New Bounds and Constraint Programming Models for the Weighted Vertex Coloring Problem

Olivier Goudet, Cyril Grelier and David Lesaint
LERIA, Université d’Angers, 2 Boulevard Lavoisier, 49045 Angers, France
{olivier.goudet,cyril.grelier,david.lesaint}@univ-angers.fr

Abstract
This paper addresses the weighted vertex coloring problem (WVCP) which is an NP-hard variant of the graph coloring problem with various applications. Given a vertex-weighted graph, the problem consists of partitioning vertices into independent sets (colors) so as to minimize the sum of the maximum weights of the colors. We first present an iterative procedure to reduce the size of WVCP instances and prove new upper bounds on the objective value and the number of colors. Alternative constraint programming models are then introduced which rely on primal and dual encodings of the problem and use symmetry breaking constraints. A large number of experiments are conducted on benchmark instances. We analyze the impact of using specific bounds to reduce the search space and speed up the exact resolution of instances. New optimality proofs are reported for some benchmark instances.

1 Introduction
Given a vertex-weighted graph, the weighted vertex coloring problem (WVCP) consists of partitioning vertices into independent sets (colors) so as to minimize the sum of the maximum weights of the colors. This problem has applications in different domains ranging from metropolitan area network design [Halldórsson and Shachnai, 2008] and batch scheduling in distributed computing [Liu et al., 2006] to traffic assignment in telecommunications [Prais and Ribeiro, 2000].

Formally, a WVCP instance \( P = (G, w) \) is defined by an undirected graph \( G = (V, E) \) with vertex set \( V \) and edge set \( E \) and a function \( w: V \rightarrow \mathbb{N}^* \) assigning a strictly positive weight \( w(v) \) to each vertex \( v \). A (legal) coloring of \( P \) is a partition \( \{V_1, \ldots, V_k\} \) of \( V \) into \( k \) independent sets of vertices, that is, no pair of vertices in each color \( V_i \) is connected in \( G \). The objective is to find a coloring \( S = \{V_1, \ldots, V_k\} \) whose score \( f(S) = \sum_{i=1}^{k} \max_{v \in V_i} w(v) \) is minimum.

WVCP generalizes the graph coloring problem (GCP) which is NP-hard and consists in determining the chromatic number \( \chi_G \) of a graph \( G \), i.e., the minimum size of its colorings. Indeed GCP is the class of WVCP instances whose vertex weights are all equal. Therefore WVCP is also NP-hard.

Various complexity and approximability results have been established for specific classes of WVCP (see e.g. [Boudhar and Finke, 2000; Demange et al., 2007]). New exact methods have been proposed more recently, notably [Cornaz et al., 2017] which is effective for dense instances, as well as powerful metaheuristics [Wang et al., 2020; Nogueira et al., 2021; Grelier et al., 2022] which have produced new upper bounds for medium and large instances of all types. Constraint Programming (CP) and hybrid CP methods have also been proposed for the graph coloring problem (e.g. [Gualandi and Malucelli, 2012; Hébrard and Katsirelos, 2020]) but not extended to WVCP to the best of our knowledge.

In this paper, we propose new vertex reduction rules, upper bounds and CP models for WVCP. Building on the work of [Wang et al., 2020], we first present two vertex reduction rules and an iterative procedure to reduce instance size (Section 2). We then establish new upper bounds on the score and number of colors based on the chromatic numbers of the subgraphs induced by the different weight values of an instance (Section 3). Next, we present three CP models for WVCP: a primal model enforcing solution compactness using a dedicated global constraint, a dual model based on a reduction of WVCP to the maximum weighted stable set problem following [Cornaz et al., 2017], and a joint model (Section 4). Lastly, we report experimental results on the CP models and study the impact of pre-computed bounds. The results show that our approach is competitive across a wide variety of benchmark instances (Section 5).

The reader is referred to the supplementary material (https://github.com/Cyril-Grelier/gc_wvcp_cp) for proofs, complete experimental results, and source code.

2 Vertex Reduction Procedure
This section presents an iterative vertex reduction procedure for WVCP using two rules. We shall denote by \( N(v) = \{u \in V \mid \{u, v\} \in E\} \) the neighborhood of vertex \( v \) in \( G = (V, E) \), and \( \Delta = \max_{v \in V} |N(v)| \) the degree of \( v \). A clique \( C \) of \( G \) is a subset of \( V \) whose vertices are pairwise adjacent.

2.1 Reduction Rules
[Wang et al., 2020] proposed a reduction rule (R0) consisting in comparing the weight of a vertex \( u \) with the weights of the vertices of a given clique \( C \) of the graph \( G \), such that \( u \notin C \). We propose an improvement R1 to this rule, which takes into
Algorithm 1 POSICLIQUE

1: Input: WVCP instance \( P = (G, w) \), vertex \( u \in V \) and clique \( C = \{c_1, \ldots, c_{|C|}\} \) s.t. \( w(c_i) \geq w(c_j) \) (1 \( \leq i < j \leq |C|)\).

2: Output: \( d \in \mathbb{N}^* \)
3: \( d \leftarrow |N(u)| + 1 \); \( l_C \leftarrow |C| - 1 \)
4: for \( i \) from 0 to \( l_C \) do
5: if \( c_{|C| - i} \in N(u) \) and \( |C| - i \geq d \) then
6: \( d \leftarrow d - 1 \)
7: return \( d \)

Algorithm 1 POSICLIQUE procedure.

Rule 1. Let \( P = (G, w) \) be a WVCP instance with \( G = (V, E) \), \( C \) a clique of \( G \), \( u \in V \setminus C \), and \( d = \text{POSICLIQUE}(u, C) \). If \( d \leq |C| \) and \( w(u) \leq w(c_d) \), the optimal score of \( P \) is unchanged after removing \( u \) from \( G \).

Proof. See supplementary material.

This first rule (R1) is equivalent to the rule R0 when \( N(u) \cap C = \emptyset \). In this case, \( d = |N(u)| + 1 \) and \( u \) is removed if \( d \leq |C| \) and \( w(u) \leq w(c_d) \). However, our rule is stronger when \( N(u) \cap C \neq \emptyset \) since the value \( d \) returned by POSICLIQUE may be lower than \( |N(u)| + 1 \) which improves the chance of a successful check \( w(u) \leq w(c_d) \) due to the weight-based decreasing order assumed on the vertices of the cliques. Note that the worst-case time complexity of the rule is \( O(|\Delta|^2) \).

Figure 1 sketches two graphs including a clique \( C = \{c_1, c_2, c_3, c_4\} \) and a vertex \( u \) of weight 6 to illustrate the evaluation of the rule. In both cases, \( |N(u)| = 2 \). On the left, \( c_2 \in N(u) \). Thus, given \( u \) and \( C \) as inputs, POSICLIQUE returns \( d = 3 \) and \( u \) cannot be deleted since it may have to take the same color as \( c_3 \) in the worst case and its weight is greater than \( w(c_3) = 4 \). On the right, \( c_3 \in N(u) \). In this case, POSICLIQUE returns \( d = 2 \) and \( u \) can be removed since it can take the blue or red color, and in either case, there is no impact on the WVCP score because its weight of 6 is lower than \( w(c_1) = 8 \) and \( w(c_2) = 7 \). Note that the rule R0 does not allow to remove \( u \) in these two scenarios since \( w(u) \) is strictly greater than \( w(c_{|N(u)| + 1}) = 4 \).

Our second reduction rule (R2) is an adaptation of the second reduction operator originally proposed by [Cheeseman et al., 1991] for the \( k \)-coloring problem. Its worst-case time complexity is \( O(|\Delta|^2) \).

Figure 1: Application of rule R1 to a vertex \( u \) of weight \( w(u) = 6 \) and a clique \( C = \{c_1, c_2, c_3, c_4\} \) in two different cases. Left: \( u \) cannot be removed. Right: \( u \) can be removed.

Rule 2. Given a WVCP instance \( P = (G, w) \) with \( G = (V, E) \) and two vertices \( u, v \in V \) such that \( N(u) \subset N(v) \) and \( w(u) \leq w(v) \) then the optimal score of \( P \) is unchanged after removing vertex \( u \) from \( G \).

Proof. See supplementary material.

2.2 Maximum Weighted Cliques Extraction and Iterative Reduction

R1 considers the deletion of a vertex relatively to a single clique. We propose to widen its scope by applying it to different cliques of \( G = (V, E) \). Ideally, each clique \( C \) should have maximum weight \( \sum_{c \in C} w(c) \). However, finding a clique of maximum weight in a graph is NP-hard so we rely on a fast heuristic to perform this task, namely, the FastWClq algorithm [Cai and Lin, 2016]. FastWClq iteratively builds a clique with greedy moves using any vertex as a starting point. To keep run time acceptable while aiming for diversity across cliques, we compute \( |V| \) cliques by generating a single clique per vertex \( v \in V \) using \( v \) as the starting point for FastWClq. This procedure is \( O(|V|^3) \).

Once the cliques generated, the two reduction rules are applied to each vertex by increasing order of weights. Each time a vertex is deleted, it is stored in a list \( L \) and the graph \( G \) and the set of cliques are updated to take the deletion into account. The process is repeated until no vertex can be removed. This iterative procedure is \( O(|\Delta|^2|V|^3) \). When a solution is found for the instance \( P' = (G', w) \) produced by the reduction procedure, it is possible to obtain a solution of the same score for the original instance \( P = (G, w) \) by coloring each vertex of the list \( L \) with a greedy algorithm in the reverse order of arrival in \( L \).

3 Upper Bounds on Score and Number of Colors

Theorem 1 introduces new upper bounds on the score and number of colors to solve a WVCP to optimality. These bounds are based on the chromatic numbers of the subgraphs induced by each weight value. We denote by \( W' = \{w(v) \mid v \in V\} \) the set of weight values used in \( G, G_w = (V_w, E_w) \) the subgraph of \( G \) induced by weight \( w \) where \( V_w = \{v \in V \mid w(v) = w\} \) and \( E_w = \{u, v \in E \mid u, v \in V_w\} \), and \( \chi_{G_w} \) the chromatic number of \( G_w \).

Theorem 1. Given a WVCP instance \( P = (G, w) \) with \( G = (V, E) \) and an optimal solution \( S^* = \{V_1, \ldots, V_k\} \) of \( P \) corresponding to a partition of \( V \) into non-empty independent sets, then \( k \leq \sum_{w \in W} \chi_{G_w} \) and \( f(S^*) \leq \sum_{w \in W} w \times \chi_{G_w} \).

Proof. See supplementary material.

Consider the WVCP instance shown in Figure 2. The upper bounds derived from Theorem 1 on the number of colors and the score are respectively equal to \( 3 = \chi_3 + \chi_4 = 1 + 2 \) and 5. Both bounds are exact in this case and the solution shown is optimal. Computing the upper bounds involves solving \( |W| \) GCP sub-problems to obtain the chromatic numbers \( \chi_{G_w} \) of the \( w \in W \). Since GCP is NP-hard, the chromatic numbers may be upper-bounded using heuristics for GCP such as TabuCol [Hertz and de Werra, 1987] (see Section 5.2).
4 Constraint Programming Models

This section introduces three alternative CP models for WVCP called primal, dual, and joint. We shall use the following notations. Given a set $S$, $|S|$ denotes the range $\{1, \ldots, |S|\}$. For a function $f : X \rightarrow Y$ and $X' \subseteq X$, $f(X')$ denotes the image of $X'$ by $f$ and $f^{-1} : Y \rightarrow 2^X$ the function defined by $f^{-1}(y) = \{x \in X | f(x) = y\}$. For a vertex $v$ of a graph $G$, $N(v), \Delta(v)$ and $\Delta(G)$ denote respectively the set of neighbours of $v$, its degree and the maximum vertex degree in $G$.

Let $\kappa \in \mathbb{N}^+$ and $P$ be a WVCP instance of graph $G = (V, E)$ and weight function $w$, $P_{\kappa}$ denotes the problem of determining the existence of a solution to $P$ that uses a number of colors smaller than or equal to $\kappa$. Given a pair $(P, \kappa)$, each model searches for an optimal solution to $P_{\kappa}$, i.e., a solution $s$ to $P_{\kappa}$ whose score $f(s)$ is the lowest among all solutions to $P_{\kappa}$. The primal model supports variability in the number of colors. Note that an optimal solution for $P_{\kappa}$ is based on a reduction of compute optimal and compact $d$-sorted solutions. The dual model is based on a reduction of optimal and compact $d$-sorted solutions. The dual representations of a $d$-solution.

In order to break symmetries induced by color permutation, $d$-sorted models require a total ordering consistent with the descending order of weights ($v_1, v_2, \ldots, v_n$). Formally, $v_i \leq v_j$ if $i < j$ for $i, j \in \{1, \ldots, n\}$. Solutions to $P_{\kappa}$ are thus modeled as maps $s : V \rightarrow K$ where $K = \{1, \ldots, \kappa\}$ is the range of colors. In order to break symmetries induced by color permutation, the computation is restricted to $d$-sorted solutions also called $d$-solutions. A solution is $d$-sorted if non-empty colors start from rank 1 and are sorted consistently with the ordering $\geq w$ of their dominant vertices. Formally, $s : V \rightarrow K$ is $d$-sorted if $s([V]) = [s([V])]$ and $\min(s^{-1}(j)) < \min(s^{-1}(k))$ for $1 \leq j < k \leq |s([V])|$. Clearly, the set of solutions to $P_{\kappa}$ is in one-to-one correspondence with the set of $d$-solutions which we shall denote by $S_{P_{\kappa}}$.

The primal model of $P_{\kappa}$ represents vertex coloring decisions as variables and uses symmetry breaking constraints to compute optimal and compact $d$-sorted solutions. The dual model is based on a reduction of WVCP to the maximum weighted stable set problem which turns the complement of $G$ into a directed graph using the chosen ordering $\geq w$ [Cornaz et al., 2017]. Dual variables represent decisions to keep or remove arcs in this digraph so as to construct pairwise vertex-disjoint simplicial stars. The simplicial stars of a dual solution map one-to-one with the colors of size $\geq 2$ in the primal solution, the center of a star being the dominant vertex in the corresponding color. A dual solution is scored by summing the weights of the target nodes in the simplicial stars. The sum of the dual and primal scores of a solution is therefore equal to the sum of the weights over all vertices. The joint model is essentially a combination of the primal and dual models using channeling constraints. Figure 2 illustrates the primal and dual representations of a $d$-solution.

4.1 Primal Model

The primal model supports variability in the number of colors used across solutions and includes symmetry breaking constraints. To allow for the possibility that some colors of $K$ may have no vertices in a solution, $G$ is extended with virtual vertices. Specifically, a disconnected vertex $u_k$ of weight 0 is introduced for each color $k \in K$ and systematically assigned to $k$ in any solution. The weight function $w$ is extended accordingly to cover the whole set of vertices $U = V \cup \{u_k | k \in K\}$ and so is dominance ordering $\geq w$. The latter is encoded by a consistent indexing of vertices over $U$ ($v_i \geq w v_j \leftrightarrow i \leq j$ for $i, j \in \{1, \ldots, N\}$). Solutions to $P_{\kappa}$ are thus modeled as maps $s : V \rightarrow K$ where $K = \{1, \ldots, \kappa\}$ is the range of colors. In order to break symmetries induced by color permutation, the computation is restricted to $d$-sorted solutions also called $d$-solutions. A solution is $d$-sorted if non-empty colors start from rank 1 and are sorted consistently with the ordering $\geq w$ of their dominant vertices. Formally, $s : V \rightarrow K$ is $d$-sorted if $s([V]) = [s([V])]$ and $\min(s^{-1}(j)) < \min(s^{-1}(k))$ for $1 \leq j < k \leq |s([V])|$. Clearly, the set of solutions to $P_{\kappa}$ is in one-to-one correspondence with the set of $d$-solutions which we shall denote by $S_{P_{\kappa}}$.

The primal model of $P_{\kappa}$ represents vertex coloring decisions as variables and uses symmetry breaking constraints to compute optimal and compact $d$-sorted solutions. The dual model is based on a reduction of WVCP to the maximum weighted stable set problem which turns the complement of $G$ into a directed graph using the chosen ordering $\geq w$ [Cornaz et al., 2017]. Dual variables represent decisions to keep or remove arcs in this digraph so as to construct pairwise vertex-disjoint simplicial stars. The simplicial stars of a dual solution map one-to-one with the colors of size $\geq 2$ in the primal solution, the center of a star being the dominant vertex in the corresponding color. A dual solution is scored by summing the weights of the target nodes in the simplicial stars. The sum of the dual and primal scores of a solution is therefore equal to the sum of the weights over all vertices. The joint model is essentially a combination of the primal and dual models using channeling constraints. Figure 2 illustrates the primal and dual representations of a $d$-solution.

\[ \text{minimize } \mathbf{x}^o \text{ s.t.} \]
\[ \mathbf{x}^o \in \{ \max_{i \in V} (w(v_i)), \ldots, \sum_{i \in V} w(v_i) \} \]  
\[ \forall v_i \in U : x^o_{i} \in K \]  
\[ \forall k \in K : x^K_{i} \in 2^U \]  
\[ \forall k \in K : x^K_{i} \in U \]  
\[ \text{INT\_SET\_CHANNEL}(\{x^K_k | k \in K\}, \{x^U_i | v_i \in U\}) \]  
\[ \forall k \in K : x^K_{i|v_{i+k}=k} = k \]  
\[ \forall \{v_i, v_j\} \in E : x^U_i \neq x^U_j \]  
\[ \forall k \in K : x^D_k = \min_k(x^K_k) \]  
\[ \sum_{k \in K} w(x^D_k) \]  
\[ \text{STRICLY\_INCREEASING}(x^D) \]
Proof. See supplementary material.

An immediate corollary of Theorem 2 is that every satisfiable instance $P_e$ has an optimal d-solution which is compact. Therefore, every WVC $P$ has an optimal solution which is a compact and optimal d-solution for $P_e$ with $\kappa = |V|$. By definition of $\mu_{P_e}$, no vertex $v$ is colored beyond the first $\Delta(v) + 1$ colors in this solution. It follows that the maximum degree of $G$ plus 1 is a safe upper-bound on the number of colors needed to optimally solve $P$ as already proved in [Demange et al., 2007]. Hence $P_e$ with $\kappa = \Delta(G) + 1$ has an optimal and compact d-solution which is also optimal for $P$.

Corollary 1. Let $P$ be a WVC instance. $P_{\Delta(G)+1}$ has an optimal and compact d-solution which is optimal for $P$.

Proof. See supplementary material.

Thanks to Corollary 1, the primal model may be safely instantiated with $\Delta(G) + 1$ colors and, by definition of $\mu_{P_e}$, be tightened with constraints upper-bounding the domain of each vertex variable $x_i^D$ with $\Delta(v_i) + 1$. We now introduce the global constraint MAX_LEFT_SHIFT to enforce solution compactness. MAX_LEFT_SHIFT applies to a vector of $n$ positive integer variables and determines the lowest value in the range $\{1, \ldots, n + 1\}$ that is not assigned to any of these variables.

Definition 2. Let $y$ be an integer domain variable and $[x_1, \ldots, x_n]$ be a vector of positive integer domain variables ($n \geq 0$). $\text{MAX LEFT SHIFT}(y, [x_1, \ldots, x_n])$ holds iff $y = \text{min}_{k=1 \ldots n+1}\{k | \forall i \in 1 \ldots n : x_i \neq k\}$.

Applying MAX_LEFT_SHIFT in the primal model to the variable of a vertex $v_i$ and those of its neighbors as formulated in constraint (P11) clearly amounts to enforcing compactness equation $\mu_{P_e}(s, v_i) = s(v_i)$ on any d-solution $s$. (P11) thus ensures only compact d-solutions may be generated with the primal model. Note that (P11) makes all coloring constraints (P7) redundant by definition of MAX_LEFT_SHIFT.

$$\forall v_i \in V : \text{MAX LEFT SHIFT}(x_i^D, [x_j^D | v_j \in N(v_i)])$$ (P11)

We now provide a decomposition of MAX_LEFT_SHIFT usings constraints (M1-M4). The decomposition relies on global constraint NVALUE [Pachet and Roy, 1999; Bessiere et al., 2006] which counts the number of different values assigned to a vector of variables: $\text{NVALUE}(g, [z_1, \ldots, z_n])$ holds iff $g = |\{z_i | k = 1 \ldots n\})$. Constraint (M1) enforces that $y$ be different from each $x_i$ (equivalent to (P7)). (M2) associates to each $x_i$ a variable $z_i$ ranging over $\{0, \ldots, n + 1\}$. (M3) either sets $z_i$ to $x_i$ if the latter is strictly smaller than $y$ or to 0 otherwise. This is achieved by reifying constraint $y > x_i$ with an implicit 0/1 variable. (M4) sets $y$ to the number of different values taken in the vector including all variables $z_i$ and a ground variable of value 0. Any variable $x_i$ strictly greater than $y$ does not contribute to the count (i.e. the value of $y$ since its variable $z_i$ is absorbed by value 0 in the vector. Hence $y$ is the number of different values taken by the variables it is strictly greater than plus 1 (value 0). Since all variables are positive, this number is necessarily the lowest possible value for $y$ ($n + 1$ in the worst-case).

$$\text{MAX LEFT SHIFT}(y, [x_1, \ldots, x_n]) \equiv$$

$$\forall i \in \{1, \ldots, n\} : y \neq x_i \quad (M1)$$

$$\forall i \in \{1, \ldots, n\} : z_i \in \{0, \ldots, n + 1\} \quad (M2)$$

$$\forall i \in \{1, \ldots, n\} : z_i = (y > x_i) \quad (M3)$$

$$\text{NVALUE}(y, [0, z_1, \ldots, z_n]) \quad (M4)$$

(P5) links vertex and color variables through a channeling constraint involving auxiliary boolean variables that reify domain membership ($x_i^D = k \leftrightarrow i \in x_i^K$ for all $i \in U, k \in K$). (P6) enforces a one-to-one mapping between virtual vertices and colors thereby ensuring the existence of a dominant vertex for each color. (P7) models the coloring constraints induced by $G$. (P8) defines the dominant vertex (possibly virtual) of each color. (P9) models the scoring function as the sum of the weights of the dominant vertices using element constraints. Note that empty colors get cancelled out in this sum. Lastly, (P10) enforces the dominance ordering on colors ($x_i^{D-1} < x_i^D$ for $2 \leq k \leq |K|$). No lexicographic ordering constraint [Frisch et al., 2002] is needed here due to the indexing of vertices using $u_i$.

We now formalize a compactness property for d-solutions and show that every WVC instance $(G, w)$ has optimal d-solutions which are compact. Such solutions are computable with a maximum of $\Delta(G) + 1$ colors. We then propose a global constraint that guarantees solution compactness when it is applied to the neighborhood of each vertex. Informally, a d-solution is compact if the color of any vertex is the lowest in $K$ that is left free by its neighbors. In other words, no vertex may be “left-shifted” to a lower-ranked color without violating coloring constraints. For instance, the d-solution in Figure 2 is compact since neither $v_3, v_4$ nor $v_5$ may be left-shifted. If $(v_3, v_4)$ and $(v_5, v_6)$ were not part of the graph, then $v_6$ and $v_4$ could be left-shifted to colors $K_1$ and $K_2$ respectively to compact the solution. Solution compactness is defined using a function which computes the lowest possible color for a vertex in a d-solution.

Definition 1. Let $P_x$ be a satisfiable WVC instance and $\mu_{P_x} : S_{P_x} \times \n \rightarrow K$ such that, for all $s \in S_{P_x}$, $v \in V$, $\mu_{P_x}(s, v) = \min_{k=1 \ldots \Delta(v)+1}\{k | \forall u \in N(v) : s(u) \neq k\}$.

$s \in S_{P_x}$ is compact if $\mu_{P_x}(s, v) = s(v)$ for all $v \in V$.

$\mu_{P_x}$ clearly exists and is uniquely defined since the neighbors of a vertex $v$ may not use more than $\Delta(v)$ colors in the range $\{1, \ldots, \Delta(v) + 1\}$. Besides, left-shifting a vertex $v$ using $\mu_{P_x}$ may only decrease the score of a d-solution $s$ or leave it unchanged. However, the resulting solution may not be de-sorted if $v$ was the dominant vertex of an intermediate color $k < s(V)$ and the latter gets empty or dominated by the next color after the shift. The resulting solution may also include a greater number of vertices that can be left-shifted compared to the original solution. Theorem 2 shows there actually exists an idempotent function that turns any d-solution into a compact d-solution with no score increase. The proof is based on a recursive algorithm which, given a d-solution, combines vertex left-shifting ($\mu_{P_x}$) with color left-shifting and swap operations to converge towards a compact d-solution.

Theorem 2. Let $P_x$ be a satisfiable WVC instance. There exists $g_{P_x} : S_{P_x} \rightarrow S_{P_x}$ such that, for all $s \in S_{P_x}$, $g_{P_x}(s)$ is compact. $f(g_{P_x}(s)) \leq f(s)$ and $g_{P_x}(g_{P_x}(s)) = g_{P_x}(s)$.

Proof. See supplementary material.

1930
4.2 Dual Model

The dual model for problem $P_2$ is a CP adaptation of the MIP model of [Cornaz et al., 2017]. The latter applies to a dual graph based on a reduction of WVCP to the maximum weighted stable set problem [Cornaz and Jost, 2008]. Given a WVCP $P = (G, w)$ and a dominance ordering $\succeq_w$, the dual graph is built by turning each edge $\{v_i, v_j\}$ of the complement of $G$ such that $v_i \succeq_w v_j$ into the arc $(v_i, v_j)$ of source node $v_i$ and target node $v_j$. We denote by $\mathcal{E}^s = \{ij \mid v_i, v_j \in V \land \{v_i, v_j\} \notin \mathcal{E} \land v_i \succeq_w v_j\}$ the set of arcs of the dual graph, and $T$ its set of target nodes (nodes with incoming arcs).

The reduction relies on the notion of a simplicial stellar forest. A pair of arcs $(ij, ik)$ is simplicial if an arc exists between $j$ and $k$ in the dual graph. A star (i.e., a set of arcs with the same source node) is simplicial if each pair of arcs in the star is simplicial in the dual graph. A simplicial stellar forest is a set of simplicial stars that span disjoint subsets of nodes. [Cornaz et al., 2017] show that the legal colorings of a WVCP instance map one-to-one with the simplicial stellar forests of its dual graph. Note that singleton colors in a WVCP solution map to the disconnected nodes in the corresponding primal forest. A simplicial stellar forest is scored by summing the weights of the target nodes of its arcs. This score added to the WVCP score of the corresponding primal solution is therefore equal to the sum of the weights of all the vertices (see Figure 2).

\[
\text{maximize } y^o \text{ s.t.} \\
\forall ij \in \mathcal{E}^s : y_{ij}^A \in \{0, 1\} \quad (D1) \\
y^o = \sum_{v_i \in V} (w(v_i)) \quad (D2) \\
y^o = \sum_{ij \in \mathcal{E}^s} w(v_j) \times y_{ij}^A \quad (D3) \\
\forall ij, ik \in \mathcal{E}^s : \{j, k\} \cap \mathcal{E}^s = \emptyset : y_{ij}^A + y_{ik}^A \leq 1 \quad (D4) \\
\forall ij, ik \in \mathcal{E}^s : y_{ij}^A + y_{ik}^A \leq 1 \quad (D5) \\
\forall h, i, j \in \mathcal{E}^s : y_{hj}^A + y_{ij} \leq 1 \quad (D6) \\
\forall v_i \in V : z^o_i = 1 - \max_{(h, i) \in \mathcal{E}^s} (y_{hi}^A) \quad (D7) \\
\forall v_i \in V \setminus T : z^o_i = 1 \quad (D8) \\
\sum_{v_i \in V} z^o_i \leq \kappa \quad (D9) \\
\minimize x^o \text{ s.t.} \\
\forall ij \in \mathcal{E}^s : y_{ij}^A \leq (x^o_j = x^o_j) \quad (J1) \\
\text{GCC}[x_k^D \mid k \in K], V, [z^V_i \mid v_i \in V] \quad (J2) \\
x^o + y^o = \sum_{v_i \in V} w(v_i) \quad (J3) \\
\forall v_i \in V, v_j \in N(v_i) \text{ s.t. } v_j \succeq_w v_i : \left( \bigwedge_{v_h \in N(v_j) \cap N(v_j)} x^o_h \neq x^o_j \right) \Rightarrow x^o_j \leq x^o_j \quad (J4)
\]

5 Experiments

We first introduce the benchmark instances used for experiments and analyse the impact of our reduction procedures on instance size. We then discuss the impact of using pre-computed bounds on problem-solving efficiency. Lastly, we compare the results of our CP models with the state-of-the-art.
Benchmark instances. A total of 188 WVCP instances were considered: 30 rxx graphs and 35 pxx graphs from matrix decomposition problems [Prais and Ribeiro, 2000] and 123 graphs coming from the DIMACS and COLOR competitions [Sun et al., 2018]. Due to lack of space, we report detailed results on 12 instances of different sizes, graph densities, and weight and degree distributions. Complete benchmark instances, source code, and results are available in the supplementary material.

Experimental settings. Experiments were performed on an Intel Xeon ES 2630, 2.66 GHz, Broadwell. CP models were programmed using Minizinc [Nethercote et al., 2007] and solved using OR-Tools [Perron and Furnon, 2022]. We used heuristics first-fail combined with domain bisection to solve primal and joint instances and a static heuristics sorting arcs by descending order of weights to solve dual instances. A time limit of 1 hour using a single CPU was set for each run.

5.1 Reduction of Instance Sizes
Table 1 reports the impact of the different reduction rules on the whole set of 188 instances. As for R0 and R1, we pre-computed a set of cliques C as discussed in Section 2.1 then applied each rule individually to each pair (v, c) with v ∈ V and c ∈ C (lines R0 and R1). As for R2, we applied the rule to each pair of non-adjacent vertices. (R1 + R2) corresponds to the joint application of the two reduction rules. Lastly Iterative corresponds to the iterative reduction procedure (see Section 2.2) which applies R1 and R2 until no vertex can be removed.

We see that R1 works slightly better than R0 in terms of number of reduced instances (# R1) and average percentage of reduction (% RV) for these reduced instances. Overall, we observe a great improvement due to the iterative procedure. The computational time required to perform the different reductions remains reasonable in average in comparison with rule R0 (see column 7).

We will only consider the reduced instances in the experiments described in the rest of the paper. Note that using the reduced instances allows to greatly improve the results compared to the original ones (see supplementary material).

5.2 Bounds on Number of Colors and Score
Table 2 shows lower and upper bounds on the score and the number of colors required for an optimal solution given by Theorem 1 or coming from previous studies.

Columns 1-4 show the different characteristics of the instances. Column 4 introduces a measure of weight heterogeneity defined by $h_W = \frac{1}{\|W\|}$. Column 5 reports the value $\Delta + 1$ ($\Delta$ denotes the maximum vertex degree in the graph) which is a baseline upper bound on the number of colors required to obtain an optimal solution [Demange et al., 2007]. Column 6 reports a lower bound on the number of colors which corresponds to the maximum size of a clique in G found with the clique extraction procedure applied during the first reduction pre-processing step (see Section 2.2). Column 7 corresponds to the upper bound on the number of colors given by Theorem 1. This bound was obtained by solving $\|W\|\mbox{ CCP sub-problems with decreasing number of colors using the C++ implementation of the TabuCol algorithm [Hertz and de Werra, 1987] proposed in [Moalic and Gondran, 2018] with a limit of 1000 iterations without improvement for each graph k-coloring sub-problem. It does not take more than 0.1 second per instance to compute the global upper bound with this method. This upper bound is written in bold when it is better than the bound $\Delta + 1$, which happens almost all the time, except for specific instances such as r30 characterized with a high weight heterogeneity $h_W$. Column 8 is a lower bound on the score computed with the method proposed by [Wang et al., 2020] (see Proposition 1) and using the set of maximum weighted cliques computed with the FastWCliq algorithm. Column 9 reports the upper bound on the score computed according to Theorem 1.

5.3 Impact of Color and Score Bounds
Table 3 shows the impact on the primal model of the upper bound on the number of colors (see Theorem 1) as well as the impact of introducing simultaneously all the bounds presented in the last section. The impact of the other bounds each taken separately is presented in the supplementary material.

Column 1 is the name of the instance. Column 2 reports the best-known score (BKS) obtained in the literature. Some of these BKS were obtained under specific and relaxed conditions, such as one day of computation in parallel on Graphic Processing Device (GPU) in [Goudet et al., 2022], and are therefore very difficult to reach. When a star is added to this score, it means that it has been proven optimal. Most of these proofs of optimality were obtained with the MIP formulation of [Cornaz et al., 2017] solved during 10h using CPLEX [Nogueira et al., 2021].

A score is written in bold in columns 3-8 when it corresponds to the BKS. When the instance is solved to optimality, a star is added and the time in seconds required to prove optimality is reported. Otherwise “i” is indicated meaning that the time limit of 1 hour has been reached. The score is

| Instance | $|V|/|W|$ | density $h_W$ | $\Delta + 1$ | colors bounds | score bounds |
|----------|----------|-------------|-------------|---------------|--------------|
| DSJC125.1g | 125 | 0.1 | 0.04 | 24 | 4 | 19 | 42 |
| DSJC125.5g | 125 | 0.5 | 0.04 | 76 | 10 | 42 | 105 |
| DSJC125.9g | 125 | 0.9 | 0.04 | 121 | 32 | 72 | 220 |
| DSJC500.1 | 125 | 1.0 | 0.04 | 26 | 26 | 166 | 477 |
| DSJC125.1g | 125 | 0.1 | 0.04 | 24 | 4 | 19 | 42 |
| DSJC125.5g | 125 | 0.5 | 0.04 | 76 | 10 | 42 | 105 |
| DSJC125.9g | 125 | 0.9 | 0.04 | 121 | 32 | 72 | 220 |
| DSJC500.1 | 125 | 1.0 | 0.04 | 26 | 26 | 166 | 477 |
| DSJC125.1g | 125 | 0.1 | 0.04 | 24 | 4 | 19 | 42 |
| DSJC125.5g | 125 | 0.5 | 0.04 | 76 | 10 | 42 | 105 |
| DSJC125.9g | 125 | 0.9 | 0.04 | 121 | 32 | 72 | 220 |
| DSJC500.1 | 125 | 1.0 | 0.04 | 26 | 26 | 166 | 477 |

Table 1: Impact of the reduction procedures. # R1: number of reduced instances (out of 188 instances), # RV: number of reduced vertices, %RV: percentage of reduction and time t(s) in seconds.

Table 2: Lower and upper bounds on the score and colors.
underlined if this optimality has never been proved before in the literature.

We see that the upper bound on the number of colors (Columns 5-6) deriving from Theorem 1 can significantly reduce the time spent by the solver on each instance, because it reduces the domain of available colors for each vertex.

Columns 7-8 correspond to the results when all the lower and upper bounds presented in Table 2 are simultaneously activated, which allows to increase the number of BKS reached (107 out of 188), as well as the number of optimality proofs (95 out of 188) for the whole set of instances.

5.4 Comparison of CP Models

Table 4 compares the results obtained by the three CP models described in Section 4 (primal, dual and joint) on the same set of reduced instances, without using pre-computed bounds. Columns 3-4 report the results of the primal model (P1-P10) and Columns 5-6 correspond to the results of the primal model extended with compactness constraints (P11) and decomposed using constraints (M1-M4). When enforcing compactness, the number of instances solved to optimality goes from 72 to 76, indicating that dynamically reducing the color domain of each vertex during the search can be beneficial. Columns 7-8 report the results of the dual model presented in Section 4.2 (D1-D10). We observe that the primal and dual models obtain different results depending on the density of the instance. Unsurprisingly, the primal model is better for instances characterized by a low graph density (in particular it can reach a new optimality proof for instance DSJC125.1.g), while the dual is better for instances with high-density graphs such as DSJC125.9.g. In Columns 9-10, we observe that the joint model coupling the primal and dual models (P1-P10, D1-D10 and J1-J4), performs well on both low and high density instances, as it can take advantage of both models. It solves 100 instances over 188 to optimality, which is the highest number. Moreover, results on some instances such as inithx.i.1, musolie.i.5 and p42, which go beyond the results obtained by the primal and dual models alone, show that it can benefit from a synergy of the two representations.

5.5 New Optimality Proofs

We launched the primal, dual, and joint models with compactness constraints and using pre-computed bounds on a server with 10 threads and a time limit of 1 hour. Table 5 reports all the new optimality proofs obtained for difficult instances during all our experiments (see supplementary material for complete tables).

6 Conclusion

We proposed an iterative reduction procedure and established new upper bounds on the score and the number of colors needed to optimally solve WVCP. We highlighted their practical value in reducing the search space through experiments carried out on benchmark instances. Three CP models were also investigated together with global constraints to break symmetries. We provided empirical evidence to shed light on their advantages and limits. The results showed that the models are competitive for most of the small- and medium-size instances, leading in particular to solving some instances to optimality. In our future work, we would like to investigate possible hybridizations of the CP models with metaheuristics based in particular on the proposed compactness algorithm.

Acknowledgements

We are grateful to the reviewers for their feedback. We thank B. Nogueira for sharing his CPLEX code used in [Nogueira et al., 2021] and A. Goeffon for fruitful exchanges on the reduction rules. We acknowledge support from the Centre Régional de Calcul Intensif des Pays de la Loire (CCIPL) for hosting experiments.

References


