10th International Conference of the Practice and Theory of Automated Timetabling
PATAT 2014, 26-29 August 2014, York, United Kingdom

The Connectedness of Clash-free Timetables

Moritz Mühlenthaler · **Rolf Wanka**

Abstract We investigate the connectedness of clash-free timetables with respect to the Kempe-exchange operation. This investigation is related to the connectedness of the search space of timetabling problem instances, which is a desirable property, for example for two-step algorithms using the Kempe-exchange during the optimization step. The theoretical framework for our investigations is based on the study of reconfiguration graphs, which model the search space of timetabling problems. We contribute to this framework by including period availability requirements in the analysis and we derive improved conditions for the connectedness of clash-free timetables in this setting. We further show that the diameter of the reconfiguration graphs increases only linearly due to the period availability requirements. We apply the theoretical insights to establish the connectedness of clash-free timetables for a number of benchmark instances.

1 Introduction

According to the classification of heuristic optimization algorithms for timetabling problems in [17], many approaches in the literature fall in the category of *two-step optimization algorithms*. The general procedure is the following: In the first step, the underlying search problem is solved and the resulting feasible solution is used as a starting point for the second step, during which the optimization is performed. In the second step only feasible solutions are considered. A recent example of a state-of-the-art two-step approach is [19], numerous other examples can be found in [17]. During the optimization step, feasible timetables are modified using Kempe-exchanges or similar operations that preserve their feasibility. It is natural to ask whether any feasible timetable, in particular an optimal one, can be reached from an initial feasible timetable. We give a partial answer to this question by investigating conditions that establish the connectedness of the search space of *clash-free* timetables.

A timetable is clash-free, if no two conflicting events are scheduled simultaneously. In our analysis, we model the structure of the search space of clash-free timetables in terms

Research funded in parts by the School of Engineering of the University of Erlangen-Nuremberg.

Moritz Mühlenthaler, Rolf Wanka
Department of Computer Science, University of Erlangen-Nuremberg, Germany
E-mail: {moritz.muehlenthaler,rolf.wanka}@cs.fau.de

of *reconfiguration graphs*. Such graphs have been studied in the context of *reconfiguration problems*. Given an instance $\mathscr{I}$ of a combinatorial search problem, the corresponding reconfiguration problem asks whether one feasible solution to $\mathscr{I}$ can be transformed into another feasible solution in a step-by-step manner by making local changes, such that each intermediate solution is also feasible. Reconfiguration variants of classical combinatorial problems have been studied for example in [4, 13, 14, 15]. The heart of the matter of timetabling problems in the academic context is the vertex coloring problem: A clash-free timetable corresponds to a proper coloring of the event conflict graph, see e.g. [9]. The connectedness of the (proper) vertex colorings of a graph has been investigated for example in [5, 8, 2]. The local change applied to a coloring in these works is an *elementary recoloring*, which changes the color of an individual node of the graph. In [16], Las Vergas and Meyniel establish conditions for the connectedness of vertex colorings using on a more general local change, the Kempe-exchange. The Kempe-exchange is a popular operation used by algorithms for timetabling problems for exploring the search space, including many of the two-step algorithms cited above. Therefore, their results can be applied in the timetabling context. Clash-freeness is typically necessary but not sufficient for a timetable to be feasible.

In many timetabling problem formulations (see e.g. [7, 27, 6]) a set of available time periods is given for each event, and all events are required to be placed strictly in their available time periods. We extend the techniques from [16] to derive conditions for the connectedness of clash-free timetables that satisfy period availability requirements. We further show that the diameter of the corresponding reconfiguration graphs increases only linearly (in the number of events) due to the period availability requirements. Our evaluation indicates the connectedness of clash-free timetables for a number of benchmark instance sets, with and without period availability requirements.

The remainder of this work is organized as follows: In Section 2 we provide the basic formalisms required for our analysis of the connectedness of clash-free timetables presented in Section 3. In Section 4 we investigate the connectedness of the clash-free timetables for number of standard benchmarking instance sets.

2 Background

2.1 The University Timetabling Problem

The University Timetabling Problem (UTP) formalizes in terms of a search problem the task of creating a course or examination schedule at a university.

Definition 1 (University Timetabling Problem (UTP))
INSTANCE:

- a set of events $E = \{e_1, \dots, e_n\}$
- a set of rooms $R = \{r_1, \dots, r_\ell\}$
- a set of time periods $P = \{p_1, \dots, p_k\}$
- a graph $G = (E, L)$ with nodes E and edges $L \subseteq \{\{u, v\} \mid u, v \in E\}$

The graph G is referred to as the *conflict graph*. Two events are called *conflicting* if they are adjacent in G. The set $P \times R$ contains the *resources*. A *timetable* τ is an assignment $\tau : E \to P \times R$. Two events e, e' are *overlapping*, if $e \neq e'$ and $\tau(e) = \tau(e')$. A timetable is called *overlap-free* if no two events overlap. Two events e, e' are *clashing* in τ, if they are conflicting and they are assigned to the same period. A timetable is *feasible*, if it is clash-free

and overlap-free.
TASK: Find a feasible timetable.

The UTP as defined above is equivalent to the problem given in [9, Section 3.4]. The clash-freeness requirement and its relation to the vertex coloring problem is the heart of the matter of timetabling problems in the academic context, see generally [9, 26]. Other kinds of requirements such as *availability requirements* and *precedence requirements* often occur in practice, see e.g. [7, 27], and in the benchmarking problem models, see e.g. [6, 12, 22]. Later, we will consider the UTP above with additional period availability requirements. These requirements mandate that only specific periods can be assigned to an event. We formalize period availability requirements in terms of an availability function α, which determines for each event the set of available periods:

$$\alpha : E \to \mathscr{P}(P) \ .$$

An important subproblem of the UTP is the *room assignment problem*. Given a period $p \in P$, then events $E' \subseteq E$ admit a room assignment, if there is an assignment $\rho : E' \to R$ such that $(p, \rho(e))$ is available for each $e \in E'$.

2.2 Vertex Coloring

A graph $G = (V(G), E(G))$, for short $G = (V, E)$, consists of a set of *vertices* V and a set of *edges* $E \subseteq \{\{u, v\} \mid u, v \in V\}$. Unless stated otherwise, we assume that graphs are loopless and finite. We denote by $u — v$ that the vertices u and v are adjacent, i.e., $\{u, v\} \in E$. The graph $G[U]$ denotes the subgraph of G induced by the vertices $U \subseteq V(G)$. G is a mapping $c : V \to \{1, \dots, k\}$ that assigns one of the colors $\{1, \dots, k\}$ to each vertex of G. A coloring is called *proper*, if no two adjacent nodes have the same color. Unless stated otherwise, we will use the term *coloring* as a shorthand for *proper coloring*. The *vertex coloring problem* asks, whether a graph admits a k-coloring. A k-coloring c of G decomposes the vertices of G into k independent sets called *color classes*. A color class $a \in \{1, \dots, k\}$ contains all vertices of color a. We denote by $G(a, b)$ the bipartite subgraph induced by the color classes a and b. A connected component in $G(a, b)$ is referred to as *Kempe-component*.

Given a set $L(v)$ (called *list*) of available colors for each $v \in V$, a *list coloring* $c : V \to \bigcup_{v \in V} L(v)$ of G is a coloring of G such that $c(v) \in L(v)$ for each $v \in V$. Graph coloring is a special case of list coloring, where all colors are available for each node. By using a standard technique, see e.g. [9, Proposition 3.2], list coloring can be reduced to vertex coloring: Let the colors be labeled $1, \dots, k$, where $k = |\bigcup_{v \in V} L(v)|$. Now, let the graph G' be a copy of G to which we add a clique C on k (new) nodes $v_1, \dots, v_k$. For each $v \in V(G)$, we add an edge $v — v_i$ to G', whenever $i \notin L(v)$. Clearly, G' admits a k-coloring if and only if G admits a list coloring. The problem of deciding if a given UTP instance admits a clash-free timetable that satisfies period availability requirements is equivalent to deciding if the conflict graph admits a list coloring, where the $L(e) = \alpha(e)$ for each event e.

2.3 The Vertex Coloring Reconfiguration Problem

Reconfiguration problems formalize the question, if a solution to a problem instance can be transformed into another solution in a step-by-step manner by some reconfiguration operation, such that each intermediate solution is feasible [14]. Reconfiguration variants of

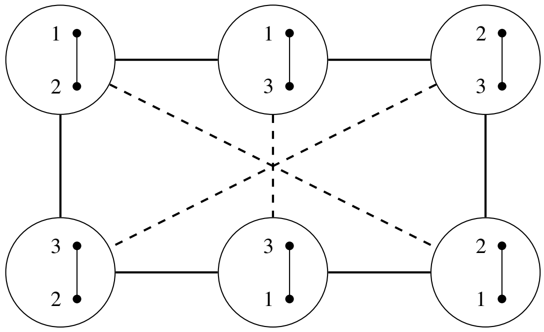


Fig. 1: The (Kempe-)3-coloring graph of the graph K_2. Solid edges correspond to elementary recolorings. Dashed edges correspond to Kempe-exchanges that are not equivalent to an elementary recoloring.

the vertex coloring problem have been studied for example in [5, 2, 23, 3]. In this context, *elementary recolorings* and *Kempe-exchanges* have been considered as reconfiguration operations. Given a coloring c of a graph G, an elementary recoloring changes the color of a single vertex of G. Two k-colorings c_1 and c_2 of G are adjacent, $c_1 \sim_E c_2$, if there is an elementary recoloring that transforms c_1 into c_2. The Kempe-exchange is a generalization of the elementary recoloring operation. Given two colors a and b, a Kempe-exchange switches the colors of a Kempe-component, i.e., a connected component in $G(a,b)$. The result of this operation is a new coloring, such that, within the Kempe-component, all vertices of the of color a are assigned to color b and vice versa. Two colorings c_1 and c_2 of G are adjacent with respect to the Kempe-exchange, $c_1 \sim_K c_2$, if there is a Kempe-exchange that transforms c_1 into c_2. Each of the two adjacency relations $\sim_E$ and $\sim_K$ gives rise to a graph structure on the set of k-colorings of G.

Definition 2 ((Kempe-)k-coloring graph) For a graph $G = (V,E)$ and $k \in \mathbb{N}$ let

$$\begin{aligned} \mathscr{V} &:= \{c : V \to \{1,\dots,k\} \mid c \text{ is a } k\text{-coloring of } G\} \\ \mathscr{E}_E &:= \{\{c_1,c_2\} \mid c_1,c_2 \in \mathscr{V} \text{ and } c_1 \sim_E c_2\} \\ \mathscr{E}_K &:= \{\{c_1,c_2\} \mid c_1,c_2 \in \mathscr{V} \text{ and } c_1 \sim_K c_2\} \quad . \end{aligned}$$

Then the k-coloring graph is the graph $\mathscr{C}_k(G) = (\mathscr{V},\mathscr{E}_E)$. The Kempe-$k$-coloring graph is the graph $\mathscr{K}_k(G) = (\mathscr{V},\mathscr{E}_K)$.

Figure 1 shows $\mathscr{C}_3(K_2)$ and $\mathscr{K}_3(K_2)$, where K_2 is the graph consisting of two vertices connected by an edge. The diameter and the connectedness of (Kempe-)k-coloring graphs have been investigated in [23, 2, 3]. The analysis of the UTP search space will follow this line of research. A graph G is called k-degenerate, if its vertices can be ordered such that each vertex has at most k neighbors preceding it. The smallest k for which G admits such an ordering is the *degeneracy* $\deg(G)$. A witness vertex ordering of the degeneracy $\deg(G)$ can be found by repeatedly removing vertices of minimal degree [21, 28]. Equivalently, the degeneracy is the largest minimum degree of any subgraph. Let $S(G)$ be the set of permutations of the vertices of G and let $\text{pred}(v,\sigma)$ denote the number of neighbors of the vertex $v \in V(G)$ that precede v in the ordering $\sigma \in S(G)$. In formal terms, the two characterizations

Algorithm 1: KEMPERECONFIGURATION

input : graph G, labeling $v_1, \dots, v_n$ of the vertices, k-colorings c_1, c_2 of G
output: list of Kempe-exchanges transforming c_1 into c_2
data : array c of length n storing the current color of each vertex, list K of Kempe-exchanges

$K \longleftarrow$ empty list;
for $i \longleftarrow 1$ **to** n**:**
    $c[i] \longleftarrow c_1(v_i)$

for $i \longleftarrow 1$ **to** n**:**
    $H \longleftarrow G[v_1, \dots, v_i]$;
    `/* Kempe-exchange` $\kappa = (a,b,u)$`, where` a,b `are colors and` $u \in V(H)$ `*/`
    for $\kappa = (a,b,u) \in K$**:**
        without loss of generality $c[i] \neq a$;
1   **if** $c[i] = b$ *and* v_i *has exactly one neighbor of color* a *in* H**:**
            $c[i] \longleftarrow a$;
2   **else if** $c[i] = b$ *and* v_i *has at least two neighbors of color* a *in* H**:**
            choose color $b' \neq b$, which is not used by any neighbor of v_i in H;
            insert Kempe-exchange (b,b',v_i) right before κ in K;
            $c[i] \longleftarrow b'$;
3   append Kempe-exchange $(c_2(v_i), c[i], v_i)$ to K;
return K;

of $\deg(G)$ can be stated as follows:

$$\deg(G) := \max_{H \subseteq G} \min_{v \in V(H)} \{d_H(v)\} = \min_{\sigma \in S(G)} \max_{v \in V(G)} \operatorname{pred}(v, \sigma) \ , \tag{1}$$

where $d_H(v)$ denotes the degree of v in H. The degeneracy of a graph is an upper bound on its chromatic number. Furthermore, the degeneracy has been used to establish the connectedness of Kempe-k-coloring graphs:

Theorem 1 ([16, Proposition 2.1]) *For any graph G, the Kempe-k-coloring graph $\mathscr{K}_k(G)$ is connected if $k > \deg(G)$.* □

The proofs given in [16, 23] are essentially an analysis of the algorithm KEMPERECONFIGURATION shown in Algorithm 1. This algorithm transforms a source coloring c_1 into a destination coloring c_2 by a sequence of Kempe-exchanges, provided that a sufficient number of colors is available. The vertices are processed one-by-one according to the given labelling. The general idea is to prevent the current vertex from interfering with the Kempe-exchanges dealing with the previously processed vertices.

3 The Connectedness of Clash-free Timetables

In the following, let G be the conflict graph G of a UTP instance $\mathscr{I}$ with periods $\{1, \dots, p\}$ and let α be the period availability function. Further, let G' be the graph derived from G by the reduction from list coloring to vertex coloring from Section 2.2. In our analysis, we consider timetables that differ only with respect to how rooms are assigned as equivalent. Each p-coloring of G corresponds to an equivalence class of clash-free timetables. Thus, the adjacency relation $\sim_K$ on the p-colorings of G induces an adjacency relation on the clash-free timetables and therefore, $\mathscr{K}_p(G)$ models the search space of clash-free timetables connected by Kempe-exchanges. If $\mathscr{K}_p(G)$ is connected, then a two-step algorithm that uses Kempe-exchanges in order to explore the search space can reach an optimal solution from any starting point. If not, then the algorithm may fail to find an optimal solution due to the structure of the search space.

In most applications, clash-freeness is not the only requirement a timetable needs to satisfy. Additional types of requirements such as period availability requirements, room availability requirements, and overlap-freeness requirements restrict the set of feasible timetables, and, as a consequence, an equivalence class corresponding to a coloring may be empty. Let C be the set of colorings of G that correspond to non-empty equivalence classes of timetables. Then the search space of $\mathscr{I}$ is connected if $\mathscr{K}_p(G)[C]$ is connected. In particular, for the additional requirements above, the corresponding reconfiguration graphs are the subgraphs of $\mathscr{K}_p(G)$ induced by the following sets of nodes:

1. period availability requirements:
$$C_\pi = \{c \in V(\mathscr{K}_p(G)) \mid \forall v \in V(G) : c(v) \text{ is available for event } v\}$$
2. overlap freeness and room availability requirements:
$$C_\rho = \{c \in V(\mathscr{K}_p(G)) \mid \forall i \in P : \text{ color class } i \text{ admits a room assignment}\}$$

Conditions establishing the connectedness of $\mathscr{K}_p(G)$ result directly from Theorem 1.

Corollary 1 *The search space of clash-free timetables is connected if $p > \deg(G)$.* □

Regarding overlap freeness and room availability requirements, to the best of our knowledge, the properties of the corresponding reconfiguration graphs have not been studied so far. The *bounded vertex k-coloring problem* with bound $b \in \mathbb{N}$ is the problem of coloring a graph with k colors such that each color is used at most b times. The bounded vertex coloring problem has been studied for example by Lucarelli [20], and Baker and Coffmann [1] in the setting of unit-time task scheduling on multiple processors and by de Werra in the timetabling context [10]. If overlap freeness is required and no particular room availability requirements are present, then the graph $\mathscr{K}_P(G)[C_\rho]$ is the reconfiguration graph of a bounded vertex coloring instance. The reconfiguration variant of the bounded vertex coloring problem seems to be an interesting problem which deserves further investigation. The situation gets more involved if room availability requirements are present. Checking if the k events in a color class admit a room assignment is equivalent to checking if a bipartite graph admits a matching of cardinality k.

We will now focus on structural properties of $\mathscr{K}_p(G)[C_\pi]$. First, we show that $\mathscr{K}_p(G)[C_\pi]$ is connected if and only if $\mathscr{K}_p(G')$ is connected. The main obstacle is that there is no Kempe-exchange on $\mathscr{K}_p(G)[C_\pi]$ corresponding to a Kempe-exchange on G' involving any of the nodes $v_1, \dots, v_p$. We construct a graph K, which is a copy of $\mathscr{K}_p(G)[C_\pi]$ with a self loop added to each node. Additionally, we add to K an edge between two colorings $u, v \in V(K)$, if there are two colors i and j such that u can be transformed into v by swapping the colors in all except a single connected component of $G(i, j)$. These additional edges are merely shortcuts for several individual Kempe-exchanges. Therefore, $\mathscr{K}_p(G)[C_\pi]$ is connected if and only if K is connected. Figure 2 shows the various graphs under consideration for a list-coloring instance consisting of a graph $G = (\{u, v\}, \{u - v\})$ and color lists $L(u) = \{1\}$ and $L(v) = \{2\}$. The nodes 1 and 2 of G' were added by the reduction from list to graph coloring.

Lemma 1 *There is a graph homomorphism $f : \mathscr{K}_p(G') \to K$.*

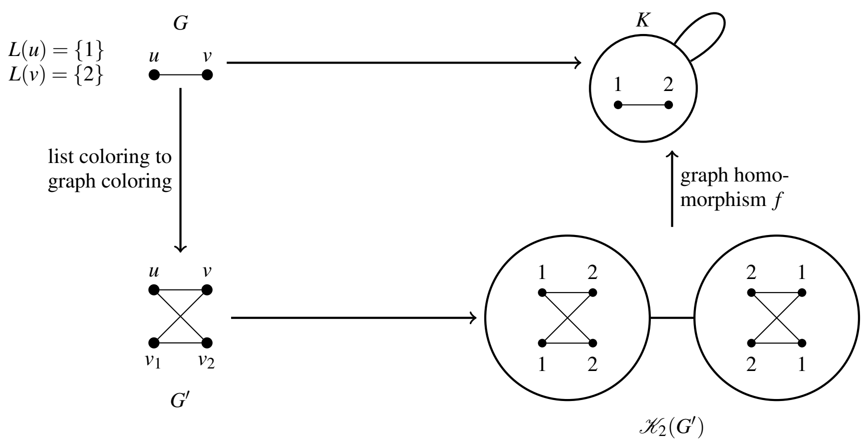


Fig. 2: Relations between the graphs G, G', K and $\mathscr{K}_p(G')$. The choice of G and the available colors determines the other graphs as described in the text. The existence of the graph homomorphism f is established by Lemma 1.

Proof We construct the mapping $f : V(\mathscr{K}_p(G')) \to V(K)$. Let $c \in V(\mathscr{K}_p(G'))$. First, we swap the colors of the p color classes such that v_i has color i for each $i \in \{1,\dots,p\}$. This can be achieved by applying a sequence of Kempe-exchanges to the coloring c: For each color $j \in \{1,\dots,p\}$, if the current color of v_j is $i \neq j$ we swap the colors in $G'(i, j)$. One Kempe-exchange is required for each Kempe-component of $G'(i, j)$. Let c' be the resulting coloring. Except for the vertices $v_1,\dots,v_p$ and their incident edges, G' is just a copy of G. Now, pick $f(c) = \tilde{c}$, where $\tilde{c}$ is equivalent to c' restricted to the vertices $V(G) \subset V(G')$. Clearly, $\tilde{c}$ is a coloring of G. Due to the construction of G', $\tilde{c}$ satisfies the list coloring requirements for G, i.e., for each $v \in V(G)$ we have $c(v) \in \alpha(v)$. Therefore, $\tilde{c} \in V(K)$.

We show that the mapping f is a graph homomorphism as required. Let c, d be colorings of G' such that $c — d$ in $\mathscr{K}_p(G')$. Further, let κ be a witness of $c \sim_K d$. There are two cases to consider:

1. The Kempe-exchange κ does not involve any of the nodes $v_1,\dots,v_p$. Then f renames the color classes of the colorings c and d if required and there is a Kempe-exchange corresponding to κ that establishes $f(c) — f(d)$ in K.
2. The Kempe-exchange κ involves two nodes $u, v \in \{v_1,\dots,v_p\}$. We need to consider following two subcases. If $G'(c(u), c(v))$ is connected then $f(c) = f(d)$ and therefore, $f(c) — f(d)$, since each node of K has a self-loop. Otherwise, $f(c)$ and $f(d)$ differ with respect to the color classes $a(u)$ and $c(v)$. We show that $f(c)$ and $f(d)$ are connected by a sequence of Kempe-exchanges that swaps the colors in all except a single Kempe-component of $G'(c(u), c(v))$ and thus $f(c) — f(d)$ by the construction of K above. To obtain $f(b)$, we first apply κ to c on G' and then apply f to the resulting coloring. The Kempe-exchange κ swaps the colors of the connected component of $G'(c(u), c(v))$ containing u and v, and then f swaps the colors in $G'(c(u), c(v))$. As a result, $f(d)$ can be obtained from $f(c)$ by swapping the colors in $G'(c(u), d(v))$ except the one containing u and v in the preimage $f^{-1}(V(G(c(u), d(v))))$.

In summary, for all $c, d \in V(\mathscr{K}_p(G')) : c — c$ implies $f(c) — f(d)$. □

The graph homomorphism f induces the equivalence relation $\sim_f$ on $V(\mathscr{K}_p(G'))$: for $a,b \in V(\mathscr{K}_p(G')) : a \sim_f b$ if $f(a) = f(b)$.

Theorem 2 *$\mathscr{K}_p(G)[C_\pi]$ is connected if and only if $\mathscr{K}_p(G')$ is connected.*

Proof We noted above that $\mathscr{K}_p(G)[C_\pi]$ is connected if and only if K is connected. Let $f : \mathscr{K}_p(G') \to K$ be the graph homomorphism from Lemma 1.
"Only if" part: Let $\mathscr{K}_p(G')$ be connected. Then K is connected since there is a graph homomorphism $\mathscr{K}_p(G') \to K$, and graph homomorphisms preserve connectedness. Therefore, $\mathscr{K}_p(G)[C_\pi]$ is connected.
"If" part: Let $\mathscr{K}_p(G)[C_\pi]$ be connected. Then K is connected. Due to the first isomorphism theorem, $K \cong \mathscr{K}_p(G')_{/\sim_f}$ and thus, $\mathscr{K}_p(G')_{/\sim_f}$ is also connected. Any two colorings u, v of G' such that $u \sim_f v$ are connected by Kempe-exchanges since one can be obtained from the other by permuting the colors of the color classes. □

For general graphs, not much is known about the diameter of their corresponding Kempe-k-coloring graphs. Using the graph homomorphism from Lemma 1, we show that the reduction from list to graph coloring increases the (possibly unknown) diameter only moderately:

Theorem 3 $\operatorname{diam}(\mathscr{K}_p(G)[C_\pi]) \leq \lfloor \frac{|V(G)|-1}{2} \rfloor \cdot \operatorname{diam}(\mathscr{K}_p(G'))$.

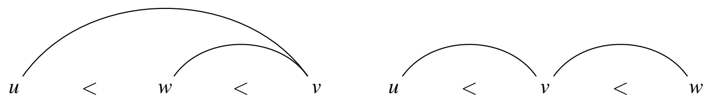
Proof For any adjacent nodes $c,d \in \mathscr{K}_p(G')$, we count how many Kempe-exchanges are required to get from $f(c)$ to $f(d)$ in $\mathscr{K}_p(G)[C_\pi]$. Let κ be the Kempe-exchange that is a witness of $c \,—\, d$, and let i and j be the involved color classes. If $c \sim_f d$ then, in the worst case, all except one connected component of $G(i,j)$ need to be switched to get from c to d for the reasons stated in cases 1 and 2 in the proof of Lemma 1. There are at most $\lfloor (|V(G)|-1)/2 \rfloor$ components and at most one Kempe-exchange is required for each of them. If $c \not\sim_f d$ then there is a single Kempe-exchange on G that establishes $f(c) \,—\, f(d)$. Thus, a shortest path of maximum length t in $\mathscr{K}_p(G')$ corresponds to a path of length at most $t \cdot \lfloor (|V(G)|-1)/2 \rfloor$ in $\mathscr{K}_p(G)[C_\pi]$. □

Given two colorings c and c' of G', the algorithm KEMPERECONFIGURATION transforms c into c' as long as there is a sufficient number of colors available. However, G' contains K_p as a subgraph therefore $\deg(G') \geq p$. According to Theorem 1, at least $p+1$ colors are needed by KEMPERECONFIGURATION and therefore Theorem 1 is not useful for proving the connectedness of clash-free timetables in the presence of period availability requirements. To overcome the limitations of Theorem 1, we fix the colors of the clique vertices $v_1, \ldots, v_p$ of G'. As a consequence, if we exclude the clique from the recoloring process, the number of colors required by KEMPERECONFIGURATION is no longer dominated by the clique.

We will first consider the general case, where the colors of some vertices $F \subseteq V(G)$ are assumed to be fixed. We denote by $\overline{F} = V(G) \setminus F$ be the remaining vertices. Further, let $S' \subset S(G)$ be the vertex orderings satisfying

$$\forall u,v \in \overline{F}, w \in F : u < v \wedge u \,—\, v \wedge v \,—\, w \Rightarrow w < v \quad . \tag{2}$$

That is, if v is a successor of u and they are adjacent, then all neighbors of v in F must precede v. Figure 3 shows two examples of vertex orderings of the graph $u \,—\, v \,—\, w$. For $F = \{w\}$, ordering 3a satisfies the condition in Eq. 2 and 3b does not. We will prove next that KEMPERECONFIGURATION does not change the color of any vertex in F if the vertices



(a) Ordering compatible with Eq. (2) (b) Ordering incompatible with Eq. (2)

Fig. 3: Two vertex orderings of the graph $u — v — w$, $F = \{w\}$.

$V(G)$ are processed according to an ordering in S'. In order to bound the number of colors required for our analysis, we introduce the following generalization of the degeneracy of a graph:

Definition 3 (Subdegeneracy) Let G be a graph and let $F \subseteq V(G)$. The *subdegeneracy* $\text{subdeg}(F,G)$ of G relative to F is defined as:

$$\text{subdeg}(F,G) = \min_{\sigma \in S'} \max_{v \in V(G) \setminus F} \text{pred}(v,\sigma)$$

Note that $\text{subdeg}(F,G) = \deg(G)$ if F is empty. If F is not empty then $\text{subdeg}(F,G) \leq \deg(G)$. Intuitively, we are looking for a vertex ordering in S' that minimmizes the maximum number of adjacent predecessors of any vertex, however, the number of predecessors of any vertex in F is irrelevant.

Theorem 4 *Let c, c' be k-colorings of G that agree on F. Then* KEMPERECONFIGURATION *returns a sequence of Kempe-exchanges such that*

1. *all intermediate colorings also agree on F, and*
2. *no more than* $\text{subdeg}(F,G)+1$ *colors are required.*

Proof We first show that the colors of the vertices F are not changed by KEMPERECONFIGURATION. Assume for a contradiction that in some intermediate coloring a vertex $w \in F$ has a color different from $c(w)$. Then w has been recolored because a neighbor u of w preceding it in σ received color $c(w)$. There are two possible reasons: Either u was recolored to $c(w)$ because $c'(u) = c(w)$, but then $c'(w) \neq c(w)$, a contradiction. If this is not the case, then u was recolored in case 1 or 2 of KEMPERECONFIGURATION, because of a neighbor v preceding it. But this is a contradiction to $\sigma \in S'$.

We now show that $\text{subdeg}(F,G)+1$ colors are sufficient. Since the vertices in F are never recolored, we consider only the vertices $\overline{F}$. An unused color may be picked for a vertex $v \in \overline{F}$ in case 2 of Algorithm 1. For each $v \in \overline{F}$, there are at most $\text{subdeg}(F,G)$ neighbors of v preceding it, and there are at most $\text{subdeg}(F,G)-1$ colors different from the color of v present among these vertices. Thus, there is at least one other color available for v. □

We propose a heuristic approach to finding a witness vertex ordering of $\text{subdeg}(F,G)$. Let $\tilde{S} \subseteq S'$ be the vertex orderings such that the vertices F precede all other vertices. Recall that for any graph G a witness vertex ordering of the degeneracy $\deg(G)$ can be found by repeatedly removing vertices of minimal degree. In a similar fashion, we can determine an optimal solution to:

$$\lambda(F,G) := \min_{\sigma \in \tilde{S}} \max_{v \in \overline{F}} \text{pred}(v,\sigma)$$

Moreover, $\lambda(F,G)$ is equivalently characterized by a max-min expression and the min-max expression above, analogous to the characterizations of the degeneracy shown in Eq. (1):

Algorithm 2: VERTEXELIMINATION

input : graph G, vertices $F \subseteq V(G)$
output: ordering $v_1, \dots, v_{|\overline{F}|}$ of the vertices $\overline{F} = V(G) \setminus F$

$G_{|D|} \longleftarrow G$;
for $i \longleftarrow |\overline{F}|$ **downto** 1 **do**
  choose v_i from $\mathrm{argmin}_{v \in \overline{F}} \{\delta(v, G_i)\}$;
  $G_{i-1} \longleftarrow G_i - v_i$.
return $v_1, \dots, v_{|\overline{F}|}$;

Theorem 5 *For any graph G and $F \subseteq V(G)$,*

$$\lambda(F,G) = \min_{\sigma \in \tilde{S}} \max_{v \in V(G) \setminus F} \mathrm{pred}(v,\sigma) = \max_{G[F] \subseteq H \subseteq G} \min_{v \in V(H) \setminus F} \{d_H(v)\} \enspace .$$

Furthermore, VERTEXELIMINATION *produces a witness vertex ordering for the min-max expression.*

Proof The proof is based on the remark on the optimality of VERTEXELIMINATION in [21]. Let $\ell = |\overline{F}|$ and for an ordering $v_1, \dots, v_\ell$ of $\overline{F}$ let $G_i = G[F \cup \{v_1, \dots, v_\ell\}]$. Further, let

$$\hat{\delta} := \max_{G[F] \subseteq H \subseteq G} \min_{v \in V(H) \setminus F} \{d(v,H)\} \enspace .$$

Intuitively, $\hat{\delta}$ is analogous to the degeneracy of G, but the vertices F are irrelevant. If an ordering $\sigma = v_1, \dots, v_\ell$ of $\overline{F}$ is an output of VERTEXELIMINATION then

$$\begin{aligned} \max_{1 \le i \le \ell} \mathrm{pred}(v_i, \sigma) &= \max_{1 \le i \le \ell} \{d(v_i, G_i)\} \\ &= \max_{1 \le i \le \ell} \min_{v \in V(G_i) \setminus F} \{d(v, G_i)\} \le \hat{\delta} \enspace . \end{aligned}$$

The graphs G_i coincide with those in Algorithm 2.

Now let H^* be a graph such that $G[F] \subseteq H^* \subseteq G$ and

$$\min_{v \in V(H^*) \setminus F} \{d(v,H)\} = \hat{\delta} \enspace .$$

Let $v_1, \dots, v_\ell$ be any ordering of $\overline{F}$ and let i be the smallest index such that $H^* \subseteq G_i$. Then v_i must be a vertex of H^* and $d(v_i, G_i) \ge \hat{\delta}$. Therefore, for any ordering $v_1, \dots, v_\ell$ of $\overline{F}$, $\max_{1 \le j \le \ell} \{d(v_j, G_j)\} \ge \hat{\delta}$, with equality if the vertex ordering is an output of VERTEX-ELIMINATION. □

Certainly, the optimality of VERTEXELIMINATION is only established with respect to the subset $\tilde{S} \subseteq S'$. The vertex ordering obtained from the algorithm can potentially be improved by the following post-processing step: Let $v_1, \dots, v_{|\overline{F}|}$ be an output of VERTEX-ELIMINATION and let k be the largest number such that $v_1, \dots, v_k$ are independent. Then the vertices $v_1, \dots, v_k$ can be moved before the vertices F in the ordering without violating condition (2). The resulting ordering $\sigma' \in S'$ is not in $\tilde{S}$ and can thus not be generated by VERTEXELIMINATION. There is a potential advantage because the construction guarantees that $\max_{v \in \overline{F}} \mathrm{pred}(v, \sigma') \le \max_{v \in \overline{F}} \mathrm{pred}(v, \sigma)$.

In summary, the heuristic for computing a vertex ordering $\sigma \in S'(G)$ such that the value $\max_{v \in \overline{F}} \mathrm{pred}(v, \sigma)$ is close to $\mathrm{subdeg}(F, G)$ performs the following two steps:

1. Run VERTEXELIMINATION to generate an ordering $v_1,\dots,v_{|\overline{F}|}$ of the vertices $\overline{F}$.
2. Let $k \in \mathbb{N}$ be the largest number such that $v_1,\dots,v_k$ are independent in G. Move the vertices $v_1,\dots,v_k$ before the vertices F in the ordering.

We apply this heuristic to prove the connectedness of the clash-free timetables that satisfy the availability constraint for a number of benchmark instances. First, we use the reduction from list to graph coloring described above to construct from a conflict graph G the graph G', which contains a clique $v_1,\dots,v_p$. Then we choose $F \subseteq V(G')$ to include every vertex with $p-1$ neighbors in $\{v_1,\dots,v_p\}$, that is

$$F = \{v \in V(G') \mid |\Gamma(v) \cap \{v_1,\dots,v_p\}| = p-1\} \ . \tag{3}$$

Now we can apply the heuristic to obtain an orderign $\sigma \in S'$ and thus an upper bound $\text{subdeg}'(F,G') = \max_{v \in V(G')\setminus F} \text{pred}(v,\sigma) \geq \text{subdeg}(F,G')$. If $p \geq \text{subdeg}'(F,G')+1$ then Theorem 4 implies that the clash-free timetables are connected.

4 Results

We use the theory developed in the previous section to establish the connectedness of clash-free timetables for a range of UTP benchmark instances. By Theorem 1, reconfiguration graphs of clash-free timetables are connected if $p > \deg(G)$ and by Theorem 4, the reconfiguration graphs of the clash-free timetables that satisfy availability requirements are connected if $p > \text{subdeg}(F,G')$ for a suitably chosen $C \subseteq V(G')$. We use the heuristic from the previous section to determine a bound $\text{subdeg}'(F,G') \geq \text{subdeg}(F,G')$. The set F of "fixed" vertices is chosen as shown in Eq. (3).

Table 1 indicates the connectedness of the clash-free timetables according to theorems 1 and 4 for instances from the CB-CTT, PE-CTT benchmark sets, as well as instances from the University of Erlangen-Nürnberg. All instances can be obtained from the SaTT group website at the University of Udine [11]. The instances `comp01`,...,`comp21` are from the CB-CTT track of the International Timetabling Competition 2007 (ITC2007) competition. The instances `ITC2_i01`,...,`ITC2_i24` are from the PE-CTT track of the same competition. The `erlangen` instances are large real-world instances from the engineering department of the University of Erlangen-Nürnberg. The `toy` instance is a small example instance from the website [11]. For each instance we give the number of periods p, the degeneracy of the conflict graph $\deg(G)$, and the bound $\text{subdeg}'(F,G') \geq \text{subdeg}(F,G')$. Table entries in bold face indicate that the corresponding value $\deg(G)$ or $\text{subdeg}'(F,G')$ certifies the connectedness of the clash-free timetables.

According to the data in Table 1 the clash-free timetables for all CB-CTT and `erlangen` instances are connected, while the conditions imposed by Theorem 1 are not satisfied for any of the PE-CTT instances. For eight CB-CTT instances, the upper bound on $\text{subdeg}(F,G')$ is sufficient to show that the reconfiguration graphs are connected in the presence of availability constraints. The situation is quite different for the PE-CTT instances, since neither $\deg(G)$ nor $\text{subdeg}'(C,G')$ is sufficient to show the connectedness of the reconfiguration graphs, better bounds on $\text{subdeg}(F,G')$ are of no use here since $\text{subdeg}(F,G') \geq \deg(G)$. Therefore, new techniques are needed for proving the connectedness (or disconnectedness) of the reconfiguration graphs for these instances.

In Tables 2 and 3, the degeneracy values of the corresponding conflict graphs are given for the Lewis/Paechter [18] and the Metaheuristic Network [24] instance sets. On these

Table 1: For each instance from the CB-CTT, PE-CTT, and Erlangen instance sets, we give the number p of periods, $\deg(G')$ and an upper bound $\text{subdeg}'(C,G') \geq \text{subdeg}(F,G')$ produced by the heuristic. All instances are available from the website [11].

instance	p	$\deg(G)$	$\text{subdeg}'(C,G')$	instance	p	$\deg(G)$	$\text{subdeg}'(C,G')$
`comp01`	30	**23**	**24**	`ITC2_i01`	45	91	109
`comp02`	25	**23**	30	`ITC2_i02`	45	99	119
`comp03`	25	**22**	27	`ITC2_i03`	45	73	92
`comp04`	25	**17**	25	`ITC2_i04`	45	78	100
`comp05`	36	**26**	43	`ITC2_i05`	45	81	99
`comp06`	25	**17**	28	`ITC2_i06`	45	80	100
`comp07`	25	**20**	**24**	`ITC2_i07`	45	80	106
`comp08`	25	**20**	**24**	`ITC2_i08`	45	69	97
`comp09`	25	**22**	25	`ITC2_i09`	45	89	108
`comp10`	25	**18**	27	`ITC2_i10`	45	97	116
`comp11`	45	**27**	**27**	`ITC2_i11`	45	75	93
`comp12`	36	**22**	40	`ITC2_i12`	45	91	109
`comp13`	25	**17**	**22**	`ITC2_i13`	45	87	106
`comp14`	25	**17**	**23**	`ITC2_i14`	45	87	107
`comp15`	25	**22**	27	`ITC2_i15`	45	79	106
`comp16`	25	**18**	25	`ITC2_i16`	45	55	83
`comp17`	25	**17**	25	`ITC2_i17`	45	50	71
`comp18`	36	**14**	**32**	`ITC2_i18`	45	91	112
`comp19`	25	**23**	27	`ITC2_i19`	45	101	120
`comp20`	25	**19**	**23**	`ITC2_i20`	45	73	92
`comp21`	25	**23**	28	`ITC2_i21`	45	72	90
`erl.2011-2`	30	**22**	32	`ITC2_i22`	45	98	118
`erl.2012-1`	30	**14**	31	`ITC2_i23`	45	117	128
`erl.2012-2`	30	**20**	32	`ITC2_i24`	45	77	97
`erl.2013-1`	30	**16**	30	`toy`	20	**10**	**11**

instances, each period is available for each event. Values in bold face indicate the connectedness of clash-free timetables is established by Theorem 1.

Finally, we will show that for the instance `toy`, the proposed heuristic yields a vertex ordering that is a witness for $\text{subdeg}(F,G')$. Let G be the conflict graph for this instance and let G' be the graph that results from the reduction from list to graph coloring. In the CB-CTT formulation, the events are grouped into courses and for each course, events of the course are a clique in G and G'. Similarly, if two courses are in conflict, then the events of both courses are a clique in G and G'. If certain periods are unavailable for a course, then the events of the course and the periods are a clique in G'. In the `toy` instance, there are four courses which consist of 16 events in total. Figure 4 shows a succinct representation of an optimal vertex ordering of the graph G'. Each node of the shown graph is a clique, as noted below the nodes, and the cliques are ordered from left to right. Two nodes of the shown graph are connected if all nodes of the corresponding cliques are connected. The nodes T, A, S and G correspond to the courses labeled SceCosC, ArcTec, TecCos and Geotec, respectively. The node P_1 represents to the periods marked unavailable for course ArcTec and the node P_2 represents the periods unavailable for SceCosC. Any two conflicting courses are connected. Let $C = V(P_1) \cup V(P_2)$.

Let $\sigma \in S(G')$ such that the cliques are arranged in the order P_1, P_2, T, A, S, G with some arbitrary choice of the relative ordering of the vertices within each clique. This ordering is a

Table 2: The connectedness of the clash-free timetables for the Lewis/Paechter instances [18]. For each instance we give the degeneracy $\deg(G)$ of the conflict graph G. Values in bold face indicate the connectedness of clash-free timetables is established by Theorem 1.

instance	deg(G)	instance	deg(G)	instance	deg(G)
`small_1`	54	`med_1`	59	`big_1`	60
`small_2`	**41**	`med_2`	67	`big_2`	68
`small_3`	98	`med_3`	67	`big_3`	64
`small_4`	69	`med_4`	69	`big_4`	80
`small_5`	84	`med_5`	87	`big_5`	75
`small_6`	**24**	`med_6`	101	`big_6`	93
`small_7`	68	`med_7`	120	`big_7`	111
`small_8`	84	`med_8`	98	`big_8`	82
`small_9`	124	`med_9`	121	`big_9`	77
`small_10`	136	`med_10`	64	`big_10`	77
`small_11`	**34**	`med_11`	97	`big_11`	76
`small_12`	**22**	`med_12`	78	`big_12`	76
`small_13`	146	`med_13`	105	`big_13`	84
`small_14`	100	`med_14`	92	`big_14`	74
`small_15`	79	`med_15`	101	`big_15`	127
`small_16`	118	`med_16`	145	`big_16`	115
`small_17`	120	`med_17`	126	`big_17`	184
`small_18`	60	`med_18`	188	`big_18`	131
`small_19`	141	`med_19`	173	`big_19`	159
`small_20`	**28**	`med_20`	153	`big_20`	144

Table 3: The connectedness of the clash-free timetables for the Metaheuristic Network instances [24]. For each instance we give the degeneracy $\deg(G)$ of the conflict graph G. Values in bold face indicate the connectedness of clash-free timetables is established by Theorem 1.

instance	deg(G)	instance	deg(G)	instance	deg(G)
`easy01`	**15**	`medium01`	49	`hard01`	68
`easy02`	**19**	`medium02`	53	`hard02`	67
`easy03`	**13**	`medium03`	52		
`easy04`	**12**	`medium04`	51		
`easy05`	**20**	`medium05`	47		

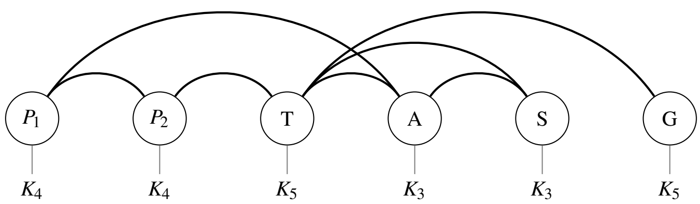


Fig. 4: Succinct representation of an optimal vertex ordering of the graph G' obtained from the conflict graph of the instance `toy` by the reduction to graph coloring. All nodes represent cliques as denoted below the nodes.

possible output of the algorithm VERTEXELIMINATION. From

$$\max_{v\in V(G')\setminus C} \text{pred}(v,\sigma) = 11 \ ,$$

we can conclude that $\text{subdeg}(F,G') \leq 11$.

Proposition 1 *For the instance* `toy`*,* $\text{subdeg}(F,G') = 11$.

Proof Let σ be an ordering of $V(G')$ and $V' \subseteq V(G') \setminus C$. The maximum number of predecessors adjacent to any vertex of V' in G' is denoted by

$$p(V',\sigma) = \max_{v\in V'} \text{pred}(v,\sigma) \ .$$

Note that for a clique $K \in \{T,A,S,G\}$, the value $p(K,\sigma)$ is determined by the last vertex of K in σ. Thus, the value of $p(K,\sigma)$ depends only on the relative order of the last vertices of the cliques $\{T,A,S,G\}$ in σ. Let $\tilde{S}$ be the vertex orderings of G' such the vertices C precede all other vertices of G' and let $\hat{S}$ be the total orderings of $\{T,A,S,G\}$. For each ordering $\sigma' \in \hat{S}$ we can pick an ordering $\ell(\sigma')$ of G' that is compatible with σ' in the sense that the relative ordering of the last vertices of the cliques is in accordance with σ'. We have,

$$\text{subdeg}(F,G') = \min_{\sigma\in\tilde{S}} \max_{K\in\{T,A,S,G\}} p(K,\sigma) = \min_{\sigma'\in\hat{S}} \max_{K\in\{T,A,S,G\}} p(K,\ell(\sigma')) \ .$$

For any ordering $\sigma' \in \hat{S}$ such that $G < T$, we have $p(T,\ell(\sigma')) \geq 13$, because the last vertex of T has at least 13 adjacent predecessors in G'. Thus, we only need to consider orderings such that $G > T$. Furthermore, since no vertex of G is adjacent to any vertex of A or S, changing the relative order of A and G or S and G does not change the number of adjacent predecessors. Hence, we can assume G is a maximum in any ordering of interest. We enumerate the values of $p(K,\ell(\sigma'))$ all for $K \in \{T,A,S,G\}$ for the 6 permutations of $\{T,A,S\}$:

clique ordering $\sigma' \in \hat{S}$	$p(T,\ell(\sigma'))$	$p(A,\ell(\sigma'))$	$p(S,\ell(\sigma'))$	$p(G,\ell(\sigma'))$
T,A,S,G	8	11	10	9
T,S,A,G	8	14	7	9
A,T,S,G	11	6	10	9
S,T,A,G	11	11	2	9
A,S,T,G	14	6	5	9
S,A,T,G	14	9	2	9

Thus,

$$\text{subdeg}(F,G') = \min_{\sigma'\in\hat{S}} \max_{K\in\{T,A,S,G\}} p(K,\ell(\sigma')) = 11$$

We can conclude that the proposed heuristic produces a witness of $\text{subdeg}(F,G') = 11$ on the instance `toy`.

5 Conclusions

We investigated the connectedness of clash-free timetables with respect to the Kempe-exchange operation. This investigation is related to the connectedness of the search space of timetabling problem instances, which is a desirable property, for example for two-step algorithms using the Kempe-exchange during the optimization step. We include period availability requirements in our analysis and derive improved conditions for the connectedness of clash-free timetables in this setting. We further show that the diameter of the reconfiguration graphs increases only linearly due to the period availability requirements. Our results indicate the connectedness of the clash-free timetables for a number of benchmark instances.

For future research, other properties of feasible timetables such as overlap-freeness may be considered as well. Furthermore, two kinds of possible improvements may be considered with respect to establishing the connectedness of clash-free timetables in the presence of period availability requirements: Both, a better analysis of Algorithm 1 and a better heuristic approach (or exact algorithm) for determining the subdegeneracy may lead to a lower number of periods required to certify the connectedness of clash-free timetables.

References

1. Brenda S. Baker and Edward G. Coffman, Jr. Mutual exclusion scheduling. *Theoretical Computer Science*, 162(2):225–243, 1996. doi:10.1016/0304-3975(96)00031-X.
2. Marthe Bonamy, Matthew Johnson, Ioannis Lignos, Viresh Patel, and Daniël Paulusma. On the diameter of reconfiguration graphs for vertex colourings. *Electronic Notes in Discrete Mathematics*, 38:161–166, 2011. doi:10.1016/j.endm.2011.09.028.
3. Marthe Bonamy, Matthew Johnson, Ioannis Lignos, Viresh Patel, and Daniël Paulusma. Reconfiguration graphs for vertex colourings of chordal and chordal bipartite graphs. *Journal of Combinatorial Optimization*, 247(1):1–12, 2014. doi:10.1007/s10878-012-9490-y.
4. Paul Bonsma. The complexity of rerouting shortest paths. In *Proc. 37th Int. Symp. on Mathematical Foundations of Computer Science (MFCS)*, pages 222–233, 2012. doi:10.1007/978-3-642-32589-2_22.
5. Paul Bonsma and Luis Cereceda. Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. *Theoretical Computer Science*, 410: 5215–5226, 2009. doi:10.1016/j.tcs.2009.08.023.
6. Alex Bonutti, Fabio De Cesco, Luca Di Gaspero, and Andrea Schaerf. Benchmarking curriculum-based course timetabling: Formulations, data formats, instances, validation, and results. *Annals of Operations Research*, 194(1):59–70, 2012. doi:10.1007/s10479-010-0707-0.
7. Michael W. Carter. A comprehensive course timetabling and student scheduling system at the University of Waterloo. In *Selected Papers from the Third International Conference on Practice and Theory of Automated Timetabling III (PATAT)*, pages 64–82, 2001. doi:10.1007/3-540-44629-X_5.
8. Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Connectedness of the graph of vertex-colourings. *Discrete Mathematics*, 308(5–6):913 – 919, 2008. doi:10.1016/j.disc.2007.07.028.
9. Dominique de Werra. An introduction to timetabling. *European Journal of Operational Research*, 19(2):151–162, 1985. doi:10.1016/0377-2217(85)90167-5.

10. Dominique de Werra. Restricted coloring models for timetabling. *Discrete Mathematics*, 165–166:161–170, 1997. doi:10.1016/S0012-365X(96)00208-7.
11. Luca Di Gaspero and Andrea Schaerf. Curriculum-based course timetabling web-site. `http://satt.diegm.uniud.it/ctt/`. Accessed September, 2013.
12. Luca Di Gaspero, Barry McCollum, and Andrea Schaerf. The second international timetabling competition (ITC-2007): Curriculum-based Course Timetabling (Track 3). In *Proceedings of the 1st International Workshop on Scheduling, a Scheduling Competition (SSC)*, 2007. URL `http://pst.istc.cnr.it/RCRA07/articoli/P08-digaspero-etal-RCRA07.pdf`.
13. Parikshit Gopalan, Phokion G. Kolaitis, Elitza Maneva, and Christos H. Papadimitriou. The connectivity of Boolean satisfiability: Computational and structural dichotomies. *SIAM Journal on Computing*, 38(6):2330–2355, 2009. doi:10.1137/07070440X.
14. Takehiro Ito, Erik D. Demaine, Nicholas J. A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12–14):1054–1065, 2011. doi:10.1016/j.tcs.2010.12.005.
15. Marcin Kamiński, Paul Medvedev, and Martin Milanič. Shortest paths between shortest paths and independent sets. In *Proc. 21st Int. W'shop. on Combinatorial Algorithms (IWOCA)*, pages 56–67, 2011. doi:10.1007/978-3-642-19222-7_7.
16. Michel Las Vergnas and Henri Meyniel. Kempe classes and the Hadwiger conjecture. *Journal of Combinatorial Theory, Series B*, 31(1):95–104, 1981. doi:10.1016/S0095-8956(81)80014-7.
17. Rhydian Lewis. *Metaheuristics for University Course Timetabling*. PhD thesis, Napier University, Edinburgh, Scotland, 2006.
18. Rhydian Lewis and Ben Paechter. New "harder" instances for the university course timetabling problem. `http://www.soc.napier.ac.uk/~benp/centre/timetabling/harderinstances.htm`. Accessed September, 2013.
19. Zhipeng Lü and Jin-Kao Hao. Adaptive tabu search for course timetabling. *European Journal of Operational Research*, 200(1):235–244, 2010. doi:10.1016/j.ejor.2008.12.007.
20. Giorgio Lucarelli. *Scheduling in Computer and Communication Systems and Generalized Graph Coloring Problems*. PhD thesis, Athens University of Economics and Business, 2009.
21. David W. Matula. A min-max theorem for graphs with application to graph coloring. *SIAM Review*, 10(4):467–490, 1968. doi:10.1137/1010115.
22. Barry McCollum, Paul McMullan, Edmund K. Burke, and Rong Parkes, Andrew J.and Qu. The second International Timetabling Competition: Examination timetabling track. Technical Report QUB/IEEE/Tech/ITC2007/Exam/v4.0/17, Queen's University, Belfast, Sep 2007. Available from `http://www.cs.qub.ac.uk/itc2007/`.
23. Bojan Mohar. Kempe equivalence of colorings. In Adrian Bondy, Jean Fonlupt, Jean-Luc Fouquet, Jean-Claude Fournier, and Jorge L. Ramírez Alfonsín, editors, *Graph Theory in Paris*, Trends in Mathematics, pages 287–297. Birkhäuser, 2007. doi:10.1007/978-3-7643-7400-6_22.
24. Olivia Rossi-Doria, Michael Sampels, Mauro Birattari, Marco Chiarandini, Marco Dorigo, Luca M. Gambardella, Joshua Knowles, Max Manfrin, Monaldo Mastrolilli, Ben Paechter, Luis Paquete, and Thomas Stützle. Supporting material for the paper [25]. `http://iridia.ulb.ac.be/supp/IridiaSupp2002-001/index.htm`. Accessed February, 2014.

25. Olivia Rossi-Doria, Michael Sampels, Mauro Birattari, Marco Chiarandini, Marco Dorigo, Luca M. Gambardella, Joshua Knowles, Max Manfrin, Monaldo Mastrolilli, Ben Paechter, Luis Paquete, and Thomas Stützle. A comparison of the performance of different metaheuristics on the timetabling problem. In Edmund Burke and Patrick Causmaecker, editors, *Practice and Theory of Automated Timetabling IV*, volume 2740 of *Lecture Notes in Computer Science*, pages 329–351. Springer Berlin Heidelberg, 2003. doi:10.1007/978-3-540-45157-0_22.
26. Andrea Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13 (2):87–127, 1999. doi:10.1023/A:1006576209967.
27. Katja Schimmelpfeng and Stefan Helber. Application of a real-world university-course timetabling model solved by integer programming. *OR Spectrum*, 29(4):783–803, 2007. doi:10.1007/s00291-006-0074-z.
28. George Szekeres and Herbert S. Wilf. An inequality for the chromatic number of a graph. *Journal of Combinatorial Theory*, 4(1):1–3, 1968. doi:10.1016/S0021-9800(68)80081-X.