Les Cahiers du GERAD ISSN: 0711–2440

About Equivalent Interval Colorings of Weighted Graphs

M. Bouchard, M. Čangalović,
A. Hertz

G–2008–03

January 2008

La publication de ces rapports de recherche bénéficie d'une subvention du Fonds québécois de la recherche sur la nature et les technologies.

About Equivalent Interval Colorings of Weighted Graphs

Mathieu Bouchard

GERAD and École Polytechnique de Montréal
C.P. 6179, Succ. Centre-ville
Montréal (Québec) Canada, H3C 2A7
mathieu.bouchard@gerad.ca

Mirjana Čangalović

Faculty of Organizational Sciences
University of Belgrade
Belgrade, Serbia
canga@fon.bg.ac.yu

Alain Hertz

GERAD and École Polytechnique de Montréal
C.P. 6179, Succ. Centre-ville
Montréal (Québec) Canada, H3C 2A7
alain.hertz@gerad.ca

January 2008

Les Cahiers du GERAD

G–2008–03

Abstract

Given a graph $G = (V, E)$ with strictly positive integer weights ω_i on the vertices $i \in V$, a k-interval coloring of G is a function I that assigns an interval $I(i) \subseteq \{1, \cdots, k\}$ of ω_i consecutive integers (called colors) to each vertex $i \in V$. If two adjacent vertices x and y have common colors, i.e. $I(i) \cap I(j) \neq \emptyset$ for an edge $[i, j]$ in G, then the edge $[i, j]$ is said *conflicting*. A k-interval coloring without conflicting edges is said *legal*. The interval coloring problem (ICP) is to determine the smallest integer k, called *interval chromatic number* of G and denoted $\chi_{int}(G)$, such that there exists a legal k-interval coloring of G. For a fixed integer k, the k-interval graph coloring problem (k-ICP) is to determine a k-interval coloring of G with a minimum number of conflicting edges. The ICP and k-ICP generalize *classical vertex coloring problems* where a single color has to be assigned to each vertex (i.e., $\omega_i = 1$ for all vertices $i \in V$).

Two k-interval colorings I_1 and I_2 are said *equivalent* if there is a permutation π of the integers $1, \cdots, k$ such that $\ell \in I_1(i)$ if and only if $\pi(\ell) \in I_2(i)$ for all vertices $i \in V$. As for classical vertex coloring, the efficiency of algorithms that solve the ICP or the k-ICP can be increased by avoiding considering equivalent k-interval colorings. To this purpose, we define and prove a necessary and sufficient condition for the equivalence of two k-interval colorings. We then show how a simple tabu search algorithm for the k-ICP can possibly be improved by forbidding the visit of equivalent solutions.

Résumé

Étant donné un graphe $G = (V, E)$ avec des poids strictement positifs ω_i sur chaque sommet $i \in V$, une k-coloration par intervalles de G est une fonction I qui attribue un intervalle $I(i) \subseteq \{1, \cdots, k\}$ de ω_i entiers consécutifs (appelés couleurs) à chaque sommet $i \in V$. Si deux sommets adjacents x et y ont une couleur en commun, i.e. $I(i) \cap I(j) \neq \emptyset$ pour une arête $[i, j]$ dans G, on dit que l'arête $[i, j]$ est *conflictuelle*. Une k-coloration par intervalles sans arête conflictuelle est dite *légale*. Le problème de la coloration par intervalles (PCI) est de déterminer le plus petit entier k, noté $\chi_{int}(G)$, tel qu'il existe une k-coloration par intervalles légale de G. Pour un entier k fixé, le problème de la k-coloration par intervalles (k-PCI) est de déterminer une k-coloration par intervalles de G avec un nombre minimum d'arêtes conflictuelles. Le PCI et le k-PCI généralisent le problème classique de la coloration des sommets d'un graphe dans lequel une unique couleur doit être attribuée à chaque sommet du graphe (i.e., $\omega_i = 1$ pour tous les sommets $i \in V$).

Deux k-colorations par intervalles I_1 et I_2 sont dites *équivalentes* s'il existe une permutation π des entiers $1, \cdots, k$ telle que $\ell \in I_1(i)$ si et seulement si $\pi(\ell) \in I_2(i)$ pour tous les sommets $i \in V$. Comme pour le problème classique de la coloration des sommets d'un graphe, l'efficacité des algorithmes de résolution du PCI ou du k-PCI peut être accrue en évitant de considérer des k-colorations par intervalles équivalentes. Dans ce but, nous définissons et prouvons une condition nécessaire et suffisante pour l'équivalence de deux k-colorations par intervalles. Nous montrons ensuite comment un algorithme tabou simple pour le k-PCI peut être amélioré en interdisant la visite de solutions équivalentes.

1 Introduction

Given a graph $G = (V, E)$ with vertex set V and edge set E, the classical graph coloring problem is to assign a color to each vertex so that no two adjacent vertices have the same color and the total number of different colors is minimized. This is one of the most studied NP-hard combinatorial optimization problems [8] with various practical applications [15]. A number of different variations and generalizations of the classical graph coloring problem arise when modeling and solving real-life problems. For example, the number of colors assigned to a vertex can be more than one, and conditions can be imposed on the colors assigned to the vertices.

One such generalization is the so-called interval coloring problem of a vertex-weighted graph [11] where a strictly positive integer weight ω_i is associated with each vertex $i \in V$, and an interval of ω_i consecutive integers must be assigned to each vertex $i \in V$ such that the intervals assigned to adjacent vertices are disjoint. More formally, let $G = (V, E)$ be an undirected graph with vertex set V and edge set E, and with strictly positive integer weights ω_i on the vertices $i \in V$. A k-interval coloring of G is a function I that assigns an interval $I(i) \subseteq \{1, \cdots, k\}$ of ω_i consecutive integers (called colors) to each vertex $i \in V$. Without loss of generality, we will always assume that a k-interval coloring of G uses all colors in $\{1, \cdots, k\}$. If two adjacent vertices x and y have common colors, i.e. $I(i) \cap I(j) \neq \emptyset$ for an edge $[i, j]$ in G, then the edge $[i, j]$ is said *conflicting*. A k-interval coloring without conflicting edges is said *legal*. The interval coloring problem (ICP) is to determine the smallest integer k, called *interval chromatic number* of G and denoted $\chi_{int}(G)$, such that there exists a legal k-interval coloring of G. The special case with $\omega_i = 1$ for all vertices $i \in V$ is equivalent to the classical graph coloring problem, and a k-interval coloring is simply called k-coloring in this case. For illustration, a legal 6-interval coloring of a graph G is represented in Figure 1, where the numbers into boxes correspond to weights on vertices. Note that $\chi_{int}(G) = 6$ for this graph since the total weight of the edge $[a, b]$ or of the triangle with vertices b, d, e is equal to 6.

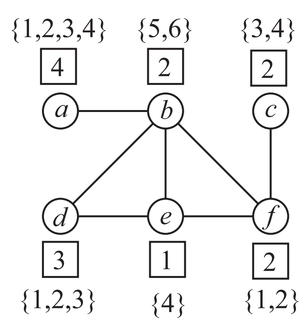

Figure 1: A legal 6-interval coloring I of a graph G.

For a fixed integer k, the k-interval graph coloring problem (k-ICP) is to determine a k-interval coloring of G with a minimum number of conflicting edges. If the minimum value is zero, this means that G admits a legal k-interval coloring, hence $\chi_{int}(G) \leq k$.

The ICP has a fairly long history dating back, at least to the 1970s. For example, Stockmeyer showed in 1976 that the interval-coloring problem is NP-hard, even when restricted to interval graphs and vertex weights in $\{1, 2\}$ (see problem SR2 in [8]). In 1976, Punter has formulated and solved a school timetabling problem with non-preemptive multiple period lessons using an interval coloring model. Another early application of the interval coloring problem was in the compile-time memory-allocation problem [5].

While the ICP is NP-hard, it can be solved in polynomial time for special classes of graphs. For example, if G is a clique then, $\chi_{int}(G)$ is equal to $\sum_{i\in V}\omega_i$, while for a bipartite graph G, we have $\chi_{int}(G) = \max_{[i,j]\in E}\{\omega_i + \omega_j\}$. More general graphs G for which $\chi_{int}(G)$ can be computed in polynomial time are studied in [11, 20].

Upper bounds on the interval chromatic number $\chi_{int}(G)$ are studied in [20], while general upper and lower bounds on $\chi_{int}(G)$ are given in [14] when vertices possibly have forbidden colors. An exact algorithm for the ICP is proposed in [4] and used to solve a real-life timetabling problem with multiple period lessons. Approximation algorithms are known for special classes of graphs such as interval graphs [2, 9] or chordal graphs [16]. Heuristic algorithms for the ICP are proposed for example in [3, 1].

In the classical graph coloring problem, every k-coloring is equivalent, up to a permutation of the colors, to $k!$ other k-colorings. In order to increase the efficiency of graph coloring algorithms, it is important to avoid visiting equivalent k-colorings when exploring the search space. A solution to the classical k-coloring problem is in fact a partition of the vertex set into k subsets called *color classes*, and the total number of non equivalent k-colorings is equal to the number of possible partitions of the vertex set into k subsets. Such considerations have inspired many researchers, including Galinier and Hao [6] who have designed a very effective genetic algorithm for the classical graph coloring problem in which new k-colorings are created from a population of k-colorings by combining color classes of two parents instead of copying color assignments. Also, the most effective local search algorithms for classical graph coloring generate neighbor k-colorings by moving a vertex from a color class to another [7].

In the next section we generalize the above equivalence relation to k-interval colorings. We then prove a necessary and sufficient condition for the equivalence of two k-interval colorings. Such a condition makes it easy to recognize equivalent k-interval colorings. We will use the following terminology. A *clique* in a graph $G = (V, E)$ is a subset $W \subseteq V$ of pairwise adjacent vertices. An *interval graph* [12, 11] is the intersection graph of a set of intervals. It has one vertex for each interval in the set, and an edge between every pair of vertices corresponding to intervals that intersect. Subsets $C_1, \cdots, C_p$ of V define a *cover* of V if $\bigcup_{i=1}^{p} C_i = V$. Moreover, if the sets C_i of the cover are mutually disjoint, they define a *partition* of V.

2 Equivalent k-interval colorings

Intuitively, two k-interval colorings are equivalent if one can be obtained from the other by permuting the colors $1, \cdots, k$. More formally, the equivalence of k-interval colorings can be defined as follows.

Definition 1 *Two k-interval colorings I_1 and I_2 are said equivalent if there is a permutation π of the integers $1, \cdots, k$ such that $\ell \in I_1(i)$ if and only if $\pi(\ell) \in I_2(i)$ for all vertices $i \in V$.*

Note that given a k-interval coloring I of graph $G = (V, E)$ and a permutation π of the integers $1, \cdots, k$, it may happen that $\bigcup_{\ell\in I(i)} \pi(\ell)$ is not an interval for some vertex $i \in V$. For example, considering the graph of Figure 1, permutation π with $\pi(1) = 6$, $\pi(6) = 1$ and $\pi(\ell) = \ell$ for $\ell \neq 1, 6$ gives colors $2, 3, 4$ and 6 to vertex a, and this is not an interval.

For the classical graph coloring problem (i.e., when $\omega_i = 1$ for all $i \in V$), Definition 1 means that two k-colorings are equivalent if their corresponding partition into color classes are identical. A concept similar to color classes in the case of interval coloring is what we call *interval color classes*, with the following formal definition.

Definition 2 *Given a k-interval coloring I of a graph $G = (V, E)$, a subset $W \subseteq V$ of vertices is an interval color class for I if*

(a) $\bigcap_{i \in W} I(i) \neq \emptyset$, *and*

(b) $\bigcap_{i \in W} I(i) \cap I(j) = \emptyset$ *for all* $j \notin W$.

The above definition can be interpreted in terms of graphs. Indeed, given an k-interval coloring I of a graph $G = (V, E)$, let $H_{G,I}$ be the interval graph with vertex set V and where two vertices i and j are linked by an edge if and only if $I(i) \cap I(j) \neq \emptyset$. Then, the interval color classes for I correspond to the maximal cliques in $H_{G,I}$. For example, the graph of Figure 2 is the interval graph $H_{G,I}$ associated with the k-interval coloring of Figure 1. It contains 4 maximal cliques (i.e. interval color classes), namely $W_1 = \{a, c, d\}$, $W_2 = \{a, c, e\}$, $W_3 = \{a, d, f\}$ and $W_4 = \{b\}$.

When $\omega_i = 1$ for all $i \in V$ (i.e., for the classical graph coloring problem), the interval graph $H_{G,I}$ is made of vertex-disjoint cliques, each one corresponding to a color class. Observe that the color classes in classical graph coloring induce a partition of the vertex set, while the interval color classes for a k-interval coloring induce a cover of the vertex set, which means that some vertices possibly belong to several interval color classes. In the example of Figure 2, vertex a belongs to three different interval color classes, and vertex c belongs to two of them.

Since two k-colorings in classical graph coloring are equivalent if and only if they induce the same partition of the vertex set into color classes, it is tempting to think that two k-interval colorings I_1 and I_2 of a graph G are equivalent if and only if they have exactly the same interval color classes, i.e. if their associated interval graphs H_{G,I_1} and H_{G,I_2} are equal. Figure 3 illustrates with an example that such a statement is not correct. Indeed, the two 7-interval colorings I_1 and I_2 on this figure have the same associated interval graph. However, if I_1 and I_2 were equivalent, then $I_2(f) = \{1, 2\} = \{\pi(3), \pi(4)\}$, which means that $\{1, 2\} \subset I_2(d)$, a contradiction.

Hence, in order to determine whether two k-interval colorings of a graph $G = (V, E)$ are equivalent, it is not sufficient to compare their corresponding cover of V with interval color classes. We will prove the following Theorem.

Theorem 1 *Two k-interval colorings I_1 and I_2 of a graph $G = (V, E)$ are equivalent if and only if*

$$| I_1(i) \cap I_1(j) | = | I_2(i) \cap I_2(j) | \quad \text{for all } i, j \text{ in } V \tag{1}$$

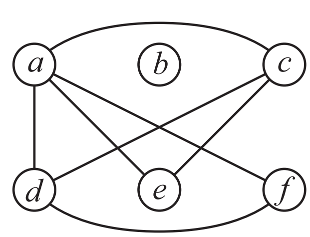

Figure 2: The interval graph $H_{G,I}$ associated with the k-interval coloring of Figure 1.

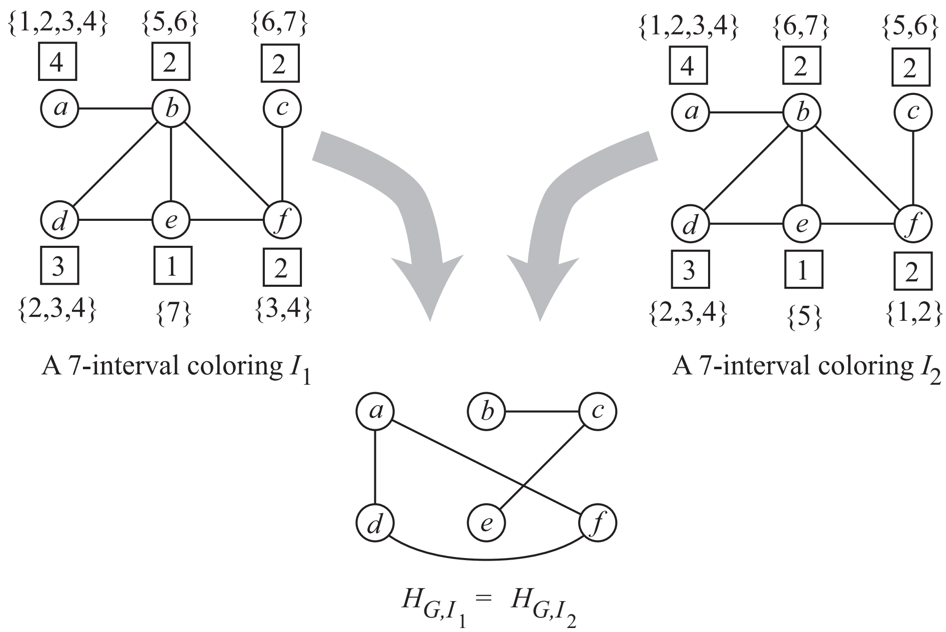

Figure 3: Two non-equivalent 7-interval colorings of a graph G, and their identical associated interval graph.

We first prove the following Lemma about k-interval colorings satisfying Property (1) of Theorem 1. For a subset $W \subseteq V$ of vertices and a k-interval coloring I, we denote $I(W) = \bigcap_{i \in W} I(i)$.

Lemma 1 *Let I_1 and I_2 be two k-interval colorings of a graph $G = (V, E)$ satisfying Property (1), and let $W \subseteq V$ be a subset of vertices. Then*

(a) $\mid I_1(W) \mid = \mid I_2(W) \mid$, and

(b) W is an interval color class for I_1 if and only if it is an interval color class for I_2.

Proof. We first prove (a). If W contains a single vertex i (i.e. $W = \{i\}$), then $\omega_i = \mid I_1(W) \mid = \mid I_2(W) \mid$. So assume $\mid W \mid \geq 2$.

- If $\mid I_1(W) \mid = 0$ then there exist at least two vertices i and j in W such that $I_1(i) \cap I_1(j) = \emptyset$. By Property (1), we then have $\mid I_2(i) \cap I_2(j) \mid = 0$, which means that $\mid I_2(W) \mid = 0$.
- If $\mid I_1(W) \mid > 0$, then $I_1(W)$ is an integer interval (since $I_1(i)$ is an integer interval for all $i \in V$), and there are at least two vertices u and v in W such that $I_1(u) \cap I_1(v) = I_1(W)$. By Property (1), we have $\mid I_2(u) \cap I_2(v) \mid = \mid I_1(W) \mid$, which means that $\mid I_2(W) \mid \leq \mid I_1(W) \mid$ since $I_2(W) \subseteq I_2(u) \cap I_2(v)$.

In all cases, we have $\mid I_2(W) \mid \leq \mid I_1(W) \mid$. By permuting the roles of I_1 and I_2, the same proof gives $\mid I_1(W) \mid \leq \mid I_2(W) \mid$. Hence, $\mid I_2(W) \mid = \mid I_1(W) \mid$.

To prove (b), assume that W is an interval color class for one of the two k-interval colorings, say I_1. Then, by definition, we have $I_1(W) \neq \emptyset$, and it follows from (a) that $I_2(W) \neq \emptyset$. If there exists a vertex $u \notin W$ such that $I_2(W \cup \{u\}) \neq \emptyset$, then we know from (a) that $I_1(W \cup \{u\}) \neq \emptyset$, which means that W is not an interval color class for I_1, a contradiction.

In summary, $I_2(W) \neq \emptyset$ and $I_2(W \cup \{u\}) = \emptyset$ for all $u \notin W$, which means that W is also an interval color class for I_2. □

We now prove that Property (1) is a necessary and sufficient condition for the equivalence of two k-interval colorings.

Proof of Theorem 1

Let I_1 and I_2 be two equivalent k-interval colorings of a graph $G = (V, E)$. By definition 1, there exists a permutation π of the integers $1, \cdots, k$ such that $\ell \in I_1(i)$ if and only if $\pi(\ell) \in I_2(i)$ for all vertices $i \in V$. Hence, for every i and j in V and for every $\ell \in \{1, \cdots, k\}$ we have $\ell \in I_1(i) \cap I_1(j)$ if and only if $\pi(\ell) \in I_2(i) \cap I_2(j)$, which means that $\mid I_1(i) \cap I_1(j) \mid = \mid I_2(i) \cap I_2(j) \mid$ for all i and j in V. Property (1) is therefore a necessary condition for the equivalence of two k-interval colorings.

We now prove that Property (1) is also a sufficient condition. The proof is by induction on the number k of colors used in I_1 and I_2. For $k = 1$, we have $I_1(i) = I_2(i) = \{1\}$ for all vertices $i \in V$ (since a k-interval coloring uses all colors in $\{1, \cdots, k\}$). Hence, permutation π with $\pi(1) = 1$ defines the equivalence between I_1 and I_2.

So, assume that $k > 1$ and Property (1) is a sufficient condition for the equivalence of two ℓ-interval colorings for all $\ell = 1, \cdots, k-1$. Consider any interval color class W for I_1. We know from Lemma 1 that $\mid I_1(W) \mid = \mid I_2(W) \mid$ and W is also an interval color class for I_2. So let π_1 be a bijective mapping from $I_1(W)$ to $I_2(W)$ (i.e., $\bigcup_{\ell \in I_1(W)} \pi_1(\ell) = I_2(W)$). For all vertices $i \in V$ and $r = 1, 2$ define

$$I'_r(i) = \begin{cases} I_r(i) - I_r(W) & \text{if } i \in W \\ I_r(i) & \text{if } i \in V - W \end{cases}$$

and

$$\omega'_i = \begin{cases} \omega_i - \mid I_1(W) \mid & \text{if } i \in W \\ \omega_i & \text{if } i \in V - W \end{cases}$$

Since W is an interval color class for I_r $(r = 1, 2)$, we have $\bigcup_{i \in V} I'_r(i) = \{1, \cdots, k\} - I_r(W)$ and $\mid I'_r(i) \mid = \omega'_i$ for all $i \in V$ and $r = 1, 2$. Note that $I'_r(i)$ is not necessarily an interval. Indeed, if the smallest color in $I_r(W)$ is strictly larger than the smallest color in $I_r(i)$ while the largest color in $I_r(W)$ is strictly smaller than the largest color in $I_r(i)$, then $I'_r(i)$ is the union of two integer intervals. So, let f_r be a function that relabels the colors in $\{1, \cdots, k\} - I_r(W)$ from 1 to $k - \mid I_r(W) \mid$ so that $f_r(i) < f_r(j)$ if and only if $i < j$ and define $I''_r(i) = \bigcup_{\ell \in I'_r(i)} f_r(\ell)$. The sets $I''_r(i)$ are integer intervals for all $i \in V$ and $r = 1, 2$. Note that if $I_r(i) = I_r(W)$ for a vertex $i \in W$, then $\omega'_i = 0$ and $I''_r(i)$ is empty.

Let $G' = (V', E')$ be the weighted graph obtained from G by removing all vertices with $\omega'_i = 0$, and by assigning weight ω'_i to all vertices $i \in V'$. By denoting $k' = k - \mid I_1(W) \mid = k - \mid I_2(W) \mid$, we have shown that I''_1 and I''_2 are two k'-interval colorings of G' with $k' < k$. In order to use the induction hypothesis, we now show that I''_1 and I''_2 satisfy Property (1).

- If i and j are two vertices in $V' \cap W$, then $I_r(W) \subseteq I_r(i) \cap I_r(j)$ for $r = 1, 2$, which means that $\mid I''_r(i) \cap I''_r(j) \mid = \mid I_r(i) \cap I_r(j) \mid - \mid I_r(W) \mid$. Since $\mid I_1(i) \cap I_1(j) \mid = \mid I_2(i) \cap I_2(j) \mid$ and $\mid I_1(W) \mid = \mid I_2(W) \mid$, we have $\mid I''_1(i) \cap I''_1(j) \mid = \mid I''_2(i) \cap I''_2(j) \mid$.

- If i and j are two vertices in V' with at least one not in W, then $I_r(W) \cap I_r(i) \cap I_r(j) = \emptyset$ for $r = 1, 2$, which means that $\mid I''_r(i) \cap I''_r(j) \mid = \mid I_r(i) \cap I_r(j) \mid$. Since $\mid I_1(i) \cap I_1(j) \mid = \mid I_2(i) \cap I_2(j) \mid$, we have $\mid I''_1(i) \cap I''_1(j) \mid = \mid I''_2(i) \cap I''_2(j) \mid$.

By induction hypothesis, we know that there exists a permutation π_2 of the colors in $\{1, \cdots, k'\}$ such that $\ell \in I''_1(i)$ if and only if $\pi_2(\ell) \in I''_2(i)$ for all vertices $i \in V'$. Consider finally permutation π of the colors in $1, \cdots, k$ such that

$$\pi(\ell) = \begin{cases} \pi_1(\ell) & \text{if } \ell \in I_1(W) \\ f_2^{-1}(\pi_2(f_1(\ell))) & \text{if } \ell \in \{1, \cdots, k\} - I_1(W) \end{cases}$$

For a vertex $i \in V$ and a color $\ell \in I_1(i)$, we have proved that

- if $\ell \in I_1(W)$, then $\pi(\ell) = \pi_1(\ell) \in I_2(W) \subseteq I_2(i)$
- if $\ell \notin I_1(W)$, then $f_1(\ell) \in I''_1(i)$. Since $\pi_2(f_1(\ell)) \in I''_2(i)$, we have $\pi(\ell) = f_2^{-1}(\pi_2(f_1(\ell))) \in I_2(i)$.

In summary, we have $\ell \in I_1(i)$ if and only if $\pi(\ell) \in I_2(i)$, which proves that I_1 and I_2 are equivalent. □

An illustration of the above construction of permutation π for two equivalent k-interval colorings is given in Figure 4. Vertex set $W = \{a, d, f\}$ is an interval color class for both 7-interval colorings I_1 and I_2 of G. We have $I_1(W) = \{2, 3\}$ and $I_2(W) = \{4, 5\}$. We can therefore consider π_1 such that $\pi_1(2) = 4$ and $\pi_1(3) = 5$. Hence, $f_1(1) = 1, f_1(4) = 2, f_1(5) = 3, f_1(6) = 4, f_1(7) = 5$ and $f_2(1) = 1, f_2(2) = 2, f_2(3) = 3, f_2(6) = 4, f_2(7) = 5$. Since $\omega'_f = 0$, vertex f does not belong to G'. All vertices in G' have the same weight as in G, except a and d for which there is reduction of two units. While $I'_1(a) = \{1, 4\}$ is not an interval, $I''_1(a) = \{f_1(1), f_1(4)\} = \{1, 2\}$. The two 5-interval colorings I''_1 and I''_2 of G' are equivalent, which can be observed with permutation π_2 such that $\pi_2(1) = 5, \pi_2(2) = 4, \pi_2(3) = 3, \pi_2(4) = 2$ and $\pi_2(5) = 1$. A proof of the equivalence of I_1 and I_2 in G is provided by permutation π with $\pi(1) = 7, \pi(2) = 4, \pi(3) = 5, \pi(4) = 6, \pi(5) = 3, \pi(6) = 2$ and $\pi(7) = 1$. For example, $\pi(4) = f_2^{-1}(\pi_2(f_1(4))) = f_2^{-1}(\pi_2(2)) = f_2^{-1}(4) = 6$.

For a k-interval coloring I of G, let us associate a weight $\mid I(i) \cap I(j) \mid$ to each edge $[i, j]$ in the interval graph $H_{G,I}$. Theorem 1 states that two k-interval colorings I_1 and I_2 of a graph G are equivalent if and only if the corresponding weighted graphs H_{G,I_1} and H_{G,I_2} are identical (i.e., they have the same edge set and the same weights on the edges). For example, the weighted interval graph associated with the 7-interval colorings of Figure 4 is represented in Figure 5. The two 7-interval colorings of Figure 3 are not equivalent since $\mid I_1(d) \cap I_1(f) \mid = 2 > 1 = \mid I_2(d) \cap I_2(f) \mid$. The corresponding weighted interval graphs are identical, except for the weight of the edge $[d, f]$ which is 2 in H_{G,I_1} and 1 in H_{G,I_2}.

3 Comparison of two algorithms for the ICP

In order to illustrate how the theoretical results of the previous section can help in the design of efficient algorithms for the ICP, we have developed two tabu search algorithms, the second one being based on Theorem 1 for avoiding the visit of equivalent solutions. We will show on a limited set of instances that the second tabu search algorithm possibly finds better solutions than the first one. The computational experiments are not meant to be exhaustive, but rather indicative and should help to orient future research on the development of more elaborate algorithms for the ICP.

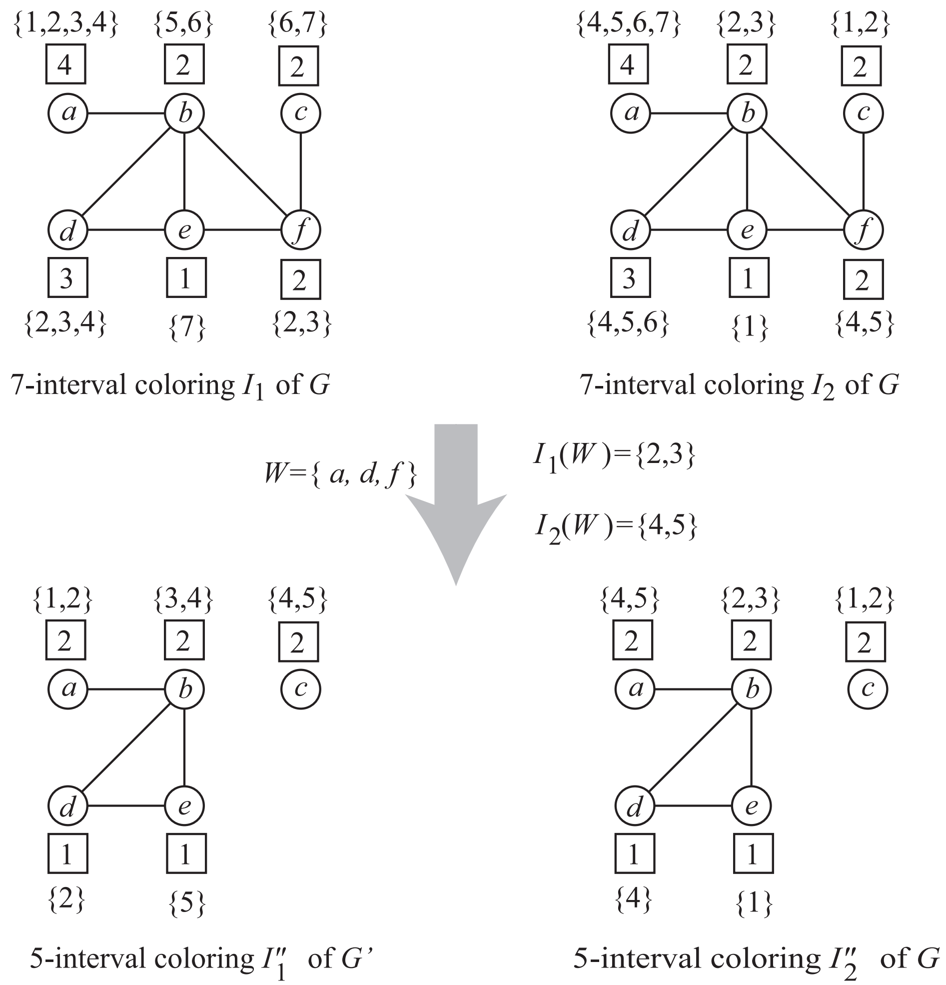

Figure 4: Illustration of the proof of Theorem 1.

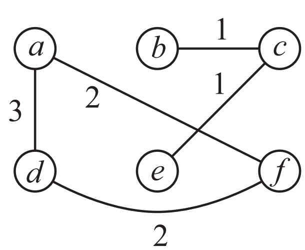

Figure 5: The weighted interval graph associated with the 7-interval colorings of Figure 4.

3.1 Two tabu search algorithms for the ICP

Let S be the set of solutions to a combinatorial optimization problem, and f a function to be minimized over S. For a solution $s \in S$, let $N(s)$ denote the *neighborhood* of s which is defined as the set of solutions in S obtained from s by performing a local change, called *move*. A local search is an algorithm that generates a sequence $s_0, s_1, \ldots, s_r$ of solutions in S, where s_0 is an initial solution and each s_i ($i \geq 1$) belongs to $N(s_{i-1})$. Tabu search is one of the most famous local search algorithms. In order to avoid cycling, tabu search uses a *tabu list* T that contains forbidden moves. Hence, a move m from s_{i-1} to s_i can only be performed if m does not belong to the tabu list T, unless $f(s_i) < f(s^*)$, where s^* is the best solution encountered so far. The general scheme of a tabu search algorithm is given in Figure 6. For more details on tabu search, the reader may refer to [10].

Tabu search

Generate an initial solution $s \in S$, set $T \leftarrow \emptyset$ and $s^* \leftarrow s$;
while no stopping criterion is met **do**
  Determine a solution $s' \in N(s)$ with minimum value $f(s')$ such that the move from s to s' does not belong to T or $f(s') < f(s^*)$;
  if $f(s') < f(s^*)$ **then**
    set $s^* \leftarrow s'$;
  Set $s \leftarrow s'$ and update T.

Figure 6: General scheme of a tabu search algorithm.

In order to illustrate how the theoretical results of the previous section can help in the design of efficient algorithms for the ICP, we have developed two tabu search algorithms, the second one being based on Theorem 1 for avoiding the visit of equivalent solutions. Both tabu search algorithms are heuristic methods for the k-ICP. They are used to solve the ICP with the following scheme.

1. Determine an upper bound k on $\chi_{int}(G)$.
2. Apply tabu search for the $(k-1)$-ICP; if the output is a legal $(k-1)$-interval coloring then set $k \leftarrow k-1$ and repeat step 2, else return k.

Tabucol [13] is a tabu search algorithm for the classical k-coloring problem (i.e., for the k-ICP with $\omega_i = 1$ for all vertices $i \in V$). The search space S is the set of (not necessary legal) k-colorings of G. A solution $c \in S$ is therefore a partition of the vertex set into k subsets $V_1, \ldots, V_k$. The evaluation function f measures the number of conflicting edges. Hence, for a solution $c = (V_1, \ldots, V_k)$ in S, $f(c)$ is equal to $\sum_{i=1}^{k} \mid E_i \mid$, where E_i denotes the set of edges with both endpoints in V_i. The goal of *Tabucol* is to determine a k-coloring c such that $f(c) = 0$. Given a k-coloring c, a neighbor k-coloring $c' \in N(c)$ is obtained by choosing an endpoint i of a conflicting edge and assigning a new color $c'(i) \neq c(i)$ to i. When modifying the color $c(i)$ of a vertex i, the tabu list stores the ordered triple $(i, c(i), f(c))$, which means that for some number of iterations, all moves to a solution c' with $f(c') \geq f(c)$ and $c'(i) = c(i)$ have a tabu status.

The first proposed tabu search algorithm, called TABU_1, is a simple adaptation of *Tabucol* to the k-ICP. The search space S is the set of (not necessary legal) k-interval colorings of G. The evaluation function f measures the total overlap of intervals on adjacent vertices. More

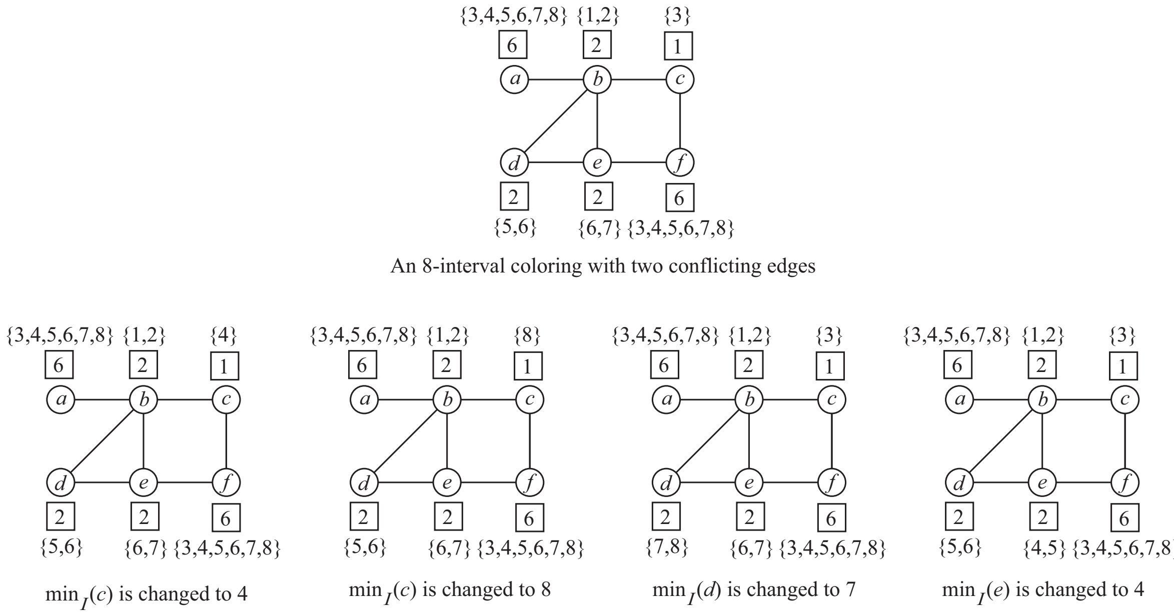

Figure 7: An 8-interval coloring with 4 equivalent neighbors.

precisely, given a k-interval coloring I of G, we define

$$f(I) = \sum_{[i,j]\in E} \mid I(i) \cap I(j) \mid .$$

Hence, a k-interval coloring is legal if and only if $f(I) = 0$. Given a k-interval coloring I, a neighbor solution I' is obtained by choosing an endpoint i of a conflicting edge and assigning a new interval $I'(i) \neq I(i)$ to i. We denote $N_1(I)$ the set of such neighbors of I. Let $\min_I(i)$ denote the smallest integer in the interval $I(i)$. When modifying the interval $I(i)$ of a vertex i, the tabu list TL_1 stores the ordered triple $(i, \min_I(i), f(I))$, which means that for some number of iterations, a move to a solution I' with $f(I') \geq f(I)$ and $\min_{I'}(i) = \min_I(i)$ (i.e., with $I(i) = I'(i)$) has a tabu status.

Consider again the *Tabucol* algorithm for the classical k-coloring problem. Let $c' \in N(c)$ be a neighbor solution of a k-coloring c obtained by modifying the color of vertex i. Then c and c' are equivalent if and only if $c(j) \notin \{c(i), c'(i)\}$ for all vertices $j \neq i$, which is most unlikely. The situation is totally different for the k-ICP. For example, consider the 8-interval coloring I at the top of Figure 7. It contains three conflicting edges, namely $[c, f]$, $[d, e]$ and $[e, f]$. Hence, a neighbor of I is obtained by changing the interval associated with vertex c, d, e or f, which gives a total of 21 neighbors in $N_1(I)$. Four of these neighbors are equivalent to I and represented at the bottom of Figure 7.

The second tabu search algorithm, called TABU$_2$, has only two differences with TABU$_1$. The first difference is on the definition of the neighborhood of a k-interval coloring. In order to avoid visiting equivalent solutions, the neighborhood $N_2(I)$ of a k-interval colorings I is defined as the subset of solutions $I' \in N_1(I)$ such that there exists at least one conflicting edge $[i, j]$ with $\mid I(i) \cap I(j) \mid \neq \mid I'(i) \cap I'(j) \mid$. Hence, given a k-interval coloring I, a neighbor solution $I' \in N_2(I)$ is obtained by choosing a vertex $i \in V$ that is the endpoint of a conflicting edge and assigning a new interval $I'(i) \neq I(i)$ to i such that there is at least one vertex j

adjacent to i with $| I(i) \cap I(j) | > 0$ and $| I(i) \cap I(j) | \neq | I'(i) \cap I(j) |$. According to Theorem 1 this is sufficient to ensure that no solution in $N_2(I)$ is equivalent to I. For the 8-interval coloring I at the top of Figure 7, $N_1(I)$ contains 21 neighbors while $N_2(I)$ contains only 13 neighbors obtained by setting $\min_{I'}(c) = 1$ or 2, $\min_{I'}(d) = 1, 2, 3, 4$ or 6, $\min_{I'}(e) = 1, 2, 3, 5$ or 7, or $\min_{I'}(f) = 1$. The four neighbors $I' \in N_1(I)$ represented at the bottom of Figure 7 do not belong to $N_2(I)$ since they are equivalent to I. In addition, $N_2(I)$ does not contain the neighbors I' with $\min_{I'}(c) = 5, 6$ or 7 or $\min_{I'}(f) = 2$ since these values do not change the size of the overlap of the intervals on the endpoint of a conflicting edge.

The second difference between TABU$_1$ and TABU$_2$ is on the definition of a tabu move. We consider a second tabu list TL_2, and a move is declared tabu if both tabu lists assign a tabu status to the move. The second tabu list is defined as follows. When modifying the interval associated with a vertex i for moving from a solution I to a neighbor solution $I' \in N_2(I)$, we consider all vertices j such that $| I(i) \cap I(j) | > 0$ and $| I(i) \cap I(j) | \neq | I'(i) \cap I(j) |$, and for each such vertex we insert the ordered quadruple $(i, j, | I(i) \cap I(j) |, f(I))$ in a second tabu list TL_2. The move from a solution I to a solution I' is considered as tabu according to TL_2 if there exists $(i, j, q, r) \in TL_2$ such that $| I(i) \cap I(j) | \neq | I'(i) \cap I'(j) | = q$ and $f(I') \geq r$. Notice that an ordered quadruple (i, j, q, r) is introduced in TL_2 only if $q > 0$, which means that we never impose that two intervals should not overlap since this would forbid the visit of too many solutions. For illustration, if we move from the 8-interval coloring I at the top of Figure 7 to a neighbor solution I' by setting $I'(f) = \{1, 2, 3, 4, 5, 6\}$, then $(a, f, 6, 4)$ and $(e, f, 2, 4)$ are introduced in the tabu list. While $| I(b) \cap I(f) | \neq | I'(b) \cap I'(f) |$, the ordered quadruple $(b, f, 0, 4)$ does not enter the tabu list since $| I(b) \cap I(f) | = 0$.

3.2 Computational experiments

We report computational experiments on 12 DIMACS benchmark graphs having up to 125 vertices. These instances have also been considered in [1], and therefore constitute a test set on which comparisons can be made with other algorithms. For a detailed description of these instances, the reader can refer to [19].

For measuring the performance of the proposed algorithms, we also report known lower and upper bounds on the optimal solution. More precisely, Čangalović and Schreuder in [4] have described an exact algorithm for finding the interval chromatic number. It is based on the Branch-and-Bound principle. An initial lower bound $LB(G)$ on $\chi_{int}(G)$ is obtained by determining a clique of maximum total weight, using a variation of the algorithm proposed in [18]. Also, an initial upper bound $UB(G)$ on $\chi_{int}(G)$ is obtained by using the heuristic algorithm proposed in [3]. Moreover, two truncated Branch-and-Bound algorithms for the ICP are proposed in [1]. Both algorithms are run with a time limit of one hour. Their output is either the optimal value (i.e., the interval chromatic number), or an upper bound on $\chi_{int}(G)$.

When using TABU$_1$ or TABU$_2$, we start the search with $k = UB(G)$. Then, as explained at the beginning of Section 3, we decrease k by one unit if a legal $(k-1)$-interval coloring is found, and this process is repeated until a time limit of one hour is reached. All tests were performed on an Intel(R) Core(TM)2 cpu 6400/2.13GHz. According to preliminary experiments, the duration of a tabu status for the first tabu list is randomly chosen at each iteration in the interval $[\sqrt{k \mid V \mid}, 3\sqrt{k \mid V \mid}\,]$ while the interval $[\frac{1}{2} \mid V \mid \sqrt{\max_{i \in V} \omega_i}\,,\ \frac{3}{2} \mid V \mid \sqrt{\max_{i \in V} \omega_i}\,]$ is used for the second tabu list. A better tuning of these parameters is certainly possible, but the chosen values turned to be the best for our limited test set.

Table 1: Results for DIMACS benchmark graphs.

instance							TABU$_1$			TABU$_2$		
name	n	m	max ω	LB	UB	Trunc BB	Best	Worse	Average	Best	Worse	Average
DSJC125.1g	125	736	5	19	23	**19**	**19**	**19**	**19**	**19**	**19**	**19**
DSJC125.5g	125	3891	5	40	72	68	62	63	62.8	62	63	62.8
DSJC125.9g	125	6961	5	122	166	163	153	154	153.4	152	154	152.8
R50_5g	50	612	5	27	34	32	31	32	31.6	31	32	31.4
R50_9g	50	1092	5	64	73	68	66	66	66	66	66	66
R75_1g	70	251	5	14	19	**16**	**16**	**16**	**16**	**16**	**16**	**16**
R75_5g	75	1407	5	31	50	46	43	44	43.2	43	44	43.2
R75_9g	75	2513	5	85	104	101	95	96	95.4	96	96	96
R100_1g	100	509	5	15	19	**17**	**17**	**17**	**17**	**17**	**17**	**17**
R100_5g	100	2456	5	35	58	54	49	50	49.8	49	50	49.6
R100_9g	100	4438	5	108	140	134	126	127	126.8	125	127	126.4
average				50.91	68.91	65.27	61.54	62.18	61.90	61.45	62.18	61.83

The three first columns of Table 1 contain the name of the instances, their number n of vertices, and their number m of edges. The next column indicates the largest vertex weight (column labeled "max ω") which also corresponds to the number of different vertex weights. We then report the value of the lower and upper bounds $LB(G)$ and $UB(G)$ (columns labeled "LB" and "UB") mentioned above. Column labeled "Trunc BB" contains the best upper bound obtained in [1] with one of the two truncated Branch-and-Bound algorithms. When a proof of optimality was obtained, we use bold numbers. The next three columns contain the results obtained using TABU$_1$. We ran TABU$_1$ five times on each graph, and columns "Best", "Worse" and "Average" contain the best, the worse and the average solution values we have reached. Again, we use bold numbers when a solution produced with TABU$_1$ is known to be optimal (because it reaches the lower bound $LB(G)$ or is equal to an optimal value reported in column "Trunc BB"). The last three columns contain the same information for TABU$_2$. The last line contains average numbers for each column.

We observe that both algorithms produce better results than those reported in [1] (column "Trunc BB"). We cannot conclude from these 12 benchmark problems that TABU$_2$ is better than TABU$_1$. However, we can observe that the use of Theorem 1 for avoiding the visit of equivalent solutions has helped to reduce the number of colors to 152 for instance DSJC125.9g and to 125 for R100_9g. Although TABU$_2$ is in average slightly better than TABU$_1$, it could never reach a 95-interval coloring for R75_9g, while such a solution was obtained on three of the five runs with TABU$_1$.

4 Conclusion

Two k-interval colorings I_1 and I_2 are said *equivalent* if there is a permutation π of the integers $1, \cdots, k$ such that $\ell \in I_1(i)$ if and only if $\pi(\ell) \in I_2(i)$ for all vertices $i \in V$. We have shown that a necessary and sufficient condition for such an equivalence is to have $\mid I_1(i) \cap I_1(j) \mid = \mid I_2(i) \cap I_2(j) \mid$ for all vertices i and j. Hence, equivalent solutions to the k-ICP are easy to recognize and we have shown that a tabu search algorithm for the k-ICP can possibly be improved by forbidding the visit of equivalent solutions.

While the two proposed tabu search algorithms produce reasonably good results in comparisons with those published in [1], we do not argue that they constitute the best possible algorithms for the ICP. The experiments reported in Section 3.2 should help to orient future research on the development of more elaborate algorithms for the ICP.

References

[1] M. Bouchard, M. Čangalović, A. Hertz. On a reduction of the interval coloring problem to a series of bandwidth coloring problems. Technical report, *Les Cahiers du GERAD, G–2007–69*, Montréal, Canada, 2007.

[2] A.L. Buchsbaum, H. Karloff, C. Kenyon, N. Reingold, M. Thorup. OPT versus LOAD in dynamic storage allocation. *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, 632–646, 2003.

[3] A.T. Clementson A.T., C.H. Elphick. Approximate coloring algorithms for composite graphs. *Journal of Operational Research Society*, 34/6:503–509, 1983.

[4] M. Čangalović, J.A.M. Schreuder. Exact colouring algorithm for weighted graphs applied to timetabling problems with lectures of different lengths. *European Journal of Operational Research*, 51:248–258, 1991.

[5] J. Fabri. Automatic storage optimization. *ACM SIGPLAN Notices: Proceedings of the 1979 SIGPLAN symposium on Compiler construction*, 14/8:83–91, 1979.

[6] P. Galinier, J.K. Hao. Hybrid evolutionary algorithms for graph colorings. *Journal of Combinatorial Optimization*, 3:379–397, 1999.

[7] P. Galinier, A. Hertz. A survey of local search methods for graph coloring. *Computers & Operations Research*, 33:2547–2562, 2006.

[8] M.R. Garey, D.S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W.H. Freman and Company, NY, 1979.

[9] J. Gergov. Algorithms for compile-time memory optimization. *Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms*, 907–908, 1999.

[10] F. Glover, M. Laguna. *Tabu Search*. Kluwer Academic Publishers, Boston, 1997.

[11] M. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, NY, 1980.

[12] G. Hajós. Über eine Art von Graphen. *Int. Math. Nachrichtern*, 11, 1957.

[13] A. Hertz and D. de Werra, Using Tabu Search Techniques for Graph Coloring, *Computing*, 39:345–351, 1987.

[14] M. Kubale. Interval vertex-coloring of a graph with forbidden colors. *Discrete Mathematics*, 74:125–136, 1989.

[15] P.M. Pardalos, T. Mavridou, J. Xue,. The graph coloring problem: A bibliographic survey. *Handbook of Combinatorial Optimization*, Kluwer Academic Publishers, 2:331–395, 1998.

[16] S.V. Pemmaraju, S. Penumatcha, R. Raman. Approximating Interval Coloring and Max-Coloring in Chordal Graphs *ACM Journal of Experimental Algorithmics*, 10:1–19, 2005.

[17] A. Punter. *Systems for timetabling by computer based on graph coloring*. Ph.D. Thesis, C.N.A.A., Hatfield Polytechnic, 1976.

[18] T. Sakaki, K. Nakashima, Y. Hattori. Algorithms for finding in the lump both bounds of the chromatic number of a graph. *The Computer Journal*, 19:329–332, 1976.

[19] M.A. Trick. *Computational symposium: Graph coloring and its generalizations*. Cornell University, Ithaca, Ny, 2002. `http://mat.gsia.cmu.edu/COLOR04`.

[20] D. de Werra, A. Hertz. Consecutive colorings of graphs. *Zeitschrift für Operations Research*, 32/1:1–8, 1988.