free ($t \geqslant 2$), then $\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant (t-1) \cdot \chi(\mathbb{G})$. After a more careful analysis of their proof we obtain the following.

Proposition 4.1.1 *Let $t \geqslant 2$. If $\mathbb{G}$ is $\mathbb{K}_{1,t}$-free, then*

$$\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant (t-1) \cdot \chi(\mathbb{G}) - t + 2.$$

Proof. First, observe that the only way for the spoiler to force any greedy algorithm (in particular, the $\mathcal{FF}$ one) to introduce $(d+1)$-th color is to present a new point x connected to d vertices $x_1, \ldots, x_d$ with $c_{\mathcal{FF}}(x_i) = i$. Thus, the degree of this new vertex x is equal to at least d and consequently

$$\chi_{\mathcal{FF}}(\mathbb{H}) \leqslant \triangle(\mathbb{H}) + 1 \text{ for every graph } \mathbb{H}.$$

On the other hand, consider an optimal coloring of a $\mathbb{K}_{1,t}$-free graph $\mathbb{G}$ and a vertex $v \in G$ such that $deg_{\mathbb{G}}(v) = \triangle(\mathbb{G})$, i.e., v has the largest degree possible. The set $N_{\mathbb{G}}(v)$ of all neighbors of v splits into several independent sets, say $N_{\mathbb{G}}(v) = V_1 \cup \ldots \cup V_k$. Without loss of generality assume that $C(V_i) = \{i\}$ for each i. Since $\mathbb{G}$ is $\mathbb{K}_{1,t}$-free and each V_i is independent we know that $|V_i| \leqslant t-1$.

Graph $\mathbb{G}$	**Presentation method**	**Coloring**	**Clique covering**
$\mathbb{K}_{1,2}$-free	-	$opt(\mathbb{G})$ 4.1.1 [7]	$opt(\mathbb{G})$ 4.3.1
$\mathbb{K}_{1,t}$-free $t \geq 3$	-	$(t-1)opt(\mathbb{G}) - t + 2$ 4.1.1 [7] and 4.1.5	∞ 4.3.2 based on [17]
$\mathbb{K}_{1,t}$-free $t \geq 3$	connected	$(t-1)opt(\mathbb{G}) - t + 2$ 4.1.1 [7] and 4.2.1	∞ 4.3.2 based on [17]
$\mathbb{K}_{2,2}$-free $=\ \mathbb{C}_4$-free	-	∞ 4.1.7 based on [17]	$\lfloor \frac{opt(\mathbb{G})\cdot(opt(\mathbb{G})+4)}{4} \rfloor \leq \ldots \leq 2^{opt(\mathbb{G})} - 1$ 4.3.5 and [17]
			for greedy algorithms $\binom{opt(\mathbb{G})+1}{2} \leq \ldots \leq 2^{opt(\mathbb{G})} - 1$ 4.3.4 and [17]
$\mathbb{K}_{2,2}$-free $\mathbb{C}_4$-free	connected	∞ 4.2.2 based on [17]	$\lfloor \frac{opt(\mathbb{G})\cdot(opt(\mathbb{G})+4)}{4} \rfloor \leq \ldots \leq 2^{opt(\mathbb{G})} - 1$ 4.3.6 and [17]
			for greedy algorithms $\binom{opt(\mathbb{G})+1}{2} \leq \ldots \leq 2^{opt(\mathbb{G})} - 1$ 4.3.4 and [17]
$\mathbb{K}_{s,t}$-free $s \geq 2,\ t \geq 3$	-	∞ 4.1.7 based on [17]	∞ 4.3.3 based on [17]
$\mathbb{K}_{s,t}$-free $s \geq 2,\ t \geq 3$	connected	∞ 4.2.2 based on [17]	∞ 4.3.3 based on [17]

Table 4.1: Possible competitive functions for on-line problems

On the other hand, there are at least $k+1$ colors used to color $N_{\mathbb{G}}(v) \cup \{v\}$ and $\triangle(\mathbb{G}) = deg_{\mathbb{G}}(v) = |V_1| + \ldots + |V_k| \leqslant k \cdot (t-1)$. This immediately gives

$$\chi(\mathbb{G}) \geqslant k+1 \geqslant \frac{\triangle(\mathbb{G})}{t-1} + 1.$$

Combining the two displayed inequalities we get $\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant (t-1) \cdot \chi(\mathbb{G}) - t + 2$ as required. □

One aim of this section is to show that in fact the $\mathcal{FF}$ algorithm is the best on-line coloring algorithm for $\mathbb{K}_{1,t}$-free graphs and that the bound shown by Proposition 4.1.1 is tight. We start with showing that this upper bound can be achieved by the spoiler playing against $\mathcal{FF}$ for $k = 3$ (Proposition 4.1.2). Then we generalize this technique first for arbitrary $k \geqslant 3$ (Theorem 4.1.3) and then relax the restriction on the algorithms (Proposition 4.1.4 and Theorem 4.1.5).

Proposition 4.1.2 *For every positive integer χ there exists a $\mathbb{K}_{1,3}$-free graph $\mathbb{G}$ with $\chi(\mathbb{G}) = \chi$ such that*

$$\chi_{\mathcal{FF}}(\mathbb{G}) \geqslant 2 \cdot \chi(\mathbb{G}) - 1.$$

Proof. We describe a strategy for the spoiler to construct a χ-colorable $\mathbb{K}_{1,3}$-free graph $\mathbb{G}$ for which $\chi_{\mathcal{FF}}(\mathbb{G}) \geqslant 2 \cdot \chi - 1$. It proceeds as follows.

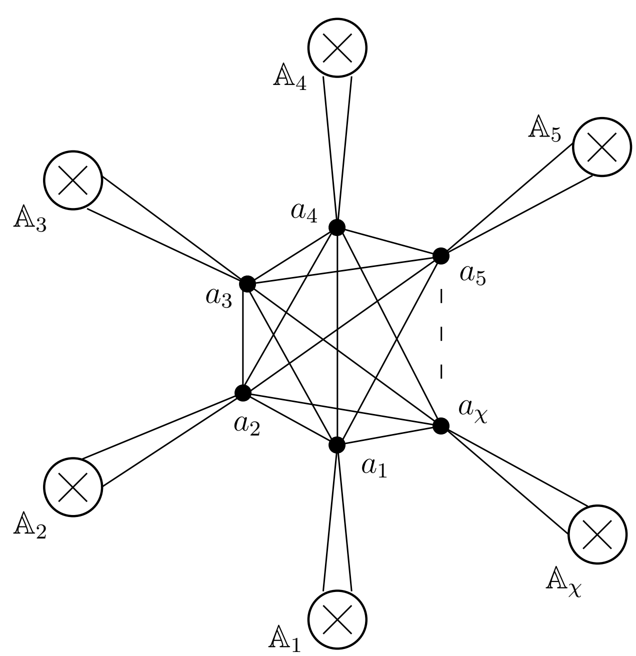


Figure 4.1: Construction of $\mathbb{G}$ with $\chi_{\mathcal{FF}}(\mathbb{G}) \geqslant 2 \cdot \chi(\mathbb{G}) - 1$.

- First, the spoiler constructs χ separated cliques $\mathbb{A}_1, \ldots, \mathbb{A}_\chi$, each with $\chi - 1$ vertices. The $\mathcal{FF}$ algorithm assigns to them colors from the set $\{1, \ldots, \chi - 1\}$.
- Next, the spoiler presents a clique $\mathbb{B}$ with χ vertices $a_1, \ldots, a_\chi$ such that for each index $i = 1, \ldots, \chi$, the subgraph induced by $\{a_i\} \cup A_i$ is a clique with χ vertices, see Figure 4.1. These vertices receive colors from the set $\{\chi, \ldots, 2\chi - 1\}$.

The constructed graph $\mathbb{G}$ is $\mathbb{K}_{1,3}$-free, because for each vertex $a_j \in G$, a subgraph induced by $N_{\mathbb{G}}(a_j)$ can be partitioned into two cliques. Therefore $N_{\mathbb{G}}(a_j)$ contains no independent set with 3 vertices. It is easy to see that $\mathbb{G}$ is χ-colorable: first, we color the clique $\mathbb{B}$ using χ colors; next, since $A_i \cup \{a_i\}$ induces a clique with χ vertices, we color $\mathbb{A}_i$ with colors from the set $\{1, \ldots, \chi\} \setminus \{c(a_i)\}$. We have got the $\mathbb{K}_{1,3}$-free χ-colorable graph $\mathbb{G}$ for which $\chi_{\mathcal{FF}}(\mathbb{G}) \geqslant 2 \cdot \chi(\mathbb{G}) - 1$ as required. □

Modifying the technique used in the proof of Proposition 4.1.2 we show the following.

Theorem 4.1.3 *Let $t \geqslant 3$. For every positive integer χ there exists a $\mathbb{K}_{1,t}$-free graph $\mathbb{G}$ with $\chi(\mathbb{G}) = \chi$ such that*

$$\chi_{\mathcal{FF}}(\mathbb{G}) \geqslant (t-1) \cdot \chi(\mathbb{G}) - t + 2.$$

Proof. To show this lower bound we describe a way the spoiler builds a χ-colorable, $\mathbb{K}_{1,t}$-free graph $\mathbb{S}_t$ forcing $\mathcal{FF}$ to use $(t-1) \cdot \chi(\mathbb{S}_t) - t + 2$ colors. The construction recursively generalizes the one used for $\mathbb{K}_{1,3}$-free graphs described in the proof of Proposition 4.1.2. To present this recursive construction we need a notion of an amalgam $\mathbb{G}[\mathbb{H}] \times \mathbb{K}_m$ of a graph $\mathbb{G}$ with $\mathbb{K}_m$ over a subgraph $\mathbb{H}$ of $\mathbb{G}$. Its vertex set consists of $|G| \cdot m + m$ points, say

$$V = (G \times \{1, \ldots, m\}) \cup \{1, \ldots, m\},$$

while the edges are given by

$$\begin{aligned} E = \quad & \{\{(a,i),(b,i)\} : i = 1, \ldots, m \text{ and } a, b \in G,\ aE_{\mathbb{G}}\, b\} \\ \cup \quad & \{\{i,j\} : i,j = 1, \ldots, m \text{ and } i \neq j\} \\ \cup \quad & \{\{(a,i), i\} : i = 1, \ldots, m \text{ and } a \in H\}. \end{aligned}$$

While building an amalgam $\mathbb{G}[\mathbb{H}] \times \mathbb{K}_m$ the spoiler first presents m copies of $\mathbb{G}$ and then the vertices of the clique $\mathbb{K}_m$. Note that $\mathbb{S}_3 := \mathbb{K}_{\chi-1}[\mathbb{K}_{\chi-1}] \times \mathbb{K}_\chi$ is the graph produced in the proof of Proposition 4.1.2. In our recursive procedure we will also need $\mathbb{S}'_3 := \mathbb{K}_{\chi-1}[\mathbb{K}_{\chi-1}] \times \mathbb{K}_{\chi-1}$. The graph $\mathbb{S}_4$ is now defined to be $(\mathbb{S}'_3 \cup \mathbb{K}_{\chi-1})[\mathbb{T}'_3 \cup \mathbb{K}_{\chi-1}] \times \mathbb{K}_\chi$, where

- $\mathbb{S}'_3 \cup \mathbb{K}_{\chi-1}$ is a disjoint sum of the graphs $\mathbb{S}'_3$ and $\mathbb{K}_{\chi-1}$,
- $\mathbb{T}'_3$ is the clique consisting of the last $\chi - 1$ vertices the spoiler used to build $\mathbb{S}'_3$,
- and finally $\mathbb{T}'_3 \cup \mathbb{K}_{\chi-1}$ denotes the disjoint sum of $\mathbb{T}'_3$ sitting in $\mathbb{S}'_3$ and $\mathbb{K}_{\chi-1}$ treated as an induced subgraph of the second summand of $\mathbb{S}'_3 \cup \mathbb{K}_{\chi-1}$.

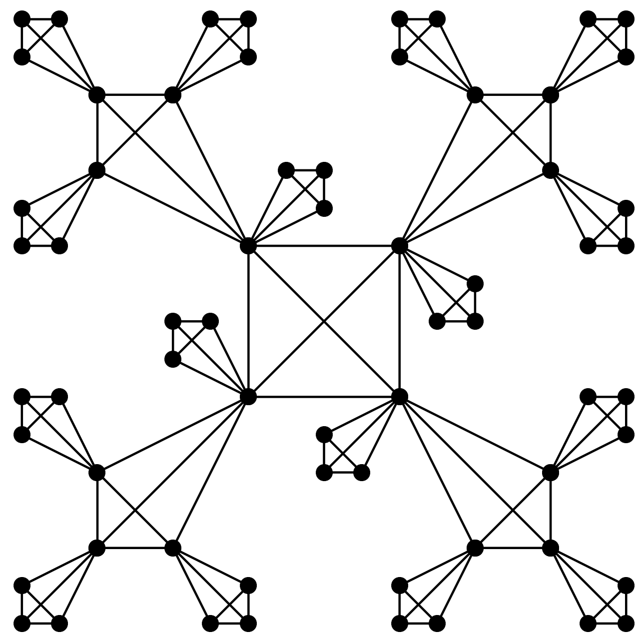


Figure 4.2: Graph $\mathbb{S}_4$ with $\chi(\mathbb{S}_4) = 4$.

Figure 4.2 presents the graph $\mathbb{S}_4$ for $\chi = 4$. It should be clear how a very similar graph $\mathbb{S}'_4 := (\mathbb{S}'_3 \cup \mathbb{K}_{\chi-1}) [\mathbb{T}'_3 \cup \mathbb{K}_{\chi-1}] \times \mathbb{K}_{\chi-1}$ is built. Summing up, the spoiler starts with

$$\mathbb{S}'_2 := \mathbb{K}_{\chi-1},$$

$$\mathbb{T}'_2 := \mathbb{S}'_2$$

and proceeds recursively by putting

$$\mathbb{S}_{t+1} := \quad (\mathbb{S}'_t \cup \ldots \cup \mathbb{S}'_2) [\mathbb{T}'_t \cup \ldots \cup \mathbb{T}'_2] \times \mathbb{K}_{\chi},$$

$$\mathbb{S}'_{t+1} := \quad (\mathbb{S}'_t \cup \ldots \cup \mathbb{S}'_2) [\mathbb{T}'_t \cup \ldots \cup \mathbb{T}'_2] \times \mathbb{K}_{\chi-1},$$

where $\mathbb{T}'_i$ is an induced subgraph of $\mathbb{S}'_i$ consisting of the last $\chi - 1$ vertices the spoiler used building $\mathbb{S}'_i$. Note that $\mathbb{T}'_i$ is itself a clique. To check that $\chi(\mathbb{S}_t) = \chi$ we will need the following

$$\text{if } \chi(\mathbb{G}) > |H| \text{ then } \chi(\mathbb{G}[\mathbb{H}] \times \mathbb{K}_1) = \chi(\mathbb{G}) \tag{4.1}$$

Our assumption ensures us that coloring $\mathbb{G}$ with $\chi(\mathbb{G})$ colors, one of them, say c, is not used to color the vertices of $\mathbb{H}$. Therefore, coloring the vertex of $\mathbb{K}_1$ with c, we end up using $\chi(\mathbb{G})$ colors on $\mathbb{G}[\mathbb{H}] \times \mathbb{K}_1$.

$$\chi(\mathbb{G}[\mathbb{H}] \times \mathbb{K}_m) = \max\{\chi(\mathbb{G}[\mathbb{H}] \times \mathbb{K}_1), m\} \tag{4.2}$$

Indeed $\mathbb{G}[\mathbb{H}] \times \mathbb{K}_m$ consists of m disjoint copies of $\mathbb{G}[\mathbb{H}] \times \mathbb{K}_1$ with extra edges connecting all vertices from m copies of $\mathbb{K}_1$. Obviously, for any vertex v of an arbitrary graph $\mathbb{F}$, each coloring of v can be extended to an optimal coloring of $\mathbb{F}$ (with $\chi(\mathbb{F})$ colors). This means that the coloring of $\mathbb{K}_m$ with m colors can be extended in each copy of $\mathbb{G}[\mathbb{H}] \times \mathbb{K}_1$ to the coloring of $\mathbb{G}[\mathbb{H}] \times \mathbb{K}_m$ using only $\chi(\mathbb{G}[\mathbb{H}] \times \mathbb{K}_1)$ colors.

Now we induct on t to show that $\mathbb{S}'_t$ and $\mathbb{S}_t$ are χ-colorable. From the proof of Proposition 4.1.2 we know that the graph $\mathbb{S}_3$ (and therefore its subgraph $\mathbb{S}'_3$) is χ-colorable. On the other hand, for $t \geqslant 3$ the graph $\mathbb{S}'_t$ (and also $\mathbb{S}_t$) contains a clique with χ elements ($\mathbb{S}'_{t-1}[\mathbb{T}'_{t-1}] \times \mathbb{K}_1$) and $\chi(\mathbb{S}'_{t-1}) = \chi$, while $\mathbb{T}'_{t-1}$ have only $\chi - 1$ elements. This, together with (4.1) and (4.2), gives that $\chi(\mathbb{S}'_t) = \chi$ and $\chi(\mathbb{S}_t) = \chi$.

Despite the fact that $\mathbb{S}_t$ (or $\mathbb{S}'_t$) is χ-colorable, the order the spoiler presents the graph to be colored forces the $\mathcal{FF}$ algorithm to use $(t-1)\chi - t + 2$ (or $(t-1)\chi - t + 1$) colors. Obviously, the graph $\mathbb{S}'_2$ has to be colored with $\chi - 1$ colors. Now forming $\mathbb{S}_3 = \mathbb{S}'_2[\mathbb{T}'_2] \times \mathbb{K}_\chi$ (or $\mathbb{S}'_3 = \mathbb{S}'_2[\mathbb{T}'_2] \times \mathbb{K}_{\chi-1}$) each of the vertices of $\mathbb{K}_\chi$ (or $\mathbb{K}_{\chi-1}$) has to be colored with a new color, so that $\mathcal{FF}$ has to use $(\chi-1)+\chi$ (or $(\chi-1)+(\chi-1)$) colors on $\mathbb{S}_3$ (or $\mathbb{S}'_3$), as claimed for $t=3$. Now, an obvious induction on t shows that $\mathcal{FF}$ is forced by the spoiler to use $(t-1)\cdot\chi(\mathbb{S}_t) - t + 2$ colors on $\mathbb{S}_t$.

Finally we have to show that each $\mathbb{S}_t$ is $\mathbb{K}_{1,t}$-free. This can be done by observing that for each vertex $v \in S_t$ the set $N_{\mathbb{S}_t}(v) = \{w : wE_{\mathbb{S}_t}v\}$ can be partitioned into at most $t-1$ cliques. □

Theorem 4.1.3 shows that the upper bound for the competitiveness of on-line coloring algorithms for $\mathbb{K}_{1,t}$-free graphs (presented in Proposition 4.1.1) is the lower bound for the $\mathcal{FF}$ algorithm. Further modification of our technique shows that the upper bound shown in Proposition 4.1.1 is also tight for every kind of algorithms.

Proposition 4.1.4 *Every on-line coloring algorithm $\mathcal{A}$ can be forced to use*

$$\chi_{\mathcal{A}}(\mathbb{G}) \geqslant 2\cdot\chi(\mathbb{G}) - 1$$

colors on $\mathbb{K}_{1,3}$-free graphs $\mathbb{G}$ with arbitrarily large $\chi(\mathbb{G})$.

Proof. For every positive integer χ we show a way the spoiler constructs a $\mathbb{K}_{1,3}$-free graph $\mathbb{G}$ with $\chi(\mathbb{G}) = \chi$ for which $\chi_{\mathcal{A}}(\mathbb{G}) \geqslant 2\cdot\chi(\mathbb{G}) - 1$. First, the spoiler generates the graph $\mathbb{G} := \mathbb{A}_1 \cup \ldots \cup \mathbb{A}_d$ consisting of as many $(\chi-1)$–element disjoint cliques $\mathbb{A}_1, \ldots, \mathbb{A}_d$ as is needed to force the algorithm $\mathcal{A}$

- either to use at least $2\chi - 1$ colors on $\mathbb{G}$
- or to color at least χ of those cliques, say $\mathbb{A}_{i_1}, \ldots, \mathbb{A}_{i_\chi}$ by using, all together, only $\chi - 1$ colors, i.e., $|C_{\mathcal{A}}(\mathbb{A}_{i_1} \cup \ldots \cup \mathbb{A}_{i_\chi})| = \chi - 1$.

Note that the pigeon-hole principle ensures us that $d = \binom{2\chi-1}{\chi-1}(\chi - 1) + 1$ cliques is sufficient to get what is required above.
If still less than $2\chi - 1$ colors are used on those cliques $\mathbb{A}_1, \ldots, \mathbb{A}_d$, the spoiler chooses $\mathbb{H} = \mathbb{A}_{i_1} \cup \ldots \cup \mathbb{A}_{i_\chi}$ with $|\, C_{\mathcal{A}}(\mathbb{A}_{i_1} \cup \ldots \cup \mathbb{A}_{i_\chi})| = \chi - 1$ and presents clique $\mathbb{K}_\chi$ in order to build $\mathbb{G}'$ (see the proof of Theorem 4.1.3 for the definition of an amalgam $\mathbb{G}\,[\mathbb{H}] \times \mathbb{K}_m$):

$$\mathbb{G}' := (\mathbb{K}_{\chi-1}\,[\mathbb{K}_{\chi-1}] \times \mathbb{K}_\chi) \cup (\mathbb{G} \setminus H).$$

Obviously, the algorithm $\mathcal{A}$ has to use χ new colors to color the last χ vertices of this new clique $\mathbb{K}_\chi$, witnessing that

$$\chi_{\mathcal{A}}(\mathbb{G}') \geqslant 2\chi - 1.$$

The graph $\mathbb{G}'$ is a disjoint sum of $\mathbb{K}_{\chi-1}\,[\mathbb{K}_{\chi-1}] \times \mathbb{K}_\chi$ and some number of $\mathbb{K}_{\chi-1}$ cliques. The graph $\mathbb{K}_{\chi-1}\,[\mathbb{K}_{\chi-1}] \times \mathbb{K}_\chi$ has been shown to be $\mathbb{K}_{1,3}$-free and χ-colorable in the proof of Proposition 4.1.2. This, together with the obvious fact that the cliques $\mathbb{K}_{\chi-1}$ are $\mathbb{K}_{1,3}$-free and χ-colorable, gives that $\mathbb{G}'$ is $\mathbb{K}_{1,3}$-free and $\chi(\mathbb{G}') = \chi$ as claimed. □

Theorem 4.1.5 *Let $t \geqslant 3$. Every on-line coloring algorithm $\mathcal{A}$ can be forced to use*

$$\chi_{\mathcal{A}}(\mathbb{G}) \geqslant (t-1) \cdot \chi(\mathbb{G}) - t + 2$$

colors on $\mathbb{K}_{1,t}$-free graphs $\mathbb{G}$ with arbitrarily large $\chi(\mathbb{G})$.

Proof. For every positive integer χ we describe a way the spoiler builds a χ-colorable $\mathbb{K}_{1,t}$-free graph $\mathbb{G}$ that forces $\mathcal{A}$ to use $(t-1) \cdot \chi(\mathbb{G}) - t + 2$ colors. The construction generalizes the one described in the proof of Proposition 4.1.4 and is similar to the one used in the proof of Theorem 4.1.3. We follow the notation of the proof of Theorem 4.1.3. In particular, we need amalgams of the form $\mathbb{G}\,[\mathbb{H}] \times \mathbb{K}_m$ and graphs $\mathbb{S}'_2, \ldots, \mathbb{S}'_t$, $\mathbb{S}_t$ defined as follows:

$$\mathbb{S}'_2 := \mathbb{K}_{\chi-1},$$

$$\mathbb{S}'_j := (\mathbb{S}'_{j-1} \cup \ldots \cup \mathbb{S}'_2)\,[\mathbb{T}'_{j-1} \cup \ldots \cup \mathbb{T}'_2] \times \mathbb{K}_{\chi-1} \quad \text{for } j = 3, \ldots, t,$$

$$\mathbb{S}_t := (\mathbb{S}'_{t-1} \cup \ldots \cup \mathbb{S}'_2)\,[\mathbb{T}'_{t-1} \cup \ldots \cup \mathbb{T}'_2] \times \mathbb{K}_\chi,$$

where $\mathbb{T}'_i$ is the induced subgraph of $\mathbb{S}'_i$ consisting of the last $\chi - 1$ vertices the spoiler used building $\mathbb{S}'_i$. Note that $\mathbb{T}'_i$ is itself a clique. To force the algorithm $\mathcal{A}$ to use $(t-1) \cdot \chi(\mathbb{G}) - t + 2$ colors, the spoiler produces the graph $\mathbb{G}$ with

$\chi(\mathbb{G}) = \chi$, by the following procedure.

procedure *STRATEGY(χ, t)*;
begin
required := $(t-1) \cdot \chi - t + 2$;
$\mathbb{G} := \emptyset$;
$\mathbb{GC} := \emptyset$; {$\mathbb{GC}$ is going to be a "garbage can", i.e., it is going to include graphs produced by the spoiler, but not needed in the following construction.}

{The spoiler produces families $\mathcal{FS}_i$ consisting of $d_{i,t}$ graphs of the form $\mathbb{S}'_i$ colored in the same way.}
CLIQUES(χ, t);
for $i = 3, \ldots, t-1$ **do** *PRODUCE(χ, i, t)*;

$\mathbb{S}_t := (\mathbb{S}'_{t-1} \cup \ldots \cup \mathbb{S}'_2)\,[\mathbb{T}'_{t-1} \cup \ldots \cup \mathbb{T}'_2] \times \mathbb{K}_\chi$;
$\mathbb{G} := \mathbb{G} \cup \mathbb{K}_\chi$; {$\mathbb{K}_\chi$ consists of χ new vertices presented in order to build $\mathbb{S}_t$.}
end;

While building an amalgam $(\mathbb{S}'_i \cup \ldots \cup \mathbb{S}'_2)\,[\mathbb{H}] \times \mathbb{K}_m$ first, the spoiler chooses m graphs from $\mathcal{FS}_i, \ldots, \mathcal{FS}_2$ isomorphic to $\mathbb{S}'_i, \ldots, \mathbb{S}'_2$, respectively; removes them from $\mathcal{FS}_i, \ldots, \mathcal{FS}_2$ and then adds the vertices of the clique $\mathbb{K}_m$. Procedures *CLIQUES(χ, t)* and *PRODUCE(χ, i, t)* are defined as follows. The numbers $d_{i,t}$ for $i = 2, \ldots, t-1$ used in these procedures denote the numbers of graphs isomorphic with $\mathbb{S}'_i$ and colored in the same way that suffices to produce the resulting graph $\mathbb{S}_t$ by the procedure *STRATEGY*. They will be calculated afterwards.

procedure *CLIQUES(χ, t)*;
begin
$\mathcal{FS}_2 := \emptyset$;
{$\mathcal{FS}_2$ is going to be the family of separated cliques (graphs of the form $\mathbb{S}'_2$).}
$d := d_{2,t}$;
while $|C_\mathcal{A}(\mathcal{FS}_2)| <$ required **and**
  $\max\{k : \exists\, \mathbb{A}_1, \ldots, \mathbb{A}_k \in \mathcal{FS}_2 : |C_\mathcal{A}(\mathbb{A}_1 \cup \ldots \cup \mathbb{A}_k)| = \chi - 1\} < d$ **do**
  {the maximum number of cliques from $\mathcal{FS}_2$,
  on which the algorithm $\mathcal{A}$ uses only $\chi - 1$ colors all together}
  begin
  $\mathbb{P} := \mathbb{K}_{\chi-1}$;
  $\mathbb{G} := \mathbb{G} \cup \mathbb{P}$;
  $\mathcal{FS}_2 := \mathcal{FS}_2 \cup \{\mathbb{P}\}$;
  WAIT for a response of $\mathcal{A}$ to color $\mathbb{P}$;
  end;
if $|C_\mathcal{A}(\mathcal{FS}_2)| \geqslant$ required **then halt**;
choose $\mathbb{A}_{i_1}, \ldots, \mathbb{A}_{i_d} \in \mathcal{FS}_2$ **such that** $|C_\mathcal{A}(\mathbb{A}_{i_1} \cup \ldots \cup \mathbb{A}_{i_d})| = \chi - 1$;

$\mathbb{GC} := \mathbb{GC} \cup \bigcup(\mathcal{FS}_2 \setminus \{\,\mathbb{A}_{i_1}, \ldots, \mathbb{A}_{i_d}\});$
$\mathcal{FS}_2 := \{\,\mathbb{A}_{i_1}, \ldots, \mathbb{A}_{i_d}\};$
end;

procedure *PRODUCE*(χ, i, t);
begin
$\mathcal{FS}_i := \emptyset;$
$\{\mathcal{FS}_i$ is going to be the family of graphs of the form $\mathbb{S}'_i$.$\}$
$d := d_{i,t};$
while $|C_{\mathcal{A}}(\mathcal{FS}_i)| <$ required **and**
$\max\{k : \ \exists\, \mathbb{A}_1, \ldots, \mathbb{A}_k \in \mathcal{FS}_i : |C_{\mathcal{A}}(\mathbb{A}_1 \cup \ldots \cup \mathbb{A}_k)| = (i-1)(\chi-1)\} < d$ **do**
{the maximum number of graphs from $\mathcal{FS}_i$, on which
the algorithm $\mathcal{A}$ uses only $(i-1)(\chi-1)$ colors all together}
begin
$\mathbb{P} := (\mathbb{S}'_{i-1} \cup \ldots \cup \mathbb{S}'_2)\,[\mathbb{T}'_{i-1} \cup \ldots \cup \mathbb{T}'_2] \times \mathbb{K}_{\chi-1};$
$\mathbb{G} := \mathbb{G} \cup \mathbb{K}_{\chi-1};$
$\{K_{\chi-1}$ consists of $\chi - 1$ new vertices presented in order to build $\mathbb{P}$.$\}$
$\mathcal{FS}_i := \mathcal{FS}_i \cup \{\mathbb{P}\};$
WAIT for a response of $\mathcal{A}$ to color $\mathbb{P}$;
end;
if $|C_{\mathcal{A}}(\mathcal{FS}_i)| \geqslant$ required **then halt**;
choose $\mathbb{A}_{i_1}, \ldots, \mathbb{A}_{i_d} \in \mathcal{FS}_i$ **such that** $|C_{\mathcal{A}}(\mathbb{A}_{i_1} \cup \ldots \cup \mathbb{A}_{i_d})| = (i-1)(\chi-1);$
$\mathbb{GC} := \mathbb{GC} \cup \bigcup(\mathcal{FS}_i \setminus \{\,\mathbb{A}_{i_1}, \ldots, \mathbb{A}_{i_d}\});$
$\mathcal{FS}_i := \{\mathbb{A}_{i_1}, \ldots, \mathbb{A}_{i_d}\};$
end;

To estimate the numbers $d_{i,t}$'s we use the pigeon-hole principle. To construct the graph $\mathbb{S}_t$ the spoiler needs χ graphs isomorphic to $\mathbb{S}'_{t-1}$ - all colored in the same way, χ graphs isomorphic to $\mathbb{S}'_{t-2}$ - all colored in the same way, etc. Therefore $d_{t-1,t} = \chi$. To construct each $\mathbb{S}'_{t-1}$ the spoiler needs $\chi - 1$ graphs isomorphic to $\mathbb{S}'_{t-2}$ - all colored in the same way. Therefore the pigeon-hole principle yields the number of graphs isomorphic to $\mathbb{S}'_{t-2}$ sufficient to get $\mathbb{S}_t$ to be

$$\begin{aligned} d_{t-2,t} &= \binom{2\chi-1}{\chi-1} \cdot (d_{t-1,t} \cdot (\chi-1) + \chi - 1) + 1 \\ &= \binom{2\chi-1}{\chi-1} \cdot (\chi-1) \cdot (d_{t-1,t} + 1) + 1. \end{aligned}$$

By the same token one can calculate that

$$d_{t-j,t} = \binom{j\chi - j + 1}{\chi - 1} \cdot (\chi-1) \cdot (d_{t-j+1,t} + \ldots d_{t-1,t} + 1) + 1$$

for $j = 2, \ldots, t-2$.

Now, note that after the performance of the procedure *PRODUCE*(χ, i, t) the following holds for each graph $\mathbb{P}$ from the family $\mathcal{FS}_i$, where $2 \leqslant i < t$.

- $C_{\mathcal{A}}(\mathbb{P}) = C_{\mathcal{A}}(\mathcal{FS}_i)$ and
- $|C_{\mathcal{A}}(\mathbb{K}_{\chi-1})| = \chi - 1$, where $\mathbb{K}_{\chi-1} \subseteq \mathbb{P}$ is a clique consisting of the last $\chi - 1$ vertices the spoiler used building the graph $\mathbb{P}$.

Building subgraphs of the graph $\mathbb{G}$ in the above way guarantees that new vertices presented during the construction of $\mathbb{S}_t$ receive from the algorithm $\mathcal{A}$ colors outside the set $C_{\mathcal{A}}(\mathcal{FS}_2 \cup \ldots \cup \mathcal{FS}_{t-1})$, so that

$$\begin{aligned} \chi_{\mathcal{A}}(\mathbb{S}_t) &= \chi + |C_{\mathcal{A}}(\mathcal{FS}_2 \cup \ldots \cup \mathcal{FS}_{t-1})| = \chi + (t-2)(\chi - 1) \\ &= (t-1)\chi - t + 2 \end{aligned}$$

and

$$\chi_{\mathcal{A}}(\mathbb{G}) = \begin{cases} (t-1) \cdot \chi - t + 2, & \text{if } \textit{CLIQUES} \text{ or } \textit{PRODUCE} \\ & \text{had halted,} \\ \chi_{\mathcal{A}}(\mathbb{S}_t \cup \mathbb{GC}) \geqslant (t-1)\chi - t + 2, & \text{otherwise.} \end{cases}$$

Finally, we show that the resulting graph $\mathbb{G}$ is exactly χ-colorable and $\mathbb{K}_{1,t}$-free. This can be done by observing that the graph $\mathbb{S}_t$ is exactly χ-colorable and $\mathbb{K}_{1,t}$-free for the very same reason as in the proof of Theorem 4.1.3 and the graph $\mathbb{GC}$ consists of disjoint subgraphs of the form $\mathbb{S}'_{t'}$ for various $t' < t$. Note that we can obtain an optimal coloring of the graph $\mathbb{G}$ by coloring vertices of $\mathbb{G}$ in the order reverse to the one in which vertices are presented by the procedure *STRATEGY(χ, t).* □

Combining Proposition 4.1.1 with Theorem 4.1.5 we get

Corollary 4.1.6 *Let $t \geqslant 3$. The on-line coloring algorithm with the First-Fit strategy for $\mathbb{K}_{1,t}$-free graphs is the best possible and it has the competitive function $(t-1) \cdot \chi(\mathbb{G}) - t + 2$, where $\mathbb{G}$ is an input graph.* □

From Corollary 4.1.6 we know that the graphs $\mathbb{K}_{1,t}$ for $t \geqslant 3$ are the forcing structures for on-line coloring problem. On the other hand, relaxing $\mathbb{K}_{1,t}$ to $\mathbb{K}_{s,t}$ (with $s, t \geqslant 2$) does not prevent the spoiler from destroying any on-line coloring algorithms. Indeed, Corollary 2.1.2 says that there is no on-line coloring algorithm for trees with a competitive function. Since each tree is $\mathbb{K}_{2,2}$-free we get

Corollary 4.1.7 *Let $s, t \geqslant 2$. There is no on-line coloring algorithm with a competitive function for $\mathbb{K}_{s,t}$-free graphs.* □

4.2 On-line coloring of $\mathbb{K}_{s,t}$-free graphs presented in a connected way

In section 4.1 we have seen that the on-line coloring algorithm $\mathcal{FF}$ for $\mathbb{K}_{1,t}$-free graphs has the competitive ratio $t-1$ and that there exists no competitive on-line coloring algorithm for $\mathbb{K}_{s,t}$-free graphs $(s, t \geqslant 2)$. In this section we show that this situation does not change even if these graphs are supposed to be presented in a connected way.

Theorem 4.2.1 *Let $t \geqslant 3$. Every on-line coloring algorithm $\mathcal{A}$ can be forced to use*

$$\chi_{\mathcal{A}}(\mathbb{G}) \geqslant (t-1) \cdot \chi(\mathbb{G}) - t + 2$$

colors on $\mathbb{K}_{1,t}$-free graphs $\mathbb{G}$ presented in a connected way with arbitrarily large $\chi = \chi(\mathbb{G})$.

Proof. To prove this theorem we modify spoiler's strategy from the proof of Theorem 4.1.5. The only change is to replace procedure *CLIQUES(χ, t)* by its slight modification *CON_CLIQUES(χ, t)*.

procedure *CON_CLIQUES(χ, t)*;
begin
$\mathbb{P}_1 := \mathbb{K}_{\chi-1}$;
$\mathbb{G} := \mathbb{P}_1$;
$\mathcal{FS}_2 := \{P_1\}$; {$\mathcal{FS}_2$ is going to be the family of cliques}
WAIT for a response of $\mathcal{A}$ to color $\mathbb{P}_1$;
$d := d_{2,t}$;
$i := 2$;
while $|C_{\mathcal{A}}(\mathcal{FS}_2)| <$ required **and**
 $\max\{k : \exists\, \mathbb{A}_1, \ldots, \mathbb{A}_k \in \mathcal{FS}_2 : |C_{\mathcal{A}}(\mathbb{A}_1 \cup \ldots \cup \mathbb{A}_k)| = \chi - 1\} < d$ **do**
 {the maximum number of cliques from $\mathcal{FS}_2$,
 on which the algorithm $\mathcal{A}$ uses only $\chi - 1$ colors all together}
 begin
 $\mathbb{P}_i := \mathbb{K}_{\chi-1}$;
 $\mathbb{G} := (\, G \cup P_i,\ E_{\mathbb{G}} \cup E_{\mathbb{P}_i} \ \cup\ \{xy :\ x$ is the last vertex of $\mathbb{P}_{i-1}$,
 y is the first vertex of $\mathbb{P}_i\}\ )$;
 $\mathcal{FS}_2 := \mathcal{FS}_2 \cup \{\mathbb{P}_i\}$;
 $i := i + 1$;
 WAIT for a response of $\mathcal{A}$ to color $\mathbb{P}_i$;
 end;
if $|C_{\mathcal{A}}(\mathcal{FS}_2)| \geqslant$ required **then halt**;
choose $\mathbb{A}_{i_1}, \ldots, \mathbb{A}_{i_d} \in \mathcal{FS}_2$ **such that** $|C_{\mathcal{A}}(\mathbb{A}_{i_1} \cup \ldots \cup \mathbb{A}_{i_d})| = \chi - 1$;
$\mathbb{GC} := \mathbb{GC} \cup \bigcup(\mathcal{FS}_2 \setminus \{\mathbb{A}_{i_1}, \ldots, \mathbb{A}_{i_d}\})$;
$\mathcal{FS}_2 := \{\mathbb{A}_{i_1}, \ldots, \mathbb{A}_{i_d}\}$;
end;

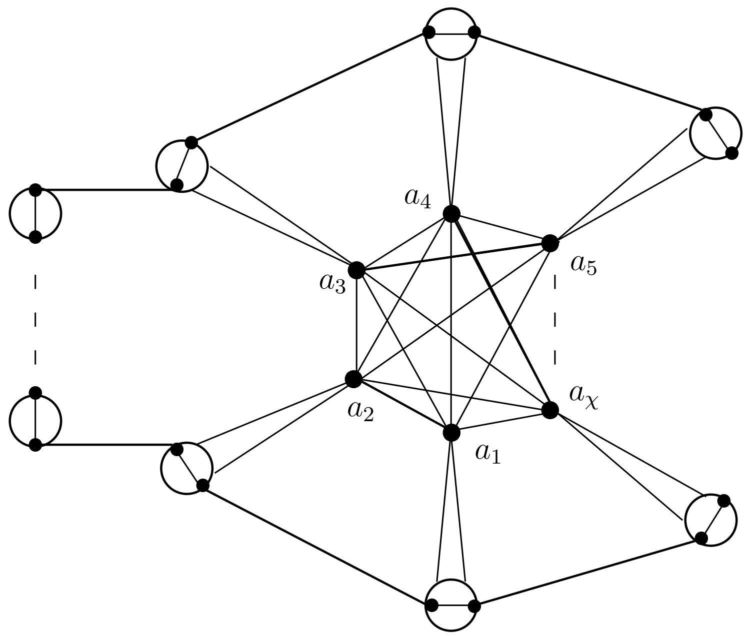


Figure 4.3: The $\mathbb{K}_{1,3}$-free graph $\mathbb{G}$ presented in a connected way

The additional edges used in line 8 to join the new clique with the one previously created, guarantee that the graph $\mathbb{G}$ is connected in each step of the game. Adding these edges does not introduce subgraph of the form $\mathbb{K}_{1,t}$. Indeed, observe that for every $\mathbb{P} \in \mathcal{FS}_2$ and for every $v \in P$ the neighborhood $N_{\mathbb{G}}(v)$ contains no independent set with 3 vertices.

Finally, we have to show that the resulting graph $\mathbb{G}$ which now has more edges is still χ-colorable. Note that for every vertex v of a graph $\mathbb{F}$ each coloring of v can be extended to an optimal coloring of $\mathbb{F}$. Similarly for every edge vw of the graph $\mathbb{F}$ each coloring of v and w can be extended to an optimal coloring of $\mathbb{F}$. This means that we obtain an optimal coloring if we color vertices of $\mathbb{G}$ in the order reverse to the one in which the vertices are presented. □

The additional restriction to present an input graph in a connected way, does not help also in coloring of $\mathbb{K}_{s,t}$-free graphs for $s, t \geqslant 2$.

Proposition 4.2.2 *Let $s, t \geqslant 2$. There is no on-line coloring algorithm with a competitive function for $\mathbb{K}_{s,t}$-free graphs presented in a connected way.*

Proof. In order to prove this, it is sufficient to observe that the construction from the proof of Proposition 3.1.1 works. Indeed, the graphs used there were presented in a connected way. Moreover, they are $\mathbb{K}_{2,2}$-free ($\mathbb{C}_4$-free) as each their cycle consists of at least 6 vertices. □

4.3 On-line clique covering of $\mathbb{K}_{s,t}$-free graphs

In this section we analyze the on-line clique covering algorithms for $\mathbb{K}_{s,t}$-free graphs ($s \geqslant 1$, $t \geqslant 2$) presented in a connected way. In particular, we prove that if $s \geqslant 1$ and $t \geqslant 3$ none of them has a competitive function. Therefore the graphs $\mathbb{K}_{s,t}$ (for $s \geqslant 1$ and $t \geqslant 3$) are not forcing structures for clique covering problem.

On the other hand, $\mathbb{K}_{2,2}$ (i.e., $\mathbb{C}_4$) is a forcing structure for clique covering. The only known (see A.Gyárfás, J.Lehel [17]) upper bound for the competitive function for $\mathbb{K}_{2,2}$-free graphs is exponential and no proper lower bound was known. The problem of determining the exact competitiveness for $\mathbb{K}_{2,2}$-free graphs is still open. All we can show here, is to increase the trivial lower bound to a quadratic one.

The remaining case of $\mathbb{K}_{1,2}$-free graphs is trivial. Indeed, each $\mathbb{K}_{1,2}$-free graph is a disjoint sum of cliques.

Corollary 4.3.1 *If $\mathbb{G}$ is $\mathbb{K}_{1,2}$-free, then $v_{\mathcal{FF}}(\mathbb{G}) = v(\mathbb{G})$.* □

Proposition 4.3.2 *Let $t \geqslant 3$. There is no on-line clique covering algorithm with a competitive function for $\mathbb{K}_{1,t}$-free graphs presented in a connected way.*

Proof. We argue as in the proof of Theorem 2.4.3, that is, we use complements $\overline{\mathbb{T}}_n$ of the trees $\mathbb{T}_n$ constructed in the proof of Theorem 2.1.1. We have already seen that $\overline{\mathbb{T}}_n$ were presented in a connected way. Additionally we have to explain why the graphs $\overline{\mathbb{T}}_n$ are $\mathbb{K}_{1,3}$-free. This is true because each tree $\mathbb{T}_n$ contains no clique with 3 vertices (therefore $\overline{\mathbb{T}}_n$ contains no independent set with 3 vertices). □

Since the $\mathbb{K}_{1,t}$-free graphs are $\mathbb{K}_{s,t}$-free for $s \geqslant 1$, we immediately get:

Corollary 4.3.3 *Let $s \geqslant 1$ and $t \geqslant 3$. There is no on-line clique covering algorithm with a competitive function for $\mathbb{K}_{s,t}$-free graphs presented in a connected way.* □

The remaining, and in fact the most challenging situation is for $\mathbb{K}_{2,2}$-free graphs. A.Gyárfás and J.Lehel [17] considered $2\mathbb{K}_2$-free graphs, i.e., graphs not containing two independent edges as an induced subgraph. They showed that $\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant 2^{\omega(\mathbb{G})} - 1$ for all $2\mathbb{K}_2$-free graphs. Since $\omega(\mathbb{G}) \leqslant \chi(\mathbb{G})$ for all $\mathbb{G}$ and the complement of a $2\mathbb{K}_2$-free graph is $\mathbb{K}_{2,2}$-free we get $v_{\mathcal{FF}}(\mathbb{G}) \leqslant 2^{v(\mathbb{G})} - 1$ for $\mathbb{K}_{2,2}$-free graphs $\mathbb{G}$. Moreover, no non-trivial lower bound for the on-line clique covering algorithms was known.

Now, we show that none of the on-line clique covering algorithms has a competitive ratio. More precisely, we show that a competitive function for the clique covering algorithms is at least $\lfloor \frac{v(\mathbb{G})\cdot(v(\mathbb{G})+4)}{4} \rfloor$ whenever $\mathbb{G}$ is a $\mathbb{K}_{2,2}$-free graph. Moreover, we are able to get a better lower bound $\binom{v(\mathbb{G})+1}{2}$ for on-line greedy clique covering algorithms.

Theorem 4.3.4 *For every on-line greedy clique covering algorithm $\mathcal{A}$ and for every positive integer $v \geqslant 2$ there exists a $\mathbb{K}_{2,2}$-free graph $\mathbb{G}$ with $v(\mathbb{G}) = v$, presented in a connected way and such that $v_{\mathcal{A}}(\mathbb{G}) \geqslant \binom{v(\mathbb{G})+1}{2}$.*
In particular, there is no on-line greedy clique covering algorithm with a competitive ratio for $\mathbb{K}_{2,2}$-free graphs presented in a connected way.

Proof. For every positive integer $v \geqslant 4$ we show a way the spoiler constructs a $\mathbb{K}_{2,2}$-free graph $\mathbb{G}$ with $v(\mathbb{G}) = v$ for which $v_{\mathcal{A}}(\mathbb{G}) \geqslant \binom{v+1}{2}$. The spoiler's strategy is based on the one described in the proof of Theorem 3.4.3.

The first step of the present strategy is just the same as in the proof of Theorem 3.4.3: the spoiler produces a cycle $\mathbb{C}_{2v}$ consisting of $2v$ vertices (see Figure 4.4). The greedy algorithm $\mathcal{A}$ covers it with v cliques: $\mathbb{A}_1, \ldots, \mathbb{A}_v$ with $|A_i| = 2$. During the construction the spoiler keeps, for himself, and updates an optimal clique covering $\mathbb{O}_1, \ldots, \mathbb{O}_v$. After first step he decides that each clique consists of two vertices (see Figure 4.4) such that

$$cl_{\mathcal{A}}(x) \in cl_{\mathcal{A}}(\mathbb{G} \setminus O_i) \quad \text{whenever } x \in O_i. \tag{4.3}$$

The spoiler keeps the property (4.3) during the construction for all but the last steps. Moreover,

$$\text{for } i \neq j, \text{there is at most one edge } vw \text{ with } v \in O_i \text{ and } w \in O_j. \tag{4.4}$$

The property (4.4) is going to be kept as an invariant throughout the construction. The next steps of spoiler's strategy connect some "unjoined" $\mathbb{O}_i$ and $\mathbb{O}_j$ by the procedure *BIJOIN(i,j)* described in the proof of Theorem 3.4.3. This procedure adds two new vertices to the graph $\mathbb{G}$. Since (4.3) holds, the algorithm $\mathcal{A}$ is forced to use for these two new vertices a completely new clique. However, the action *BIJOIN(i,j)* does not increase the number of cliques in an optimal covering and does not change (4.3). It only increases sizes of $\mathbb{O}_i$ and $\mathbb{O}_j$.

To force the algorithm $\mathcal{A}$ to use $\binom{v+1}{2}$ cliques, the spoiler performs the following procedure (see Figure 4.4).

Procedure *STRATEGY(v)*;
begin
2a: **for** $j := 3, \ldots, v-1$ **do** *BIJOIN*$(1, j)$;
2b: **for** $i := 2, \ldots, v-2$ **do**
  for $j := i+2, \ldots, v$ **do** *BIJOIN*(i, j);
3: **for** $i := 1, \ldots, v$ **do**
  begin
  {the spoiler adds a new vertex a}
4: $\mathbb{G} := (\ G \cup \{a\}\ ,\ E_{\mathbb{G}} \cup \{ax : x \in O_i\}\ )$;
5: $\mathbb{O}_i := \mathbb{O}_i \cup \{a\}$;
  end;
end;

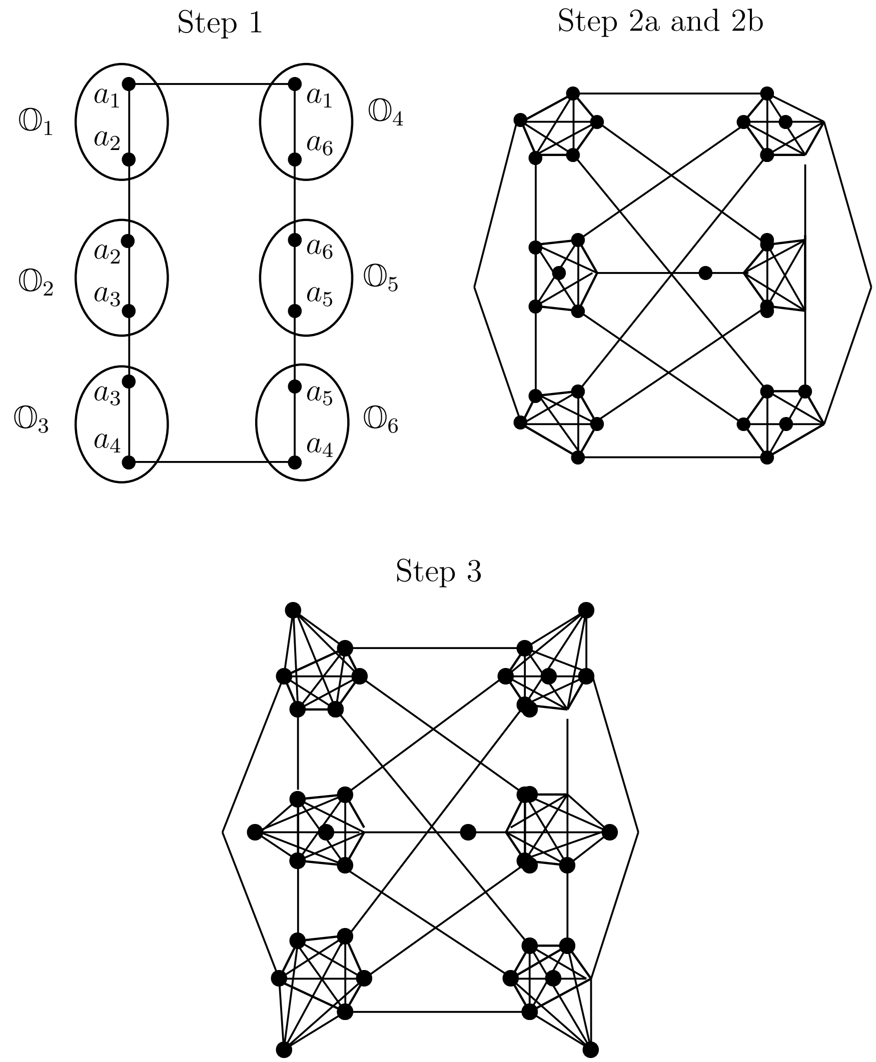


Figure 4.4: Construction of $\mathbb{G}$ with $v(\mathbb{G}) = 6$, $a_i \in A_i$.

Note that the spoilers performs $(v-3)$ joins in line 2a and $(1+2+\ldots+v-3)$ joins in line 2b. Therefore, his joins force $\mathcal{A}$ to use $(v-3) + \frac{(v-3)(v-2)}{2} = \frac{v(v-3)}{2}$ additional cliques. Up to this point, $\mathcal{A}$ was forced to use $v + \frac{v(v-3)}{2} = \frac{v(v-1)}{2}$ cliques.

In the last step (line 3) the spoiler adds one vertex to each $\mathbb{O}_i$. Such vertices force $\mathcal{A}$ to use new cliques. Summing up, the algorithm $\mathcal{A}$ was forced to use

$$v_{\mathcal{A}}(\mathbb{G}) = \frac{v(v-1)}{2} + v = \frac{v(v+1)}{2} \text{ cliques.}$$

Before we finish the proof by showing that the resulting graph $\mathbb{G}$ is $\mathbb{K}_{2,2}$-free, we observe that (4.4) is indeed an invariant. After the first step the condition (4.4) holds and for each $i < j$ there is an edge vw with $v \in O_i$ and $w \in O_j$

only if $j = i + 1$ or ($i = 1$ and $j = v$). The only way to introduce new edge vw with $v \in O_i$ and $w \in O_j$ is to perform *BIJOIN*(i, j) or *BIJOIN*(j, i). Since *BIJOIN*$(1, v)$ is not performed at all and for each i, j procedure *BIJOIN*(i, j) is performed at most once and only for $j \geqslant i + 2$, the invariant (4.4) is kept. Now, suppose that there is an induced subgraph $\mathbb{C}_4 \subseteq \mathbb{G}$ consisting of vertices x_1, x_2, x_3, x_4. Since there are two successive vertices of $\mathbb{C}_4$ in different cliques we may, without loss of generality, assume that $x_1 \in O_i$ and $x_2 \in O_j$. It is easy to see that $N_{\mathbb{G}}(x_1) = O_i \setminus \{x_1\} \cup \{x_2\}$ and $N_{\mathbb{G}}(x_2) = O_j \setminus \{x_2\} \cup \{x_1\}$. This means that $x_3 \in O_j$ and $x_4 \in O_i$. Consequently there are two edges between $\mathbb{O}_i$ and $\mathbb{O}_j$, namely x_1x_2 and x_4x_3, contrary to (4.4). □

Now, we show that also none of the on-line clique covering algorithms for $\mathbb{K}_{2,2}$-free graphs has a competitive ratio and in fact that the lower bound for a competitive function is $\lfloor \frac{v(\mathbb{G})\cdot(v(\mathbb{G})+4)}{4} \rfloor$, whenever $\mathbb{G}$ is a $\mathbb{K}_{2,2}$-free graph.

Theorem 4.3.5 *For every on-line clique covering algorithm $\mathcal{A}$ and for every positive integer $v \geqslant 2$ there exists a $\mathbb{K}_{2,2}$-free graph $\mathbb{G}$ with $v(\mathbb{G}) = v$ such that $v_{\mathcal{A}}(\mathbb{G}) \geqslant \lfloor \frac{v(\mathbb{G})\cdot(v(\mathbb{G})+4)}{4} \rfloor$.*
In particular, there is no on-line clique covering algorithm with a competitive ratio for $\mathbb{K}_{2,2}$-free graphs.

Proof. For every positive integer $v \geqslant 4$ we describe a way the spoiler constructs a $\mathbb{K}_{2,2}$-free graph $\mathbb{G}$ with $v = v(\mathbb{G})$ for which $v_{\mathcal{A}}(\mathbb{G}) \geqslant \lfloor \frac{v(v+4)}{4} \rfloor$. It is convenient to assign colors to cliques created by $\mathcal{A}$ (called $\mathcal{A}$-cliques) and talk about colors instead of cliques.

The spoiler starts with v one element cliques $\mathbb{O}_1, \ldots, \mathbb{O}_v$. In each step he keeps, for himself, an updated optimal clique covering $\mathbb{O}_1, \ldots, \mathbb{O}_v$ of the graph presented up to this point. In order to force $\mathcal{A}$ to use additional colors he uses and modifies the tool described in the proof of the Theorem 3.2.7 called a join of two cliques. In that proof a join is performed by the procedure *JOIN*, which, for two cliques $\mathbb{O}_i$ and $\mathbb{O}_j$ adds a new vertex $x \notin G$ with edges to all vertices from $O_i \cup O_j$ and possibly an additional vertex depending on the response of the algorithm $\mathcal{A}$. The procedure *JOIN* joins two disjoint cliques $\mathbb{O}_i$ and $\mathbb{O}_j$ in the three possible ways depending on algorithm behavior, that is, by classical join, wasteful join and binding join (see Theorem 3.2.7 for definitions). The new modified procedure will be called *MULTI_JOIN*. It acts in the same way as the procedure *JOIN* except for wasteful joins. If the new vertex x (added in line 2) is colored with a completely new color, the spoiler performs next join between $\mathbb{O}_i$ and $\mathbb{O}_j$, until the algorithm $\mathcal{A}$ colors new added vertex with some old one (line 7) or until $\mathcal{A}$ wastes REQUIRED $= \lfloor \frac{v(v+4)}{4} \rfloor$ colors (line 6). In procedure *MULTI_JOIN* we also use a concept of a tied clique defined in the proof of Theorem 3.2.7.

procedure *MULTI_JOIN*(i, j);
begin
if $|O_i| > |O_j|$ **then begin** *MULTI_JOIN*(j, i); exit; **end**;

repeat
$\mathbb{G}' := (G \cup \{x\}\,,\ E_{\mathbb{G}} \cup \{xy:\ y \in O_i \cup O_j\})$;
WAIT for a response of $\mathcal{A}$ to color x
if $cl_{\mathcal{A}}(x) \in Cl_{\mathcal{A}}(\mathbb{O}_i)$ **then** $\mathbb{O}_j := \mathbb{O}_j \cup \{x\}$
else $\mathbb{O}_i := \mathbb{O}_i \cup \{x\}$;
if $cl_{\mathcal{A}}(x) \geqslant$ REQUIRED **then halt**;
until $cl_{\mathcal{A}}(x) \in Cl_{\mathcal{A}}(\mathbb{O}_i)$;

if $\mathbb{O}_i$ is tied **then**
begin {the spoiler adds a new vertex z_1}
$\mathbb{G}' := (G' \cup \{z_1\}\,,\ E_{\mathbb{G}'} \cup \{z_1 y:\ y \in O_i\})$;
$\mathbb{O}_i := \mathbb{O}_i \cup \{z_1\}$;
end
else if $\mathbb{O}_j$ is tied **then**
begin {the spoiler adds a new vertex z_2}
$\mathbb{G}' := (G' \cup \{z_2\}\,,\ E_{\mathbb{G}'} \cup \{z_2 y:\ y \in O_j\})$;
$\mathbb{O}_j := \mathbb{O}_j \cup \{z_2\}$;
end;
$\mathbb{G} := \mathbb{G}'$;
end;

To present the spoiler's strategy we need a notion of a graph $\mathbb{G}^*$. Let $\mathbb{G}$ be an arbitrary graph with $v(\mathbb{G}) = v$ and let $\mathbb{O}_1, \ldots, \mathbb{O}_v$ be cliques in an optimal clique covering of $\mathbb{G}$. The graph $\mathbb{G}^*$ consists of v vertices $\mathbb{O}_1, \ldots, \mathbb{O}_v$ and edges according to the following.

$$\mathbb{O}_i\mathbb{O}_j \in E_{\mathbb{G}^*} \ \Leftrightarrow\ \exists x \in O_i\ \exists y \in O_j \ \text{ such that }\ xy \in E_{\mathbb{G}}.$$

The spoiler builds the forcing graph $\mathbb{G}$ by the following strategy.

1. First, the spoiler presents v separated vertices (one element cliques $\mathbb{O}_1, \ldots, \mathbb{O}_v$).
2. Next, he performs the procedure *MULTI_JOIN* $\lfloor \frac{v^2}{4} \rfloor$ times in order to get a graph $\mathbb{G}$ for which $\mathbb{G}^*$ is a complete bipartite graph $(V_1 \cup V_2, E)$ such that
   - $|V_1| = |V_2| = \frac{v}{2}$, if v is even,
   - $|V_1| = \frac{v+1}{2}$ and $|V_2| = \frac{v-1}{2}$, if v is odd.

This can be done, for example, by the following strategy:

procedure *STRATEGY*;
begin
for $i := 1$ **to** $\lceil \frac{v}{2} \rceil$ **do**
  for $j := 1$ **to** $\lfloor \frac{v}{2} \rfloor$ **do** *MULTI_JOIN*$(2i-1, 2j)$;
end;

Every performance of *MULTI_JOIN* stops either when $\mathcal{A}$ uses REQUIRED colors (line 6) or when $\mathcal{A}$ colors new vertex x with some old color (in other words, when some 1-vertex $\mathcal{A}$-clique becomes a 2-vertex $\mathcal{A}$-clique, line 7). The number of all 2-vertex $\mathcal{A}$-cliques is therefore at least as big as the number of performing joins. Note that *MULTI_JOIN*(i, j) is performed only for i odd and j even and that it is performed exactly one time. This ensures us that all $\mathcal{A}$-cliques have at most 2 vertices. At the same time, none of the $\mathbb{O}_i$'s is tied, therefore each $\mathbb{O}_i$ contains at least one one-element $\mathcal{A}$-clique. Summing up, the number of all $\mathcal{A}$-cliques is

$$v_{\mathcal{A}}(\mathbb{G}) \geqslant \begin{cases} \text{REQUIRED}, & \text{if some MULTI\_JOIN was halted in line 6,} \\ \lfloor \frac{v^2}{4} \rfloor + v, & \text{otherwise.} \end{cases}$$

Therefore $v_{\mathcal{A}}(\mathbb{G}) \geqslant \lfloor \frac{v(v+4)}{4} \rfloor$.

Finally, we check that $\mathbb{G}$ is $\mathbb{K}_{2,2}$-free. This follows from the fact that $\mathbb{G}^*$ is bipartite and therefore $\mathbb{K}_3$-free. Suppose otherwise, i.e., there is an induced subgraph $\mathbb{C}_4 \subseteq \mathbb{G}$ with vertices x, y, z, u. There are three cases (see Figure 4.5):

1. $\mathbb{C}_4$ is contained in 2 cliques: $\mathbb{C}_4 \subseteq \mathbb{O}_{i_1} \cup \mathbb{O}_{i_2}$.

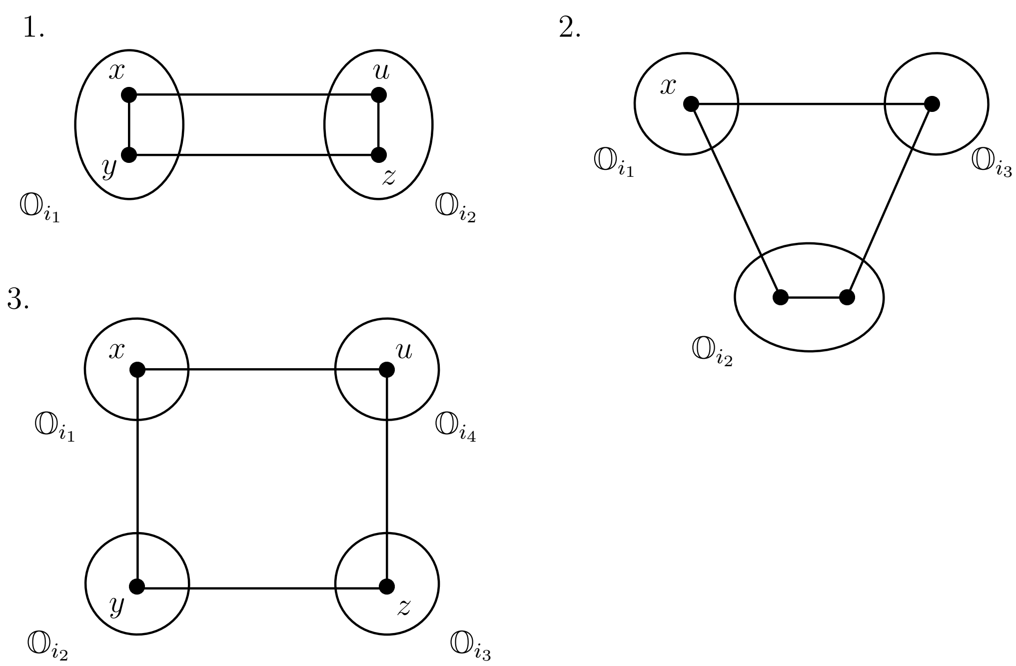


Figure 4.5: The possible cases of $\mathbb{G}$ containing $\mathbb{C}_4$.

2. $\mathbb{C}_4$ is contained in 3 cliques: $\mathbb{C}_4 \subseteq \mathbb{O}_{i_1} \cup \mathbb{O}_{i_2} \cup \mathbb{O}_{i_3}$.
3. $\mathbb{C}_4$ is contained in 4 cliques: $\mathbb{C}_4 \subseteq \mathbb{O}_{i_1} \cup \mathbb{O}_{i_2} \cup \mathbb{O}_{i_3} \cup \mathbb{O}_{i_4}$.

First, note that the case 2 can not happen as $\mathbb{G}^*$ is $\mathbb{K}_3$-free. To exclude remaining cases (1 and 3) let x be the vertex that was presented by the spoiler as the last vertex of $\mathbb{C}_4$. Without loss of generality we may assume that $x \in O_{i_1}$. Obviously, x is not one of the first v vertices presented as they are totally separated. Thus x was created either by line 2 or by line 8/10. In the last case, all edges starting in x have to end in $\mathbb{O}_{i_1}$. This however, is not the case as there is an edge xu with $u \in O_{i_2}$. If x was produced by line 2 then x was connected with all vertices in exactly two cliques. On the other hand,

- in case 1, x is not connected with $z \in O_{i_2}$,
- in case 3, x is connected with vertices in three different cliques, namely, with $y \in O_{i_2}$, $u \in O_{i_4}$ and previously existing ones in O_{i_1}. □

The graph constructed in the proof of Theorem 4.3.5 was not presented in a connected way. Now we improve spoiler's presentation technique to make it connected.

Theorem 4.3.6 *For every on-line clique covering algorithm $\mathcal{A}$ and for every positive integer $v \geqslant 2$ there exists a $\mathbb{K}_{2,2}$-free graph $\mathbb{G}$ with $v(\mathbb{G}) = v$ presented in a connected way such that $v_{\mathcal{A}}(\mathbb{G}) \geqslant \lfloor \frac{v(\mathbb{G})\cdot(v(\mathbb{G})+4)}{4} \rfloor$. In particular, there is no on-line clique covering algorithm with a competitive ratio for $\mathbb{K}_{2,2}$-free graphs presented in a connected way.*

Proof. To show this lower bound the spoiler constructs a graph which is the same as the one described in the proof of Theorem 4.3.5, but presented in a different order. To keep connectivity, we modify the procedure *MULTI_JOIN* from the proof of Theorem 4.3.5 in the same way as we have modified the procedure *JOIN* from the proof of Theorem 3.2.7 in the proof of Theorem 3.2.8.

The spoiler starts with one element set $O_1 = \{x\}$ and empty sets $O_2, \ldots, O_v$. These sets are supposed to form an optimal clique covering of $\mathbb{G}$. Next, the spoiler performs new procedure *MULTI_CON_JOIN* many times as he performs the procedure *MULTI_JOIN* in the proof of Theorem 4.3.5. *MULTI_CON_JOIN*(i, j) adds the first vertex of $\mathbb{O}_i$ or $\mathbb{O}_j$ when necessary and joins two cliques $\mathbb{O}_i$ and $\mathbb{O}_j$.

procedure $MULTI\_CON\_JOIN(i, j)$;
begin
if $|O_i| > |O_j|$ **then begin** *MULTI_CON_JOIN*(j, i); exit;**end**;
if $|O_i| \neq \emptyset$ **then begin** *MULTI_JOIN*(i, j); exit; **end**;

{if $|O_i| = \emptyset$ then the spoiler adds two new vertices x_1 and x_2}

$\mathbb{G}' := (G \cup \{x_1, x_2\}\,,\ E_{\mathbb{G}} \cup \{x_1x_2\} \cup \{x_1y:\ y \in O_j\})$;
$\mathbb{O}_i := \{x_2\}$;

WAIT for a response of $\mathcal{A}$ to color x_1 and x_2
if $cl_{\mathcal{A}}(x_1) = cl_{\mathcal{A}}(x_2)$ **then** $\mathbb{O}_j := \mathbb{O}_j \cup \{x_1\}$
else begin
    $\mathbb{O}_i := \mathbb{O}_i \cup \{x_1\}$;
    {if it is a wasteful join then .. }
    if $cl_{\mathcal{A}}(x_1) \notin Cl_{\mathcal{A}}(\mathbb{G} \setminus \{x_1, x_2\})$ **then** *MULTI_JOIN(i, j)*;
    end;

if $\mathbb{O}_i$ is tied **then**
    begin {the spoiler adds a new vertex z_1}
    $\mathbb{G}' := (G' \cup \{z_1\}\,,\ E_{\mathbb{G}'} \cup \{z_1y:\ y \in O_i\})$;
    $\mathbb{O}_i := \mathbb{O}_i \cup \{z_1\}$;
    end
else if $\mathbb{O}_j$ is tied **then**
    begin {the spoiler adds a new vertex z_2}
    $\mathbb{G}' := (G' \cup \{z_2\}\,,\ E_{\mathbb{G}'} \cup \{z_2y:\ y \in O_j\})$;
    $\mathbb{O}_j := \mathbb{O}_j \cup \{z_2\}$;
    end;
$\mathbb{G} := \mathbb{G}'$;
end;

The spoiler builds the graph $\mathbb{G}$ by the procedure *STRATEGY* described in the proof of Theorem 4.3.5: First, he performs $MULTI\_CON\_JOIN(1, j)$ for each even $2 \leqslant j \leqslant v$ and then $MULTI\_CON\_JOIN(i, j)$ for each even $2 \leqslant j \leqslant v$ and some i. Therefore, each $MULTI\_CON\_JOIN(i, j)$ is applied to at least one non-empty set, say O_j. If the second set O_i is empty, the procedure makes it non-empty (in a connected way). This guarantees that the resulting graph $\mathbb{G}$ is presented in a connected way.

The graph $\mathbb{G}$ is actually the same as (i.e., isomorphic to) the one described in the proof of Theorem 4.3.5, but the spoiler presents its vertices in a different order. Call by $\mathcal{S}_0$ and $\mathcal{A}_0$ the spoiler and the algorithm from the proof of Theorem 4.3.5, and by $\mathcal{S}$ and $\mathcal{A}$ the ones in this proof. A careful inspection of the procedure *MULTI_CON_JOIN* gives that for each game played by the algorithm $\mathcal{A}$ with the spoiler $\mathcal{S}$ there is a way for $\mathcal{A}_0$ to play with the spoiler $\mathcal{S}_0$ such that the graph built by $\mathcal{S}$ (which is the same as the one built by $\mathcal{S}_0$) is colored in the same way as the one built by $\mathcal{S}_0$. Therefore, as in the proof of Theorem 4.3.5, we have

$$v_{\mathcal{A}} \geqslant \lfloor \frac{v(\mathbb{G}) \cdot (v(\mathbb{G}) + 4)}{4} \rfloor. \qquad \square$$

4.4 Conclusions

As we have seen in this chapter, if a graph $\mathbb{G}$ does not contain an induced subgraph $\mathbb{K}_{1,t}$ ($t \geqslant 2$), the spoiler is able to force at most $(t-1)\cdot\chi(\mathbb{G})-t+2$ colors (Proposition 4.1.1). Theorem 4.1.5 shows that this bound is tight. A graph $\mathbb{K}_{1,t}$ is a forcing structure for the on-line coloring problem.

On the other hand, $\mathbb{K}_{s,t}$ ($s \geqslant 2$, $t \geqslant 2$) are not the forcing structures in this sense: there are graphs without these induced subgraphs, on which the spoiler can arbitrarily cheat the on-line algorithms (Corollary 4.1.7). The additional restriction, to present graphs in a connected way, does not change obtained results (Theorem 4.2.1 Proposition 4.2.2).

The graphs $\mathbb{K}_{s,t}$ for $s \geqslant 1$, $t \geqslant 3$ are not the forcing structures for clique covering problem (Corollary 4.3.3). The remaining case of $\mathbb{K}_{2,2}$-free graphs for which there is a huge gap between our lower bound and an exponential upper bound for the competitive function leads to the open problem. At this moment we know that this competitive function is between $\lfloor\frac{v(\mathbb{G})\cdot(v(\mathbb{G})+4)}{4}\rfloor$ (Theorem 4.3.5) and $2^{v(\mathbb{G})}-1$ [17].

Problem 4 *Is there an on-line clique covering algorithm with a competitive function better than exponential for $\mathbb{K}_{2,2}$-free graphs?*

Chapter 5

$\mathbb{P}_k$-free graphs

5.1 On-line coloring of $\mathbb{P}_k$-free graphs

One of interesting and often occurred structures in graphs are paths. A path with k vertices is denoted by $\mathbb{P}_k$. The on-line coloring algorithms for graphs without induced paths were studied by A.Gyárfás, J.Lehel [17, 18] and H.A.Kierstead, S.G.Penrice, W.T.Trotter [22]. A.Gyárfás and J.Lehel have shown that the on-line $\mathcal{FF}$ coloring algorithm is optimal for $\mathbb{P}_4$-free graphs.

Proposition 5.1.1 (A.Gyárfás, J.Lehel [17]) *If $\mathbb{G}$ is $\mathbb{P}_4$-free, then $\chi_{\mathcal{FF}}(\mathbb{G}) = \chi(\mathbb{G})$.* □

Since the complement of $\mathbb{P}_4$ is $\mathbb{P}_4$, we have

Corollary 5.1.2 *If $\mathbb{G}$ is $\mathbb{P}_4$-free, then $v_{\mathcal{FF}}(\mathbb{G}) = v(\mathbb{G})$.* □

This means that $\mathbb{P}_4$ is a forcing structure in a very strong sense. If a graph does not contain it, the spoiler is unable to force any additional colors. On the other hand, a longer path $\mathbb{P}_6$ was shown by A.Gyárfás and J.Lehel not to be a forcing structure for coloring.

Theorem 5.1.3 (A.Gyárfás, J.Lehel [17]) *For every on-line coloring algorithm $\mathcal{A}$ and for every positive integer n there exists a bipartite $\mathbb{P}_6$-free graph $\mathbb{G}_n$ such that $\chi_{\mathcal{A}}(\mathbb{G}_n) \geqslant n$.* □

Obviously, graphs $\mathbb{G}_n$, mentioned in Theorem 5.1.3, satisfy $\chi(\mathbb{G}_n) \leqslant 2$. On the other hand, each on-line coloring algorithm can be forced by some $\mathbb{G}_n$ to use at least n colors. Therefore, there is no on-line coloring algorithm with a competitive function for $\mathbb{P}_6$-free graphs. The graphs $\mathbb{G}_n$ constructed in the proof of Theorem 5.1.3 are not presented in a connected way. But there is an easy way to modify them to make this theorem work for $\mathbb{P}_6$-free graphs presented in a connected way.

Let $\mathcal{A}$ be an on-line coloring algorithm. The very first vertex, say x, presented by the spoiler is going to keep the graph connected. After presenting x, the

Graph $\mathbb{G}$	**Presentation method**	**Coloring**	**Clique covering**
$(\mathbb{C}_4, 2\mathbb{K}_2)$-free	connected	$opt(\mathbb{G})+1$ 5.4.4	$opt(\mathbb{G})+1$ 5.4.6
$(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free	-	$\binom{opt(\mathbb{G})+1}{2}$ 5.3.9	$2opt(\mathbb{G})-1$ 5.3.17 and [17]
$(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free	connected	$\binom{opt(\mathbb{G})}{2}+1 \leq \ldots \leq \binom{opt(\mathbb{G})+1}{2}$ 5.3.14 and 5.3.8	$2opt(\mathbb{G})-1$ 5.3.17 and [17]
$(\mathbb{C}_4, \mathbb{P}_5)$-free	-	$\binom{opt(\mathbb{G})+1}{2}$ 5.3.13	$2opt(\mathbb{G})-1$ 5.3.20 and [17]
$(\mathbb{C}_4, \mathbb{P}_5)$-free	connected	$\binom{opt(\mathbb{G})}{2}+1 \leq \ldots \leq \binom{opt(\mathbb{G})+1}{2}$ 5.3.15 and 5.3.12	$2opt(\mathbb{G})-1$ 5.3.20 and [17]
$\mathbb{P}_4$-free	connected	$opt(\mathbb{G})$ [17], 5.1.1	$opt(\mathbb{G})$ [17], 5.1.2
$\mathbb{P}_5$-free	-	$\binom{opt(\mathbb{G})+1}{2} \leq \ldots \leq \frac{4^{opt(\mathbb{G})}-1}{3}$ 5.3.4 and [22]	∞ 5.2.1 based on [17]
$\mathbb{P}_5$-free	connected	$\binom{opt(\mathbb{G})}{2}+1 \leq \ldots \leq \frac{4^{opt(\mathbb{G})}-1}{3}$ 5.3.16 and [22]	∞ 5.2.1 based on [17]
$\mathbb{P}_k$-free $k \geq 6$	connected	∞ [17], 5.1.5	∞ 5.2.2 based on [17]

Table 5.1: Possible competitive functions for on-line problems

spoiler proceeds as in the construction of $\mathbb{G}_n$ needed to prove Theorem 5.1.3 and additionally joins all vertices to x. Note that the color assigned to x is not used by $\mathcal{A}$ to color other vertices. The algorithm $\mathcal{A}$ is forced to use new colors on a constructed graph $\mathbb{G}'_n$ as it has been forced to use new colors during the construction of $\mathbb{G}_n$. Thanks to all edges between x and all other vertices and since a graph $\mathbb{G}_n$ is $\mathbb{P}_6$-free, $\mathbb{G}'_n$ is $\mathbb{P}_6$-free as well. The new graphs $\mathbb{G}'_n$ are not longer bipartite, but they are obviously 3-colorable. Since they are presented in a connected way and force $\mathcal{A}$ to use arbitrarily many colors, we immediately get

Theorem 5.1.4 *There is no on-line coloring algorithm with a competitive function for $\mathbb{P}_6$-free graphs presented in a connected way.* □

Since $\mathbb{P}_6$-free graphs are $\mathbb{P}_k$-free for $k \geqslant 6$, therefore we immediately get:

Corollary 5.1.5 *Let $k \geqslant 6$. There is no on-line coloring algorithm with a competitive function for $\mathbb{P}_k$-free graphs presented in a connected way.* □

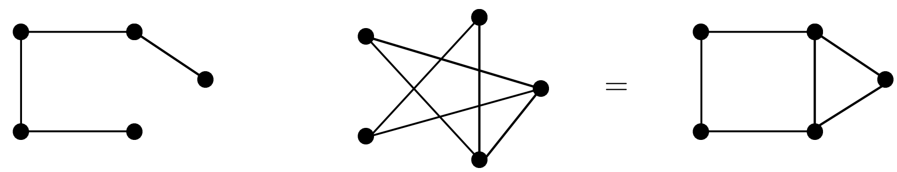
The remaining, and in fact the most challenging situation is for $\mathbb{P}_5$-free graphs. A.Gyárfás and J.Lehel [18] have shown a quite complicated on-line coloring algorithm which gives a superexponential competitive function for $\mathbb{P}_5$-free graphs. Next, H.A.Kierstead, S.G.Penrice, W.T.Trotter [22] have invented different algorithm $\mathcal{A}$ for which $\chi_{\mathcal{A}}(\mathbb{G}) \leqslant (4^{\omega(\mathbb{G})} - 1)/3$, where $\mathbb{G}$ is $\mathbb{P}_5$-free. Since the chromatic number $\chi(\mathbb{G})$ is at least $\omega(\mathbb{G})$, the competitiveness of coloring problem for $\mathbb{P}_5$-free graphs is bounded by an exponential function $(4^{\chi(\mathbb{G})} - 1)/3$. However, no non-trivial lower bound for the competitive function was known. We establish, in Theorem 5.3.2 and Corollary 5.3.4, a lower bound for this problem to be quadratic. There is a huge gap between lower and upper bounds for this problem. Determining the exact competitiveness here is still open. The results of this chapter are summarized in Table 5.

5.2 On-line clique covering of $\mathbb{P}_k$-free graphs

We have already mentioned in section 5.1, that the path $\mathbb{P}_4$ is a forcing structure for clique covering problem and that the on-line $\mathcal{FF}$ clique covering algorithm is optimal for $\mathbb{P}_4$-free graphs. Now, we show that, in contrast to $\mathbb{P}_4$, the path $\mathbb{P}_5$ is not a forcing structure for clique covering problem.

Proposition 5.2.1 *There is no on-line clique covering algorithm with a competitive function for $\mathbb{P}_5$-free graphs presented in a connected way.*

Proof. We argue as in the proof of Theorem 2.4.3, that is, we use complements $\overline{\mathbb{T}}_n$ of the trees $\mathbb{T}_n$ constructed in the proof of Theorem 2.1.1. We have already seen that $\overline{\mathbb{T}}_n$ were presented in a connected way. Additionally, we have to show that graphs $\overline{\mathbb{T}}_n$ are $\mathbb{P}_5$-free. Suppose that in some graph $\overline{\mathbb{T}}_n$ there is an induced path $\mathbb{P}_5$. Therefore, there exists the complement $\overline{\mathbb{P}}_5$ in the tree



$\mathbb{P}_5$ $\overline{\mathbb{P}}_5$

Figure 5.1: $\mathbb{P}_5$ and its complement $\overline{\mathbb{P}}_5$.

$\mathbb{T}_n$. This cannot happen as $\overline{\mathbb{P}}_5$ contains cycles $\mathbb{C}_3$ and $\mathbb{C}_4$ (see Figure 5.1), while $\mathbb{T}_n$ contains no cycles. □

Since $\mathbb{P}_5$-free graphs are $\mathbb{P}_k$-free for $k \geqslant 5$, we have the following.

Corollary 5.2.2 *Let $k \geqslant 5$. There is no on-line clique covering algorithm with a competitive function for $\mathbb{P}_k$-free graphs presented in a connected way.* □

5.3 $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs

We have already mentioned that there is a huge gap between a trivial lower bound and an upper bound for coloring of $\mathbb{P}_5$-free graphs. In order to narrow this gap we analyze how forcing graphs would look like to improve the lower bound. In this section we show that additionally forbidding the occurrence of $\mathbb{C}_4$ and $\mathbb{C}_5$ we are able to determine an exact competitive function: we show that the competitive function for $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graphs is $\binom{\chi(\mathbb{G})+1}{2}$. This means that the lower bound for $\mathbb{P}_5$-free graphs is quadratic and each graph forcing coloring algorithms to use more than $\binom{\chi(\mathbb{G})+1}{2}$ colors contains $\mathbb{C}_4$ or $\mathbb{C}_5$. Moreover, we show that the competitive function for a wider family of $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs is $\binom{\chi(\mathbb{G})+1}{2}$ as well. This means that each graph forcing coloring algorithms to use more than $\binom{\chi(\mathbb{G})+1}{2}$ colors contains in fact $\mathbb{C}_4$.

We have also mentioned the huge gap between a quadratic lower bound and an exponential upper bound for clique covering of $\mathbb{C}_4$-free graphs (Section 4.4, Problem 4). The analysis of $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs, that is going to be done in this section, establishes the competitive function for this family to be $2v(\mathbb{G})-1$. This means that each graph which forces covering algorithms to use at least quadratic number of cliques contains $\mathbb{P}_5$.

5.3.1 On-line coloring of $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs

We start with showing the lower bound $\binom{\chi(\mathbb{G})+1}{2}$ for the $\mathcal{FF}$ algorithm on $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graphs $\mathbb{G}$. Next, we generalize this, by showing that this is also a lower bound for arbitrary algorithms. At the end, we show that this lower bound is tight.

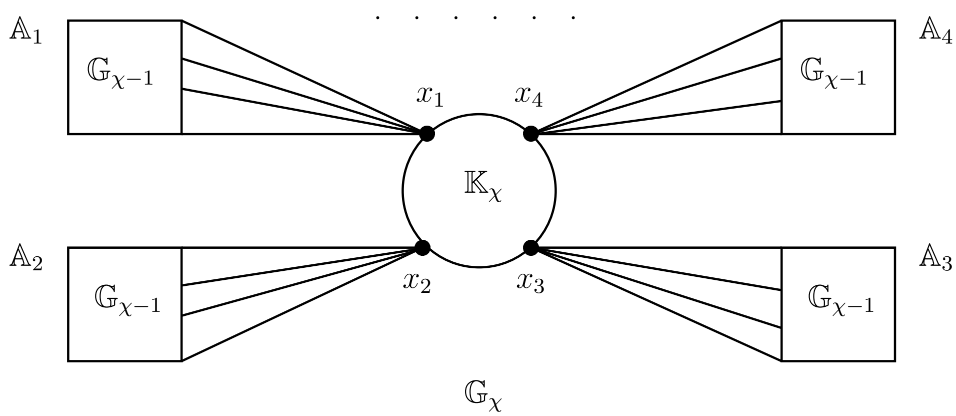


Figure 5.2: Construction of $\mathbb{G}_\chi$.

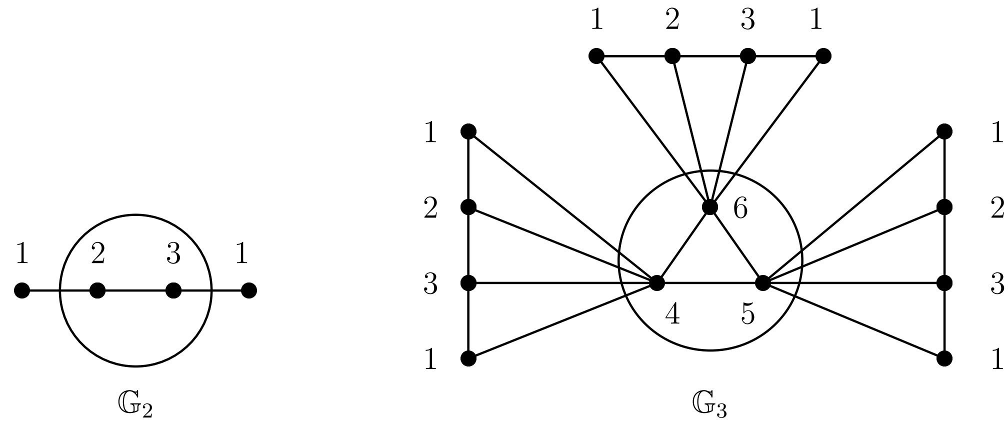


Figure 5.3: Examples

Theorem 5.3.1 *For every positive integer χ there exists a $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graph $\mathbb{G}$ with $\chi(\mathbb{G}) = \chi$ such that*

$$\chi_{\mathcal{FF}}(\mathbb{G}) \geqslant \binom{\chi(\mathbb{G})+1}{2}.$$

Proof. We describe a strategy for the spoiler to construct a $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graph $\mathbb{G}_\chi$ with $\chi(\mathbb{G}_\chi) = \chi$ for which the $\mathcal{FF}$ algorithm have to use $\binom{\chi+1}{2}$ colors. To construct $\mathbb{G}_\chi$ we induct on χ. Let $\mathbb{G}_1$ be a single vertex and assume that for $i = 1, \ldots, \chi - 1$ there is a strategy for the spoiler which forces $\mathcal{FF}$ to use at least $\binom{i+1}{2}$ distinct colors on $\mathbb{G}_i$. In order to build $\mathbb{G}_\chi$ the spoiler first constructs χ copies of $\mathbb{G}_{\chi-1}$, namely $\mathbb{A}_1, \ldots, \mathbb{A}_\chi$. The $\mathcal{FF}$ algorithm colors each $\mathbb{A}_i$ in the same way with $\chi_{\mathcal{FF}}(\mathbb{G}_{\chi-1})$ colors. The graph $\mathbb{G}_\chi$ is formed by adding a clique $\mathbb{K}_\chi$ consisting of vertices $x_1, \ldots, x_\chi$ such that each vertex x_i is joined to all vertices of $\mathbb{A}_i$ (see Figures 5.2 and 5.3). $\mathcal{FF}$ is forced to color new vertices $x_1, \ldots, x_\chi$

with completely new colors. Inducting on χ we get

$$\chi_{\mathcal{FF}}(\mathbb{G}_\chi) = \chi + \chi_{\mathcal{FF}}(\mathbb{G}_{\chi-1}) = \chi + \binom{\chi}{2} = \binom{\chi+1}{2}.$$

It is easy to see that $\chi(\mathbb{G}_\chi) = \chi$. We obtain an optimal coloring of this graph by coloring vertices of $\mathbb{G}_\chi$ in the order reverse to the one in which vertices were presented. In other words, first, we color with χ colors the vertices of the clique $\mathbb{K}_\chi$ added at the end of the construction. Next, we color each $\mathbb{A}_i$, which is $(\chi-1)$–colorable, with colors from the set $C(\mathbb{K}_\chi) \setminus c(x_i)$. Finally, inducting on χ, one can easily notice that $\mathbb{G}_\chi$ is $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free. □

Theorem 5.3.2 *For every on-line coloring algorithm $\mathcal{A}$ and for every positive integer χ there exists a $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graph $\mathbb{H}$ with $\chi(\mathbb{H}) = \chi$ such that*

$$\chi_{\mathcal{A}}(\mathbb{H}) \geqslant \binom{\chi(\mathbb{H})+1}{2}.$$

Proof. We describe a strategy for the spoiler to construct a $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graph $\mathbb{H}_\chi$ with $\chi(\mathbb{H}_\chi) = \chi$ for which the algorithm $\mathcal{A}$ uses at least $\binom{\chi+1}{2}$ colors. The construction improves the one described in the proof of Theorem 5.3.1. In that proof the spoiler constructs the graphs $\mathbb{G}_\chi$ with $\chi(\mathbb{G}_\chi) = \chi$ forcing $\mathcal{FF}$ to use $\binom{\chi+1}{2}$ colors.

To construct $\mathbb{H}_\chi$ we induct on χ. Let $\mathbb{H}_1$ be a single vertex and assume that for $i = 1, \ldots, \chi-1$ there is a strategy for the spoiler which forces $\mathcal{A}$ to use at least $\binom{i+1}{2}$ distinct colors to color $\mathbb{H}_i$. In order to obtain the graph $\mathbb{G}_\chi$ in the proof of Theorem 5.3.1, the spoiler has needed χ graphs $\mathbb{A}_1, \ldots, \mathbb{A}_\chi$ isomorphic to $\mathbb{G}_{\chi-1}$ colored with the same colors, i.e., $C_{\mathcal{FF}}(\mathbb{A}_1) = \ldots = C_{\mathcal{FF}}(\mathbb{A}_\chi)$. This was guaranteed by the rules of the First Fit strategy. In our modified construction the spoiler applies his strategy for $\mathbb{H}_{\chi-1}$ to generate isolated graphs $\mathbb{B}_1, \ldots, \mathbb{B}_t$, so that $\mathcal{A}$ is forced to use on each of them $\chi_{\mathcal{A}}(\mathbb{B}_i) = \binom{\chi}{2}$ colors, while $\chi(\mathbb{B}_i) \leqslant \chi-1$. Since this can be accomplished with $\mathbb{B}_i$'s satisfying $|B_i| \leqslant |H_{\chi-1}|$, repeating this procedure long enough the spoiler ends up with at least χ graphs $\mathbb{B}_{i_1}, \ldots, \mathbb{B}_{i_\chi}$ with

$$C_{\mathcal{A}}(\mathbb{B}_{i_1}) = \ldots = C_{\mathcal{A}}(\mathbb{B}_{i_\chi}) \quad \text{or} \tag{5.1}$$

$$|C_{\mathcal{A}}(\mathbb{B}_1 \cup \ldots \cup \mathbb{B}_t)| \geqslant \binom{\chi+1}{2}. \tag{5.2}$$

If (5.2) happens, in order to get $\chi(\mathbb{H}_\chi) = \chi$, the spoiler presents a few additional vertices joined to all vertices presented previously; the construction is done. Otherwise, the graph $\mathbb{H}_\chi$ is formed as $\mathbb{G}_\chi$ in the proof of Theorem 5.3.1: the spoiler adds a clique $\mathbb{K}_\chi$ consisting of vertices $x_1, \ldots, x_\chi$ such that each vertex x_j is joined to all vertices of $\mathbb{B}_{i_j}$. Since (5.1) holds, the algorithm $\mathcal{A}$ is forced to color new vertices $x_1, \ldots, x_\chi$ with colors outside $C_{\mathcal{A}}(\mathbb{B}_{i_j})$. Inducting on χ we get

$$\chi_{\mathcal{A}}(\mathbb{H}_\chi) = \chi + |C_{\mathcal{A}}(\mathbb{B}_{i_1} \cup \ldots \cup \mathbb{B}_{i_\chi})| = \chi + \binom{\chi}{2} = \binom{\chi+1}{2}.$$

A careful inspection of the graph $\mathbb{H}_\chi$ shows that $\mathbb{H}_\chi$ is $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free and χ-colorable. □

Enlarging the family considered in Theorem 5.3.2 we get the following corollaries.

Corollary 5.3.3 *For every on-line coloring algorithm $\mathcal{A}$ and for every positive integer χ there exists a $(\mathbb{C}_4, \mathbb{P}_5)$-free graph $\mathbb{G}$ with $\chi(\mathbb{G}) = \chi$ such that*

$$\chi_{\mathcal{A}}(\mathbb{G}) \geqslant \binom{\chi(\mathbb{G}) + 1}{2}.$$ □

Corollary 5.3.4 *For every on-line coloring algorithm $\mathcal{A}$ and for every positive integer χ there exists a $\mathbb{P}_5$-free graph $\mathbb{G}$ with $\chi(\mathbb{G}) = \chi$ such that*

$$\chi_{\mathcal{A}}(\mathbb{G}) \geqslant \binom{\chi(\mathbb{G}) + 1}{2}.$$ □

To prove an upper bound for a competitive function on $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graphs, we make use of the concept of dominating sets in graphs. A set $D \subseteq G$ is dominating in $\mathbb{G}$ if $D \cap N_{\mathbb{G}}(x) \neq \emptyset$ for every $x \in G$. An induced subgraph $\mathbb{H}$ of $\mathbb{G}$ is dominating in $\mathbb{G}$ if the set H is dominating in $\mathbb{G}$. We will need the following properties of dominating subgraphs in $\mathbb{P}_5$-free graphs obtained by G.Bascó, Z.Tuza [1] and M.Cozzens, L.Kelleher [10].

Theorem 5.3.5 (G.Bascó, Z.Tuza [1]) *If $\mathbb{G}$ is connected and $\mathbb{P}_5$-free, then $\mathbb{G}$ contains either a dominating clique or a dominating path $\mathbb{P}_3$.* □

Theorem 5.3.6 (G.Bascó, Z.Tuza [1], M.Cozzens, L.Kelleher [10]) *If $\mathbb{G}$ is connected and $(\mathbb{C}_5, \mathbb{P}_5)$-free, then $\mathbb{G}$ contains a dominating clique.* □

Besides these two theorems we will need the following lemma.

Lemma 5.3.7 *If a connected graph $\mathbb{G}$ is $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free and $\mathbb{D}$ is a dominating clique in $\mathbb{G}$, then*

$$\omega(\mathbb{G} \setminus D) < \omega(\mathbb{G}).$$

Proof. Since $\mathbb{G}$ is $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free, $\mathbb{G}$ does not contain induces cycles with more than 3 vertices. This means $\mathbb{G}$ is chordal and therefore $\mathbb{G}$ has a perfect elimination order of its vertices: $x_1, x_2, \ldots, x_n$ [23]. In other words, for every vertex $x_i \in G$, its neighbors in the set $\{x_{i+1}, \ldots, x_n\}$ form a clique. For a fixed PEO we need the following notation:

- $PEO(x) =$ number of the vertex x in the perfect elimination order, i.e., $PEO(x_i) = i$,

- $N^+_{\mathbb{G}}(x) = \{y \in G : \ PEO(y) > PEO(x) \text{ and } yx \in E_{\mathbb{G}}\}$.

Let $\omega = \omega(\mathbb{G})$ and suppose to the contrary that $\mathbb{D}$ is a dominating clique in $\mathbb{G}$ with $\omega(\mathbb{G} \setminus D) = \omega(\mathbb{G})$. This gives a clique $\mathbb{D}' \subseteq \mathbb{G} \setminus D$ with $|D'| = \omega(\mathbb{G}) = \omega$. Let

- $d_1, \ldots, d_m$
- $a_1, \ldots, a_\omega$

be the lists of elements in D and D' (respectively) according to the perfect elimination order, i.e., $PEO(d_i) < PEO(d_j)$ and $PEO(a_i) < PEO(a_j)$ whenever $i < j$. Note that $N^+_{\mathbb{G}}(a_1) = \{a_2, \ldots, a_\omega\} \cup A$ for some $A \subseteq G \setminus D'$. Since the set $N^+_{\mathbb{G}}(a_1) \cup \{a_1\}$ forms a clique of size ω, we have $A = \emptyset$ and therefore $N^+_{\mathbb{G}}(a_1) \cup \{a_1\} = D'$.

Now, we show that each a_j is a neighbor of d_m, i.e., $d_m a_j \in E_{\mathbb{G}}$. Since $\mathbb{D}$ is dominating in $\mathbb{G}$, for each a_j there must be $d_{i_j} \in D$ with $d_{i_j} a_j \in E_{\mathbb{G}}$. We have $PEO(d_{i_1}) < PEO(a_1)$, as otherwise $d_{i_1} \in N^+_{\mathbb{G}}(a_1) = \{a_2, \ldots, a_\omega\} \subseteq D'$ in contrast to $D \cap D' = \emptyset$. Moreover, $N^+_{\mathbb{G}}(d_{i_1}) \supseteq \{d_{i_1+1}, \ldots, d_m, a_1\}$. This, together with the fact that sets of the form $N^+_{\mathbb{G}}(x)$ induce cliques, gives $d_m a_1 \in E_{\mathbb{G}}$. Therefore we have

$$PEO(d_m) < PEO(a_1) \tag{5.3}$$

as otherwise $d_m \in N^+_{\mathbb{G}}(a_1) = \{a_2, \ldots, a_\omega\}$.

From (5.3) we get $PEO(d_{i_j}) < PEO(a_j)$ for each j and $N^+_{\mathbb{G}}(d_{i_j}) \supseteq \left\{d_{i_j+1}, \ldots, d_m, a_j\right\}$. This, together with the fact that sets of the form $N^+_{\mathbb{G}}(x)$ induce cliques, gives that $d_m a_j \in E_{\mathbb{G}}$ for each j. Therefore the set $\{d_m, a_1, \ldots, a_\omega\}$ forms a clique of size $\omega + 1$, which contrasts to $\omega(\mathbb{G}) = \omega$. □

Now we are ready to prove the following theorem:

Theorem 5.3.8 *If* $\mathbb{G}$ *is* $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$*-free, then*

$$\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant \binom{\chi(\mathbb{G}) + 1}{2}.$$

Proof. We prove that $\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant \binom{\omega(\mathbb{G})+1}{2}$ which will finish the proof in view of $\omega(\mathbb{G}) \leqslant \chi(\mathbb{G})$ for each $\mathbb{G}$. Let $\omega = \omega(\mathbb{G})$ and suppose $\mathbb{G}$ is connected. Since $\mathbb{G}$ is $(\mathbb{C}_5, \mathbb{P}_5)$-free, Theorem 5.3.6 supplies a dominating clique $\mathbb{D}^1$ in $\mathbb{G}$. First Fit strategy ensures us that there is a connected component $\mathbb{G}^1$ of $\mathbb{G} \setminus D^1$ with $C_{\mathcal{FF}}(\mathbb{G}^1 \cup \mathbb{D}^1) = C_{\mathcal{FF}}(\mathbb{G})$, so that

$$\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant |C_{\mathcal{FF}}(\mathbb{D}^1)| + |C_{\mathcal{FF}}(\mathbb{G}^1)|.$$

Since $\mathbb{G}^1$ is an induced subgraph of $\mathbb{G}$, it is $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free as well. Note that Lemma 5.3.7 guarantees that $\omega(\mathbb{G} \setminus D^1) < \omega(\mathbb{G})$. Consequently $\omega(\mathbb{G}^1) < \omega(\mathbb{G})$ as $\mathbb{G}^1 \subseteq \mathbb{G} \setminus D^1$. Thus $\omega(\mathbb{G}^1) \leqslant \omega - 1$.

Now, we apply the same argument to the $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graph $\mathbb{G}^1$, i.e., using Theorem 5.3.6 we get a dominating clique $\mathbb{D}^2$ in $\mathbb{G}^1$. As previously there is a connected component $\mathbb{G}^2$ of $\mathbb{G}^1 \setminus D^2$ with

$$\chi_{\mathcal{FF}}(\mathbb{G}^1) \leqslant |C_{\mathcal{FF}}(\mathbb{D}^2)| + |C_{\mathcal{FF}}(\mathbb{G}^2)| \text{ and } \omega(\mathbb{G}^2) < \omega(\mathbb{G}^1).$$

Thus $\omega(\mathbb{G}^2) \leqslant \omega - 2$. Repeating this argument sufficiently many times we end up with components $\mathbb{G}^1, \mathbb{G}^2, \ldots, \mathbb{G}^{\omega-1}$ for which $\omega(\mathbb{G}^i) \leqslant \omega - i$ and with dominating cliques $\mathbb{D}^1, \mathbb{D}^2, \mathbb{D}^3, \ldots, \mathbb{D}^\omega$ for which $|D^i| \leqslant \omega - i + 1$. Note that $\omega(G^{\omega-1}) = |D^\omega| \leqslant 1$. Summing up, we get

$$\begin{aligned}\chi_{\mathcal{FF}}(\mathbb{G}) &\leqslant |C_{\mathcal{FF}}(\mathbb{D}^1)| + |C_{\mathcal{FF}}(\mathbb{D}^2)| + \ldots + |C_{\mathcal{FF}}(\mathbb{D}^\omega)| \\ &\leqslant \omega + (\omega - 1) + \ldots + 1 = \binom{\omega+1}{2}.\end{aligned}$$

If $\mathbb{G}$ is not connected, it consists of connected $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free components. The behavior of $\mathcal{FF}$ ensures us that there exists a component, say $\mathbb{G}_i$, for which $C_{\mathcal{FF}}(\mathbb{G}_i) = C_{\mathcal{FF}}(\mathbb{G})$. Since $\chi_{\mathcal{FF}}(\mathbb{G}_i) \leqslant \binom{\omega(\mathbb{G}_i)+1}{2}$, we have $\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant \binom{\omega(\mathbb{G})+1}{2}$. □

Combining Theorems 5.3.2 and 5.3.8 we get

Corollary 5.3.9 *The on-line coloring algorithm with the First-Fit strategy for* $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$*-free graphs is the best possible and it has the competitive function* $\binom{\chi(\mathbb{G})+1}{2}$*, where* $\mathbb{G}$ *is an input graph.* □

Now we show that $\binom{\chi(\mathbb{G})+1}{2}$ is also the competitive function for $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs. By Corollary 5.3.3 it is a lower bound. To show that $\binom{\chi(\mathbb{G})+1}{2}$ is an upper bound we need to understand structural properties of $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs. We need also the following notions modelled after [12].

A subset $S \subseteq G$ which satisfies $1 < |S| < |G|$ is *homogeneous* in a graph $\mathbb{G}$ if every vertex in $G \setminus S$ is adjacent either to all vertices in S or to none of them (in other words, $N_{\mathbb{G}}(x) \setminus S = N_{\mathbb{G}}(y) \setminus S$ for $x, y \in S$). A graph is a buoy if it is an inflated cycle $\mathbb{C}_5$, more precisely, if G can be decomposed into $G = A_0 \cup \ldots \cup A_4$ such that $xy \in E_{\mathbb{G}}$ whenever $x \in A_i$ and $y \in A_{i+1}$ (modulo 5) and all other edges of $\mathbb{G}$ have both their ends in the same A_i. A buoy is *complete* if each $\mathbb{A}_i$ can be taken to be a clique.

Theorem 5.3.10 (J.Fouquet, V.Giakoumakis, F.Maire, H.Thuillier [12]) *A connected graph* $\mathbb{G}$ *is* $(\mathbb{C}_4, \mathbb{P}_5)$*-free if and only if there exists a partition* $G = Q \cup B$ *with the following properties* (see Figure 5.4)*:*

- $\mathbb{Q} := \mathbb{G}|_Q$ *is chordal and* $\mathbb{P}_5$*-free.*
- *If* $\mathbb{B} := \mathbb{G}|_B \neq \emptyset$ *then it can be partitioned into disjoint complete buoys* $\mathbb{B}_1, \ldots, \mathbb{B}_n$*. Each* $\mathbb{B}_i$ *is a homogeneous part of* $\mathbb{G}$ *and for* $b \in B_i$ *the set* $Q_i := N_{\mathbb{G}}(b) \setminus B_i = N_{\mathbb{G}}(b) \setminus B$ *forms a clique in* $\mathbb{Q}$*. Moreover, there is a clique* $\mathbb{K} \subseteq \mathbb{Q}$ *for which* $B \subseteq N_{\mathbb{G}}(K)$*.* □

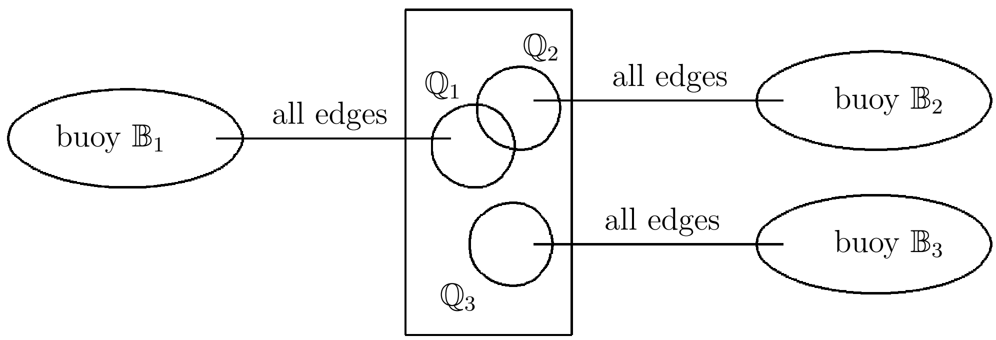


Figure 5.4: $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs. $\mathbb{Q}_1$, $\mathbb{Q}_2$, $\mathbb{Q}_3$ are cliques.

The number n of buoys appearing in the decomposition of $\mathbb{B}$ is called the buoy number of $\mathbb{G}$. With the help of Theorem 5.3.10 we will show that $\mathcal{FF}$ strategy is the best possible on $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs. In order to analyze the $\mathcal{FF}$ algorithm we need the following lemma, that allows as to reduce the buoy number of $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs, eventually to zero.

Lemma 5.3.11 *Let $\mathbb{G}$ be a connected $(\mathbb{C}_4, \mathbb{P}_5)$-free graph presented on-line. Then there exists a connected $(\mathbb{C}_4, \mathbb{P}_5)$-free graph $\mathbb{G}'$ presented on-line such that*

- $\chi(\mathbb{G}) = \chi(\mathbb{G}')$,
- $\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant \chi_{\mathcal{FF}}(\mathbb{G}')$

and

- *the buoy number of $\mathbb{G}'$ is strictly smaller than the one of $\mathbb{G}$, provided the last one is non zero.*

Proof. Let $\mathbb{Q}$ and $\mathbb{B}_1, \ldots, \mathbb{B}_n$ $(n \geqslant 1)$ be a decomposition supplied for $\mathbb{G}$ by Theorem 5.3.10 and let $Q_i = N_{\mathbb{G}}(B_i) \setminus B_i$ $(i = 1, \ldots, n)$. First of all, we will construct a $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graph $\mathbb{B}_1'$ such that $\chi(\mathbb{B}_1) = \chi(\mathbb{B}_1')$ and $\chi_{\mathcal{FF}}(\mathbb{B}_1) \leqslant \chi_{\mathcal{FF}}(\mathbb{B}_1')$. Afterwards, we will show that the graph obtained from $\mathbb{G}$ by a special replacement of $\mathbb{B}_1$ by $\mathbb{B}_1'$ can serve as $\mathbb{G}'$.

Being a complete buoy we know that $\mathbb{B}_1$ is $\mathbb{K}_{1,3}$-free and from Proposition 4.1.1 we get $\chi_{\mathcal{FF}}(\mathbb{B}_1) \leqslant 2 \cdot \chi(\mathbb{B}_1) - 1$. On the other hand, from Theorem 5.3.1 we know that for every positive integer χ, especially for $\chi = \chi(\mathbb{B}_1)$, there exists a $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free on-line graph $\mathbb{B}_1'$ with $\chi(\mathbb{B}_1') = \chi$ and $\chi_{\mathcal{FF}}(\mathbb{B}_1') \geqslant \binom{\chi+1}{2}$. Note that $\chi_{\mathcal{FF}}(\mathbb{B}_1) \leqslant \chi_{\mathcal{FF}}(\mathbb{B}_1')$. Now supposing that $\mathbb{B}_1$ and $\mathbb{B}_1'$ are disjoint we define $\mathbb{G}' := ((G \setminus B_1) \cup B_1', E_{\mathbb{G}'})$, where

$$\begin{aligned} E_{\mathbb{G}'} &= \{xy:\ xy \in E_{\mathbb{G}} \ \text{ and } \ x, y \notin B_1\} \cup E_{\mathbb{B}_1'} \\ &\quad \cup \{xy:\ x \in Q_1 \ \text{ and } \ y \in B_1'\}\,. \end{aligned} \tag{5.4}$$

From Theorem 5.3.10 we know that $Q_1 = N_{\mathbb{G}}(B_1)\backslash B_1$ induces a clique. Moreover, B_1 is a homogeneous part of $\mathbb{G}$. Thus, there exist all possible edges between Q_1 and B_1 in $\mathbb{G}$ and there are no edges between $G\backslash(Q_1 \cup B_1)$ and B_1. Similarly in $\mathbb{G}'$ there are all edges between Q_1 and B_1' and there is no edge between $G'\backslash(Q_1 \cup B_1')$ and B_1'.

To describe the order of vertices in which the graph $\mathbb{G}'$ is presented we first consider the order of B_1', say $x_1, \ldots, x_n$, which forces $\mathcal{FF}$ to use at least $\chi_{\mathcal{FF}}(\mathbb{B}_1)$ colors. Let $\mathbb{G}$ and $\mathbb{B}_1'$ be colored by the $\mathcal{FF}$ algorithm. We define:

- i_j = index of the first vertex of $\mathbb{B}_1'$ colored with j, for $j = 1, \ldots, \chi_{\mathcal{FF}}(\mathbb{B}_1)$,
- k = index of the first vertex of $\mathbb{B}_1'$ colored with $\chi_{\mathcal{FF}}(\mathbb{B}_1)+1$, if there is such a vertex.

To construct (step by step) the on-line graph $\mathbb{G}'$ we follow the order in which the vertices of $\mathbb{G}$ were presented, say $z_1, \ldots, z_m$. Let $\mathbb{G}_i'$ be the on-line subgraph of $\mathbb{G}'$ constructed after the first i steps, i.e., after the analysis of the first i vertices of $\mathbb{G}$. We start with $\mathbb{G}_0' = \emptyset$. For each $i = 1, \ldots, m$ we proceed as follows.

- If $z_i \in G \setminus B_1$, then to get $\mathbb{G}_i$ we add z_i to $\mathbb{G}_{i-1}'$ with respective edges according to (5.4).
- If $z_i \in B_1$ and z_i is colored by $\mathcal{FF}$ with a color already used, then z_i is not added to $\mathbb{G}_{i-1}'$ and we do not add any replacing vertices instead of it, i.e., $\mathbb{G}_i' = \mathbb{G}_{i-1}'$.
- If $z_i \in B_1$ and z_i is colored with a new color, then in place of z_i, we add the first vertex of $\mathbb{B}_1'$ colored with $c = |C_{\mathcal{FF}}(\mathbb{B}_1', \mathbb{B}_1' \cap \mathbb{G}_{i-1}')| + 1$ and the next vertices of $\mathbb{B}_1'$ colored with $c_1 \leqslant c$, i.e., we add vertices $x_{i_c}, \ldots, x_{i_{c+1}-1}$ with respective edges according to (5.4). It is easy to observe that the first added vertex, i.e., x_{i_c} is colored by $\mathcal{FF}$ with a new color. Moreover for each $x_1,\ x_2 \in B_1'$ if $C_{\mathcal{FF}}(\mathbb{B}_1', x_1) = C_{\mathcal{FF}}(\mathbb{B}_1', x_2)$ then $C_{\mathcal{FF}}(\mathbb{G}_i', x_1) = C_{\mathcal{FF}}(\mathbb{G}_i', x_2)$. Therefore the algorithm $\mathcal{FF}$ uses only one new color on vertices $x_{i_c}, \ldots, x_{i_{c+1}-1}$.

At the end of the construction, that is, when all vertices of $\mathbb{G}$ have been already analyzed and still $G_m' \neq (G \setminus B_1) \cup B_1'$, we add to $\mathbb{G}_m'$ the remaining vertices of $\mathbb{B}_1'$, i.e., $x_k, \ldots, x_n$ with respective edges according to (5.4).

Now we verify the properties of $\mathbb{G}'$. In this verification we use the fact that all possible edges between B_1' and Q_1 are presented in $\mathbb{G}'$ (as it was for B_1 and Q_1 in $\mathbb{G}$), and that there are no other edges between B_1' and $G_1' \setminus B_1'$ - the old part of $\mathbb{G}'$ (again as it was for $\mathbb{B}_1$ in $\mathbb{G}$).

1. $\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant \chi_{\mathcal{FF}}(\mathbb{G}')$.
   For every vertex $x_i \in G_m' \setminus B_1'$ the set of colors of vertices in $\{x_j \in G' :\ j < i \ \text{ and } x_j E_{\mathbb{G}'} x_i\}$ is the same as the set of colors of vertices in $\{x_j \in G :\ j < i \ \text{ and } x_j E_{\mathbb{G}} x_i\}$. Therefore the vertices of $\mathbb{G}_m' \setminus B_1'$ are

colored by $\mathcal{FF}$ in the same way as in $\mathbb{G} \setminus B_1$, i.e., $C_{\mathcal{FF}}(\mathbb{G}', \mathbb{G}'_m \setminus B'_1) = C_{\mathcal{FF}}(\mathbb{G}, \mathbb{G} \setminus B_1)$. Moreover, note that $C_{\mathcal{FF}}(\mathbb{G}', \mathbb{B}'_1 \setminus \{x_k, \ldots, x_n\}) = C_{\mathcal{FF}}(\mathbb{G}, \mathbb{B}_1)$. Therefore $|C_{\mathcal{FF}}(\mathbb{G})| = |C_{\mathcal{FF}}(\mathbb{G}'_m)| \leqslant |C_{\mathcal{FF}}(\mathbb{G}')|$.

2. $\chi(\mathbb{G}) = \chi(\mathbb{G}')$.
Thanks to all edges between Q_1 and B_1 every optimal coloring of $\mathbb{G}$ is optimal on $\mathbb{G} \setminus B_1$ and on $\mathbb{B}_1$. On the other hand every optimal coloring of $\mathbb{G}' \setminus B'_1$ can be extended to an optimal coloring of $\mathbb{G}'$. Therefore

$$\begin{aligned} \chi(\mathbb{G}) &= |C(\mathbb{Q}_1)| + \max\{|C(\mathbb{G} \setminus B_1) \setminus C(\mathbb{Q}_1)|, |C(\mathbb{B}_1)|\} \\ &= |C(\mathbb{Q}_1)| + \max\{|C(\mathbb{G}' \setminus B'_1) \setminus C(\mathbb{Q}_1)|, |C(\mathbb{B}'_1)|\} = \chi(\mathbb{G}'). \end{aligned}$$

3. $\mathbb{G}'$ is $(\mathbb{C}_4, \mathbb{P}_5)$-free with the smaller buoy number than the one of $\mathbb{G}$.
Suppose there exists an induced path y_1, y_2, y_3, y_4, y_5 in $\mathbb{G}'$. Since the graphs $\mathbb{G}$ and $\mathbb{B}'_1$ are $\mathbb{P}_5$-free, without loss of generality, we may assume that $y_1 \in B'_1$ and $y_5 \in G' \setminus B'_1$. Now, thanks to all edges between $\mathbb{B}'_1$ and $\mathbb{Q}_1$ one can see that $y_2 \in Q_1$ and $y_3, y_4, y_5 \notin Q_1 \cap B'_1$. But then, for every vertex $b \in B_1$, the vertices b, y_2, y_3, y_4, y_5 induce a path in a $\mathbb{P}_5$-free graph $\mathbb{G}$. In the similar way, it is easy to see that if there was an induced cycle $\mathbb{C}_4$ in $\mathbb{G}'$, then there would be some induced cycle with 4 vertices in $\mathbb{G}$.

The graph $\mathbb{G}'$ consists of chordal and $\mathbb{P}_5$-free graph $\mathbb{Q}$, $n-1$ buoys: $\mathbb{B}_2, \ldots, \mathbb{B}_n$ and $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graph $\mathbb{B}'_1$. Therefore, obviously, the buoy number of $\mathbb{G}'$ is smaller than the one of $\mathbb{G}$ as $\mathbb{B}'_1$, being also chordal and $\mathbb{P}_5$-free, can then be incorporated into the $\mathbb{Q}$-part of $\mathbb{G}'$ (defined by Theorem 5.3.10). □

Now we are ready to prove an upper bound for a competitive function for coloring of $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs.

Theorem 5.3.12 *If $\mathbb{G}$ is $(\mathbb{C}_4, \mathbb{P}_5)$-free then*

$$\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant \binom{\chi(\mathbb{G})+1}{2}.$$

Proof. First, assume that a $(\mathbb{C}_4, \mathbb{P}_5)$-free graph $\mathbb{G}$ is connected. We show that there exists a $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graph $\mathbb{G}'$ such that $\chi(\mathbb{G}) = \chi(\mathbb{G}')$ and $\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant \chi_{\mathcal{FF}}(\mathbb{G}')$. Then from Theorem 5.3.8 we get

$$\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant \chi_{\mathcal{FF}}(\mathbb{G}') \leqslant \binom{\chi(\mathbb{G}')+1}{2} = \binom{\chi(\mathbb{G})+1}{2}.$$

We know that there exists a partition of $\mathbb{G}$ into $\mathbb{Q}$ and $\mathbb{B}$ which satisfies the itemized properties from Theorem 5.3.10. If $\mathbb{B} = \emptyset$ then $\mathbb{G}' := \mathbb{G}$ and we are done. Otherwise, Lemma 5.3.11 supplies a graph $\mathbb{G}'$ with the smaller buoy number and for which $\chi(\mathbb{G}) = \chi(\mathbb{G}')$, $\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant \chi_{\mathcal{FF}}(\mathbb{G}')$. Repeating this lemma sufficiently many times we end up with a graph $\mathbb{G}^{(n)}$, which does not contain any buoys and for which $\chi(\mathbb{G}) = \chi(\mathbb{G}^{(n)})$ and $\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant \chi_{\mathcal{FF}}(\mathbb{G}^{(n)})$. The graph $\mathbb{G}^{(n)}$ is $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free, thus it can serve as $\mathbb{G}'$.

If $\mathbb{G}$ is not connected, it consists of connected $(\mathbb{C}_4, \mathbb{P}_5)$-free components. The behavior of $\mathcal{FF}$ ensures us that there exists a component, say $\mathbb{G}_i$, for which $C_{\mathcal{FF}}(\mathbb{G}_i) = C_{\mathcal{FF}}(\mathbb{G})$. Since $\chi_{\mathcal{FF}}(\mathbb{G}_i) \leqslant \binom{\chi(\mathbb{G}_i)+1}{2}$, thus $\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant \binom{\chi(\mathbb{G})+1}{2}$. □

Combining Theorems 5.3.2 and 5.3.12 we get

Corollary 5.3.13 *The on-line coloring algorithm with the First-Fit strategy for $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs is the best possible and it has the competitive function $\binom{\chi(\mathbb{G})+1}{2}$, where $\mathbb{G}$ is an input graph.* □

As we have seen, the competitive function for coloring of $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graphs is $\binom{\chi(\mathbb{G})+1}{2}$, where $\mathbb{G}$ is an input graph. The forcing graph constructed in the proof of Theorem 5.3.2 is not presented in a connected way. To obtain a lower bound for a competitive function for $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graphs presented in a connected way, we slighty modify the construction described in the proof of Theorem 5.3.2.

Proposition 5.3.14 *For every on-line coloring algorithm $\mathcal{A}$ and for every positive integer χ there exists a $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graph $\mathbb{G}$ with $\chi(\mathbb{G}) = \chi$ presented in a connected way and such that*

$$\chi_{\mathcal{A}}(\mathbb{G}) \geqslant \binom{\chi(\mathbb{G})}{2} + 1.$$

Proof. To obtain this lower bound, the spoiler builds a $\mathbb{G}$ with $\chi(\mathbb{G}) = \chi$ which is similar to the one described in the proof of Theorem 5.3.2. First, the spoiler presents a single vertex x. The algorithm $\mathcal{A}$ colors it with some color c. Next, the spoiler constructs a graph $\mathbb{G}_{\chi-1}$ as in the proof of Theorem 5.3.2. To maintain connectivity he joins every vertex of $\mathbb{G}_{\chi-1}$ to x. The algorithm $\mathcal{A}$ is forced to use $\chi_{\mathcal{A}}(\mathbb{G}_{\chi-1})$ new colors to color $G = \{x\} \cup G_{\chi-1}$. Note that the new graph $\mathbb{G}$ is χ-colorable and it is still $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free. Therefore we have

$$\chi_{\mathcal{A}}(\mathbb{G}) = \chi_{\mathcal{A}}(\mathbb{G}_{\chi-1}) + 1 \geqslant \binom{\chi(\mathbb{G}_{\chi-1}) + 1}{2} + 1 = \binom{\chi(\mathbb{G})}{2} + 1. \qquad \square$$

Enlarging the family considered in Theorem 5.3.14 we get

Corollary 5.3.15 *For every on-line coloring algorithm $\mathcal{A}$ and for every positive integer χ there exists a $(\mathbb{C}_4, \mathbb{P}_5)$-free graph $\mathbb{G}$ with $\chi(\mathbb{G}) = \chi$ presented in a connected way and such that*

$$\chi_{\mathcal{A}}(\mathbb{G}) \geqslant \binom{\chi(\mathbb{G})}{2} + 1. \qquad \square$$

In the obvious way we obtain

Corollary 5.3.16 *For every on-line coloring algorithm $\mathcal{A}$ and for every positive integer χ there exists a $\mathbb{P}_5$-free graph $\mathbb{G}$ with $\chi(\mathbb{G}) = \chi$ presented in a connected way and such that*

$$\chi_{\mathcal{A}}(\mathbb{G}) \geqslant \binom{\chi(\mathbb{G})}{2} + 1. \qquad \square$$

There is a gap between lower and upper bounds for a competitive function for both $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs and $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graphs presented in a connected way. Namely,

$$\binom{\chi(\mathbb{G})}{2} + 1 \leqslant \ldots \leqslant \binom{\chi(\mathbb{G})+1}{2}.$$

The problem of determining the exact competitiveness for coloring of $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs and $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graphs presented in a connected way remains open.

5.3.2 On-line clique covering of $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs

Before we show a competitive function for clique covering of $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs, consider the smaller family of $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graphs. One can easily see that the $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graphs are chordal. From Corollary 2.2.5 we know that the competitive function for clique covering of chordal graphs presented in a connected way is $2 \cdot v(\mathbb{G}) - 1$, where $\mathbb{G}$ is an input graph. Obviously, it is also an upper bound for a competitive function for $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graphs. At the same time, the graphs presented in the proof of Theorem 2.1.9 are $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free. It is easy to see that they were presented in a connected way, therefore we get

Corollary 5.3.17 *The on-line clique covering algorithm with the First-Fit strategy for $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graphs presented in a connected way is the best possible and it has the competitive function $2 \cdot v(\mathbb{G}) - 1$, where $\mathbb{G}$ is an input graph.* $\square$

Now, we show that $2 \cdot v(\mathbb{G}) - 1$ is also the competitive function for $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs. The subgraph $\mathbb{C}_5$ is not a useful structure for the spoiler. To prove this result we will need the structural property of $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs presented in Theorem 5.3.10. It says that a $(\mathbb{C}_4, \mathbb{P}_5)$-free graph can be partitioned into a chordal graph and disjoint complete buoys. Remind that a complete buoy is a graph composed of 5 cliques, $\mathbb{A}_0, \ldots, \mathbb{A}_4$, such that $xy \in E_{\mathbb{G}}$ whenever $x \in A_i$, $y \in A_{i+1}$ and all other edges of $\mathbb{G}$ have both their ends in the same A_i. First of all, we prove an easy lemma.

Lemma 5.3.18 *If $\mathbb{B}$ is a complete buoy then $v_{\mathcal{FF}}(\mathbb{B}) = v(\mathbb{B}) = 3$.*

Proof. Let $\mathbb{A}_0, \ldots, \mathbb{A}_4$ be cliques to which a complete buoy $\mathbb{B}$ is partitioned, according to the definition of a buoy. Obviously, $v(\mathbb{B}) = 3$. We show that

$v_{\mathcal{FF}}(\mathbb{B}) = 3$. Suppose that there exists $x \in B$ to which the $\mathcal{FF}$ algorithm assigns the clique with number 4. Thus there exist vertices $x_1, x_2, x_3 \in B$ such that $cl_{\mathcal{FF}}(x_i) = i$ for $i = 1, 2, 3$ and $\{xx_1, xx_2, xx_3\} \notin E_{\mathbb{B}}$. Without loss of generality assume that $x \in A_0$, and $x_1, x_2 \in A_2$. Then there exists a vertex $y \in B$ such that $cl_{\mathcal{FF}}(y) = cl_{\mathcal{FF}}(x_1)$ and $yx_2 \notin E_{\mathbb{B}}$. Such vertex y is not in $A_1 \cup A_2 \cup A_3$, therefore $yx_1 \notin E_{\mathbb{B}}$, a contradiction. □

Now we show that the upper bound for the competitiveness of clique covering problem for $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs is also $2 \cdot v(\mathbb{G}) - 1$, where $\mathbb{G}$ is an input graph.

Theorem 5.3.19 *If $\mathbb{G}$ is $(\mathbb{C}_4, \mathbb{P}_5)$-free then*

$$v_{\mathcal{FF}}(\mathbb{G}) \leqslant 2 \cdot v(\mathbb{G}) - 1.$$

Proof. The idea of the proof is very similar to the one used in the proof of Theorem 5.3.12. First, assume that a $(\mathbb{C}_4, \mathbb{P}_5)$-free $\mathbb{G}$ is connected. To prove the theorem we show that there exists a $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graph $\mathbb{G}'$ such that $v(\mathbb{G}) = v(\mathbb{G}')$ and $v_{\mathcal{FF}}(\mathbb{G}) \leqslant v_{\mathcal{FF}}(\mathbb{G}')$. From Corollary 5.3.17 we will get $v_{\mathcal{FF}}(\mathbb{G}) \leqslant v_{\mathcal{FF}}(\mathbb{G}') \leqslant 2 \cdot v(\mathbb{G}') - 1 = 2 \cdot v(\mathbb{G}) - 1$.

Let $\mathbb{Q}$ and $\mathbb{B}_1, \ldots, \mathbb{B}_n$ be a decomposition supplied for $\mathbb{G}$ by Theorem 5.3.10. If $G = Q$ then $\mathbb{G}' := \mathbb{G}$ and the construction is done. Therefore assume that $n \geqslant 1$. First, we proceeds in the way similar to the proof of Lemma 5.3.11. Since $\mathbb{B}_1$ is a complete buoy, Lemma 5.3.18 gives that $v_{\mathcal{FF}}(\mathbb{B}_1) = v(\mathbb{B}_1) = 3$. Therefore, one can easily find a $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graph $\mathbb{B}_1'$ such that $v(\mathbb{B}_1) = v(\mathbb{B}_1')$ and $v_{\mathcal{FF}}(\mathbb{B}_1) \leqslant v_{\mathcal{FF}}(\mathbb{B}_1')$. Now supposing that $\mathbb{B}_1$ and $\mathbb{B}_1'$ are disjoint we construct $\mathbb{G}^{(1)} = ((G \setminus B_1) \cup B_1', E_{\mathbb{G}^{(1)}})$, where

$$E_{\mathbb{G}^{(1)}} = \{xy : \ xy \in E_{\mathbb{G}} \ \text{ and } \ x, y \notin B_1\} \cup E_{\mathbb{B}_1'} \cup \{xy : \ x \in Q_1 \ \text{ and } \ y \in B_1'\}.$$

and $Q_1 = N_{\mathbb{G}}(B_1) \setminus B_1$. In other words, in place of vertices from $\mathbb{B}_1$, we present respective vertices from $\mathbb{B}_1'$. Additionally, we join vertices of $\mathbb{B}_1'$ to the same vertices which are joined to the vertices of $\mathbb{B}_1$, i.e., $(N_{\mathbb{G}}(x) \setminus B_1) = (N_{\mathbb{G}^{(1)}}(x') \setminus B_1')$ for each $x \in B_1$ and $x' \in B_1'$. The on-line presentation of $\mathbb{G}^{(1)}$ is defined in the very similar way to the presentation of $\mathbb{G}'$ in the proof of Lemma 5.3.11.

Thanks to all edges between Q_1 and B_1 and between Q_1 and B_1' the graph $\mathbb{G}^{(1)}$ can be covered optimally with $v(\mathbb{G})$ cliques. It is also easy to see that $Cl_{\mathcal{FF}}(\mathbb{G}, \mathbb{B}_1) \subseteq Cl_{\mathcal{FF}}(\mathbb{G}^{(1)}, \mathbb{B}_1')$. Moreover, $Cl_{\mathcal{FF}}(\mathbb{G}^{(1)}, \mathbb{G}^{(1)} \setminus B_1') = Cl_{\mathcal{FF}}(\mathbb{G}, \mathbb{G} \setminus B_1)$. Therefore $v_{\mathcal{FF}}(\mathbb{G}) \leqslant v_{\mathcal{FF}}(\mathbb{G}^{(1)})$. Note that $\mathbb{G}^{(1)}$ is $(\mathbb{C}_4, \mathbb{P}_5)$-free and it has the smaller buoy number that the one of $\mathbb{G}$.

Repeating the previous technique sufficiently many times we end up with a graph $\mathbb{G}^{(n)}$, which does not contain any buoys, and for which $v(\mathbb{G}) = v(\mathbb{G}^{(n)})$ and $v_{\mathcal{FF}}(\mathbb{G}) \leqslant v_{\mathcal{FF}}(\mathbb{G}^{(n)}) \leqslant 2 \cdot v(\mathbb{G}) - 1$. Note that $\mathbb{G}^{(n)}$ does not contain $\mathbb{C}_5$. Indeed, each $\mathbb{C}_5$ in $\mathbb{G}$ has been completely contained in some buoy and this buoy was replaced by a $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graph. Moreover, replacing $\mathbb{B}_i$ by $\mathbb{B}_i'$ does not create any of $\mathbb{C}_4$, $\mathbb{C}_5$ or $\mathbb{P}_5$. Therefore the graph $\mathbb{G}^{(n)}$ is $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free and it can serve as $\mathbb{G}'$.

If $\mathbb{G}$ is not connected, it consists of connected $(\mathbb{C}_4, \mathbb{P}_5)$-free components $\mathbb{G}_1, \ldots, \mathbb{G}_m$. Since $v_{\mathcal{FF}}(\mathbb{G}_i) \leqslant 2v(\mathbb{G}_i) - 1$ for each $\mathbb{B}_i$, the behavior of $\mathcal{FF}$ ensures as that $v_{\mathcal{FF}}(\mathbb{G}) \leqslant 2v(\mathbb{G}) - 1$. □

Combining Theorems 2.1.9 and 5.3.19 we get

Corollary 5.3.20 *The on-line clique covering algorithm with the First-Fit strategy for $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs presented in a connected way is the best possible and it has the competitive function $2 \cdot v(\mathbb{G}) - 1$, where $\mathbb{G}$ is an input graph.* □

5.4 $(\mathbb{C}_4, 2\mathbb{K}_2)$-free graphs

As we have seen in Section 5.3.1, the competitive function for the coloring of $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs is quadratic, therefore there is no on-line coloring algorithm with a competitive ratio for this family. Now, consider the smaller family of $(\mathbb{C}_4, 2\mathbb{K}_2)$-free graphs. Recall that $2\mathbb{K}_2$ denotes two disjoint edges. We show that such family is very simple and easy for on-line coloring and clique covering algorithms. The spoiler is able to force at most one additional color during coloring and also at most one additional clique during clique covering on graphs from this family. In order to prove these results we need the following structural property of the $(\mathbb{C}_4, 2\mathbb{K}_2)$-free graphs.

Theorem 5.4.1 (Z.Blásik, M.Hujter, M.Pluhár, Z.Tuza [6]) *A graph $\mathbb{G}$ is $(\mathbb{C}_4, 2\mathbb{K}_2)$-free if and only if there exists a partition $G = I \cup K \cup C$ with the following properties:*

(i) $\mathbb{I} := \mathbb{G}|_I$ *is an independent set,*

(ii) $\mathbb{K} := \mathbb{G}|_K$ *is a clique,*

(iii) $C = \emptyset$ *or* $\mathbb{C} := \mathbb{G}|_C$ *is a cycle with 5 vertices,*

(iv) $v_1v_3 \notin E_{\mathbb{G}}$ *and* $v_2v_3 \in E_{\mathbb{G}}$ *whenever* $v_1 \in I$, $v_2 \in K$, $v_3 \in C$. □

Now, with the help of Theorem 5.4.1, we prove an upper bound for competitiveness for $(2\mathbb{K}_2, \mathbb{C}_4)$-free graphs.

Theorem 5.4.2 *If $\mathbb{G}$ is $(2\mathbb{K}_2, \mathbb{C}_4)$-free then $\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant \chi(\mathbb{G}) + 1$.*

Proof. Let $G = I \cup K \cup C$ be a decomposition supplied for $\mathbb{G}$ by Theorem 5.4.1 and let $k = |K|$. Now, consider two cases: $\mathbb{C} = \emptyset$ or $\mathbb{C} = \mathbb{C}_5$.

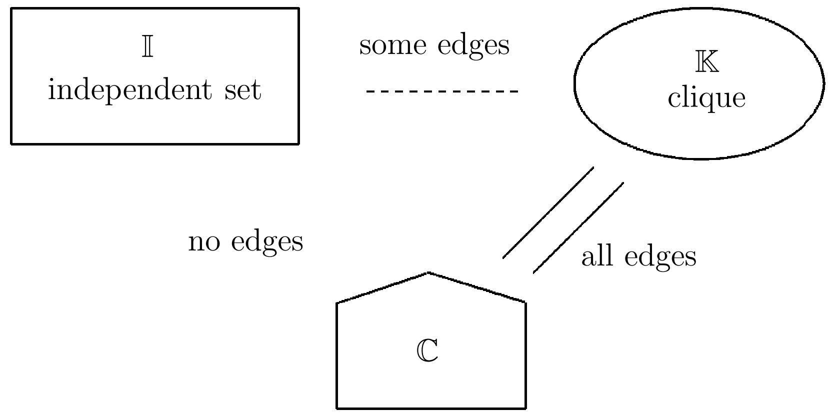


Figure 5.5: $(\mathbb{C}_4, 2\mathbb{K}_2)$-free graphs. $\mathbb{C} = \emptyset$ or $\mathbb{C} = \mathbb{C}_5$

1. Suppose that $\mathbb{C} = \mathbb{C}_5$. We show that, in this case, in fact $\chi_{\mathcal{FF}}(\mathbb{G}) = \chi(\mathbb{G})$. Note that $\chi(\mathbb{G}) = |K|+3$. Therefore, it is sufficient to show that $\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant |K| + 3$.

   Suppose that there exists $v \in G$ such that $c_{\mathcal{FF}}(v) = k + 4$. It has at least $k + 3$ neighbors with distinct colors from set $\{1, \dots, k + 3\}$. If $v \in I$ or $v \in C_5$, then it has at most k or $k + 2$ neighbors, respectively. Therefore $v \in K$. By our assumption on v we know that there is $N \subseteq N_{\mathbb{G}}(v)$ such that $|N| = k + 3$, $|C_{\mathcal{FF}}(N)| = \{1, \dots, k + 3\}$ and $N \cap K = K \setminus \{v\}$. Therefore $N \cap (I \cup C) = \{a_1, a_2, a_3, a_4\}$ and suppose that $c_{\mathcal{FF}}(a_1) \leqslant \ldots \leqslant c_{\mathcal{FF}}(a_4)$. This means that a_4 had to get earlier neighbors b_1, b_2, b_3 with $c_{\mathcal{FF}}(b_i) = c_{\mathcal{FF}}(a_i)$. As each point in $I \cup C$ has at most 2 neighbors in $I \cup C$, at least one b_i is in K. Thus $c_{\mathcal{FF}}(a_i) = c_{\mathcal{FF}}(b_i) \in C_{\mathcal{FF}}(\mathbb{K})$ for some i, which can happen only if $b_i = v$. However, $k + 4 = c_{\mathcal{FF}}(v) = c_{\mathcal{FF}}(a_i) \leqslant k + 3$, a contradiction.

2. Now, consider the case $\mathbb{C} = \emptyset$. Without loss of generality assume that $\mathbb{K}$ is a maximal clique. Then $\chi(\mathbb{G}) = |K| = k$. We show that $\chi_{\mathcal{FF}}(\mathbb{G}) \leqslant \chi(\mathbb{G}) + 1$. Suppose that there exists v such that $c_{\mathcal{FF}}(v) = k + 2$. Since vertices of $\mathbb{I}$ have at most k neighbors, $v \in K$. Let $N \subseteq N_{\mathbb{G}}(v)$ be such that $|N| = k + 1$, $C_{\mathcal{FF}}(N) = \{1, \dots, k + 1\}$ and $N \cap K = K \setminus \{v\}$. Therefore $|N \cap I| = 2$. Without loss of generality we may assume that $c_{\mathcal{FF}}(a_1) < c_{\mathcal{FF}}(a_2)$ for some $a_1, a_2 \in N \cap I$. Again there must be a neighbor b_1 of a_2 with $c_{\mathcal{FF}}(b_1) = c_{\mathcal{FF}}(a_1)$. Since $b_1 \in K$ then $c_{\mathcal{FF}}(a_1) = c_{\mathcal{FF}}(b_1) \in C_{\mathcal{FF}}(K)$, implying $b_1 = v$ and $c_{\mathcal{FF}}(a_1) = k + 2$, a contradiction. □

Now, we indicate an easy construction of $(2\mathbb{K}_2, \mathbb{C}_4)$-free graphs $\mathbb{G}$ forcing coloring algorithms to actually use $\chi(\mathbb{G}) + 1$ colors.

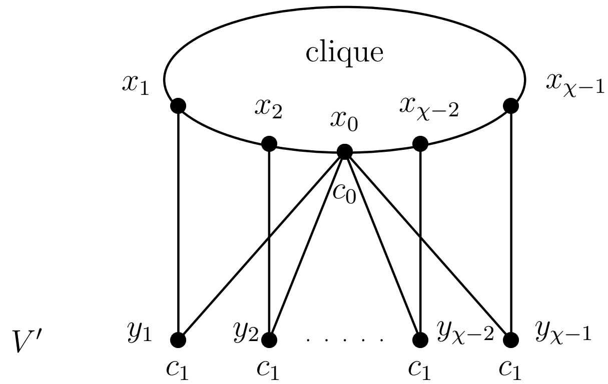


Figure 5.6: The forcing $(2\mathbb{K}_2, \mathbb{C}_4)$-free graph $\mathbb{G}$, case 1.

Proposition 5.4.3 *For every on-line coloring algorithm $\mathcal{A}$ and for every positive integer $\chi \geqslant 3$ there exists a $(2\mathbb{K}_2, \mathbb{C}_4)$-free graph $\mathbb{G}$ with $\chi(\mathbb{G}) = \chi$ presented in a connected way for which $\chi_{\mathcal{A}}(\mathbb{G}) \geqslant \chi(\mathbb{G}) + 1$.*

Proof. The construction of a forcing graph $\mathbb{G}$ with $\chi(\mathbb{G}) = \chi$ is pictured on Figure 5.6. First, the spoiler presents a vertex x_0. The algorithm $\mathcal{A}$ colors it with c_0. Next, the spoiler presents a lot of vertices joined to the vertex x_0 until there exists a subset V' such that $|V'| = \chi - 1$ and $|C_{\mathcal{A}}(V')| = 1$ or until $\mathcal{A}$ uses $\chi + 1$ colors. In the second case, in order to construct a graph with chromatic number χ, it suffices to present a clique with $\chi - 1$ vertices $x_1, \ldots, x_{\chi-1}$ and join each x_i to x_0.

In the first case, let $V' = \{y_1, \ldots, y_{\chi-1}\}$ and $C_{\mathcal{A}}(V') = \{c_1\}$. Next, the spoiler presents a clique with $\chi - 1$ vertices $x_1, \ldots, x_{\chi-1}$. He joins each x_i to vertices x_0 and y_i. The algorithm $\mathcal{A}$ is forced to color each x_i with color another than c_0 or c_1. Thus $\chi_{\mathcal{A}}(\mathbb{G}) \geqslant 2 + \chi - 1 = \chi + 1$ as required. However, we can color $\mathbb{G}$ optimally with χ colors if we first color vertices $x_0, x_1, \ldots, x_{\chi-1}$ and then the other ones. It is easy to see that $\mathbb{G}$ is $(2\mathbb{K}_2, \mathbb{C}_4)$-free. □

Summing up we have

Corollary 5.4.4 *The on-line coloring algorithm with the First-Fit strategy for $(2\mathbb{K}_2, \mathbb{C}_4)$-free graphs presented in a connected way is the best possible and it has the competitive function $\chi(\mathbb{G}) + 1$, where $\mathbb{G}$ is an input graph.* □

Since $2\mathbb{K}_2$ is a complement of $\mathbb{C}_4$, we immediately get that the upper bound for a competitive function for clique covering of $(\mathbb{C}_4, 2\mathbb{K}_2)$-free graphs is the same as for coloring these graphs, that is, $v(\mathbb{G}) + 1$. The complements of the forcing graphs constructed in the proof of Proposition 5.4.3 force clique covering algorithms to use $v(\mathbb{G}) + 1$, but they are not presented in a connected way. The following proposition describes strategy of the spoiler to construct forcing $(\mathbb{C}_4, 2\mathbb{K}_2)$-free graphs presented in a connected way.

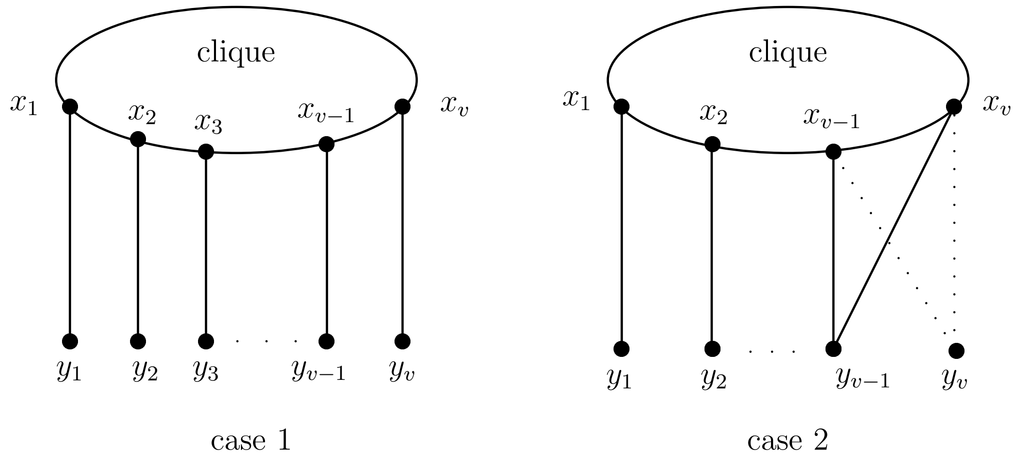


Figure 5.7: The forcing $(2\mathbb{K}_2, \mathbb{C}_4)$-free graph $\mathbb{G}$.

Proposition 5.4.5 *For every on-line coloring algorithm $\mathcal{A}$ and for every positive integer $v \geqslant 2$ there exists a $(2\mathbb{K}_2, \mathbb{C}_4)$-free graph $\mathbb{G}$ with $v(\mathbb{G}) = v$ presented in a connected way for which $v_{\mathcal{A}}(\mathbb{G}) \geqslant v(\mathbb{G}) + 1$.*

Proof. First, the spoiler presents a clique $\mathbb{K}$ with v vertices $x_1, \ldots x_v$ in some order. The algorithm $\mathcal{A}$ covers these vertices with one or more cliques. There are two cases:

1. There exist at least two vertices, say x_1 and x_2, covered with the same clique, that is, $cl_{\mathcal{A}}(x_1) = cl_{\mathcal{A}}(x_2) = c$.
   Then, the spoiler presents vertices $y_1, \ldots, y_v$ and joins each y_i to x_i. The algorithm $\mathcal{A}$ is forced to use two completely new cliques to cover y_1 and y_2, i.e., $cl_{\mathcal{A}}(y_1) \neq c$, $cl_{\mathcal{A}}(y_2) \neq c$ and $cl_{\mathcal{A}}(y_1) \neq cl_{\mathcal{A}}(y_2)$. Moreover, it covers vertices $y_3, \ldots, y_n$ with at least $v-2$ cliques other than c, $cl_{\mathcal{A}}(y_1)$ and $cl_{\mathcal{A}}(y_2)$. Summing up, $\mathcal{A}$ uses at least $v+1$ cliques to cover $\{x_1, \ldots, x_v, y_1, \ldots, y_v\}$, while presented vertices can be covered optimally with v cliques of the form $\{x_i, y_i\}$.

2. Otherwise, $\mathcal{A}$ covers vertices $x_1, \ldots, x_v$ with v cliques.
   Then, the spoiler presents vertices $y_1, \ldots, y_{v-2}$ and joins each y_i to x_i. Next, the spoiler presents a vertex y_{v-1} with the edges $y_{v-1}x_{v-1}$ and $y_{v-1}x_v$. If y_{v-1} is covered with the clique $cl_{\mathcal{A}}(x_{v-1})$ then the spoiler presents vertex y_v with an edge $y_v x_{v-1}$, otherwise, the spoiler presents vertex y_v with an edge $y_v x_v$ (see Figure 5.7). Note that at least one of the vertices y_{v-1}, y_v is covered with a new clique, i.e., $C_{\mathcal{A}}(\{y_{v-1}, y_v\}) \nsubseteq C_{\mathcal{A}}(\{x_1, \ldots, x_v\})$. Summing up, $\mathcal{A}$ uses at least $v+1$ cliques to cover $\{x_1, \ldots, x_v, y_1, \ldots, y_v\}$, while presented vertices can be covered optimally with v cliques: $\{x_1, y_1\}, \ldots \{x_v, y_v\}$ or $\{x_1, y_1\}, \ldots \{x_{v-2}, y_{v-2}\}, \{x_{v-1}, y_v\}, \{x_v, y_{v-1}\}$. □

Summing up we have

Corollary 5.4.6 *The on-line clique covering algorithm with the First-Fit strategy for* $(2\mathbb{K}_2, \mathbb{C}_4)$*-free graphs presented in a connected way is the best possible and it has the competitive function* $v(\mathbb{G}) + 1$*, where* $\mathbb{G}$ *is an input graph.* □

5.5 Conclusions

As we have seen in this chapter, the path $\mathbb{P}_4$ is a forcing structure in a very strong sense: the competitive function for coloring of $\mathbb{P}_4$-free graphs is $\chi(\mathbb{G})$ (Proposition 5.1.1) and also the competitive function for clique covering of $\mathbb{P}_4$-free graphs is $v(\mathbb{G})$ (Corollary 5.1.2). On the other hand, the paths $\mathbb{P}_k$ for $k \geqslant 6$ are forcing structures neither for coloring (Corollary 5.1.5) nor for clique covering (Corollary 5.2.2). The remaining path $\mathbb{P}_5$ is a forcing structure only for coloring. As we have seen, determining the exact competitive function for coloring of $\mathbb{P}_5$-free graphs is still open. It is between $\binom{\chi(\mathbb{G})+1}{2}$ (Corollary 5.3.4) and $\frac{4^{\chi(\mathbb{G})-1}}{3}$ [22].

Problem 5 *Is there an on-line coloring algorithm with a polynomial competitive function for* $\mathbb{P}_5$*-free graphs?*

Problem 6 *Is there a strategy for the spoiler to construct* $\mathbb{P}_5$*-free graphs forcing on-line coloring algorithms to use more than* $\binom{\chi(\mathbb{G})+1}{2}$ *colors?*

As we have seen, the competitive function for coloring of $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs is exactly $\binom{\chi(\mathbb{G})+1}{2}$ (Corollary 5.3.13). Therefore each forcing graph asked for in Problem 6 contains $\mathbb{C}_4$ as an induced subgraph. There would be interesting to search for other subgraphs which are contained in each forcing graph for Problem 6. For example, one may ask about cycle $\mathbb{C}_5$, i.e., about a competitive function for $(\mathbb{C}_5, \mathbb{P}_5)$-free graphs. Obviously, it is between $\binom{\chi(\mathbb{G})+1}{2}$ and $\frac{4^{\chi(\mathbb{G})-1}}{3}$.

Problem 7 *Is there an on-line coloring algorithm with a polynomial competitive function for* $(\mathbb{C}_5, \mathbb{P}_5)$*-free graphs?*

In this chapter we have also analyzed graphs presented in a connected way. It is a small gap between lower and upper bounds for competitive functions for coloring $(\mathbb{C}_4, \mathbb{P}_5)$-free and $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graphs presented in a connected way. They are bounded by $\binom{\chi(\mathbb{G})}{2} + 1$ (Proposition 5.3.14) and $\binom{\chi(\mathbb{G})+1}{2}$ (Theorem 5.3.12).

Problem 8 *What are the competitive functions for coloring of* $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$*-free graphs and* $(\mathbb{C}_4, \mathbb{P}_5)$*-free graphs?*

In this chapter we have also analyzed the competitiveness for clique covering of $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graphs and $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs presented in a connected way. The competitive functions for this problem are $2 \cdot v(\mathbb{G}) - 1$ for both families (Corollaries 5.3.17 and 5.3.20). In Chapter 4 we have seen that there is a huge gap between known lower and upper bounds for a competitive function for $\mathbb{C}_4$-free graphs (Problem 4). From the competitiveness for covering of $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs we get that each graph forcing clique covering algorithms to use at least quadratic cliques contains $\mathbb{P}_5$ as an induced subgraph. As in the problem of coloring $\mathbb{P}_5$-free graph there would be interesting to search for subgraphs which are contained in $\mathbb{C}_4$-free graphs forcing clique covering algorithms.

Chapter 6

Conclusions

We have seen in Corollary 2.1.2 that even on very simple graphs, like trees, there is no competitive on-line coloring algorithm. This means that the spoiler can cheat with no bounds even using very simple tricks. Actually, the tricks used in trees are the graphs $\mathbb{K}_{1,t}$. Indeed, in Proposition 4.1.1 we have seen that forbidding $\mathbb{K}_{1,t}$ ties the spoiler's hands and he cannot cheat more than $(t-1) \cdot \chi(\mathbb{G}) - t + 2$. Also in Proposition 5.1.1 and Theorems 2.1.8, 3.2.1, 5.3.19, 5.4.2 we have observed that there are many different structures which are:

- on one hand, useful in forcing many colors
- and on the other hand, necessary for such forcing.

Summarizing these results we obtain the following.

Theorem 6.1 *Let an infinite family $\mathcal{F}$ of graphs have an on-line coloring algorithm with a competitive ratio at least $c > 1$. Then there are 4 infinite families: $\mathcal{P}_4$, $\mathcal{C}_4^*$, $\mathcal{K}_{1,c}$, $\mathcal{I}_{2c-1} \subseteq \mathcal{F}$ such that:*

- *each graph in $\mathcal{P}_4$ contains $\mathbb{P}_4$ as an induced subgraph* (Prop. 5.1.1),
- *each graph in $\mathcal{C}_4^*$ contains $\mathbb{C}_4$ or $2\mathbb{K}_2$ as an induced subgraph* (Th. 5.4.2),
- *each graph in $\mathcal{K}_{1,c}$ contains $\mathbb{K}_{1,\lfloor c \rfloor}$ as an induced subgraph* (Prop. 4.1.1),
- *each graph in $\mathcal{I}_{2c-1}$ contains $\mathbb{I}_{\lfloor 2c-1 \rfloor}$ as an induced subgraph* (Th. 3.2.1).

Moreover if $c > 2$ then $\mathcal{F}$ contains additionally infinite families $\overline{\mathcal{C}}_{k\geqslant 4}$ and $\overline{\mathcal{P}}_5^*$ such that

- *each graph in $\overline{\mathcal{C}}_{k\geqslant 4}$ contains $\overline{\mathbb{C}}_k$ as an induced subgraph for some $k \geqslant 4$, i.e., complements of none graph in $\overline{\mathcal{C}}_{\geqslant 4}$ is chordal* (Th. 2.1.8),
- *each graph in $\overline{\mathcal{P}}_5^*$ contains $\overline{\mathbb{P}}_5$ or $2\mathbb{K}_2$ as an induced subgraph* (Th. 5.3.19). □

A similar theorem can be formed for clique covering:

Theorem 6.2 *Let an infinite family $\mathcal{F}$ of graphs have an on-line covering algorithm with a competitive ratio at least $c > 1$. Then there are 4 infinite families: $\mathcal{P}_4$, $\mathcal{C}_4^*$, $\overline{\mathcal{K}}_{1,c}$, $\mathcal{K}_{2c-1} \subseteq \mathcal{F}$ such that:*

- *each graph in $\mathcal{P}_4$ contains $\mathbb{P}_4$ as an induced subgraph* (Prop. 5.1.1),
- *each graph in $\mathcal{C}_4^*$ contains $\mathbb{C}_4$ or $2\mathbb{K}_2$ as an induced subgraph* (Th. 5.4.2),
- *each graph in $\overline{\mathcal{K}}_{1,c}$ contains $\mathbb{K}_1 + \mathbb{K}_{\lfloor c \rfloor}$ (i.e., $\overline{\mathbb{K}}_{1,\lfloor c \rfloor}$) as an induced subgraph* (Prop. 4.1.1),
- *each graph in $\mathcal{K}_{2c-1}$ contains $\mathbb{K}_{\lfloor 2c-1 \rfloor}$ as an induced subgraph* (Th. 3.2.1).

Moreover if $c > 2$ then $\mathcal{F}$ contains additionally infinite families $\mathcal{C}_{k\geqslant 4}$ and $\mathcal{P}_5^*$ such that

- *each graph in $\mathcal{C}_{k\geqslant 4}$ contains $\mathbb{C}_k$ as an induced subgraph for some $k \geqslant 4$, i.e., no graph in $\mathcal{C}_{\geqslant 4}$ is chordal* (Th. 2.1.8),
- *each graph in $\mathcal{P}_5^*$ contains $\mathbb{C}_4$ or $\mathbb{P}_5$ as an induced subgraph* (Th. 5.3.19). □

The lists of forbidden structures presented in Theorems 6.1 and 6.2 do not seem to be full. This obviously leads to the following project-oriented problem.

Problem 0 Classify all forcing structures for on-line coloring and clique covering problems.

Below we list other problems that were left open in this dissertation.

Problem 1 (for discussion see Chapter 3) Is there an on-line clique covering algorithm for $\mathbb{K}_s$-free graphs with a competitive function better than $\frac{s}{2} \cdot v(\mathbb{G})$?

Problem 2 (for discussion see Chapter 3) Is there an on-line clique covering algorithm for s-colorable graphs with a competitive function better than $\frac{s+1}{2} \cdot v(\mathbb{G})$?

Problem 3 (for discussion see Chapter 3) Is there an on-line clique covering algorithm for planar graphs presented in a connected way with a competitive ratio lower than $\frac{5}{2}$?

Problem 4 (for discussion see Chapter 4) Is there an on-line clique covering algorithm with a competitive function better than exponential for $\mathbb{K}_{2,2}$-free graphs?

Problem 5 (for discussion see Chapter 5) Is there an on-line coloring algorithm with a polynomial competitive function for $\mathbb{P}_5$-free graphs?

Problem 6 (for discussion see Chapter 5) Is there a strategy for the spoiler to construct $\mathbb{P}_5$-free graphs forcing on-line coloring algorithms to use more than $\binom{\chi(\mathbb{G})+1}{2}$ colors?

Problem 7 (for discussion see Chapter 5) Is there an on-line coloring algorithm with a polynomial competitive function for $(\mathbb{C}_5, \mathbb{P}_5)$-free graphs?

Problem 8 (for discussion see Chapter 5) What are the competitive functions for coloring of $(\mathbb{C}_4, \mathbb{C}_5, \mathbb{P}_5)$-free graphs and $(\mathbb{C}_4, \mathbb{P}_5)$-free graphs?

Bibliography

[1] G.Bascó, Z.Tuza, Dominating cliques in $\mathbb{P}_5$-free graphs, Periodica Math. Hungar. 21 (1990), 303-308.

[2] D.R.Bean, Effective coloration, The Journal of Symbolic Logic 41 (1976), 469-480.

[3] C.Berge, Les problèmes de colorations en théorie des graphes, Publications de l'Institut de Statistique de l'Universit de Paris 9 (1960), 123-160.

[4] C.Berge, Färbung von Graphen, deren sämtliche bzw, deren ungerade Kreise starr sind, In Wissenschaftliche Zeitschrift. Martin Luther Universitat Halle-Wittenberg 10 (1961), 114-115.

[5] A.Brandstädt, V.B.Le, J.P.Spinrad, Graph Classes: A survey, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (1999).

[6] Z.Blásik, M.Hujter, M.Pluhár, Z.Tuza, Graphs with no induced $\mathbb{C}_4$ and $2\mathbb{K}_2$, Discrete Mathematics 115 (1993), 51-55.

[7] A.Capponi, C.Pilloto, On-line coloring and on-line partitioning of graphs, manusricpt, 2003.

[8] M.Chrobak and M.Ślusarek, On some packing problems relating to Dynamical Storage Allocation, Informatique Theorique et Applications 22 (1988), 487-499.

[9] M.Chudnovsky, N.Robertson, P.D.Seymour, R.Thomas, Progress on prefect graphs, Mathematical Programming B 97 (2003), 405-422.

[10] M.Cozzens, L.Kelleher, Dominating cliques in graphs, Discrete Mathematics 86 (1990), 101-116.

[11] R.Diestel, Graph Theory, Springer, New York (1997).

[12] J.Fouquet, V.Giakoumakis, F.Maire, H.Thuillier, On graphs without $\mathbb{P}_5$ and $\overline{\mathbb{P}_5}$, Discrete Mathematics 146 (1995), 33-44.

[13] M.R.Garey, D.S.Johnson, L.Stockmeyer, Some simplified NP-complete graph problem, Theoretical Computer Science 1 (1976), 237-267.

[14] R.L.Graham, Bounds for certain multiprocessor anomalies, Bell System Technical Journal 45 (1966), 1563-1581.

[15] M.Grötschel, L.Lovász, A.Schrijver, The ellipsoid method and its consequences in combinatorial optimalization, Combinatorica 1 (1981), 169-197; Corrigendum: Combinatorica 4 (1984), 291-295.

[16] M.Grötschel, L.Lovász, A.Schrijver, Polynomial algorithms for perfect graphs, Annals of Discrete Mathematic 24 (1984), 225-256.

[17] A.Gyárfás, J.Lehel, On-line and first fit colorings of graphs, Journal of Graph Theory 12 (1988), 217-227.

[18] A.Gyárfás, Z.Király, J.Lehel, Effective on-line coloring of $\mathbb{P}_5$-free graphs, Combinatorica 11 (1991), 181-184.

[19] A.Gyárfás, J.Lehel, On-line 3-chromatic Graphs I. Triangle-Free Graphs, SIAM Journal of Discrete Mathematics 12(3) (1999), 385-411.

[20] R.M.Karp, Reducibility among combinatorial problems, In R.E.Miller and J.W.Thatcher, editors, Complexity of Computer Computations, Plenum Press, New York (1972), 85-103.

[21] H.A.Kierstead, W.T.Trotter, An extremal problem in recursive combinatorics, Congressus Numerantium 33 (1981), 143-153

[22] H.A.Kierstead, S.G.Penrice, W.T.Trotter, On-line and first-fit coloring of graphs which do not induce $\mathbb{P}_5$, SIAM Journal of Discrete Mathematics 8 (1995), 485-498.

[23] D.J.Rose, Triangulated graphs and the elimination process, Journal of Mathematical Analysis and Applications 32 (1970), 597-609.

[24] D.J.Rose, R.E.Tarjan, G.S. Lueker, Algorithmic aspect of vertex elimination on graphs, SIAM Journal of Computing 5 (1976), 266-283.

[25] M.Ślusarek, Optimal on-line coloring of circular arc graphs, Informatique Thèorique et Applications 29 (1995), 423-429.