

Graph colouring variations

A. Hertz

B. Ries

G-2019-32

May 2019

La collection *Les Cahiers du GERAD* est constituée des travaux de recherche menés par nos membres. La plupart de ces documents de travail a été soumis à des revues avec comité de révision. Lorsqu'un document est accepté et publié, le pdf original est retiré si c'est nécessaire et un lien vers l'article publié est ajouté.

Citation suggérée : A. Hertz, B. Ries (Mai 2019). Graph colouring variations, Rapport technique, Les Cahiers du GERAD G-2019-32, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (<https://www.gerad.ca/fr/papers/G-2019-32>) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2019
– Bibliothèque et Archives Canada, 2019

The series *Les Cahiers du GERAD* consists of working papers carried out by our members. Most of these pre-prints have been submitted to peer-reviewed journals. When accepted and published, if necessary, the original pdf is removed and a link to the published article is added.

Suggested citation: A. Hertz, B. Ries (May 2019). Graph colouring variations, Technical report, Les Cahiers du GERAD G-2019-32, GERAD, HEC Montréal, Canada.

Before citing this technical report, please visit our website (<https://www.gerad.ca/en/papers/G-2019-32>) to update your reference data, if it has been published in a scientific journal.

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2019
– Library and Archives Canada, 2019

Graph colouring variations

Alain Hertz ^{a,b}

Bernard Ries ^c

^a GERAD, Montréal (Québec), Canada, H3T 2A7

^b Department of Mathematics and Industrial Engineering, Polytechnique Montréal (Québec) Canada, H3C 3A7

^c Decision Support & Operations Research Group, University of Fribourg, Fribourg, Switzerland, CH-1700

alain.hertz@gerad.ca

bernard.ries@unifr.ch

May 2019

Les Cahiers du GERAD

G-2019-32

Copyright © 2019 GERAD, Hertz, Ries

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs. Les auteurs conservent leur droit d'auteur et leurs droits moraux sur leurs publications et les utilisateurs s'engagent à reconnaître et respecter les exigences légales associées à ces droits. Ainsi, les utilisateurs:

- Peuvent télécharger et imprimer une copie de toute publication du portail public aux fins d'étude ou de recherche privée;
- Ne peuvent pas distribuer le matériel ou l'utiliser pour une activité à but lucratif ou pour un gain commercial;
- Peuvent distribuer gratuitement l'URL identifiant la publication.

Si vous pensez que ce document enfreint le droit d'auteur, contactez-nous en fournissant des détails. Nous supprimerons immédiatement l'accès au travail et enquêterons sur votre demande.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*. Copyright and moral rights for the publications are retained by the authors and the users must commit themselves to recognize and abide the legal requirements associated with these rights. Thus, users:

- May download and print one copy of any publication from the public portal for the purpose of private study or research;
- May not further distribute the material or use it for any profit-making activity or commercial gain;
- May freely distribute the URL identifying the publication.

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Abstract: We consider three colouring problems which are variations of the basic vertex-colouring problem, and are motivated by applications from various domains. We give pointers to theoretical and algorithmic developments for each of these variations.

1 Introduction

A k -colouring of a graph G is an assignment of one of k integers, called its *colour*, to each vertex of G so that adjacent vertices receive different colours. The basic vertex-colouring problem is to determine the smallest integer k , called the *chromatic number* of G , for which G admits a k -colouring.

Graph colouring has been the subject of many articles and books. For example, the recent book edited by Beineke and Wilson [3] covers many topics related to graph colouring and shows its links with areas such as topology, algebra, geometry, and computer networks.

Graph colouring has many practical applications, where the vertices represent items to which a resource has to be assigned, and the edges correspond to incompatibility constraints. The aim of this chapter is to study three variations of this basic model that are motivated by situations where additional requirements are imposed. We give pointers to theoretical and algorithmic developments for each of these variations.

We first analyse the *selective graph colouring problem* which, given a graph with a partition of its vertex-set into clusters, asks us to select exactly one vertex per cluster so that the chromatic number of the subgraph induced by the selected vertices is minimum.

We then consider situations in which the vertices of the graph to be coloured are revealed one by one, together with the edges linking them to previously revealed vertices. An *online algorithm* must then irrevocably assign a colour to the vertices as they arrive, without knowing how the next ones will be linked to the revealed ones.

Scheduling problems involving precedence constraints can be modelled as a *mixed graph colouring problem*: the vertices that have to be coloured are linked not only by edges which represent the incompatibility constraints, but also by arcs (oriented edges) which represent the precedence constraints. While adjacent vertices must receive different colours, an arc linking a vertex v to a vertex w implies that the colour of v should be strictly smaller than the colour of w .

We recall that for arbitrary graphs, the basic vertex-colouring problem is NP-hard (see [14]).

2 Selective graph colouring

In this section, we consider a generalisation of the standard vertex-colouring problem, which is known in the literature as the *selective graph colouring problem* or *partition colouring problem* and is defined as follows. Let $G = (V, E)$ be an undirected graph and let $\mathcal{V} = \{V_1, V_2, \dots, V_p\}$ be a partition of its vertex-set V . The sets of \mathcal{V} are called *clusters*. We define a *selection* as a subset of vertices $V' \subseteq V$ such that $|V' \cap V_i| = 1$ for all $i \in \{1, 2, \dots, p\}$. A *selective k -colouring* of G with respect to the partition \mathcal{V} is defined by (V', c) , where V' is a selection and c is a k -colouring of $G[V']$ – that is, the graph induced by the selection V' . As for many other graph colouring problems, we may define a chromatic number related to the selective graph colouring problem. Indeed, the smallest integer k for which a graph G admits a selective k -colouring with respect to \mathcal{V} is called the *selective chromatic number*, and is denoted by $\chi_{SEL}(G)$.

The selective graph colouring problem, in its optimisation version, takes as input a graph $G = (V, E)$ and a partition \mathcal{V} , and outputs a selection V^* such that $\chi(G[V^*])$ is minimum. This problem is denoted by SEL-COL. The decision version of it, denoted by k -SEL-COL, where $k \geq 1$ is a fixed integer, takes the same input and asks whether there exists a selection V' such that $\chi(G[V']) \leq k$.

Clearly, SEL-COL generalises the standard graph colouring problem. Indeed, when each cluster has size 1, we obtain the standard graph colouring problem. Also, it is straightforward to see that $\chi_{SEL}(G) \leq \chi(G)$.

As for many other graph colouring problems, the selective graph colouring problem arises from an application. Indeed, it was introduced in [30], under the name *partition colouring problem*, and was

used to model the wavelength routing and assignment problem in optical networks (see also [31], [36]). In this problem, we are given a set of source-destination pairs in a network, and are required to find a path between each such pair and assign a wavelength to it, in such a way that any two paths which share an edge get different wavelengths. The goal consists in using a minimum number of different wavelengths.

One way of dealing with this wavelength routing and assignment problem is the so-called *path-colouring*, which consists in two steps. In Step 1, we determine a set of paths between each pair, and in Step 2, we choose one path from each set, in such a way that the number of wavelengths needed is minimised. This second step may be modelled as a selective colouring problem. Indeed, we associate a vertex with each path that we have previously determined; we add an edge between two vertices if the corresponding paths share at least one edge; and finally, we define a cluster to be the set of vertices that correspond to paths between the same source-destination pair. It is easy to see that SEL-COL in this new graph gives an optimal solution to the second step of the path-colouring approach. But the selective graph colouring problem has many other applications, as shown in [8]: dichotomy-based constraint encoding, frequency assignment, timetabling, quality test scheduling, berth allocation and vehicle routing with multiple stacks.

Before presenting complexity results regarding the selective colouring problem, we start with an important observation. SEL-COL asks us to find a selection V^* for which $\chi(G[V^*]) = \chi_{SEL}(G)$. Clearly, the value of the selective chromatic number may be hard to determine, even if an optimal selection V^* is known. So, even if we can determine an optimal solution for SEL-COL, it may still be difficult to compute the value of the selective chromatic number. Indeed, consider the case where each cluster has size 1 (which is exactly the standard colouring problem). Then finding an optimal selection is trivial, but computing the corresponding chromatic number is difficult. Note that, on the other hand, 1-SEL-COL is NP-complete even when each cluster has size 3 (see [8]). But this time, finding a partition is difficult while verifying whether the graph induced by a selection is 1-colourable is trivial.

Since SEL-COL and k -SEL-COL are difficult problems in general, there has been some interest in determining their complexities in special graph classes (see for instance [6], [7],[8]). The following result of Demange et al. [7] gives a list of graph classes for which both problems can be solved in polynomial time (with sometimes additional constraints).

Theorem 1 *Let $G = (V, E)$ be a graph, and let $\mathcal{V} = \{V_1, V_2, \dots, V_p\}$ be a partition of its vertex-set. Then SEL-COL and k -SEL-COL can both be solved in polynomial time in the following cases:*

- (i) G is a threshold graph;
- (ii) G is a bipartite graph and $|V_i| \leq 2$, for $i = 1, 2, \dots, p$;
- (iii) G is isomorphic to nC_4 and $|V_i| \geq 4$, for $i = 1, 2, \dots, p$;
- (iv) G is isomorphic to nP_3 and $|V_i| \geq 3$, for $i = 1, 2, \dots, p$;
- (v) G is a disjoint union of cliques;
- (vi) G has stability number at most 2.

Let us briefly analyse the cases above.

- (i) Let (K, S) be a split partition of G with K being maximal, and let v_1, v_2, \dots, v_n be the vertices in G with $v_1, v_2, \dots, v_j \in S$ and $v_{j+1}, v_{j+2}, \dots, v_n \in K$. Since G is a threshold graph, we may suppose, without loss of generality, that $N(v_1) \subseteq N(v_2) \subseteq \dots \subseteq N(v_j)$.

First notice that we may assume that each cluster V_i , $i = 1, 2, \dots, p$, is either contained in K or in S . Indeed, suppose that there exists a cluster V_i that intersects both K and S . If there exists a selection V^* for which $V^* \cap V_i = \{v\} \subseteq K$ and $w \in V_i \cap S$, then $V^{*'} = (V^* - \{v\}) \cup \{w\}$ is a selection with $\chi(G[V^{*'}]) \leq \chi(G[V^*])$. Now, suppose that clusters V_1, V_2, \dots, V_q are contained in

K and clusters $V_{q+1}, V_{q+2}, \dots, V_p$ are contained in S . It is not difficult to see that $\chi_{SEL}(G) = q$, since K is maximal and so, there exists a vertex in K which is not adjacent to any vertex in S .

(ii) First notice that $\chi_{SEL}(G) = 1$ or 2 . If we know that $\chi_{SEL}(G) = 2$, then we arbitrarily choose one vertex in each cluster, and this clearly gives us a selection V^* for which $\chi(G[V^*]) = 2$, since G is bipartite. Thus, we only need to be able to check whether $\chi_{SEL}(G) = 1$. This can be done using 2-SAT. Indeed, from an instance of SEL-COL in a bipartite graph G with a partition $\mathcal{V} = \{V_1, V_2, \dots, V_p\}$ for which $|V_i| \leq 2$, for $i = 1, 2, \dots, p$, we construct an instance of 2-SAT as follows:

- we associate a variable x_j with each vertex v_j , $j = 1, 2, \dots, n$;
- we associate a clause $C_i = x_j$ with each cluster V_i , $i \in \{1, 2, \dots, p\}$ for which $V_i = \{v_j\}$;
- we associate two clauses $C_i^1 = x_j \vee x_k$ and $C_i^2 = \overline{x_j} \vee \overline{x_k}$ with each cluster V_i , $i \in \{1, 2, \dots, p\}$ for which $V_i = \{v_j, v_k\}$;
- finally, we associate a clause $C = \overline{x_j} \vee \overline{x_k}$ with each edge $v_j v_k$ for which v_j, v_k belong to different clusters.

Now it is easy to see that if there exists a truth assignment such that each clause contains at least one literal that is true, then we obtain a selection V^* for which $\chi(G[V^*]) = 1$ by choosing those vertices whose corresponding variables are true. Conversely, if there exists a selection V^* for which $\chi(G[V^*]) = 1$ – that is, V^* is a stable set – then we simply set to ‘true’ the variables corresponding to the vertices in V^* . This gives us a truth assignment for which each clause contains at least one literal that is true.

(iii) Notice that if G is isomorphic to nC_4 and $|V_i| \geq 4$ for $i = 1, 2, \dots, p$, then $\chi_{SEL}(G) = 1$. Indeed, let $C_4^1, C_4^2, \dots, C_4^n$ be the n cycles of G . Construct an auxiliary bipartite graph $H = (V_X, V_Y, E_H)$ by associating a vertex $x_i \in V_X$ with each cluster V_i , for $i = 1, 2, \dots, p$, by associating a vertex $y_j \in V_Y$ with each cycle C_4^j , for $j = 1, 2, \dots, n$, and by adding $|V_i \cap C_4^j|$ parallel edges between vertex x_i and vertex y_j for $i = 1, 2, \dots, p$ and $j = 1, 2, \dots, n$. It follows from the construction of H that $d(x_i) \geq 4$ and $d(y_j) = 4$ for $i = 1, 2, \dots, p$ and $j = 1, 2, \dots, n$. So, there exists a matching M in H saturating V_X . It is easy to see that such a matching corresponds to a selection V^* such that $\chi(G[V^*]) = 1$.

(iv) If G is isomorphic to nP_3 and $|V_i| \geq 3$ for $i = 1, 2, \dots, p$, then, as in the previous case, we can prove that $\chi_{SEL}(G) = 1$.

(v) Let K^1, K^2, \dots, K^q be the cliques in G . First notice that we may assume, without loss of generality, that $|V_i \cap K^j| \leq 1$ for $i = 1, 2, \dots, p$ and $j = 1, 2, \dots, q$. Also, we clearly have $1 \leq \chi_{SEL}(G) \leq \max_{j=1, \dots, q} \{|K^j|\}$. Hence, we only need to be able to check whether $\chi_{SEL}(G) \leq k$ for each $k = 1, 2, \dots, \max_{j=1, \dots, q} \{|K^j|\}$. This can be done by solving maximum-flow problems in a network N defined as follows :

- we associate a vertex v_i with each V_i , $i = 1, 2, \dots, p$;
- we associate a vertex w_j with each K^j , $j = 1, 2, \dots, q$;
- we add a source s and an arc sv_i of capacity 1 for $i = 1, 2, \dots, p$;
- we add a sink t and an arc $w_j t$ of capacity k for $j = 1, 2, \dots, q$;
- we add an arc $v_i w_j$ of capacity 1 for each $i \in \{1, 2, \dots, p\}$ and $j \in \{1, 2, \dots, q\}$ for which $V_i \cap K^j \neq \emptyset$.

Now $\chi_{SEL}(G) \leq k$ if and only if there is a maximum flow in N of value p . Indeed, suppose that there is a selection V^* in G for which $\chi(G[V^*]) \leq k$. Then, for each vertex $v \in V^*$ for which $v \in V_i \cap K^j$ ($i \in \{1, 2, \dots, p\}$, $j \in \{1, 2, \dots, q\}$), we add one unit of flow along the path $sv_i w_j t$. Since $G([V^*])$ is k -colourable, V^* must contain at most k vertices from each clique K^j , $j \in \{1, 2, \dots, q\}$, so there exists a flow of value p in N . Conversely, if such a flow exists, then each arc sv_i , $i = 1, 2, \dots, p$ is used by exactly one flow unit. We now obtain a selection by choosing the vertex in $V_i \cap K^j$, for each arc $v_i w_j$ along which there is a flow unit. Furthermore,

since we have a capacity of k on each arc incident to t , it follows that $|V^* \cap K^j| \leq k$. So, $G[V^*]$ is k -colourable.

(vi) If G has stability number 1 (and so is a clique), we immediately have $\chi_{SEL}(G) = p$. So we may assume that $\alpha(G) = 2$. We can then solve the problem by reducing it to a maximum matching problem in an auxiliary graph $G' = (V', E')$ defined as follows:

- we associate a vertex v_i with each cluster V_i , $i = 1, 2, \dots, p$;
- we add an edge between two vertices v_i and v_j if there exists a vertex $x \in V_i$ which is not adjacent to a vertex $y \in V_j$.

Suppose now that G admits a selective $(p - q)$ -colouring, and denote by V^* the corresponding selection. Since $\alpha(G) = 2$, it follows that each colour class has size at most 2, and so, q corresponds to the number of colour classes whose size is exactly 2. We now construct a matching M of size q in G' as follows: for each pair $x, y \in V^*$ for which $x \in V_i$ and $y \in V_j$ have the same colour, we add the edge $v_i v_j$ to M . This gives us the desired matching. Conversely, if we can find a matching M of size q in G' , then we may find a selection V^* in G inducing a selective $(p - q)$ -colouring. Indeed, for each edge $v_i v_j \in M$, we add the corresponding non-adjacent vertices $x \in V_i$ and $y \in V_j$ to V^* and give them the same colour. There then remain $p - q$ clusters for which no vertex has been selected. We arbitrarily choose one vertex in each such cluster and colour it with a new colour. This gives us the desired selective colouring.

Notice that if in (iii) and (iv) the constraints regarding the size of the clusters in \mathcal{V} are relaxed, both problems become difficult, as shown by Demange et al. [7].

Theorem 2 *Let $G = (V, E)$ be a graph and let $\mathcal{V} = \{V_1, V_2, \dots, V_p\}$ be a partition of its vertex set. Then SEL-COL is NP-hard and 1-SEL-COL is NP-complete if, for $i = 1, 2, \dots, p$, G is isomorphic to nC_4 and $|V_i| = 3$, or G is isomorphic to nP_3 and $|V_i| = 2$ or 3.*

From Theorem 2, we can easily show that both problems remain difficult in paths and cycles (see also Demange et al. [7]).

Theorem 3 *Let $G = (V, E)$ be a graph, and let $\mathcal{V} = \{V_1, V_2, \dots, V_p\}$ be a partition of its vertex-set. Then SEL-COL is NP-hard and k -SEL-COL is NP-complete if G is a cycle or a path and $|V_i| = 2$ or 3 for $i = 1, 2, \dots, p$.*

As mentioned in Theorem 1, SEL-COL and k -SEL-COL are polynomial-time solvable in threshold graphs. It is therefore natural to ask for the computational complexity of both problems in a superclass of threshold graphs – namely, split graphs. It turns out that SEL-COL is difficult in this graph class, even when each cluster has size at most 2. On the other hand, it was shown in [7] that SEL-COL admits a polynomial time approximation scheme (PTAS) for split graphs, which implies that k -SEL-COL can be solved in polynomial time.

There has also been an increasing interest in exact algorithms to solve the selective graph colouring problem (see, for instance, [11], [12], [24],[41]). In a recent paper [41], the authors provide an exact cutting plane algorithm which they test on randomly generated perfect graphs, with different densities and different sizes of the clusters, and compare its performances to those of an integer programming formulation and a branch-and-price algorithm given in [12].

3. Online colouring

In real-world applications that can be modelled using a graph colouring problem, it may happen that the graph to be coloured is not known from the beginning – in other words, the input graph is only partially available, because some relevant input arrives only in the future. This is the case, for example, in dynamic storage allocation [40], or when assigning channels (colours) to users (vertices) in a telecommunication network [23]. In such situations, the vertices arrive one by one, together with

the edges linking them to previously revealed vertices. An *online algorithm* irrevocably colours the vertices as they arrive, and the *online colouring problem* is to determine such an algorithm with best possible performance. The most standard performance measures are defined and analysed below.

Online graph colouring can be viewed as a two-person game, where one of the players is the online algorithm, while the other player, called the *spoiler*, reveals the vertices and the edges of the graph. The game is played in rounds. In each round, the spoiler reveals a new vertex v and all edges joining it to previous vertices, and the online algorithm has to choose a colour for v that does not appear at a known neighbour of v . We emphasise that the considered online algorithms are restricted to be deterministic, unless otherwise stated. We suggest the work of Vishwanathan [46] for a good introduction to randomised online algorithms.

3.1 Competitive analysis

The performance of an online colouring algorithm is typically measured using worst case analysis. More precisely, for an online algorithm A and a given graph G , we are interested in how well A does, with the worst possible ordering of the vertices. In other words, let $A(G, \sigma)$ be the number of colours that A uses to colour G when the vertices are revealed in the order σ , and let

$$A(G) = \max_{\sigma} A(G, \sigma).$$

A traditional measure of the quality of A on G is the *performance ratio* $\rho_A(G)$, defined by

$$\rho_A(G) = \frac{A(G)}{\chi(G)},$$

where $\chi(G)$ is the standard chromatic number of G , which can be computed by an *offline* algorithm.

One of the simplest and most natural online colouring algorithms is the *First-Fit* algorithm (FF for short) which, given an arbitrary ordering of the vertices and the set of positive integers as its colour-set, assigns to each successive vertex the smallest feasible colour. Such an algorithm can never use more than $\Delta(G) + 1$ colours, where $\Delta(G)$ is the maximum degree in G . So, for example, FF needs at most three colours for paths, which implies that $\rho_{FF}(G) \leq \frac{3}{2}$ on each path G .

Note that the graph to be coloured may be known beforehand, but in such a case the online algorithm receives no knowledge of which vertex in the graph each revealed vertex corresponds to. For example, suppose it is known that the graph to be coloured is a path on four vertices, a, b, c, d , with edges ab, bc and cd . If the first two vertices v, w to be coloured are not adjacent, then the online algorithm has two choices: if it assigns different colours to v and w , then the spoiler can decide that $v = a$ and $w = c$, and if it assigns the same colour to v and w , then the spoiler can set $v = a$ and $w = d$. In both cases, three colours are needed to colour the path. This example shows that $\rho_{FF}(G) = \frac{3}{2}$ for all paths G of order at least 4, and no online colouring algorithm can perform better than FF on such paths.

The situation is different for other classes of graphs. For example, there are bipartite graphs with $2n$ vertices for which FF requires n colours. Indeed, let $G = (V \cup W, E)$ be a bipartite graph with $V = \{v_1, v_2, \dots, v_n\}$, $W = \{w_1, w_2, \dots, w_n\}$, and $E = \{v_i w_j : i \neq j\}$. If the vertices are revealed in the order $v_1, w_1, v_2, w_2, \dots, v_n, w_n$, then FF assigns colour i to v_i and w_i for each $i = 1, 2, \dots, n$. Hence, there are bipartite graphs G of order n for which $FF(G) \geq \lceil n/2 \rceil$. A better performance is easily achievable, for example by using the online colouring algorithm proposed by Lovász et al. [33] that uses at most $2 \log n + 1$ colours on all bipartite graphs of order n . This is the best possible performance up to an additive constant, since Gutowski et al. [15] have shown that there are bipartite graphs G of order n for which $A(G) \geq 2 \log n - 10$ for all online algorithms A .

We emphasise that the above analysis is for the worst case. While we have observed that FF can require three colours on paths and n colours on bipartite graphs of order $2n$, it is clear that no more than two colours are used by FF if the graph to be coloured is bipartite, and if the set of revealed

vertices always induces a connected graph. An online algorithm A is *competitive* on a graph class \mathcal{C} if there is a function f such that $A(G) \leq f(\chi(G))$ for all graphs $G \in \mathcal{C}$.

As shown by Bean [2], there is no competitive algorithm for forests, since some of them require at least $1 + \log n$ colours, whereas $\chi(G) \leq 2$ for all forests G of order n . But there are classes of graphs for which online algorithms might be competitive:

- Kierstead and Trotter [26] have proved that for the class of interval graphs, there is an online colouring algorithm A for which $A(G) \leq 3\chi(G) - 2$;
- Gyárfás and Lehel [16] have shown that $FF(G) \leq \chi(G) + 1$ for split graphs, $FF(G) \leq \frac{3}{2}\chi(G)$ for the complements of bipartite graphs, and $FF(G) \leq 2\chi(G) - 1$ for the complements of chordal graphs;
- P_4 -free graphs are coloured optimally by the FF algorithm (that is, $FF(G) = \chi(G)$), whatever the order in which the vertices are revealed. Moreover, Kierstead et al. [27] have shown that there is an online algorithm that colours all P_5 -free graphs with at most $(4^{\chi(G)} - 1)/3$ colours, while Gyárfás and Lehel [16] have proved that there is no competitive online colouring algorithm for P_6 -free graphs.

The *performance function* $\rho_A(n)$ of an online algorithm A is the maximum value of $\rho_A(G)$ over all graphs G of order n . Halldórson and Szegedy [20] have proved that $\rho_A(n) \geq 2n/\log^2 n$ for all online colouring algorithms A , while Lovász, Saks and Trotter [33] have designed an online algorithm A with performance function $\rho_A(n) \in O(n/\log^* n)$, where $\log^* n$ is the least integer k for which the k th iterated logarithm function $\log^{(k)}$ satisfies $\log^{(k)}(n) < 1$. This was later improved to $O(n \log \log \log n / \log \log n)$ by Kierstead [25]. As already mentioned, all these results are for deterministic online algorithms. Better performances can be achieved for randomised algorithms. For example, Halldórson [19] has devised a randomised algorithm that attains a performance function $\rho_A(n) \in O(n/\log n)$.

3.2 Online competitive analysis

So far, we have considered the standard competitive analysis, where the performance of an online algorithm is compared with the best existing colouring, which can be obtained offline. There is however another type of analysis, called *online competitive analysis*, where the performance of an online algorithm is compared with the best possible performance that an online algorithm can achieve. The *online chromatic number* $\chi^\circ(G)$ of a graph G is defined as the smallest number k for which there is an online algorithm that can colour G with k colours, for any incoming ordering of the vertices.

The above definition implies that

$$\chi^\circ(G) = \inf_A A(G),$$

where the infimum is taken over all online colouring algorithms A ; for example, we have observed that $\chi^\circ(P_4) = 3$. When viewing online colouring as a two-person game, we see that the online chromatic number is exactly the number of colours that is used if both players (the online algorithm and the spoiler) play optimally.

An online algorithm A is *online competitive* on a graph class \mathcal{C} if there is a function f such that $A(G) \leq f(\chi^\circ(G))$, for all graphs $G \in \mathcal{C}$.

We have observed that there are no competitive online algorithms for the class \mathcal{C} of forests, but Gyárfás and Lehel [17] have proved that FF is online competitive for forests, since $FF(G) = \chi^\circ(G)$ for all forests G .

It may be true that there is an online competitive algorithm for all graphs, but this is an open question, even for bipartite graphs. Micek and Wiechert [34], [35] have shown that there are online algorithms that colour P_7 -free bipartite graphs with at most $4\chi^\circ(G) - 2$ colours, P_8 -free bipartite graphs with at most $3(\chi^\circ(G) + 1)^2$ colours, and P_9 -free bipartite graphs with at most $6(\chi^\circ(G) + 1)^2$ colours.

Böhm and Veselý [4] have shown that the problem of deciding whether $\chi^o(G) \leq k$ for a given graph G and a given integer k is PSPACE-complete. However, as proved in [18], the following problems can be solved in polynomial time:

- determine the online chromatic number $\chi^o(T)$ of a tree T ;
- determine whether $\chi^o(G) \leq 3$ when G is bipartite or triangle-free or connected.

3.3 Maximum k -colourable subgraph

The maximum k -colourable subgraph problem consists in colouring as many vertices of a given graph as possible with at most k colours. The number of vertices in such a maximum k -colourable subgraph is denoted by $\alpha_k(G)$; for $k = 1$, this is equivalent to determining a maximum stable set. Also, the chromatic number of G is the smallest integer k such that the maximum k -colourable subgraph of G is G itself.

Here also we can consider an online version of the problem, the difference from the previous online colouring problem being that only k colours are available, and we can decide (or be forced) not to colour a revealed vertex. Also, the objective does not consist in using as few colours as possible (since all k colours are available), but rather to colour as many vertices as we can.

Let $n_A(G, \sigma, k)$ be the number of vertices coloured by A , when there are k available colours and the vertices of G are revealed in the order σ . Also, let

$$n_A(G, k) = \min_{\sigma} n_A(G, \sigma, k).$$

The *competitive ratio* of an online algorithm A on G is then defined as

$$q_A(G, k) = \frac{n_A(G, k)}{\alpha_k(G)}.$$

In what follows, we assume that the vertices are revealed by sets of size $t \geq 1$. The following result is proved in [23].

Theorem 4 *If the vertices of a graph G of order n are revealed by sets of size $t < n$, then*

$$q_A(G, k) \leq kt/(n - t),$$

for all online colouring algorithms A .

In particular, for $t = 1$ and $k = 1$ (that is, the vertices are revealed one by one, and we are looking for a maximum stable set), we obtain $q_A(G, 1) \leq 1/(n - 1)$; this was also proved by Escoffier and Thomas [9]. For $t = 1$, Theorem 4 shows that $q_A(G, k) \in \mathcal{O}(k/n)$. It is not difficult to design online algorithms that achieve such an asymptotic competitive ratio. We can, for example, colour the first k revealed vertices with a different colour. It then follows that such an algorithm has a competitive ratio $q_A(G, k) \geq k/n$.

Suppose that an online colouring algorithm A cannot leave a revealed vertex uncoloured when at least one of the k available colours does not appear on one of its revealed neighbours. Then all vertices that remain uncoloured by A have at least k coloured neighbours. It follows that the number $n - n_A(G, k)$ of uncoloured vertices is at most equal to $\Delta(G)n_A(G, k)/k$, which implies that $n_A(G, k) \geq kn/(\Delta(G) + k)$. Since $\alpha_k(G) \leq n$, we obtain the following lower bound on the competitive ratio of an online algorithm:

$$q_A(G, k) \geq \frac{kn/(\Delta(G) + k)}{n} = \frac{k}{\Delta(G) + k}.$$

Assume now that we are allowed to delay the colouring of the revealed vertices, but at some cost. More precisely, let $p \geq 1$ and let p^{i-j} be the profit of colouring a vertex at iteration j if it was revealed

at iteration $i \leq j$. The problem to be solved is then to determine a colouring with maximum total profit. For example, if $p = 2$, then the profit of colouring a vertex v one iteration after it was revealed is $\frac{1}{2}$, whereas it is equal to 1 if v is coloured immediately. The competitive ratio $q_A(G, k)$ considered above can be extended to this case by defining it as the ratio of the profit resulting from A to the maximum profit $\alpha_k(G)$. The following result appears in [23].

Theorem 5 *If the vertices of a graph G of order n are revealed by sets of size $t < n$, and if the profit of colouring a vertex at iteration j which is revealed at iteration $i \leq j$ is p^{i-j} , then*

$$q_A(G, k) \leq \frac{kt}{n-t} + \frac{n-t(k+1)}{p(n-t)},$$

for all online colouring algorithms A .

Note that if $p \rightarrow \infty$, then the ratio $kt/(n-t)$ of Theorem 4 is reached, but if $p = 1$ (that is, there is no penalty for waiting), then the algorithm can be considered as offline and the ratio is 1.

An online algorithm cannot decide in which order the vertices are revealed. It can only choose which colour (if any) to assign to a revealed vertex. It has been observed that the FF algorithm has a tendency to create unbalanced colour classes, since small colours in $\{1, 2, \dots, k\}$ are preferred to large ones. A possible strategy to try to avoid such imbalance is to choose a colour for the next revealed vertex on the basis of the last colour used. More precisely, if the last coloured vertex received colour $i \in \{1, 2, \dots, k\}$, then the next one will receive the first available colour in the ordered sequence $(i+1, i+2, \dots, k, 1, 2, \dots, i)$. Such a strategy is called *Next-Fit* (NF for short).

Note that, whereas we have $n_{FF}(G, k) \leq n_{FF}(G, k+1)$ for all graphs G and all $k \geq 1$, it may happen that $n_{NF}(G, k) > n_{NF}(G, k+1)$. For example, consider the graph G in Figure 1. If v_i is revealed before v_j , for $i < j$, then $n_{NF}(G, 2) = 6$, since vertices v_1, v_3, v_5 receive colour 1 while v_2, v_4, v_6 receive colour 2. With $k = 3$, v_1, v_2 and v_3 first receive colours 1, 2 and 3, respectively. Then, v_4 receives colour 1 and v_5 receives colour 2. This means that none of the 3 colours is available for v_6 , and we have $n_{NF}(G, 3) = 5 < n_{NF}(G, 2)$.

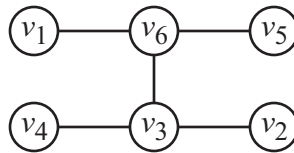


Figure 1: A graph G for which $n_{NF}(G, 2) > n_{NF}(G, 3)$.

We can colour the edges instead of the vertices of a graph, where the goal is to colour as many edges as possible using only a given number k of available colours. Favrholt and Mikkelsen [10] have shown that NF has a competitive ratio of $\frac{1}{2}$ on paths, and of $2\sqrt{3} - 3$ on trees. For vertex-colouring on general graphs, experiments reported in [23] show that FF outperforms NF .

4 Mixed graph colouring

The standard vertex-colouring problem is often used to solve scheduling problems involving incompatibility constraints. Indeed, each vertex corresponds to a job and two vertices are joined by an edge if the corresponding jobs cannot be processed at the same time. A vertex-colouring of the graph then gives a possible schedule respecting the constraints. In more general scheduling problems, there are often more requirements than just incompatibility constraints. It follows that the standard vertex-colouring model is too limited to be useful in many scheduling applications.

In this section, we discuss a graph colouring problem, that generalises the standard vertex-colouring problem and which takes into account both the incompatibility constraints and also *precedence constraints*. It is called the *mixed graph colouring problem*.

A *mixed graph* $G^{mix} = (V, A, E)$ is a graph that contains edges (set E) and arcs (set A). A *colouring* of a mixed graph G^{mix} is a mapping $c : V \rightarrow \mathbb{N}$ for which $c(v) \neq c(w)$ for each edge $vw \in E$, and $c(v) < c(w)$ for each arc $vw \in A$. If at most k distinct colours are used, then c is a k -*colouring* of a mixed graph. The minimum number of colours needed to colour the vertices of a mixed graph G^{mix} is called the *mixed chromatic number*, and is denoted by $\chi_M(G^{mix})$. In its decision version, the *mixed graph colouring problem* (MGCP) asks whether a given mixed graph G^{mix} can be coloured with at most k colours. Notice that G^{mix} must contain no directed circuits, for otherwise there exists no vertex-colouring.

How can this problem handle precedence constraints? Imagine a scheduling problem with the usual incompatibility constraints and also precedence constraints – that is, for some pairs of tasks t_1, t_2 , we know that t_1 has to be executed before t_2 . Assume that the execution time of each task is one time unit. The goal is then to execute all tasks within a minimum amount of time, taking into account the incompatibility and precedence constraints. We build a mixed graph G^{mix} as follows:

- we associate a vertex with every task;
- we add an edge between any two vertices that correspond to incompatible tasks;
- we add an arc from some vertex v to a vertex w if the task corresponding to v must be executed before the task corresponding to w .

A k -colouring of G^{mix} then corresponds to a schedule of all tasks within k time units.

Mixed graphs were first introduced in [42], and the mixed graph colouring problem has been considered by many authors – see, for instance, [1], [21], [28], [37], [44]. In what follows, we first present some bounds on the mixed chromatic number and some complexity results regarding the MGCP in special graph classes. We then discuss the *precedence-constrained class sequencing problem* (PCCSP), which can be modelled as a mixed graph colouring problem.

4.1 Bounds and complexity results

Consider a mixed graph $G^{mix} = (V, A, E)$. Let V_A be the set of vertices that are incident to at least one arc, and let ℓ be the length of a longest directed path in the directed partial graph $G_A = (V_A, A, \emptyset)$. Then $\ell + 1 \leq \chi_M(G^{mix})$. In [21], the authors considered the MGCP for the first time, and presented upper bounds on the mixed chromatic number. Denote by G' the graph induced by V_A , but where we consider all arcs as edges. Finally, let G be the undirected graph obtained from G^{mix} by considering all arcs as edges. The following two theorems are proved in [21].

Theorem 6 *Let $G^{mix} = (V, A, E)$ be a mixed graph. Then*

$$\chi_M(G^{mix}) \leq \chi(G) + |V_A| - \chi(G').$$

The upper bound in Theorem 6 is best possible. Indeed, consider a directed path on n vertices. Then $\chi_M(G^{mix}) = n$, $\chi(G) = 2$, $|V_A| = n$ and $\chi(G') = 2$, and so $\chi_M(G^{mix}) = \chi(G) + |V_A| - \chi(G')$.

Theorem 7 *Let $G^{mix} = (V, A, E)$ be a mixed graph with $A \neq \emptyset$. Then*

$$\chi_M(G^{mix}) \leq (\ell + 1)(\chi(G) - 1) + 1.$$

Again, the upper bound in Theorem 7 is best possible. Indeed, consider p copies K^1, K^2, \dots, K^p of an undirected clique of size q and let x_{ij} be the vertices of K^j for $i = 1, 2, \dots, q$ and $j = 1, 2, \dots, p$. Then add the arcs $(x_{ik}, x_{j(k+1)})$ for $i \neq j$ and $k = 1, 2, \dots, p - 1$. Clearly, $\ell + 1 = p$ and $\chi(G) = q$. Furthermore,

$$\chi_M(G^{mix}) = q + (q - 1)(p - 1) = p(q - 1) + 1.$$

We thus obtain the upper bound.

It immediately follows from Theorem 7 and from the lower bound on $\chi_M(G^{mix})$ that the mixed chromatic number of a mixed bipartite graph can take only two possible values: $\ell + 1$ or $\ell + 2$. But deciding which is the right value is difficult, as was shown in [38].

Theorem 8 *The MGCP is NP-complete for cubic planar mixed bipartite graphs and $k = 3$.*

In [21], the authors also considered mixed trees, and showed that the MGCP can be solved in quadratic time on these graphs. This was then improved in [13], where a linear-time algorithm was given to solve the MGCP for mixed trees. This result was then generalised in [39] to graphs of bounded treewidth.

Theorem 9 *The MGCP is polynomial-time solvable for graphs of bounded treewidth.*

In [43] the authors presented some further complexity result, under the assumptions that the directed partial graph $G_A = (V_A, A, \emptyset)$ is a disjoint union of paths and the graph $G_E = (V, \emptyset, E)$ is a disjoint union of cliques. This setting corresponds exactly to the unit-time, minimum-length job shop scheduling problem.

Theorem 10 *Let $G^{mix} = (V, A, E)$ be a mixed graph for which $G_A = (V_A, A, \emptyset)$ is a disjoint union of paths and $G_E = (V, \emptyset, E)$ is a disjoint union of cliques. Then the MGCP is linear-time solvable for $k = 3$.*

If we now consider the case $k = 4$, then the problem becomes NP-complete even in this very particular setting, as shown in [47].

Theorem 11 *Let $G^{mix} = (V, A, E)$ be a mixed graph for which $G_A = (V_A, A, \emptyset)$ is a disjoint union of paths and $G_E = (V, \emptyset, E)$ is a disjoint union of cliques. Then the MGCP is NP-complete for $k = 4$.*

In addition to the two assumptions mentioned above, assume now that, for each clique in $G_E = (V, \emptyset, E)$, no two vertices belong to the same directed path. In terms of scheduling theory, this additional constraint corresponds to the restriction that no two operations of the same job can be executed on a same machine. The following result appears in [22].

Theorem 12 *Let $G^{mix} = (V, A, E)$ be a mixed graph for which $G_A = (V_A, A, \emptyset)$ is a disjoint union of paths, $G_E = (V, \emptyset, E)$ is a disjoint union of two cliques, and for each clique in $G_E = (V, \emptyset, E)$, no two vertices belong to the same directed path. Then the MGCP is linear-time solvable.*

If our mixed graph has exactly three cliques, then the problem becomes NP-complete (see [29]).

Theorem 13 *Let $G^{mix} = (V, A, E)$ be a mixed graph for which $G_A = (V_A, A, \emptyset)$ is a disjoint union of paths, $G_E = (V, \emptyset, E)$ is a disjoint union of three cliques, and for each clique in $G_E = (V, \emptyset, E)$, no two vertices belong to the same directed path. Then the MGCP is NP-complete.*

4.2 A precedence-constrained sequencing problem

In this section we present the *precedence-constrained class sequencing problem* (PCCSP), which can be modelled as a colouring problem in a special mixed graph. Consider a set V of operations, a set \mathcal{C} of classes, and a set $P \subseteq V \times V$ of precedence constraints – that is, for some pairs of operations v, w , we know that v has to be executed before w . Each operation $v \in V$ belongs to exactly one class $\gamma_v \in \mathcal{C}$. The operations in V must be performed sequentially in a one-machine environment, and a set-up is required between the execution of two consecutive operations if they belong to different classes. The PCCSP asks for a sequence of operations that minimises the number of set-ups while respecting precedence constraints.

As mentioned by Tovey [45], the PCCSP is a fundamental scheduling problem in systems where processors have the flexibility to perform more than one operation. For example, Lofgren, McGinnis

and Tovey [32] consider a circuit card assembly problem, where each operation inserts an electronic component on a card at one of the assembly stations, and each component required by a card is available at exactly one assembly station. Two insertions belong to the same class if they are to be performed at the same assembly station, and a set-up therefore corresponds to moving from one station to another.

Consider the directed graph $G_A = (V, A)$, where each operation is represented by a vertex $v \in V$, and whenever $(v, w) \in P$, we introduce an arc vw . We may assume that A is acyclic and transitive. We obtain a mixed graph G^{mix} from G_A by removing all arcs vw that link two vertices v and w of the same class (so, $\gamma_v = \gamma_w$), and by adding all edges vw between vertices v, w with $\gamma_v \neq \gamma_w$. Observe that the undirected part of G^{mix} has a special structure: it is a complete q -partite graph, where $q = |\mathcal{C}|$ is the number of classes.

For each vertex $v \in V$, let $\text{pred}(v) = \{w \in V : \gamma_w \neq \gamma_v \text{ and } vw \in A\}$ be the set of predecessors of v in G^{mix} belonging to a class different from γ_v , and let $\text{succ}(v) = \{w \in V : \gamma_w \neq \gamma_v \text{ and } vw \in A\}$ be the set of successors of v in G^{mix} belonging to a class different from γ_v . Note that if $(v, w) \in P$ for any two vertices v, w of the same class $\gamma_v = \gamma_w$, then $\text{pred}(v) \subseteq \text{pred}(w)$ and $\text{succ}(v) \subseteq \text{succ}(w)$.

Each solution to the PCCSP with $k - 1$ set-ups corresponds to a k -colouring of G^{mix} , where two operations executed without any intermediate set-up have the same colour. Conversely, consider a k -colouring c of G^{mix} , and let v, w be two vertices with $(v, w) \in P$. If $\gamma_v \neq \gamma_w$, then vw is an arc in G^{mix} , which implies that $c(v) < c(w)$. If $\gamma_v = \gamma_w$ and $c(v) > c(w)$, then $\text{pred}(v) \subseteq \text{pred}(w)$, which implies that $c(u) < c(w)$ for all $u \in \text{pred}(v)$. Hence, by assigning the colour $c(w)$ to v (instead of $c(v)$), we obtain another k -colouring of G^{mix} . We can therefore assume that $c(v) \leq c(w)$ for all $(v, w) \in P$. A solution to the PCCSP can then be obtained as follows. We first execute all operations whose corresponding vertices v satisfy $c(v) = 1$, then all operations whose corresponding vertices v satisfy $c(v) = 2$, and continue this process until all operations are executed. For any group of operations whose corresponding vertices have the same colour, we execute them in an order that respects the precedence constraints P . The resulting total ordering of the operations is a solution to the PCCSP, and the number of set-ups corresponds to one less than the number of colours used in c . Indeed, all operations whose corresponding vertices have the same colour belong to the same class. As a result, a colouring of G^{mix} with a minimum number of colours corresponds to a solution of the PCCSP with a minimum number of set-ups.

It is shown in [5] that preprocessing procedures can help to simplify instances of the PCCSP. We describe some of these here. Consider two distinct vertices v and w for which $\gamma_v = \gamma_w$. If $\text{pred}(v) = \text{pred}(w)$ or $\text{succ}(v) = \text{succ}(w)$, or $\text{pred}(v) \subset \text{pred}(w)$ and $\text{succ}(v) \subset \text{succ}(w)$, then v and w can be merged into a single vertex because there is an optimal solution to the PCCSP where v and w have the same colour. This is illustrated by the example of Figure 2(a) which contains 17 operations and three classes, represented by the colours white, gray and black. Only the arcs of G_A that belong to the transitive reduction of the precedence constraints are represented – that is, uw is not represented if u precedes v and v precedes w . We can merge 1 with 6, 2 with 10, 8 with 12, 7 with 16, and 5 with 11 and 17, because $\text{pred}(1) = \text{pred}(6) = \emptyset$, $\text{pred}(2) = \text{pred}(10) = \{9\}$, $\text{pred}(8) = \text{pred}(12) = \{9\}$, $\text{succ}(7) = \text{succ}(16) = \emptyset$, and $\text{succ}(5) = \text{succ}(11) = \text{succ}(17) = \emptyset$. The graph resulting from these merge operations is shown in Figure 2(b).

Furthermore, a lower bound on the number of colours needed to colour the vertices of G^{mix} can be obtained as follows. For each class $\gamma \in \mathcal{C}$, we construct a directed graph G^γ , obtained from G_A by adding a source s , adding an arc sv of length 1 for each vertex v for which $\gamma_v = \gamma$, associating a length of 1 with each arc wv for which $\gamma_w \neq \gamma$ and $\gamma_v = \gamma$, and associating a length of 0 with each other arc.

Here, the length of an arc wv for which $\gamma_w \neq \gamma$ and $\gamma_v = \gamma$ corresponds to the one set-up that is needed when we switch from an operation w of class $\gamma_w \neq \gamma$ to an operation v of class γ . Furthermore, the length of the arcs sv for which $\gamma_v = \gamma$ accounts for the first execution of operations within the class γ . It is now easy to see that the length of a longest path in G^γ starting at s corresponds to a lower bound on the number of colours that are needed to colour the vertices whose corresponding operations

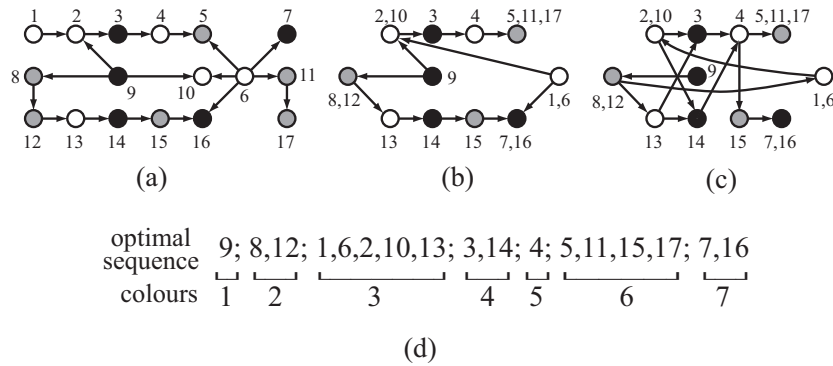


Figure 2: Preprocessing reduction procedures for the PCCSP.

belong to class γ . If r_γ is this length, then $\sum_{\gamma \in \mathcal{C}} r_\gamma$ is a lower bound on the mixed chromatic number. For the graph depicted in Figure 2 (b), the lower bound r_γ is 2 for the white vertices, 2 for the gray ones, and 3 for the black ones. This gives a total of $\sum_{\gamma \in \mathcal{C}} r_\gamma = 2 + 2 + 3 = 7$.

Consider any upper bound UB for the optimum number of colours needed to colour the vertices of G^{mix} . The next procedure tries to add precedence constraints that must be satisfied by any solution that uses fewer colours than UB . For each pair v, w of distinct vertices in G^{mix} with $vw \notin A$ and $wv \notin A$, we compute the above lower bound LB for the graph obtained from G^{mix} by adding the arc vw in A . If $LB \geq UB$, then w is executed not later than v in any feasible solution with value at most $UB - 1$. Therefore, vw can be added to A , forbidding w to be performed later than v . Similarly, if a lower bound LB is obtained after wv is added to A is at least UB , then we add wv to A to avoid w being processed before v .

For an illustration, consider again the graph depicted in Figure 2(b), and assume that we have already found a colouring with $UB = 8$ colours. We check whether we can insist that operation 4 precedes operation 15. For this purpose we temporarily add an arc from 15 to 4 to A and calculate the above lower bound; this is now equal to 8, because the longest paths have lengths 2, 3, and 3 for the white, gray, and black vertices. So, $\sum_{\gamma \in \mathcal{C}} r_\gamma = UB$, and we can add the arc from 4 to 15 to A , since, in any solution with at most 7 colours, operation 4 is executed before operation 15. We can similarly impose other additional precedence constraints to obtain the graph in Figure 2(c). An optimal sequence is then easy to obtain. Indeed, operation 9 is the only one with no predecessors, so it should be performed first. Operations 8 and 12 are the next ones, and we can proceed in this way, with no choice at each stage, to get a sequence that corresponds to a colouring with 7 colours. This sequence is represented in Figure 2(d).

References

- [1] F. S. Al-Anzi, Y. N. Sotskov, A. Allahverdi and G. V. Andreev, Using mixed graph coloring to minimize total completion time in job shop scheduling, *Appl. Math. and Comp.* 182 (2006), 1137–1148.
- [2] D. R. Bean, Effective coloration, *J. Symbolic Logic* 41 (1976), 469–480.
- [3] L. W. Beineke and R. J. Wilson, *Topics in Chromatic Graph Theory*, Cambridge University Press, 2015.
- [4] M. Böhm and P. Veselý, Online chromatic number is PSPACE-complete, *Theory of Comp. Systems* 62 (2018), 1366–1391.
- [5] R. Bürgy, P. Baptiste and A. Hertz, An exact solution approach for the precedence-constrained class sequencing problem, *Les Cahiers du Gerad*, G-2017-82, Montréal, 2017.

-
- [6] M. Demange, J. Monnot, P. Pop and B. Ries, Selective graph coloring in some special classes of graphs, *Lecture Notes in Comp. Science* 7422 (2012), 320–331.
- [7] M. Demange, J. Monnot, P. Pop and B. Ries, On the complexity of the selective graph coloring problem in some special classes of graphs, *Theoretical Computer Science* 540–541 (2014), 89–102.
- [8] M. Demange, T. Ekim, B. Ries and C. Tanasescu, On some applications of the selective graph coloring problem, *European J.Operational Research* 240 (2015), 307–314.
- [9] B. Escoffier and P. Thomas, On-line models and algorithms for max independent set, *RAIRO - Operations Research* 40 (2006), 49–42.
- [10] L. M. Favrholt and J. W. Mikkelsen, Online dual edge coloring of paths and trees, *Lecture Notes in Comp. Science* 8952 (2015), 181–192.
- [11] Y. Frota, N. Maculan, T. F. Noronha and C. C. Ribeiro, A branch-and-cut algorithm for partition coloring, *Networks* 55 (2010), 194–204.
- [12] F. Furini, E. Malaguti and A. Santini, An exact algorithm for the partition coloring problem, *Computers & Operations Research* 92 (2018), 170–181.
- [13] H. Furmańczyk, A. Kosowski and P. Zyliński, A note on mixed tree coloring, *Information Processing Letters* 106 (2008), 133–135.
- [14] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, 1979.
- [15] G. Gutowski, J. Kozik, P. Micek and X. Zhu, Lower bounds for on-line graph colorings, *Lecture Notes in Comp. Science* 8889 (2014), 507–515.
- [16] A. Gyárfás and J. Lehel, On-line and first fit colorings of graphs, *J. Graph Theory* 12 (1988), 217–227.
- [17] A. Gyárfás and J. Lehel, First fit and on-line chromatic number of families of graphs, *Ars Combinatorica* 29C (1990), 168–176.
- [18] A. Gyárfás, Z. Király and J. Lehel, On-line graph coloring and finite basis problems, *Combinatorics, Paul Erdős is Eighty* 1 (1993), 207–214.
- [19] M. M Halldórsson, Parallel and on-line graph coloring, *J. Algorithms* 23 (1997), 265–280.
- [20] M. M. Halldórsson and M. Szegedy, Lower bounds for on-line graph coloring, *Theoretical Computer Science* 130 (1994), 163–174.
- [21] P. Hansen, J. Kuplinsky and D. de Werra, Mixed graph colorings, *Math. Methods of Operations Research* 45 (1997), 145–160.
- [22] N. Hefetz and I. Adiri, An efficient optimal algorithm for the two-machines unit-time jobshop schedule-length problem, *Math. of Operations Research* 7 (1982), 354–360.
- [23] A. Hertz, R. Montagné and F. Gagnon, Online algorithms for the maximum k -colorable subgraph problem, *Computers & Operations Research* 91 (2018), 209–224.
- [24] E. A. Hoshino, Y. A. Frota and C. C. de Souza, A branch-and-price approach for the partition coloring problem, *Operations Research Letters* 39 (2011), 132–137.
- [25] H. A. Kierstead, On-line coloring k -colorable graphs, *Israel J. Math.* 105 (1998), 93–104.
- [26] H. A. Kierstead and W. T. Trotter, An extremal problem in recursive combinatorics, *Congr. Numer.* 33 (1981), 143–153.
- [27] H. A. Kierstead, S. G. Penrice and W. T. Trotter, On-line and first-fit coloring of graphs that do not induce p_5 , *SIAM J. Discrete Math.* 8 (1995), 485–498.

- [28] A. Kouider, H. Ait Haddadène, S. Ourari and A. Oulamara, Mixed graph colouring for unit-time scheduling, *International J. Production Research* 55 (2017), 1720–1729.
- [29] J. K. Lenstra and A. H. G. Rinnooy Kan, Computational complexity of discrete optimization problems, *Annals of Discrete Math.* 4 (1979), 121–140.
- [30] G. Li and R. Simha, The partition coloring problem and its application to wavelength routing and assignment, *Proceedings of the First Workshop on Optical Networks*, Dallas, 2000.
- [31] Z. Liu, W. Guo, Q. Shi, W. Hu and M. Xia, Sliding scheduled lightpath provisioning by mixed partition coloring in WDM optical networks, *Optical Switching and Networking* 10 (2013), 44–53.
- [32] Ch. B. Lofgren, L. F. McGinnis and C. A. Tovey, Routing printed circuit cards through an assembly cell, *Operations Research* 39 (1991), 992–1004.
- [33] L. Lovász, M. Saks and W. T. Trotter, An on-line graph coloring algorithm with sublinear performance ratio, *Discrete Math.* 75 (1989), 319–325.
- [34] P. Micek and V. Wiechert, An on-line competitive algorithm for coloring P_8 -free bipartite graphs, In Hee-Kap Ahn and Chan-Su Shin, editors, *Algorithms and Computation*, 516–527, 2014.
- [35] P. Micek and V. Wiechert, An on-line competitive algorithm for coloring bipartite graphs without long induced paths, *Algorithmica* 77 (2017), 1060–1070.
- [36] T. F. Noronha and C. C. Ribeiro, Routing and wavelength assignment by partition colouring, *European J. Operational Research* 171 (2006), 797–810.
- [37] B. Ries, Coloring some classes of mixed graphs, *Discrete Appl. Math.* 155 (2007), 1–6.
- [38] B. Ries, Complexity of two coloring problems in cubic planar bipartite mixed graphs, *Discrete Appl. Math.* 158 (2010), 592–596.
- [39] B. Ries and D. de Werra, On two coloring problems in mixed graphs, *European J. Combinatorics* 29 (2008), 712–725.
- [40] J. M. Robson, An estimate of the store size necessary for dynamic storage allocation, *J. ACM* 18 416–423.
- [41] O. Şeker, T. Ekim and Z. C. Taşkin, An exact cutting plane algorithm to solve the selective graph coloring problem in perfect graphs, *arXiv preprint arXiv:1811.12094* (2018).
- [42] Y. N. Sotskov and V. S. Tanaev, Chromatic polynomial of a mixed graph, *Vesti Akademii Navuk BSSR, Seryya Fizika-Matematychnykh Navuk*, Minsk (in Russian) 6 (1976), 20–23.
- [43] Y. N. Sotskov, A. Dolgui and Werner F., Mixed graph coloring for unit-time job-shop, *Int. J. Math. Algorithms* 2 (2001), 289–323.
- [44] Y. N. Sotskov, V. S. Tanaev and F. Werner, Scheduling problems and mixed graph colorings, *Optimization* 51 (2002), 597–624.
- [45] C. A. Tovey, Non-approximability of precedence-constrained sequencing to minimize setups, *Discrete Appl. Math.* 134 (2004), 351–360.
- [46] S. Vishwanathan, Randomized online graph coloring, *J. Algorithms* 13 (1992), 657–669.
- [47] D. P. Williamson, L. A. Hall, J. A. Hoogeveen, C. A. J. Hurkens, J. K. Lenstra, S. V. Sevast'janov and D. B. Shmoys, Short shop schedules, *Operations Research* 45 (1997), 288–294.