Les Cahiers du GERAD ISSN: 0711–2440

A Comparison of Integer and Constraint Programming Models for the Deficiency Problem

S. Altinakar, G. Caposossi,
A. Hertz

G–2014–25

April 2014

La publication de ces rapports de recherche bénéficie d'une subvention du Fonds de recherche du Québec – Nature et technologies.
Dépôt légal – Bibliothèque et Archives nationales du Québec, 2014.

A Comparison of Integer and Constraint Programming Models for the Deficiency Problem

Sivan Altinakar

GERAD & Département de mathématiques et de génie industriel
Polytechnique Montréal
Montréal (Québec) Canada, H3C 3A7

sivan.altinakar@gerad.ca

Gilles Caporossi

GERAD & Département des sciences de la décision
HEC Montréal
Montréal (Québec) Canada, H3T 2A7

gilles.caporossi@gerad.ca

Alain Hertz

GERAD & Département de mathématiques et de génie industriel
Polytechnique Montréal
Montréal (Québec) Canada, H3C 3A7

alain.hertz@gerad.ca

April 2014

Les Cahiers du GERAD

G–2014–25

Abstract: An edge-coloring of a graph $G = (V, E)$ is a function c that assigns an integer $c(e)$ (called color) in $\{0, 1, 2, \dots\}$ to every edge $e \in E$ so that adjacent edges share different colors. An edge-coloring is compact if the colors of the edges incident to every vertex form an interval of consecutive integers. The deficiency problem is to determine the minimum number of pendant edges that must be added to a graph such that the resulting graph admits a compact edge-coloring. We propose and analyze three integer programming models and one constraint programming model for the deficiency problem.

Key Words: Compact edge-colorings; integer linear programming; constraint programming.

Résumé : Une coloration des arêtes d'un graphe G est une fonction qui attribue un entier (appelé couleur) à chaque arête de G de telle sorte que les arêtes adjacentes aient des couleurs différentes. Une coloration des arêtes est compacte si les couleurs incidentes à chaque sommet forment un intervalle d'entiers consécutifs. Le problème de la déficience consiste à déterminer le nombre minimum d'arêtes pendantes qui doivent être rajoutées de telle sorte que le graphe résultant admette une coloration compacte de ses arêtes. Nous proposons et analysons trois modèles de programmation mathématique en nombres entiers et un modèle de programmation par contraintes pour le problème de la déficience.

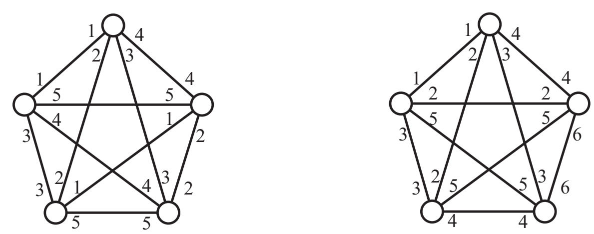
1 Introduction

All graphs considered in this paper are connected, have no loops, but may contain parallel edges. An *edge-coloring* of a graph $G = (V, E)$ is a function $c : E \to \{0, 1, 2, \dots\}$ that assigns a color $c(e)$ to every edge $e \in E$ such that $c(e) \neq c(e')$ whenever e and e' share a common endpoint. A *k-edge-coloring* is a similar function, but uses only colors in $\{0, 1, \dots, k-1\}$. Let E_v denote the set of edges incident with vertex $v \in V$. The *degree* $\deg_v$ of a vertex v is the number of edges in E_v and the maximum degree in G is denoted $\Delta(G)$. Note that all k-edge-colorings of a graph G use at least $\Delta(G)$ different colors, which means that $\Delta(G) \leq k$.

An edge-coloring of a graph $G = (V, E)$ is *compact* if $\{c(e) : e \in E_v\}$ is a set of consecutive positive integers for all vertices $v \in V$. The terms *consecutive edge-colorings* [5, 8] and *interval edge-colorings* [2, 3, 9, 10, 14, 16] are also used by some authors. A graph is *compactly colorable* if it admits a compact edge-coloring. For an edge-coloring c of a graph $G = (V, E)$, let $\underline{c}(v) = \min_{e \in E_v}\{c(e)\}$ and $\overline{c}(v) = \max_{e \in E_v}\{c(e)\}$ denote, respectively, the smallest and the largest color assigned to an edge incident to v. It follows from the above definitions that if c is compact, then $\overline{c}(v) = \underline{c}(v) + \deg_v - 1$ for all vertices $v \in V$.

The problem of determining whether or not a given graph is compactly colorable is known to be $\mathcal{NP}$-complete [16], even for bipartite graphs. Given a k-edge-coloring c of a graph G, let $d_v(G, c)$ denote the minimum number of integers that must be added to $\{c(e) : e \in E_v\}$ to form an interval of consecutive integers. The *deficiency* of c is defined as the sum $d(G, c) = \sum_{v \in V} d_v(G, c)$. Hence, c is compact if and only if $d(G, c) = 0$. The *deficiency of a graph* G, denoted $d(G)$, is the minimum deficiency $d(G, c)$ over all edge-colorings c of G. This concept, which was introduced by Giaro et al. [6], provides a measure of how close G is to be compactly colorable. Indeed, $d(G)$ is the minimum number of pendant edges that must be added to G such that the resulting graph is compactly colorable. The problem of determining the deficiency of a graph is $\mathcal{NP}$-hard [5]. This problem is also studied in [2, 6, 9, 10, 14, 15, 8, 4].

Vizing's theorem [17] guarantees the existence of a k-edge-coloring for all $k \geq \Delta(G) + 1$. But it may happen that $d(G, c) = d(G)$ only if c uses strictly more than $\Delta(G) + 1$ colors. For example, it not difficult to verify that the clique K_5 on five vertices has no edge-coloring with $\Delta(K_5) = 4$ colors, and that all 5-edge-colorings c of K_5 have a deficiency $d(K_5, c) = 3$. However, as illustrated in Figure 1, it is not difficult to color the edges of K_5 with six colors and a total deficiency of 2.



A 5-edge-coloring of K_5 with deficiency 3.   A 6-edge-coloring of K_5 with deficiency 2.

Figure 1: The minimum deficiency of K_5 can only be achieved by using at least $\Delta(G) + 2$ colors

The problem of determining a compact k-edge-coloring (if any) of a graph was introduced by Asratian and Kamalian [3]. It often arises in scheduling problems with compactness constraints [7]. For example, the open shop problem considers m processors $P_1, \cdots, P_m$ and n jobs $J_1, \cdots, J_n$. Each job J_i is a set of s_i tasks. Suppose that each task has to be processed in one time unit on a specific processor. No two tasks of the same job can be processed simultaneously and no processor can work on two tasks at the same time. Moreover, compactness requirements state that waiting periods are forbidden for every job and no idles are allowed on any processor. In other words, the time periods assigned to the tasks of a job must be consecutive, and each processor must be active during a set of consecutive periods. The existence of a feasible compact schedule with k time periods is equivalent to the existence of a compact k-edge-coloring of the graph G that contains one vertex for each job and each processor, and one edge for each task (i.e., a task of job J_i to be processed on P_j is represented by an edge between the vertices representing J_i and P_j). Each color used in the k-edge-coloring corresponds to a time period. The compactness requirements for each job and each processor are equivalent to imposing that the colors appearing on the edges of E_v must be consecutive for every vertex v in G. If the waiting periods of the jobs and the idles on the processors are not forbidden but their number has to be minimized, the problem is then to find an edge-coloring of G with minimum deficiency.

In this paper, we compare different models for computing the deficiency $d(G)$ of a graph G. In Section 2, we give an upper bound on the number of colors used in an edge-coloring with minimum deficiency. This bound is used to reduce the number of variables in the various models presented in Section 3. The performances of the proposed models are compared in Section 4.

2 An upper bound on the number of colors.

Let $s(G)$ be the smallest integer such that G admits an $s(G)$-edge-coloring c with deficiency $d(G, c) = d(G)$. Similarly, let $S(G)$ be the largest integer such that G admits an $S(G)$-edge-coloring c with deficiency $d(G, c) = d(G)$. We clearly have $\Delta(G) \leq s(G) \leq S(G)$. For example, it is not difficult to show that for G equal to a cordless cycle C_6 on six vertices, we have $s(G) = 2$ and $S(G) = 4$. A k-edge-coloring of C_6 with deficiency $d(C_6) = 0$ is shown in Figure 2 for $k = 2, 3, 4$. Note that a graph G does not necessarily admit a k-edge-coloring with deficiency $d(G)$ for all values of $k \in \{s(G), \cdots, S(G)\}$. For example, Sevastjanov [16] has given a graph G with $s(G) = 100$, $S(G) = 173$, and for which there is no k-edge-coloring with minimum deficiency when $k \in \{101, \cdots, 172\}$.

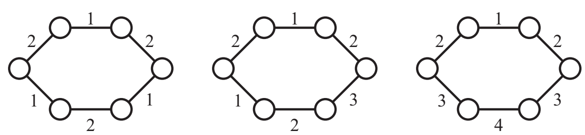


Figure 2: k-edge-colorings of C_6 with $k = 2, 3, 4$

Giaro et al. [8] have proved that if a graph G is compactly colorable (i.e., $d(G) = 0$), then $S(G) \leq 2n - 4$, where n is the number of vertices in G. We extend this result to all graphs G, using a similar proof as in [8].

Theorem 1 *If G is a graph with $n \geq 3$ vertices, then*

$$S(G) \leq 2n - 4 + d(G).$$

In order to prove this theorem we first need to define additional notions which are similar to those used in [8]. For an edge-coloring c, we denote c_{min} and c_{max} the minimum and maximum color used in c. Also, we denote V_c^{min} (V_c^{max}) the subset of vertices incident to an edge having color c_{min} (c_{max}). An e-path P for c is a simple path with vertex set $\{v_1, \cdots, v_p\}$, $p \geq 1$ such that $v_1 \in V_c^{min}$, $v_p \in V_c^{max}$, and every v_i is adjacent to v_{i+1} ($1 \leq i < p$). We denote e_i the edge that links v_i with v_{i+1}. Moreover:

- an i-hair ($1 \leq i \leq p$) of P is an edge e incident to v_i and such that
  - $c(e_i) < c(e) < c(e_{i+1})$ if $1 < i < p$,
  - $c(e) < c(e_1)$ if $i = 1$,
  - $c(e) > c(e_{p-1})$ if $i = p$;
- an i-node ($1 \leq i \leq p$) is the endvertex of an i-hair other than v_i.

Let V_P and E_P be the vertex set and the edge set of an e-path P, and let W_P be its set of i-nodes and H_P its set of i-hairs. The skeleton of P is the subgraph G_P of G with vertex set $V_P \cup W_P$ and edge set $E_P \cup H_P$. Clearly, if $d(G, c) = 0$, then the skeleton of every e-path P for c contains at least one edge of color k for each $k \in \{c_{min}, \cdots, c_{max}\}$. Giaro et al. [8] have proved that if c is an edge-coloring of G with $d(G, c) = d(G) = 0$, then G contains an e-path P for c such that the degree of every i-node in its skeleton G_P is 1 or 2. We are now ready to prove Theorem 1.

Proof of Theorem 1. Let c be an edge-coloring of G that uses $S(G)$ colors and such that $d(G,c) = d(G)$. We augment G to $\widetilde{G}$ by introducing $d_v(G,c)$ pendant edges to each deficient vertex v. Hence, a total of $d(G)$ new vertices and $d(G)$ new edges are added to G, and we can now extend c to a compact $S(G)$-edge-coloring $\widetilde{c}$ of $\widetilde{G}$ by assigning the missing colors to the new edges around each vertex v.

Consider an e-path P for $\widetilde{c}$ such that the degree of every i-node in its skeleton $\widetilde{G}_P$ is 1 or 2. Let R be the set of vertices in $\widetilde{G}_P$ that do not belong to G, and let I denote the set of i-nodes of $\widetilde{G}_P$ that do not belong to R. Let p denote the number of vertices in P and let m be the number of edges in $\widetilde{G}_P$. The number of vertices in $\widetilde{G}_P$ is then equal to $p + |I| + |R| \leq n + |R|$.

Note that the i-nodes of I can be of degree 1 or 2 in $\widetilde{G}_P$, but those in R are of degree 1. Since $\widetilde{c}$ uses $S(G)$ colors which all appear on the edges of $\widetilde{G}_P$, we have $S(G) \leq m$, where m is the number of edges in $\widetilde{G}_P$. Hence, it is sufficient to prove that m is not larger than $2n - 4 + d(G)$.

If $p = 1$, then $\widetilde{G}_P$ is a star and $m = |I| + |R| \leq (n-1) + d(G) \leq 2n - 4 + d(G)$. So assume $p \geq 2$. We then have

$$m \leq 2|I| + |R| + p - 1 \leq 2n - (p+1) + |R| \leq 2n - (p+1) + d(G).$$

If $p \geq 3$ then $m \leq 2n - 4 + d(G)$. So assume $p = 2$ while $m \geq 2n - 3 + d(G)$. Then the above inequalities become equalities. Consequently, each i-nodes in I has degree 2 in $\widetilde{G}_P$ and $n = |I| + 2$. Hence, $\widetilde{G}$ and $\widetilde{G}_P$ have the same vertex set, and $n \geq 3$ implies $|I| > 0$. It follows that the $|I|$ largest possible colors for the 1-hairs that link v_1 to the vertices in I are $\widetilde{c}(e_1) - |I|, \cdots, \widetilde{c}(e_1) - 1$, while the I smallest possible colors for the 2-hairs that link v_2 to the vertices in I are $\widetilde{c}(e_1) + 1, \cdots, \widetilde{c}(e_1) + |I|$. Since $\widetilde{c}$ is compact while every vertex in R has degree 1, $\widetilde{G}$ necessarily contains edges that link several pairs of vertices in I. The number of such edges is at least equal to

$$\frac{1}{2}\left(\sum_{j=1}^{|I|}(c(e_1) + j) - \sum_{j=1}^{|I|}(c(e_1) - j) - \sum_{j=1}^{|I|} 1\right) = \frac{|I|^2}{2}$$

But the number of such edges cannot be larger than $|I|(|I| - 1)/2$, a contradiction. □

For illustration, we have $d(K_3) = 1$ and $s(G) = S(G) = 3$. The above bound is therefore the best possible since $2n - 4 + d(G) = 3$ in this case.

3 Models

Four different models are described in this section. The first three of them are Integer Linear Programs (IP for short) while the last one is a Constraint Programming model (CP for short). The IP models use Boolean variables $c_{e,k}$ which are equal to 1 if color k is assigned to edge e. In order to reduce the number of variables, we consider an upper bound K on the number of possible colors. By choosing K such that $\Delta(G) \leq K < s(G)$, all edge-colorings produced by our models would have a deficiency strictly larger than $d(G)$. Ideally, we should use a value for K such that $s(G) \leq K \leq S(G)$: by setting $K = s(G)$, we would constrain the problem just enough to ensure the existence of at least one edge-coloring with minimum deficiency, and with $K = S(G)$ we would not reject any edge-coloring with minimum deficiency. But $s(G)$ and $S(G)$ are typically not known, which explains why we use an upper bound on $S(G)$.

As a corollary of Theorem 1, we have $S(G) \leq 2n - 4 + D$ for all $D \geq d(G)$. Since, to the best of our knowledge, no graph G with n vertices and $d(G) > n$ is known, we have decided to set $D = n$. In other words, we set the upper bound K on $S(G)$ equal to $3n - 4$. If the minimum deficiency d^* obtained with this upper bound is not larger than n, then we know that the bound is valid, which means that $d(G) = d^*$. Otherwise, if d^* is strictly larger than n, then $n < d(G) \leq d^*$, which means that G is a counter-example to the conjecture that $d(G) \leq n$ for all graphs G with n vertices. In such a case, we can solve the problem a second time, with the new upper bound $K = 2n - 4 + d^*$. The minimum deficiency obtained with this new bound is then equal to $d(G)$.

3.1 Integer linear programming models

3.1.1 First IP model (PART)

Model 1 is the most natural one. Let $C = \{0, \cdots, K-1\}$ denote the set of possible colors. As mentioned above, we use Boolean variables $c_{e,k}$ for every edge e and every color $k \in C$. We also consider integer variables $\underline{c}_v$ and $\overline{c}_v$ which correspond, respectively, to the minimal and maximal color assigned to an edge incident to vertex v. Finally, d_v is an integer variable whose value is the deficiency at vertex v.

The first two constraints define an edge-coloring, while the next three constraints define $\underline{c}_v$, $\overline{c}_v$ and d_v. The last constraint, although not strictly necessary, forces the usage of color 0 in order to eliminate equivalent solutions obtained by translating the colors in C.

3.1.2 Second IP model (COV)

Model 2 is a variation of the previous one, where we require that *at least* one color is assigned to each edge, instead of *exactly* one. We therefore solve a covering problem rather than a partitioning one.

3.1.3 Third IP model (STEP)

For Model 3, we replace integer variables $\underline{c}_v$ and $\overline{c}_v$ by Boolean variables $p_{v,k}$ and $q_{v,k}$ defined as follows:

- $p_{v,k} = 0$ if $k \geq \underline{c}_v$, and $p_{v,k-1} \geq p_{v,k}$ for $0 < k \leq \underline{c}_v$;
- $q_{v,k} = 0$ if $k \leq \overline{c}_v$, and $q_{v,k} \leq q_{v,k+1}$ for $\overline{c}_v \leq k < K$.

We therefore have $p_{v,k} + q_{v,k} + \sum_{e \in E_v} c_{e,k} \leq 1$ for all vertices v and all colors $k \in C$. Let $C_v^P = \{0, \ldots, K-1-\deg_v\}$ and $C_v^Q = \{\deg_v, \ldots, K-1\}$. In other words, $p_{v,k}$ ($q_{v,k}$) can have value 1 only if $k \in C_v^P$ (C_v^Q). At a vertex v, we then have $\underline{c}_v \geq \sum_{k \in C_v^P} p_{v,k}$ and $\overline{c}_v \geq K-1-\sum_{k \in C_v^Q} q_{v,k}$, which implies

$$\begin{aligned} d_v &= \overline{c}_v - \underline{c}_v + 1 - \deg_v \\ &\geq \left[K - 1 - \sum_{k \in C_v^Q} q_{v,k}\right] - \left[\sum_{k \in C_v^P} p_{v,k}\right] + 1 - \deg_v \\ &= K - \deg_v - \sum_{k \in C_v^P} p_{v,k} - \sum_{k \in C_v^Q} q_{v,k}. \end{aligned}$$

Note that the first, third and fourth constraints of this model impose $p_{v,k} = 0$ if $k \geq \underline{c}_v$ and $q_{v,k} = 0$ if $k \leq \overline{c}_v$. But it could happen that $p_{v,k} = 0$ for $k < \underline{c}_v$ and $q_{v,k} = 0$ for $k > \overline{c}_v$. For example, by setting $p_{v,k} = q_{v,k} = 0$ for all v and all k, we would have a feasible solution. However, minimizing $K - \deg_v - \sum_{k \in C_v^P} p_{v,k} - \sum_{k \in C_v^Q} q_{v,k}$ is equivalent to maximizing $\sum_{k \in C_v^P} p_{v,k} + \sum_{k \in C_v^Q} q_{v,k}$. Hence, at the optimal solution, we necessarily have $p_{v,k} = 1$ for $k < \underline{c}_v$ and $q_{v,k} = 1$ for $k > \overline{c}_v$, which means that

$$\sum_{v \in V} (K - \deg_v - \sum_{k \in C_v^P} p_{v,k} - \sum_{k \in C_v^Q} q_{v,k}) = \sum_{v \in V} d_v.$$

3.2 A constraint programming model (CP)

We have implemented a CP model for the deficiency problem. It uses integer variables c_e to denote the color assigned to an edge e. Using *allDifferent* constraints, it is trivial to enforce that edges incident to a vertex must have different colors. Also $\underline{c}_v$ and $\overline{c}_v$ can easily be defined by using the *max* and *min* functions.

Model 1 First IP model (PART)

$$
\begin{array}{lll}
\min & \sum_{v \in V} d_v & \\
\text{s.t.} & \sum_{e \in E_v} c_{e,k} \leq 1 & \forall v \in V, k \in C \\
& \sum_{k \in C} c_{e,k} = 1 & \forall e \in E \\
& \underline{c}_v \leq \sum_{k \in C} (k \cdot c_{e,k}) & \forall v \in V, \forall e \in E_v \\
& \overline{c}_v \geq \sum_{k \in C} (k \cdot c_{e,k}) & \forall v \in V, \forall e \in E_v \\
& d_v = \overline{c}_v - \underline{c}_v + 1 - \deg_v & \forall v \in V \\
& \sum_{e \in E} c_{e,0} \geq 1 & \\
& c_{e,k} \in \{0,1\} & \forall e \in E, k \in C \\
& \underline{c}_v \in \{0, \ldots, K - \deg_v\} & \forall v \in V \\
& \overline{c}_v \in \{\deg_v - 1, \ldots, K - 1\} & \forall v \in V \\
& d_v \in \{0, \ldots, K - \deg_v\} & \forall v \in V
\end{array}
$$

Model 2 Second IP model (COV)

$$
\begin{array}{lll}
\min & \sum_{v \in V} d_v & \\
\text{s.t.} & \sum_{e \in E_v} c_{e,k} \leq 1 & \forall v \in V, k \in C \\
& \sum_{k \in C} c_{e,k} \geq 1 & \forall e \in E \\
& \underline{c}_v \leq K - (K - k) \cdot c_{e,k} & \forall v \in V, \forall e \in E_v, \forall c \in C \\
& \overline{c}_v \geq k \cdot c_{e,k} & \forall v \in V, \forall e \in E_v, \forall c \in C \\
& d_v = \overline{c}_v - \underline{c}_v + 1 - \deg_v & \forall v \in V \\
& \sum_{e \in E} c_{e,0} \geq 1 & \\
& c_{e,k} \in \{0,1\} & \forall e \in E, k \in C \\
& \underline{c}_v \in \{0, \ldots, K - \deg_v\} & \forall v \in V \\
& \overline{c}_v \in \{\deg_v - 1, \ldots, K - 1\} & \forall v \in V \\
& d_v \in \{0, \ldots, K - \deg_v\} & \forall v \in V
\end{array}
$$

4 Computational results

4.1 Experimental setup

In order to compare the performances of the four proposed models, we consider two datasets obtained by using the NAUTY library, originally developed by Brendan McKay[12]:

Dataset D1. The first dataset is the complete collection of connected simple graphs with $4 \leq n \leq 8$ vertices.

Dataset D2. For a positive integer n and a real number $f \in [0.05, 0.95]$, let G_f^n denote the set of all connected graphs with n vertices and edge density in $[f - 0.05, f + 0.05[$. The second considered dataset is a collection of connected simple graphs, partitioned into subgroups denoted R_f^n. Each subgroup

Model 3 Third IP model (COV)

$$
\begin{aligned}
\min \quad & \sum_{v\in V}(K-\deg_v-\sum_{k\in C_v^P}p_{v,k}-\sum_{k\in C_v^Q}q_{v,k}) & \\
\text{s.t.} \quad & \sum_{e\in E_v}c_{e,k}+p_{v,k}+q_{v,k}\le 1 & \forall v\in V,k\in C\\
& \sum_{k\in C}c_{e,k}=1 & \forall e\in E\\
& p_{v,k-1}\ge p_{v,k} & \forall v\in V,k\in C_v^P,k\ne 0\\
& q_{v,k}\le q_{v,k+1} & \forall v\in V,k\in C_v^Q,k\ne K-1\\
& \sum_{e\in E}c_{e,0}\ge 1 & \\
& c_{e,k}\in\{0,1\} & \forall e\in E,k\in C\\
& p_{v,k}\in\{0,1\} & \forall v\in V,k\in C_v^P\\
& p_{v,k}=0 & \forall v\in V,k\notin C_v^P\\
& q_{v,k}\in\{0,1\} & \forall v\in V,k\in C_v^Q\\
& q_{v,k}=0 & \forall v\in V,k\notin C_v^Q
\end{aligned}
$$

Model 4 Constraint Programming (CP)

$$
\begin{aligned}
\min \quad & \sum_{v\in V}d_v & \\
\text{s.t.} \quad & \underset{e\in E_v}{\text{allDifferent}}\ c_e & \forall v\in V\\
& \underline{c}_v=\min_{e\in E_v}c_e & \forall v\in V\\
& \overline{c}_v=\max_{e\in E_v}c_e & \forall v\in V\\
& d_v=\overline{c}_v-\underline{c}_v+1-\deg_v & \\
& |\{e\in E\mid c_e=0\}|\ge 1 & \\
& c_e\in C & \forall e\in E\\
& \underline{c}_v\in\{0,\dots,K-\deg_v\} & \forall v\in V\\
& \overline{c}_v\in\{\deg_v-1,\dots,K-1\} & \forall v\in V\\
& d_v\in\{0,\dots,K-\deg_v\} & \forall v\in V
\end{aligned}
$$

R_f^n contains eight different graphs (when possible) chosen at random in G_f^n. We report results for $n\in\{4,\dots,100\}$ and $f\in\{0.1,0.2,\dots,0.9\}$. Note that for $n\le 7$ and a given f, it may happen that the number of graphs in G_f^n is strictly smaller than 8. In such a case, we set $R_f^n=G_f^n$.

We have run our models respectively with the CPLEX and CP OPTIMIZER solvers (version 12.2) by IBM/ILOG, on a laptop with an Intel Core 2 Duo CPU at 1.20GHz and 4Gb of RAM.

4.2 Model comparisons for dataset D1

We first compare the performances of the four models on dataset D1. All models have determined the deficiency of all graphs with at most 7 vertices. For $n=8$, only the CP model could determine all optimal solutions in a reasonable computing time. An analysis of the optimization process of each model clearly shows that optimal solutions are typically found instantaneously, but a proof of optimality can take minutes or hours in the most difficult cases.

Mean computing times are reported in Table 1 for $4 \leq n \leq 7$ vertices. The graphs are grouped according to the value of their deficiency, and the last line of the table indicates the number of graphs in each group. Another representation of the computing times is given in Figure 3, where we indicate the number of graphs for which each model had a computing time $t \in [2^i, 2^{i+1}[$, for various values of i. We observe that the CP model is faster than the other models, except for the clique K_7 on 7 vertices, which is the only graph with $n = 7$ vertices and deficiency 3. The PART model comes second.

Table 1: Mean computing times for dataset D1, with $n \leq 7$

n	4	5			6		7			
deficiency	0	0	1	2	0	1	0	1	2	3
PART	0.015	0.033	0.194	0.715	0.112	0.663	0.416	11.267	56.784	330.305
COV	0.019	0.061	0.298	3.051	0.188	2.724	0.803	48.505	462.991	8446.893
STEP	0.034	0.085	0.260	0.449	0.184	1.349	0.938	21.115	122.948	567.281
CP	0.000	0.003	0.004	0.050	0.002	0.011	0.003	0.297	41.842	4169.700
# of graphs	6	15	5	1	104	8	772	75	5	1

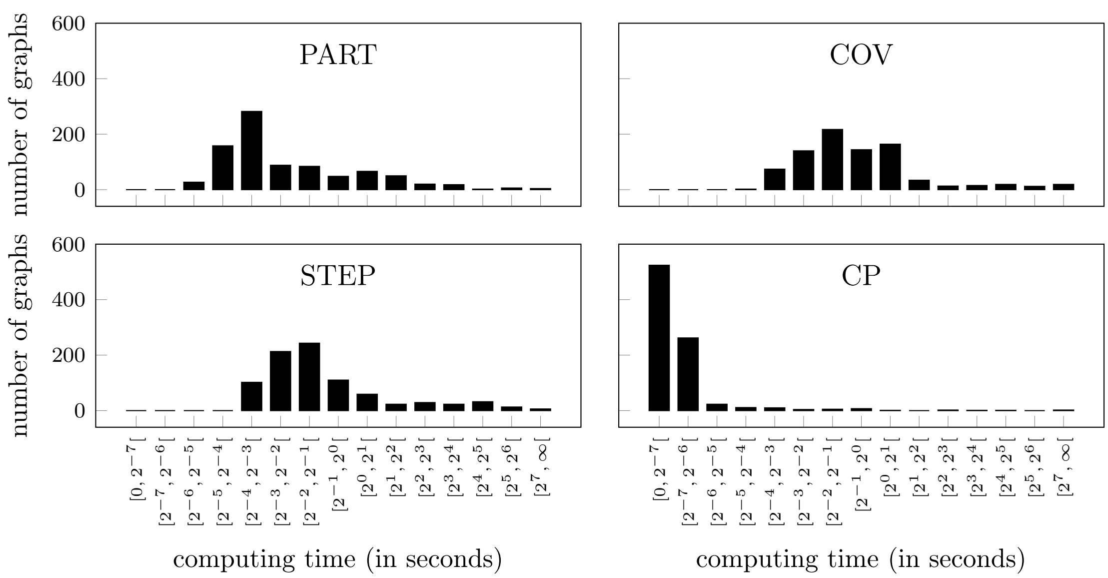


Figure 3: Histograms of computing times for dataset D1, with $n \leq 7$

We reach the same conclusion by comparing the models pairwise in Table 2, where we count the number of times a model is faster than another one, with a tolerance of 0.02 seconds. If, for a given graph, the computing time difference between two models is equal to or less than this tolerance, we consider both models equivalent for this graph. The entry at line A and column B of this table indicates the number of graphs (out of 992) for which the solution time for model A was smaller than that of model B by at least 0.02 seconds.

Table 2: Pairwise comparisons for dataset D1, with $n \leq 7$.

	PART	COV	STEP	CP
PART		830	802	1
COV	121		355	0
STEP	154	538		2
CP	961	988	990	

A more detailed view is given in Figure 4, where we indicate the number of graphs for which the difference in computing times lies in an interval $]\frac{2^i}{10}, \frac{2^{i+1}}{10}]$, or its negative equivalent, with a central bin $[-0.02, 0.02]$ for all cases within the tolerance of 0.02 seconds. We clearly see that the CP model dominates the other ones (almost all bins of the histograms in the last row are on the right side), and that PART is better than COV and STEP (most bins of the upper histograms of the first column are on the left side) . The last two models, seem to have overall comparable performances, even if on some occasions one does better than the other.

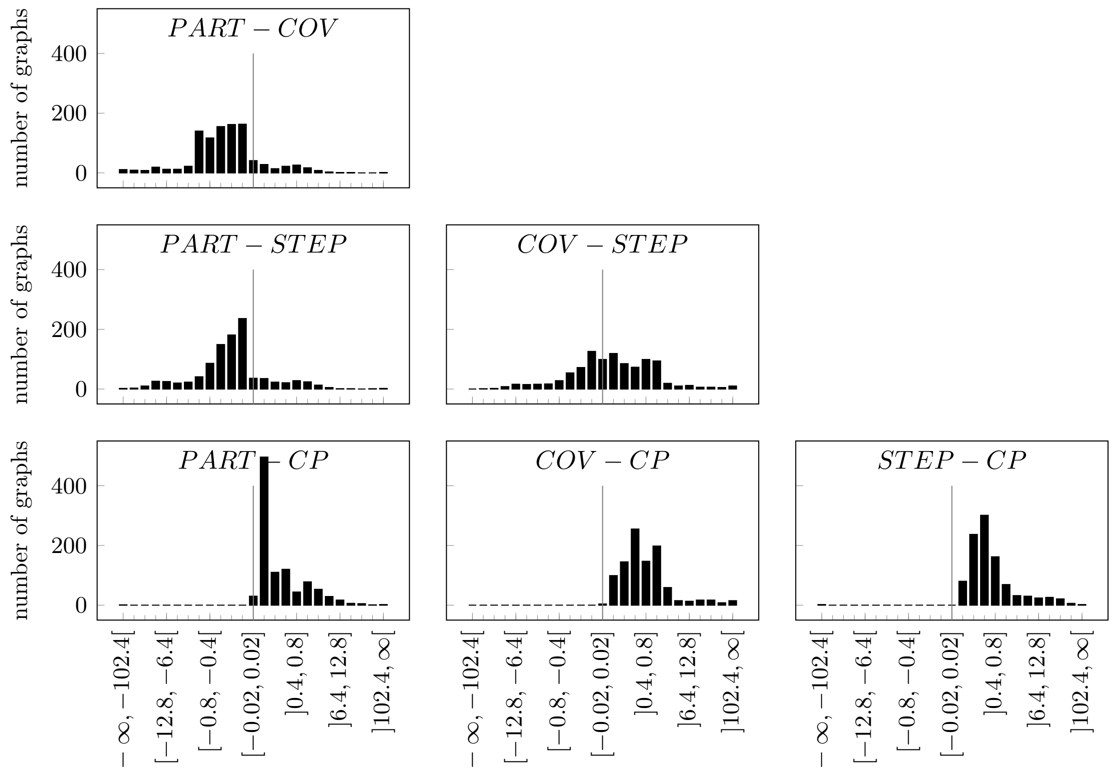


Figure 4: Histograms of the differences in computing times for dataset D1, with $n \leq 7$

As already mentioned above, the CP model is the only one that could determine, in a reasonable computing time, the deficiency of all graphs with $n = 8$ vertices. We show in Table 3 the distribution of the graphs with $4 \leq n \leq 8$ vertices, according to their number m of edges and their deficiency. The last line of the table indicates the average computing time needed by the CP model to determine an optimal solution. Also, we show in Figure 5 all graphs G with $n = 6$ and 8 vertices, and with largest deficiency. As indicated in Table 3 there are eight such graphs for $n = 6$ and four for $n = 8$. Sets of vertices forming either an independent set or a clique, and sharing the same neighborhood outside the set, are grouped in rectangles. An arbitrary permutation of the vertices in a group corresponds to an automorphism of the considered graph, and the existence of these groups therefore indicates that the addition of symmetry breaking constraints in our models could be very helpful for decreasing the computing time.

4.3 Model comparisons for dataset D2

For dataset D2, we have decided to compare the four models by limiting the computing time to 10 seconds on each graph. Given one of the models and a set R^n_f we consider the following properties:

- *all solved*: the model could determine the minimum deficiency of all graphs in R^n_f;
- *non solved*: the model has not determined any optimal solution for the graphs in R^n_f;
- *all feasible*: the model could determine a feasible solution for all graphs in R^n_f;
- *non feasible*: the model has not determined any feasible solution for the graphs in R^n_f.

Table 3: The distribution of all graphs G with $4 \leq n \leq 8$ vertices, according to their number m of edges and their deficiency

n	4	5			6		7				8		
m deficiency	0	0	1	2	0	1	0	1	2	3	0	1	2
3	2												
4	2	3											
5	1	4	1		6								
6	1	5			13		11						
7		2	2		18	1	32	1			23		
8		1	1		21	1	67				89		
9			1		19	1	101	6			234	2	
10				1	12	2	127	5			483	3	
11					7	2	125	13			797	17	
12					4	1	123	3			1165	4	
13					2		79	16			1412	42	
14					1		54	8	2		1552	27	
15					1		32	8			1483	32	
16							13	8			1274	16	
17							5	4	1		926	44	
18							3	1	1		638	19	1
19								2			376	24	
20									1		215	5	
21										1	103	10	1
22											51	3	2
23											20	4	
24											11		
25											4	1	
26											2		
27											1		
28											1		
# of graphs	6	15	5	1	104	8	772	75	5	1	10860	253	4
mean time	0.000	0.003	0.004	0.050	0.002	0.011	0.003	0.297	41.842	4169.700	0.015	6.865	700.735

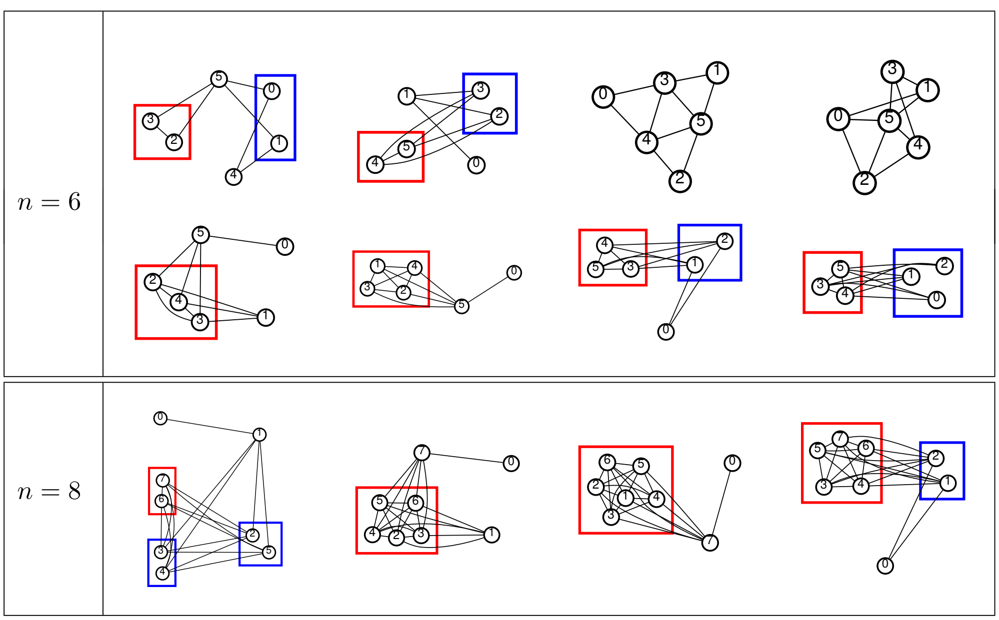


Figure 5: All graphs with $n = 6$ and 8 vertices and with largest deficiency

For every $f \in \{0.1, 0.2, \ldots, 0.9\}$, we show in Figure 6 the largest n for which the *all solved* and *all feasible* properties were satisfied, and the smallest n for which the *non solved* and *non feasible* properties were satisfied. Note that the *all feasible* and *non feasible* curves for the CP model do not appear on Figure 6, the reason being that CP was able to determine feasible solutions for all graphs with 100 vertices. Figure 7 superimposes the four *all solved* curves.

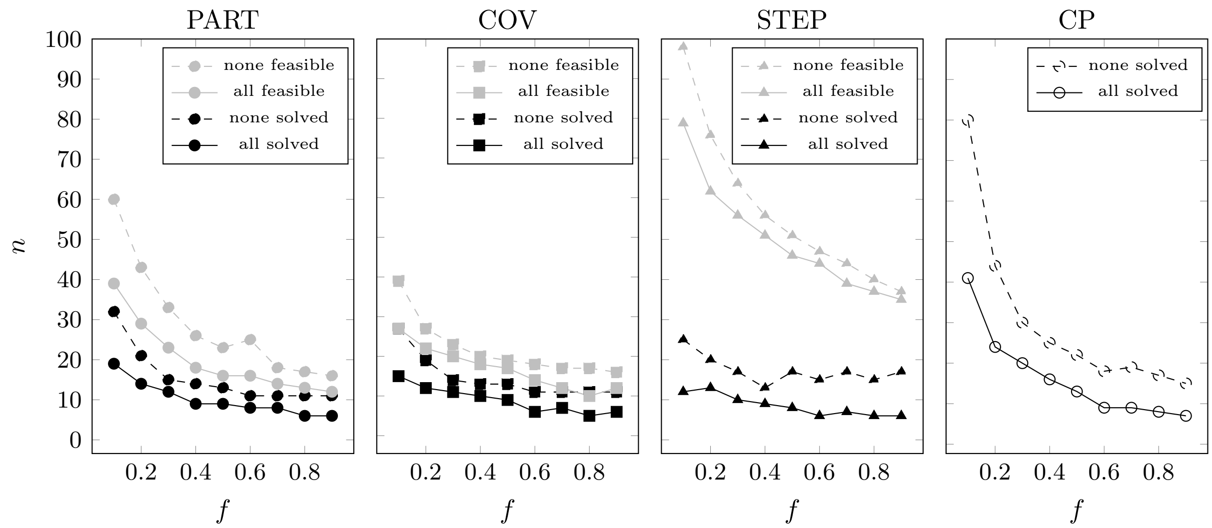


Figure 6: The none feasible, all feasible, none solved and all solved curves of the four models

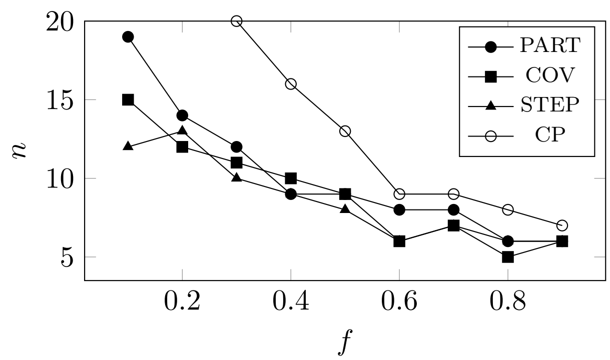


Figure 7: The all solved curves

The *all solved* curves of Figure 7 clearly show that all models struggle with denser graphs. Again, CP does sensibly better than the other models, among which PART seems to have a slight advantage. The downward dent at density 0.6 for COV and STEP is due to the existence of a 1-deficient graph in $R^7_{0.6}$ (for which both models need more than 10 seconds to determine the minimum deficiency) while $R^7_{0.5}$ and $R^7_{0.7}$ contain only graphs G with $d(G) = 0$.

When considering the *all feasible* and *non feasible* curves, we can observe that the STEP model performs better than PART and COV. However, the feasible solutions produced by STEP have a much larger deficiency than those produced by the CP model. To illustrate this fact, let $N(f)$ denote the largest number of vertices for which the *all feasible* property is satisfied by STEP for a given f, and let $N'(f) = \lfloor 0.9N(f) \rfloor$. The values of $N'(f)$ are given in Table 4. We show in Figure 8 the maximum, the minimum and the average deficiency of the feasible edge-colorings produced by STEP and CP for the graphs in R^n_f, with $n = N'(f)$.

4.4 Other remarks

As mentioned in Section 2, the upper bound $K = 3n-4$ on $S(G)$ is possibly larger than the theoretical bound $2n-4+d(G)$. For all graphs G for which we were able to determine the minimum deficiency, we have therefore

Table 4: Values of $N(f)$ and $N'(f)$

density	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$N(f)$	79	62	56	51	46	44	39	37	35
$N'(f)$	71	55	50	45	41	39	35	33	31

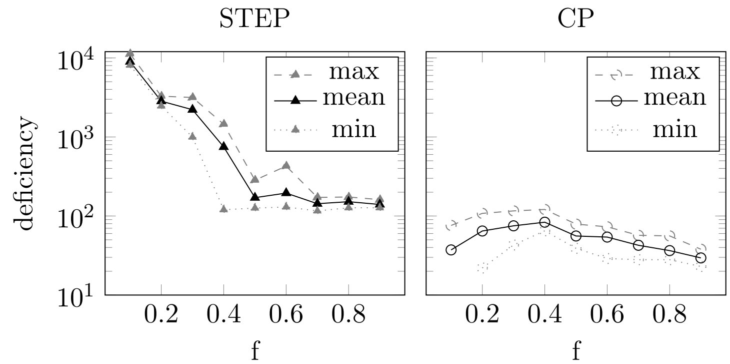


Figure 8: Maximum, minimum and average deficiencies of the feasible solutions found by STEP and CP in $R_f^{N'(f)}$

made a second run of our models, but with $K = 2n-4+d(G)$. We could observe a tiny improvement in term of computing time, but nothing meaningful enough that would justify the implementation of a technique that would dynamically decrease the upper bound on $S(G)$ each time a feasible solution is found with deficiency strictly smaller than $K-2n+4$.

Also, as already mentioned when analyzing the graphs with $n = 6$ and 8 vertices, and with largest deficiency, symmetry breaking rules can possibly help to decrease the computing time. We have implemented *manually* some of these rules for dense graphs which tend to have many automorphisms and are therefore more challenging for our models. Thanks to the addition of these constraints, we were able to drastically decrease the computing time, which confirms that this is a promising research avenue for the solution of the deficiency problem on larger graphs.

5 Conclusion

The problem of determining the deficiency of a graph is surprisingly hard. Three of our integer linear programming models could comfortably process graphs with only up to seven vertices, and the CP model performed a little bit better by determining the deficiency of all graphs with 8 vertices. It appears that the main problem is not to generate an optimal solution, but rather to prove its optimality.

To limit the number of variables, all models use an upper K on the number of colors to be used. While $K \geq \Delta(G)+1$ is sufficient to guarantee the existence of an edge-coloring, it may happen that the use of more colors allows a smaller deficiency. To address this problem, we have shown that a coloring with minimum deficiency never uses more than $2n-4+d(G)$ colors. This bound on $S(G)$ guarantees that every optimal solution can possibly be obtained by our models. A bound on $s(G)$ would guarantee the existence of at least one edge coloring with minimum deficiency. We have run our models with $K = 3n-4$, which is a valid bound, unless the optimal value d^* produced by our models is larger than n, in which case we can perform a second run with the valid bound $K = 2n-4+d^*$.

A comparison of the performances of the four proposed models has clearly shown that the CP model is significantly better than the IP models. We finally found that the addition of symmetry-breaking constraints in our models seems to be a promising research avenue for decreasing the computing time.

References

[1] S. Altinakar, G. Caporossi, A. Hertz. On compact k-edge-colorings: A polynomial time reduction from linear to cyclic. *Discrete Optimization* 8:502–512, 2011.

[2] A.S. Asratian, C.J. Casselgren. On interval edge colorings of (α, β)-biregular bipartite graphs. *Discrete Mathematics* 307:1951–1956, 2006.

[3] A.S. Asratian, R.R. Kamalian. Interval colorings of the edges of a multigraph. *Applied Math.* 5:25–34, 1987. (In Russian)

[4] M. Bouchard, A. Hertz, G. Desaulniers. Lower bounds and a tabu search algorithm for the minimum deficiency problem. *Journal of Combinatorial Optimization* 17:168–191, 2009.

[5] K. Giaro. The complexity of consecutive Δ-coloring of bipartite graphs: 4 is easy, 5 is hard. *Ars Combinatoria* 47:287–298, 1997.

[6] K. Giaro, M. Kubale, M. Malafiejski. On the deficiency of bipartite graphs. *Discrete Applied Mathematics* 94:193–203, 1999.

[7] K. Giaro, M. Kubale, M. Malafiejski. Compact scheduling in open shop with zero-one time operations. *INFOR* 37:37–47, 1999.

[8] K. Giaro, M. Kubale, M. Malafiejski. Consecutive colorings of the edges of general graphs. *Discrete Mathematics* 236:131–143, 2001.

[9] D. Hanson, C.O.M. Loten. A lower bound for interval colouring of bi-regular bipartite graphs. *Bulletin of the Institute of Combinatorics and Applications* 18:69–74, 1996.

[10] D. Hanson, C.O.M. Loten, B. Toft. On interval colorings of bi-regular bipartite graphs. *Ars Combinatoria* 50:23–32, 1998.

[11] M. Kubale, A. Nadolski Chromatic scheduling in a cyclic open shop. *European Journal of Operational Research* 164:585–591, 2005.

[12] B.D. McKay, A. Piperno. Practical graph isomorphism, II *Preprint submitted to Elsevier, 9 January 2013.*

[13] A. Nadolski. Compact cyclic edge-colorings of graphs. *Discrete Mathematics* 308:2407–2417, 2008.

[14] A.V. Pyatkin. Interval coloring of $(3, 4)$-biregular bipartite graphs having large cubic subgraphs. *Journal of Graph Theory* 47(2):122–128, 2004.

[15] A. Schwartz. The deficiency of a regular graph. *Discrete Mathematics* 306:1947–1954, 2006.

[16] S.V. Sevastianov. On interval edge colouring of bipartite graphs. *Metody Diskretnowo Analiza* 50:61–72, 1990. (In Russian)

[17] V.G. Vizing. On an estimate of the chromatic class of a p-graph *Diskret. Analiz.* 3:25–30, 1964. (In Russian)