Mathematical Programming (2020) 183:61–103
https://doi.org/10.1007/s10107-020-01491-4

FULL LENGTH PAPER

Series B

Symmetry-breaking inequalities for ILP with structured sub-symmetry

Pascale Bendotti[1,2] · Pierre Fouilhoux[1] · Cécile Rottner[1,2]

Received: 31 May 2019 / Accepted: 6 March 2020 / Published online: 30 March 2020

Abstract

We consider integer linear programs whose solutions are binary matrices and whose (sub-)symmetry groups are symmetric groups acting on (sub-)columns. Such structured sub-symmetry groups arise in important classes of combinatorial problems, e.g. graph coloring or unit commitment. For a priori known (sub-)symmetries, we propose a framework to build (sub-)symmetry-breaking inequalities for such problems, by introducing one additional variable per considered sub-symmetry group. The derived inequalities are full-symmetry-breaking and in polynomial number w.r.t. the number of sub-symmetry groups considered. The proposed framework is applied to derive such inequalities when the symmetry group is the symmetric group acting on the columns. It is also applied to derive sub-symmetry-breaking inequalities for the graph coloring problem. Experimental results give insight into how to select the right inequality subset in order to efficiently break sub-symmetries. Finally, the framework is applied to derive (sub-)symmetry breaking inequalities for Min-up/min-down Unit Commitment Problem with or without ramp constraints. We show the effectiveness of the approach by presenting an experimental comparison with state-of-the-art symmetry-breaking formulations.

Keywords Symmetry-breaking inequalities · Sub-symmetries · Graph Coloring · Unit Commitment

Mathematics Subject Classification 90C10 · 90C57 · 90C90

✉ Pierre Fouilhoux
pierre.fouilhoux@lip6.fr

Pascale Bendotti
pascale.bendotti@edf.fr

Cécile Rottner
cecile.rottner@edf.fr

1 Sorbonne Université, Laboratoire d'Informatique de Paris 6, LIP6, 4 Place Jussieu, 75005 Paris, France

2 EDF R&D, 7 Boulevard Gaspard Monge, 91120 Palaiseau, France

1 Introduction

Symmetries arising in integer linear programs can impair the solution process, in particular when symmetric solutions lead to an excessively large Branch and Bound (B&B) search tree. Various techniques, so called *symmetry-breaking techniques*, are available to handle symmetries in integer linear programs of the form

$$(ILP) \qquad \min\{cx \mid x \in \mathcal{X}\}, \text{ with } \mathcal{X} \subseteq \mathcal{P}(m,n) \text{ and } c \in \mathbb{R}^{m\times n}$$

where $\mathcal{P}(m,n)$ is the set of $m \times n$ binary matrices. A symmetry is defined as a permutation π of the indices $\{(i,j) \mid 1 \leq i \leq m,\ 1 \leq j \leq n\}$ such that for any solution matrix $x \in \mathcal{X}$, matrix $\pi(x)$ is also solution with the same cost, i.e., $\pi(x) \in \mathcal{X}$ and $c(x) = c(\pi(x))$. The *symmetry group* $\mathcal{G}$ of (ILP) is the set of all such permutations. It partitions the solution set $\mathcal{X}$ into *orbits*, i.e., two matrices are in the same orbit if there exists a permutation in $\mathcal{G}$ sending one to the other. A *subproblem* is problem (ILP) restricted to a subset of $\mathcal{X}$. In [4], symmetries arising in solution subsets of (ILP) are called *sub-symmetries*. Such sub-symmetries may not exist in $\mathcal{G}$.

In this article, we focus on structured symmetries arising from (sub-)symmetry groups containing all sub-column permutations of a given solution submatrix. These symmetry groups are assumed to be known or previously detected [6,23].

A first idea to break symmetries is to reformulate the problem using integer variables summing the variables along orbits. Such a reformulation aggregates variables, thus reducing the size of the resulting ILP [22]. However, it can be used only when aggregated solutions can be disaggregated. This is for example the case when the integer decomposition property [1] holds. A more general idea to break symmetries is, in each orbit to pick one solution, defined as the *representative*, and then restrict the solution set to the set of all representatives. The most common choice of representative is based on the lexicographical order. Column $y \in \{0,1\}^m$ is said to be *lexicographically greater than* column $z \in \{0,1\}^m$ if there exists $i \in \{1,\dots,m-1\}$ such that $\forall i' \leq i$, $y_{i'} = z_{i'}$ and $y_{i+1} > z_{i+1}$, i.e., $y_{i+1} = 1$ and $z_{i+1} = 0$. We write $y \succ z$ (resp. $y \succeq z$) if y is lexicographically greater than z (resp. greater than or equal to z). A technique is said to be *full symmetry-breaking* (resp. *partial symmetry-breaking*) if the solution set is exactly (resp. partially) restricted to the representative set. A symmetry-breaking technique is said to be *flexible* if, at any node of the B&B tree, the branching rule can be derived from any linear inequality on the variables.

Most techniques based on branching and pruning rules [12,28,33] are either full symmetry-breaking or flexible. Variable fixing [4,19] is both full symmetry-breaking and flexible. Other symmetry-breaking techniques rely on full or partial symmetry-breaking inequalities. Such techniques are flexible. Note that the size of the LP solved at each node of the branching tree is generally invariant under pruning and variable fixing techniques, whereas it is increased by the use of symmetry-breaking inequalities.

Symmetry-breaking inequalities can be derived from the linear description of the convex hull of an arbitrary representative set [16]. In most works, each chosen representative x is lexicographically maximal in its orbit, i.e., $x \succeq g(x)$, for each $g \in \mathcal{G}$. The convex hull of the latter representative set is called the *symmetry-breaking polytope*

[16]. When x is a matrix and when the symmetry group $\mathcal{G}$ acts on the columns of x, the symmetry-breaking polytope is called *orbitope*. Even if complete linear descriptions for orbitopes may be hard to reach in general [26], integer programming formulations for these polytopes still yield full symmetry-breaking inequalities [16]. Instead of considering orbits of solutions, [23,24] introduce inequalities enforcing a lexicographical order within orbits of variables.

When symmetry group $\mathcal{G}$ is the *symmetric group* $\mathfrak{S}_n$ acting on the columns, i.e., the set containing all column permutations, then the chosen representative x of an orbit may be such that its columns $x(1)$, …, $x(n)$ are lexicographically non-increasing, i.e., for all $j < n, x(j) \succeq x(j+1)$. The convex hull of all $m \times n$ binary matrices with lexicographically non-increasing columns is called the *full orbitope* [20]. Sub-symmetries and sub-orbitopes are introduced in [4] as a generalization of symmetries and full orbitopes to a given set of matrix subsets. The aim in this paper is to develop a general framework, that enables deriving sub-symmetry-breaking inequalities designed to handle simultaneously symmetries and sub-symmetries in symmetric groups.

For the particular case of packing (resp. partition) problems, i.e., problems whose solution matrix features at most (resp. exactly) one 1-entry in each row, a class of full symmetry-breaking inequalities is introduced in [20]. These inequalities lead to a complete linear description of two special cases of orbitopes: the packing (resp. partitioning) orbitope, i.e., the convex hull of all $m \times n$ binary matrices with lexicographically non-increasing columns and with at most (resp. exactly) one 1-entry per row.

For the full orbitope, a complete linear description in the x variable space seems hard to reach [26]. For the full orbitope restricted to 2-column matrices, a complete linear description in the x space is available [26]. An $O(mn^3)$ extended formulation is given in [18]. To the best of our knowledge, it has never been used in practice to handle symmetries.

Another class of symmetry-breaking inequalities aims to ensure that the integer solutions lie in the full orbitope. For instance, the following full symmetry-breaking inequalities are introduced by Friedman [13]:

$$\sum_{i=1}^{m} 2^{m-i} x_{i,j} \geq \sum_{i=1}^{m} 2^{m-i} x_{i,j+1}, \quad \forall j \in \{1, \dots, n-1\} \tag{1.1}$$

As the 2^{m-i} term might cause numerical intractability, alternative inequalities featuring ternary coefficients can be used at the expense of losing the full symmetry-breaking property, e.g. *column inequalities* [20,30,31]:

$$\sum_{k=1}^{i} x_{k,j} \geq x_{i,j+1}, \quad \forall j \in \{1, \dots, n-1\}, \quad \forall i \in \{1, \dots, m\} \tag{1.2}$$

Another option is to use a partial symmetry-breaking form of Friedman inequalities, as in [17,25]:

$$\sum_{i=1}^{m} x_{i,j} \geq \sum_{i=1}^{m} x_{i,j+1}, \quad \forall j \in \{1, \ldots, n-1\} \tag{1.3}$$

The latter inequalities enforce that the total number of ones in each column is non-increasing, thus not guaranteeing lexicographically non increasing columns for the representatives.

An alternative avoiding the exponential coefficients of Friedman's inequalities can be to use the full symmetry-breaking inequalities discussed in [16]. These inequalities ensure that any integer point is in the full orbitope. They can be separated in linear time and have ternary coefficients like inequalities (1.2) and (1.3).

In this article, sub-symmetries arising from solution subsets whose symmetry groups contain the symmetric group acting on some sub-columns are assumed to be known. We propose a general framework to build full symmetry-breaking inequalities in order to handle these sub-symmetries. One additional variable per subset Q considered may be needed in these inequalities, depending on whether variables x are sufficient to indicate that "x belongs to subset Q".

The proposed framework is applied to derive such inequalities when the symmetry group is the symmetric group $\mathfrak{S}_n$ acting on the columns.

It is also applied to derive full (sub-)symmetry-breaking inequalities for two problems: the Graph Coloring Problem (GCP) and a variant of the Unit Commitment Problem.

The GCP has a particular structure as it is a partition problem. Such structure can be exploited to derive dedicated sub-symmetry-breaking inequalities. We consider the classical IP formulation [10] which is often used as an example featuring many symmetric solutions. Note that the integer decomposition property does not apply to the graph coloring problem as aggregating a solution in this case is meaningless. We demonstrate the efficiency of the proposed framework to break symmetries. A comparison is performed with two state-of-the-art symmetry-breaking families of inequalities: column inequalities (1.2) adapted to partition structures, and inequalities completely describing the partitioning orbitope [20]. Experimental results highlight that a well-chosen subset of the proposed sub-symmetry-breaking inequalities is competitive with these two state-of-the-art techniques.

The considered variant of the Unit Commitment Problem is with constraints on the minimum up and down times of each unit. This variant is called the Min-up/min-down Unit Commitment Problem (MUCP) as defined in [35]. When the MUCP is considered, the integer decomposition property holds for the classical formulation and thus efficient aggregation techniques apply [22]. A variant of the MUCP is with constraints limiting power variations, referred to as *ramp constraints*. When the ramp-constrained MUCP is considered, the integer decomposition property does not hold anymore for the classical formulation, then the corresponding aggregated solutions can no longer be disaggregated. This emphasizes that cases for which such property holds are more the exceptions than the rule. We show that the proposed sub-symmetry-breaking inequalities outperform state-of-the-art symmetry-breaking formulations, such as the aggregated interval MUCP formulation [22] as well as the classical MUCP formulation featuring inequalities (1.3). To extend the experimental comparisons, the proposed framework is also shown to be competitive with two state-of-the-art symmetry-breaking techniques based on branching and fixing: Modified Orbital Branching (MOB) [31] derived from Orbital Branching [33], and orbitopal fixing for the full (sub)-orbitope [4].

Note that an extended abstract of this paper appeared in [5].

In Sect. 2, the framework is described. In Sect. 3, an application to the symmetric group case is presented. The framework is applied to derive sub-symmetry-breaking inequalities dedicated to the GCP in Sect. 4 and to the MUCP in Sect. 5, together with experimental results.

2 Sub-symmetry-breaking inequalities

For a given solution subset Q, the symmetry group $\mathcal{G}_Q$ of the corresponding subproblem is different from $\mathcal{G}$ and may contain symmetries not present in $\mathcal{G}$. In practice it is too expensive to compute the symmetry group for every subset $Q \subset \mathcal{X}$. However for many problems, symmetries of $\mathcal{G}$ can be deduced from the problem's structure, and so can symmetries of $\mathcal{G}_Q$, for some particular solution subsets Q. In this case, symmetries of $\mathcal{G}_Q$ are a priori known, and thus do not need to be computed. Such symmetries may be handled together with symmetries of $\mathcal{G}$. In this section, we introduce sub-symmetry-breaking inequalities designed to simultaneously handle symmetries and sub-symmetries in symmetric groups. First, we briefly recall the concepts of sub-symmetry in ILP introduced in [4].

2.1 Background on sub-symmetries

Consider a subset $Q \subset \mathcal{X}$ of solutions of (ILP). The sub-symmetry group $\mathcal{G}_Q$ relative to subset Q is defined as the symmetry group of subproblem $\min\{cx \mid x \in Q\}$. Permutations in sub-symmetry group $\mathcal{G}_Q$ are referred to as *sub-symmetries*.

Let $\{Q_s \subset \mathcal{X},\ s \in \{1,\ldots,q\}\}$ be a set of solution subsets. To each Q_s, $s \in \{1,\ldots,q\}$, there corresponds a sub-symmetry group $\mathcal{G}_{Q_s}$. Let $O_k^s, k \in \{1,\ldots,o_s\}$, be the orbits defined by $\mathcal{G}_{Q_s}$ on subset Q_s, $s \in \{1,\ldots,q\}$, and $\mathcal{O} = \{O_k^s,\ k \in \{1,\ldots,o_s\},\ s \in \{1,\ldots,q\}\}$. For given $x \in \mathcal{P}(m,n)$, let us define $\mathcal{G}(x) = \bigcup_{Q_s \ni x} \mathcal{G}_{Q_s}$, the set of all permutations π in $\bigcup_{s=1}^{q} \mathcal{G}_{Q_s}$ such that π applied to x defines a symmetric solution to x. Matrix x' is said to be in relation with $x \in \mathcal{P}(m,n)$ if there exist $r \in \mathbb{N}$ and permutations $\pi_1,\ldots,\pi_r$ such that $\pi_k \in \mathcal{G}(\pi_{k-1} \circ \cdots \circ \pi_1(x))$, $\forall k \in \{1,\ldots,r\}$, and $x' = \pi_r \circ \pi_{r-1} \circ \cdots \circ \pi_1(x)$. The *generalized orbit* $\mathbb{O}$ *of* x *with respect to* $\{Q_s, s \in \{1,\ldots,q\}\}$ is thus the set of all x' in relation with x. By definition, for any generalized orbit $\mathbb{O}$, there exist orbits σ_1, …, $\sigma_p \in \mathcal{O}$ such that $\mathbb{O} = \cup_{s=l}^{p} \sigma_l$. To each orbit σ, there corresponds a representative $\rho(\sigma)$. When dealing with sub-symmetries, the representatives should satisfy the following property to make sure they would not be eliminated while breaking symmetries. The set of representatives $\{\rho(\sigma), \sigma \in \mathcal{O}\}$ is said to be *orbit-compatible* if for any generalized orbit $\mathbb{O} = \cup_{l=1}^{p} \sigma_l$, where σ_1, …, $\sigma_p \in \mathcal{O}$, there exists j such that $\rho(\sigma_j) = \rho(\sigma_l)$ for all l

verifying $\rho(\sigma_j) \in \sigma_l$. Such a solution $\rho(\sigma_j)$ is said to be a *generalized representative* of $\mathbb{O}$.

For each orbit O_k^s, $k \in \{1, \ldots, o_s\}$, $s \in \{1, \ldots, q\}$, let its representative $x_k^s \in O_k^s$ be the lexicographically maximal element in O_k^s.

Lemma 1 ([4]) *The set of representatives* $\{x_k^s,\ k \in \{1, \ldots, o_s\},\ s \in \{1, \ldots, q\}\}$ *is orbit-compatible.*

Given $x \in \mathcal{X}$ and sets $R \subset \{1, \ldots, m\}$ and $C \subset \{1, \ldots, n\}$, we consider submatrix (R, C) of x, denoted by $x(R, C)$, obtained by considering columns C of x on rows R only. Symmetry group $\mathcal{G}_Q$ is the *sub-symmetric group* with respect to (R, C) if it is the set of all permutations of the columns of $x(R, C)$. If $\mathcal{G}_Q$ is the sub-symmetric group with respect to (R, C) then subset Q is said to be *sub-symmetric* with respect to (R, C).

Consider a set $\mathbb{S}$ of solution subsets Q_s, $s \in \{1, \ldots, q\}$, such that each subset Q_s, $s \in \{1, \ldots, q\}$, is sub-symmetric with respect to (R_s, C_s). For each orbit O_k^s, $k \in \{1, \ldots, o_s\}$ of $\mathcal{G}_{Q_s}$, $s \in \{1, \ldots, q\}$, its representative $x_k^s \in O_k^s$ is chosen to be such that submatrix $x_k^s(R_s, C_s)$ is lexicographically maximal, i.e., its columns are lexicographically non-increasing. Such x_k^s is said to be the *lex-max* of orbit O_k^s with respect to (R_s, C_s). The following holds as a direct corollary of Lemma 1.

Lemma 2 ([4]) *The set of lex-max representatives* $\{x_k^s,\ k \in \{1, \ldots, o_s\},\ s \in \{1, \ldots, q\}\}$ *is orbit-compatible.*

The *full sub-orbitope* $\mathcal{P}_{sub}(\mathbb{S})$ associated to $\mathbb{S}$ is the convex hull of binary matrices x such that for each $s \in \{1, \ldots, q\}$, if $x \in Q_s$ then the columns of $x(R_s, C_s)$ are lexicographically non-increasing.

2.2 Definition and validity of sub-symmetry-breaking inequalities

Consider a set $\mathbb{S}$ of solution subsets Q_s, $s \in \{1, \ldots, q\}$, such that each subset Q_s, $s \in \{1, \ldots, q\}$, is sub-symmetric with respect to (R_s, C_s). Consider an integer variable z_s, $s \in \{1, \ldots, q\}$, such that $z_s = 0$ if variable $x \in Q_s$, and such that $z_s \geq 1$ if $x \notin Q_s$. For any $x \in \mathcal{X}$, one can define function Z associating x to a vector $Z(x)$ such that z_s, $s \in \{1, \ldots, q\}$, is the s^{th} component of $Z(x)$ denoted by $Z_s(x)$

Note that in many cases, function Z can be chosen to be linear, i.e., each integer variable z_s is a linear expression of variables x. In such cases, no additional variable z_s is needed, as $z_s = Z(x)$. In some cases where function Z is not linear, variable z_s can be linearly expressed from variables x using a few additional inequalities or integer variables.

Given $c,\ c'$, two consecutive columns in C_s such that $c < c'$, the *sub-symmetry-breaking inequality*, denoted by $(I_s(c))$, is defined as follows.

$$x_{r_1,c'} \leq z_s + x_{r_1,c} \qquad \text{where } r_1 = \min(R_s) \tag{2.1}$$

If $\mathcal{Q}$ corresponds to a packing problem, i.e., each $x \in \mathcal{Q}$ features at most one 1-entry in each row, the sub-symmetry-breaking inequality $(I_s(c))$ simplifies to

$$x_{r_1,c'} \leq z_s \qquad \text{where } r_1 = \min(R_s) \tag{2.2}$$

The q sub-symmetric subsets contained in $\mathbb{S}$ correspond to known sub-symmetries to be broken. The total number of inequalities (2.1) or (2.2) is $O(nq)$. Note that this number can be large depending on the choice of $\mathbb{S}$.

For each orbit $O_k^s, k \in \{1, \dots, o_s\}$, of $\mathcal{G}_{Q_s}, s \in \{1, \dots, q\}$, the chosen representative is the lex-max of orbit O_k^s with respect to (R_s, C_s). Then by Lemma 2, this set of representatives is orbit-compatible. In particular, solution set $\mathcal{X}$ can be restricted to the set of representatives by considering its intersection with the full sub-orbitope $\mathcal{P}_{sub}(\mathbb{S})$. If $x \in Q_s$, inequality $(I_s(c))$ enforces that the first row of submatrix $x(R_s, C_s)$ is lexicographically non-increasing, hence the following result.

Lemma 3 (Validity) *If* $x \in \mathcal{P}_{sub}(\mathbb{S})$, *then* $(x, Z(x))$ *satisfies inequality* $(I_s(c))$ *for each* $s \in \{1, \dots, q\}$ *and* $c,\ c' \in C_s$ *such that* $c < c'$.

Note that an inequality similar to (2.1) applied to a row of R_s distinct from r_1 may not be valid when used alongside with (2.1), as shown in Example 1.

Example 1 Let $\mathbb{S} = \{Q_1\}, q = 1$, where

$$Q_1 = \left\{ x \in \mathcal{P}(4,3) \cap \mathcal{X} \mid \sum\nolimits_{c=1}^{3} x_{2,c} = 3 \right\}$$

Let us suppose the symmetry group of Q_1 is the sub-symmetric group with respect to submatrix $(\{3, 4\}, \{1, 2, 3\})$. Variable z_1 can be defined using equality $z_1 = 3 - \sum_{c=1}^{3} x_{2,c}$. Note that $z_1 = Z_1(x) = 0$ when $\sum_{c=1}^{3} x_{2,c} = 3$, i.e., $x \in Q_1$, and is positive otherwise. Here the first row in R_1 is $r_1 = \min(R_1) = 3$, thus given $c,\ c' \in \{1, 2, 3\}, c < c'$, inequality $(I_1(c))$ is

$$x_{3,c'} \leq \left(3 - \sum\nolimits_{j=1}^{3} x_{2,j}\right) + x_{3,c}.$$

This inequality enforces that row 3 of a solution matrix x is lexicographically ordered, i.e., $x_{3,1} \geq x_{3,2} \geq x_{3,3}$, whenever $\sum_{c=1}^{3} x_{2,c} = 3$.

Now consider solutions $x^1, x^2 \in Q_1$:

$$x^1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \qquad \text{and} \qquad x^2 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Inequality $(I_1(c))$ cuts off solution x^2 from the feasible set. Inequality (2.1) applied to row 4 is $x_{4,c'} \leq (3 - \sum_{j=1}^{3} x_{2,j}) + x_{4,c}$ This inequality would cut off x^1. This shows that these two inequalities cannot be used simultaneously.

Note that in the general case, inequalities (2.1) may only be partial-symmetry-breaking. Indeed, for given $s \in \{1, \ldots, q\}$ and $c,\ c' \in C_s$ such that $c < c'$, inequality $(I_s(c))$ only enforces that the first row of submatrix $x(R_s, C_s)$ is lexicographically non-increasing when $x \in Q_s$. In the case when $x_{r_1,c'} < x_{r_1,c}$, then sub-columns $x(R_s, \{c'\}) \prec x(R_s, \{c\})$. Otherwise, when $x_{r_1,c'} = x_{r_1,c}$, inequality (2.1) is not sufficient to select the lexmax representatives.

To enforce a lexicographical order, subsequent rows of submatrix $x(R_s, C_s)$ should be considered until a tie-break row is found. It is shown in the next section that inequalities $(I_s(c))$ for all $s \in \{1, \ldots, q\}$ and $c < c' \in C_s$ enforce that $x \in \mathcal{P}_{sub}(\mathbb{S})$ provided a tie-break condition on set $\mathbb{S}$ is fulfilled.

2.3 Full symmetry-breaking sufficient condition

In this section, we introduce a condition for inequalities (2.1) to be full symmetry-breaking.

For each $s \in \{1, \ldots, q\}$, consider $R_s = \{r_1^s, \ldots, r_{|R_s|}^s\}$ and $C_s = \{c_1^s, \ldots, c_{|C_s|}^s\}$, where $r_1^s < \cdots < r_{|R_s|}^s$ and $c_1^s < \cdots < c_{|C_s|}^s$. For given $s \in \{1, \ldots, q\}$ and any two columns $c_{l-1}^s, c_l^s \in C_s$, if there is a solution $x \in Q_s$ such that columns c_{l-1}^s and c_l^s are equal from row r_1^s to row r_{k-1}^s, it must be ensured that row r_k^s is lexicographically non increasing, i.e., $x_{r_k^s, c_{l-1}^s} \geq x_{r_k^s, c_l^s}$. The key idea is to exhibit another set $Q_p \in \mathbb{S}$ for quadruple (Q_s, k, l, x), such that Q_p contains x and is sub-symmetric with respect to (R_p, C_p), where the first row of R_p is r_k^s and C_p contains columns c_{l-1}^s and c_l^s. Then inequality $(I_p(c_{l-1}^s))$ will ensure that $x_{r_k^s, c_{l-1}^s} \geq x_{r_k^s, c_l^s}$. For each quadruple (Q_s, k, l, x), the existence of such a subset Q_p in $\mathbb{S}$ will be ensured by *tie-break* condition $(\mathcal{C})$, defined as follows:

$$
(\mathcal{C}) \quad \begin{cases} \forall s \in \{1, \ldots, q\}, \quad \forall k \in \{2, \ldots, |R_s|\}, \quad \forall l \in \{2, \ldots, |C_s|\} \\ \text{If } x \in Q_s \text{ such that } x_{r_{k'}^s, c_{l-1}^s} = x_{r_{k'}^s, c_l^s}, \quad \forall k' \in \{1, \ldots, k-1\}, \\ \ \text{then there exists } p \in \{1, \ldots, q\} \text{ such that } x \in Q_p, C_p \supseteq \{c_{l-1}^s, c_l^s\} \text{ and } r_k^s = \min(R_p) \end{cases}
$$

If tie-break condition $(\mathcal{C})$ holds, inequalities $(Q_s(c_{l-1}^s, c_l^s)), \forall s \in \{1, \ldots, q\}, \quad \forall l \in \{2, \ldots, |C_s|\}$ exactly restrict the solution set to the representative set $\mathcal{X} \cap \mathcal{P}_{sub}(\mathbb{S})$. They are thus full symmetry-breaking, w.r.t. the sub-symmetries defined by $\mathbb{S}$. This gives the proof idea for the following theorem.

Theorem 1 *If tie-break condition* $(\mathcal{C})$ *holds, then:*

(i) $(x, Z(x))$ *satisfies* $(I_s(c_{l-1}^s)), \forall s \in \{1, \ldots, q\}, \quad \forall l \in \{2, \ldots, |C_s|\}$
(ii) $x \in \mathcal{P}_{sub}(\mathbb{S})$

are equivalent.

For general set $\mathbb{S}$, tie-break condition $(\mathcal{C})$ may not hold. Fortunately, it will be shown that we can construct from $\mathbb{S}$ another set $\widetilde{\mathbb{S}}$ satisfying $(\mathcal{C})$ and such that $\mathcal{P}_{sub}(\widetilde{\mathbb{S}}) = \mathcal{P}_{sub}(\mathbb{S})$.

The idea is to divide each Q_s, $s \in \{1, \ldots, q\}$, in smaller subsets such that for each row $r_k^s \in R_s$ and each column $c_l^s \in C_s$, l greater than 1, there is a subset Q, which is sub-symmetric with respect to $(R, C) = (\{r_k^s, \ldots, r_{|R_s|}^s\}, \{c_{l-1}^s, c_l^s\})$.

The set $\widetilde{\mathbb{S}}$ is defined as

$$\widetilde{\mathbb{S}} = \left\{ \widetilde{Q}_s(k,l) \mid s \in \{1,\ldots,q\},\ k \in \{1,\ldots,|R_s|\},\ l \in \{2,\ldots,|C_s|\} \right\}$$

where for each $s \in \{1,\ldots,q\}$, for each $l \in \{2,\ldots,|C_s|\}$, for each $k \in \{1,\ldots,|R_s|\}$, the *tie-break* subset $\widetilde{Q}_s(k,l)$ is defined as

$$\widetilde{Q}_s(k,l) = \left\{ x \in Q_s \mid x_{r,c^s_{l-1}} = x_{r,c^s_l}, \quad \forall r \in \{r^s_1,\ldots,r^s_{k-1}\} \right\}$$

Note that for solution $x \in Q_s$ such that columns c^s_{l-1} and c^s_l are equal from row r^s_1 to row r^s_{k-1}, the set exhibited for quadruple (Q_s,k,l,x) is $\widetilde{Q}_s(k,l)$. Note also that $\widetilde{Q}_s(1,l) = Q_s, l \in \{2,\ldots,|C_s|\}$.

We thus have the following result.

Lemma 4 *Set $\widetilde{\mathbb{S}}$ satisfies $(\mathcal{C})$ and is such that $\mathcal{P}_{sub}(\widetilde{\mathbb{S}}) = \mathcal{P}_{sub}(\mathbb{S})$.*

Proof The symmetry group of tie-break subset $\widetilde{Q}_s(k,l)$ is the sub-symmetric group with respect to $(R,C) = (\{r^s_k,\ldots,r^s_{|R_s|}\},\{c^s_{l-1},c^s_l\})$. Thus if some solution $x \in Q_s$ is such that columns c^s_{l-1} and c^s_l are equal from row r^s_1 to row r^s_{k-1}, then tie-break subset $\widetilde{Q}_s(k,l)$ contains x and is such that $C \supseteq \{c^s_{l-1},c^s_l\}$ and $\min(R) = r^s_k$. Tie-break condition $(\mathcal{C})$ is therefore satisfied by $\widetilde{\mathbb{S}}$. It can be readily checked that the full sub-orbitopes defined by $\widetilde{\mathbb{S}}$ and $\mathbb{S}$ are the same. □

It follows, from Theorem 1, that inequalities $(Q(c,c'))$, $c < c' \in C$, $Q \in \widetilde{\mathbb{S}}$ are full symmetry-breaking with respect to the sub-symmetries defined by $\mathbb{S}$.

Corollary 1 *If for each $Q \in \widetilde{\mathbb{S}}$, $(x, Z(x))$ satisfies inequality $(Q(c,c'))$, $\forall c < c' \in C$, then $x \in \mathcal{P}_{sub}(\mathbb{S})$.*

Set $\widetilde{\mathbb{S}}$ can be considered instead of $\mathbb{S}$ to obtain full-symmetry-breaking inequalities. In this case, one inequality (resp. at most one variable) is added per subset $Q \in \widetilde{\mathbb{S}}$, i.e., $O(qmn)$ inequalities (resp. variables).

Partial-symmetry-breaking relaxations If for each $Q_s \in \mathbb{S}$, set $\widetilde{\mathbb{S}}$ contains sets $\widetilde{Q}_s(k,l)$ for each $l \in \{2,\ldots,|C_s|\}$ and for each $k \in \{1,\ldots,\sigma\}$, where $\sigma \in \{1,\ldots,|R_s|\}$, then the corresponding sub-symmetry-breaking inequalities are not full-symmetry-breaking anymore. In this case we say that they are σ*-symmetry-breaking.*

Variables $\widetilde{z}$ Even if problem-specific variables $\widetilde{z}$ could be more efficient, for each $s \in \{1,\ldots,q\}$ and $l \in \{2,\ldots,|C_s|\}$, variables $\widetilde{z}_s(k,l)$ associated to subsets $\widetilde{Q}_s(k,l)$, $k \in \{2,\ldots,|R_s|\}$, can always be inductively defined as

$$\begin{aligned} \widetilde{z}_s(2,l) &= z_s + x_{r^s_1,c^s_{l-1}} - x_{r^s_1,c^s_l} \\ \widetilde{z}_s(k,l) &= \widetilde{z}_s(k-1,l) + x_{r^s_{k-1},c^s_{l-1}} - x_{r^s_{k-1},c^s_l}, \quad k \in \{3,\ldots,|R_s|\} \end{aligned}$$

Indeed, for any $x \in \mathcal{P}_{sub}(\mathbb{S})$, we have that $x_{r^s_{k-1},c^s_{l-1}} \geq x_{r^s_{k-1},c^s_l}$ if $\widetilde{z}_s(k-1,l) = 0$.

For packing problems, variables $\widetilde{z}$ can be straighforwardly defined as:

$$\widetilde{z}_s(k,l) = z_s + \sum_{k'=1}^{k-1} x_{r^s_{k'},c^s_{l-1}}, \quad k \in \{2,\dots,|R_s|\}$$

Indeed, for $x \in \mathcal{P}_{sub}(\mathbb{S})$, for each $k' \in \{1,\dots,k-1\}$, $x_{r^s_{k'},c^s_l}$ cannot be 1 if $\sum_{k'=1}^{k-1} x_{r^s_{k'},c^s_{l-1}} = 0$.

Example 2 Referring to Example 1, $\widetilde{\mathbb{S}} = \left\{\widetilde{Q}_1(1,l),\ \widetilde{Q}_1(2,l),\ l \in \{2,3\}\right\}$. For each $l \in \{2,3\}$, $\widetilde{Q}_1(1,l) = Q_1$ as for any s, $\widetilde{Q}_s(k,l) = Q_s$ whenever $k = 1$. We also have $\widetilde{Q}_1(2,l) = \left\{x \in Q_1 \mid x_{3,l-1} = x_{3,l}\right\}$. For each $l \in \{2,3\}$, $\widetilde{z}_l$ associated to subset $\widetilde{Q}_1(2,l)$ can be expressed as follows: $\widetilde{z}_l = 2z_1 + (x_{3,l-1} - x_{3,l})$. Indeed, when $z_1 = 0$, inequality (2.1) becomes $x_{3,l-1} \le x_{3,l}$. Thus, $\widetilde{z}_l = 0$ if $x_{3,l-1} = x_{3,l}$ and $z_l \ge 1$ otherwise. When $z_1 = 1$, $\widetilde{z}_l \ge 1$. Hence the following inequalities are full symmetry-breaking:

$$\begin{aligned}
x_{3,l-1} &\le \left(3 - \sum_{j=1}^{3} x_{2,j}\right) + x_{3,l} && \forall l \in \{2,3\} \\
x_{4,l-1} &\le \left(6 + x_{3,l-1} - x_{3,l} - 2\sum_{j=1}^{3} x_{2,j}\right) + x_{4,l} && \forall l \in \{2,3\}
\end{aligned}$$

2.4 Scope extension of sub-symmetry-breaking inequalities

For a given Q_s, and $c < c' \in C_s$, recall that if $x \in Q_s$, then inequality (2.1) enforces that row $r_1 = \min(R_s)$ is lexicographically ordered on columns c and c'. Interestingly it is not necessary that x belongs to Q_s to impose this lexicographical order. Indeed, to enforce this lexicographical order, it suffices that x belongs to some set $\underline{Q}_s$ whose symmetry group includes the transposition π defined as: $\pi(r_1, c) = (r_1, c')$. The idea is then to enforce the lexicographical order for any solution $x \in \underline{Q}_s$, instead of any $x \in Q_s$.

When $\underline{Q}_s$ is such that $Q_s \subset \underline{Q}_s$, then we say that a *scope extension* of the corresponding sub-symmetry-breaking inequality is performed, in the sense that the lexicographical order is applied to a larger solution subset. Roughly speaking, we can say that sub-symmetry-breaking inequalities corresponding to $\underline{Q}_s$ have a larger scope than those corresponding to Q_s. Thus, considering $\underline{Q}_s$ instead of Q_s leads to break more sub-symmetries. Moreover, it may also simplify the expression of variable z_s, as done for example in the second case of Sect. 4.3.

The same argument applies to subsets $\widetilde{Q}_s(k,l)$ which can be replaced by $\underline{\widetilde{Q}}_s(k,l)$ such that $\widetilde{Q}_s(k,l) \subseteq \underline{\widetilde{Q}}_s(k,l)$ and transposition π defined as $\pi(r^s_k, c^s_{l-1}) = (r^s_k, c^s_l)$ is in the symmetry group of $\underline{\widetilde{Q}}_s(k,l)$.

The proposed framework is applied in the following three sections. Two applications are presented in Sects. 3 and 5, where inequalities (2.1) are derived in a straightforward way in the sense that set $\mathbb{S}$ already satisfies tie-break condition (C) in both applications. In Sect. 4, examples of tie-break set $\widetilde{\mathbb{S}}$ construction and of scope extensions are given.

3 Application to the symmetric group case

In this section, we apply the framework of Sect. 2 to any problem whose symmetry group $\mathcal{G}$ is the symmetric group $\mathfrak{S}_n$ acting on the columns. The collection $\mathbb{S}_{\mathfrak{S}}$ of subsets considered will lead to inequalities restricting any solution $x \in \mathcal{X}$ to be in the full orbitope. These inequalities feature variables z which can be explicitly expressed from x with $O(mn)$ linear inequalities. Here, the sub-symmetries considered are restrictions of symmetries' actions to solution subsets.

A complete linear description of the 2-column full orbitope, featuring additional integer variables, is proposed in [26]. In the general n-column case, we show that these inequalities can also be derived using the framework described in Sect. 2, and can be used as full symmetry-breaking inequalities.

We consider

$$\mathbb{S}_{\mathfrak{S}} = \Big\{ Q_{i,j},\ i \in \{0\} \cup \{1, \dots, m-1\},\ j \in \{2, \dots, n\} \Big\},$$
$$\text{where } Q_{i,j} = \Big\{ x \in \mathcal{X} \mid x_{i',j-1} = x_{i',j} \quad \forall i' \in \{1, \dots, i\} \Big\}.$$

Subset $Q_{i,j}$ is the set of feasible solutions such that columns $j-1$ and j are equal from row 1 to row i. Note that $Q_{0,j} = \mathcal{X}$. The symmetry group of $Q_{i,j}$ is then the sub-symmetric group with respect to $(R_i, \{j-1, j\})$ where $R_i = \{i+1, \dots, m\}$. It can be readily checked that in this case, $\mathbb{S}$ already satisfies condition $(\mathcal{C})$.

Let variable $z_{i,j}$ be such that $z_{i,j} = 0$ if $x \in Q_{i,j}$ and 1 otherwise. Note that for all $j \in \{2, \dots, n\}$, $Q_{0,j} = \mathcal{X}$, thus $z_{0,j} = 0$, $\forall x \in \mathcal{X}$. Note also that $\mathcal{X} \cap \mathcal{P}_{sub}(\mathbb{S}_{\mathfrak{S}})$ is a subset of the full orbitope. Thus, given that the columns of any $x \in \mathcal{X} \cap \mathcal{P}_{sub}(\mathbb{S}_{\mathfrak{S}})$ are in a non-increasing lexicographical order, function Z can be chosen such that $Z(x) = z$, where z satisfies the following linear inequalities.

$$\left\{\begin{array}{lll}
z_{1,j-1} = x_{1,j-1} - x_{1,j} & \forall j \in \{2, \dots, n\} & (3.1a)\\
z_{i,j-1} \le z_{i-1,j-1} + x_{i,j-1} & \forall i \in \{2, \dots, m\},\ j \in \{2, \dots, n\} & (3.1b)\\
z_{i,j-1} + x_{i,j} \le 1 + z_{i-1,j-1} & \forall i \in \{2, \dots, m\},\ j \in \{2, \dots, n\} & (3.1c)\\
x_{i,j-1} \le z_{i,j-1} + x_{i,j} & \forall i \in \{2, \dots, m\},\ j \in \{2, \dots, n\} & (3.1d)\\
z_{i-1,j-1} \le z_{i,j-1} & \forall i \in \{2, \dots, m\},\ j \in \{2, \dots, n\} & (3.1e)
\end{array}\right.$$

Constraint (3.1a) sets variable $z_{1,j-1}$ to 1 whenever columns $j-1$ and j are different and in a non-increasing lexicographical order on row 1, and to 0 when they are equal. Constraint (3.1b) (resp. (3.1c)) sets variable $z_{i,j-1}$ to 0 when $z_{i-1,j-1} = 0$ and columns $j-1$ and j are equal to 0 (resp. 1) on row i. Constraint (3.1d) sets variable $z_{i,j-1}$ to 1 if columns $j-1$ and j are different and in a non-increasing lexicographical order on row i. Constraint (3.1e) sets $z_{i,j-1}$ to 1 when variable $z_{i-1,j-1} = 1$, i.e., when columns $j-1$ and j are different from row 1 to $i-1$.

For each $i \in \{2, \dots, m\}$ and $j \in \{2, \dots, n\}$, sub-symmetry breaking inequality (2.1) for subset $Q_{i-1,j}$ is as follows:

$$x_{i,j} \leq z_{i-1,j} + x_{i,j-1} \tag{3.2}$$

It ensures that if columns $j-1$ and j of x are equal from row 1 to i, then row $i+1$ is in a non-increasing lexicographical order.

Note that if $z_{i-1,j} - z_{i,j} = -1$ then necessarily $x_{i,j} = 0$. Thus inequality (3.2) can be lifted to

$$x_{i,j} \leq (2z_{i-1,j} - z_{i,j}) + x_{i,j-1} \tag{3.3}$$

In the special case when $n = 2$, by replacing variable $z_{i,j}$ by $y_{i,j}$ where $z_{i,j} = 1 - \sum_{i'=1}^{i} y_{i',j}$, for each $i \in \{1,\ldots,m\}$, $j \in \{1,2\}$, inequalities (3.1a)–(3.3) yield the complete linear description of the 2-column full orbitope proposed in [26]. Note that the latter description is an extended formulation, i.e., not in the original space as additional variables are introduced.

In the general n-column case, inequalities (3.1a)–(3.3) are still full symmetry-breaking (by Theorem 1), and then can be used in practice to restrict the feasible set to any full orbitope. In this case, $O(mn)$ additional variables and constraints are needed. Possible alternatives to using additional variables also exist, see e.g., [4,26].

4 Application to the graph coloring problem

In this section, the framework of Sect. 2 is applied to the graph coloring problem.

Given an undirected graph $G = (V, E)$ with $|V| = n$, a *vertex coloring* of G is an assignment of values $\{1,\ldots,n\}$, denoted as *colors*, to the vertices so that no two adjacent vertices receive the same color. The minimum number of colors in a vertex coloring of G is called the *chromatic number* $\chi(G)$ of G. The *vertex coloring problem* is to find a vertex coloring with a minimum number of colors. Let K be an upper bound on $\chi(G)$. The classical IP formulation F [10] is the following.

$$\begin{aligned}
\min_{x,y} \quad & \sum_{k=1}^{K} y_k \\
\text{s. t.} \quad & x_{i,k} + x_{j,k} \leq y_k \quad \forall \{i,j\} \in E, \quad \forall k \in \{1,\ldots,K\} && (4.1)\\
& \sum_{k=1}^{K} x_{i,k} = 1 \quad \forall i \in V && (4.2)\\
& x_{i,k}, y_k \in \{0,1\} \quad \forall i \in V, \quad \forall k \in \{1,\ldots,K\} && (4.3)
\end{aligned}$$

A solution is a matrix $x = (x_{i,k})$ where each column corresponds to a color and each row corresponds to a vertex. Variable $x_{i,k}$ indicates that color $k \in \{1,\ldots,K\}$ is assigned to vertex $i \in \{1,\ldots,n\}$, and variable y_k indicates that color k is used to color some vertices. The feasible solution set is denoted by $\mathcal{X}_{col}$.

Formulation (F) exhibits many symmetries. As pointed out in [20], symmetries make this formulation difficult to solve in particular because they lead to the feasibility of many fractional vertices, thus resulting in a poor LP-bound.

The symmetry group associated to formulation (F) contains the symmetric group acting on the columns of solution matrices. Indeed, a column corresponds to a color and a new vertex coloring can be obtained from another by permuting color indices. Techniques to break such symmetries have been largely investigated in the literature. One option is to propose alternative formulations. For instance, an extension of the classical formulation has been devised in [8,11] using the notion of representative vertices, i.e., vertices representing a color. Moreover, a column generation based linear program, proposed in [29], provides very good lower bounds. This approach is also used to come up with exponential size ILPs [15,27]. Another option is to add symmetry-breaking inequalities to formulation (F) in order to remove non-representative solutions from the feasible set. For example, the authors of [30] propose the following full-symmetry-breaking inequalities which correspond to column inequalities dedicated to partition problems:

$$x_{i,k} \leq \sum_{i'=k-1}^{i-1} x_{i',k-1}, \qquad \forall 1 \leq k \leq i$$

where $x_{i,k} = 0$ for any k and i such that $k > i$.

In [20], a generalization of such partition-dedicated column inequalities is introduced and is as follows:

$$\sum_{k'=k}^{\min(i,n)} x_{i,k'} \leq \sum_{i'=k-1}^{i-1} x_{i',k-1}, \qquad \forall 1 \leq k \leq i \tag{4.4}$$

It is also shown in [20] that the partitioning orbitope is completely described by trivial inequalities and *shifted column inequalities*, defined as:

$$\sum_{k=j}^{\min(i,n)} x_{i,k} \leq \sum_{p=1}^{i-j+1} x_{p+c_p-1,c_p} \tag{4.5}$$

for any $(i, j) \in (n, K)$ and integers $c_1 \leq \cdots \leq c_{i-j+1} \leq j-1$ such that for $p \in \{1, \ldots, i-j+1\}$, $p + c_p - 1 \in \{1, \ldots, n\}$.

4.1 Sub-symmetries in the graph coloring problem

Formulation (F) features many sub-symmetries. For two colors c_1 and c_2, a natural sub-symmetry arises from the possibility of permuting colors c_1 and c_2 in a subset R of vertices. This permutation is a symmetry for the colorings such that all neighbors of R are colored neither by c_1 nor by c_2. Note that a convenient way to obtain such a subset R is to start selecting two subsets S_1 and S_2 and then choose R non-adjacent to them.

	c_1	c_2
S_1	(1)	(0)
S_2	(0)	(1)
R	$(*)$	$(*)$
$N(R)$	(0)	(0)
U	(*)	(*)

Fig. 1 Two columns of $Q^{S_1,S_2,R}_{c_1,c_2}$ where $U = V \backslash (S_1 \cup S_2 \cup R \cup N(R))$

Let S_1 and S_2 be two disjoint stable subsets of V and let $R \subseteq V$ such that any $r \in R$ is neither a neighbor of S_1 nor of S_2. The neighborhood of a set S is denoted by $N(S) = \{v \in V \backslash S \ : \ \exists \{u, v\} \in E \ \text{ s. t. } \ u \in S\}$

Consider solution subset

$$Q^{S_1,S_2,R}_{c_1,c_2} = \Big\{ x \in \mathcal{X}_{col} \mid x_{i,c_1} = 1 \quad \forall i \in S_1, \ x_{i,c_2} = 1 \quad \forall i \in S_2, \ x_{i,c_1} = x_{i,c_2} = 0 \quad \forall i \in N(R) \Big\}$$

Subset $Q^{S_1,S_2,R}_{c_1,c_2}$ contains all colorings such that S_1 has color c_1, S_2 has color c_2, and the neighbors of R are neither colored by c_1 nor by c_2. Therefore, in general there exists an exponential number of such subsets $Q^{S_1,S_2,R}_{c_1,c_2}$. An illustration of columns c_1 and c_2 for the solutions of $Q^{S_1,S_2,R}_{c_1,c_2}$ is given in Fig. 1. The variables that are fixed in $Q^{S_1,S_2,R}_{c_1,c_2}$ are indicated with (0) or (1), while the others are indicated with symbol $(*)$.

Subset $Q^{S_1,S_2,R}_{c_1,c_2}$ is sub-symmetric with respect to $(R, \{c_1, c_2\})$. Such sub-symmetries, referred to as *0-neighbor sub-symmetries*, correspond to permutations of a set R of vertices between colors c_1 and c_2, because neighbors of R have any colors but c_1 and c_2.

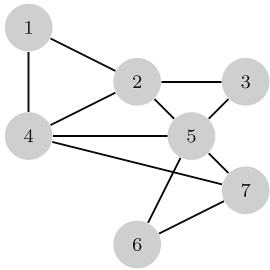

Fig. 2 Example of a graph $G = (V, E)$

4.2 0-neighbor-sub-symmetry-breaking inequalities

Variable z associated to $Q^{S_1,S_2,R}_{c_1,c_2}$ can be linearly expressed in terms of x variables, as follows

$$z = \sum_{s \in S_1}(1 - x_{s,c_1}) + \sum_{s \in S_2}(1 - x_{s,c_2}) + \sum_{r \in N(R)\setminus N(S_1)} x_{r,c_1} + \sum_{r \in N(R)\setminus N(S_2)} x_{r,c_2} \tag{4.6}$$

Note that there is no need to check that vertices of $N(S_1)$ (resp. $N(S_2)$) are not colored by color c_1 (resp. c_2). This is actually enforced by inequality (4.1) since we impose that all elements of S_1 (resp. S_2) are colored by c_1 (resp. c_2).

As there is exactly one 1-entry on each solution row, the GCP is a partitioning problem and a fortiori a packing problem. Thus packing-specific sub-symmetry-breaking inequalities (2.2) can be applied:

$$x_{r_1,c_2} \le z, \text{ where } r_1 = \min R. \tag{4.7}$$

Example 3 Figure 2 gives an example of a graph $G = (V, E)$. For $S_1 = \{2\}$ and $S_2 = \varnothing$. Let set $R = \{6, 7\}$ which does not contain any neighbor of 2. Here $N(R) = \{4, 5\}$. For given colors c_1 and c_2, set $Q^{S_1,S_2,R}_{c_1,c_2}$ contains all solutions such that vertex 2 is colored by c_1 and such that vertices 4 and 5 are colored neither by c_1 nor by c_2. Here $r_1 = \min(R) = 6$, thus sub-symmetry-breaking inequality (4.7) writes

$$x_{6,c_2} \le (1 - x_{2,c_1}) + x_{4,c_2} + x_{5,c_2}.$$

Note that considering non-empty set S_1 enables us to check that vertex 2 is colored by c_1 without checking that vertices 4 and 5 are not colored by c_1. Thus the inequality is tighter than the one with $S_1 = \varnothing$.

Particular case Note that $Q^{S_1,S_2,R}_{c_1,c_2} \subseteq Q^{\varnothing,\varnothing,R}_{c_1,c_2} = \{x \in \mathcal{X}_{col} \mid x_{i,c_1} = x_{i,c_2} = 0 \quad \forall i \in N(R)\}$. Therefore, as $Q^{\varnothing,\varnothing,R}_{c_1,c_2}$ may contain more solutions, the derived sub-symmetry-breaking inequalities will have a larger scope. However, associated variables z may be different: variable z corresponding to $Q_{c_1,c_2}^{\varnothing,\varnothing,R}$ will feature the term $\sum_{v\in N(r_1)} x_{v,c_1}$, for a given $r_1 \in R$, while variable z corresponding to $Q_{c_1,c_2}^{S_1,S_2,R}$ will feature the term $\sum_{s\in S_1\cap N(r_1)}(1-x_{s,c_1}) + \sum_{v\in N(r_1)\setminus N(S_1)} x_{v,c_1}$ instead. The former term may feature much less variables (in particular if $N(S_1)$ contains a lot of elements of $N(r_1)$) but can also lead to weaker sub-symmetry-breaking inequalities.

Condition ($\mathcal{C}$) Since set $\mathbb{S}$ contains arbitrary $Q_{c_1,c_2}^{S_1,S_2,R}$, condition ($\mathcal{C}$) is not necessarily satisfied. For each $Q_{c_1,c_2}^{S_1,S_2,R} \in \mathbb{S}$, for any $r \in \{1,\ldots,|R|\}$, set $\widetilde{Q}_{c_1,c_2}^{S_1,S_2,R}(r)$ is the tie-break set defined in Sect. 2.3:

$$\widetilde{Q}_{c_1,c_2}^{S_1,S_2,R}(r) = Q_{c_1,c_2}^{S_1,S_2,R} \cap \left\{x \mid x_{i,c_1} = x_{i,c_2}, \quad \forall i \in \{v_1,\ldots,v_{r-1}\}\right\}$$

where $R = \{v_1,\ldots,v_{|R|}\}$, $v_1 < \cdots < v_{|R|}$. Let us then consider set $\widetilde{\mathbb{S}}$ containing the sets in $\mathbb{S}$ and sets $\widetilde{Q}_{c_1,c_2}^{S_1,S_2,R}(r)$, for each $r \in \{2,\ldots,|R|\}$. By Lemma 4, set $\widetilde{\mathbb{S}}$ satisfies condition ($\mathcal{C}$) and therefore the associated sub-symmetry-breaking inequalities are full symmetry-breaking. As shown in Sect. 2.3, in the case of packing problems, variable $\widetilde{z}$ associated to $\widetilde{Q}_{c_1,c_2}^{S_1,S_2,R}(r)$ can be expressed as $\widetilde{z} = z + \sum_{i=1}^{r-1} x_{v_i,c_1}$, where z is the variable associated to set $Q_{c_1,c_2}^{S_1,S_2,R}$.

Column inequalities We can show that partition-dedicated column inequalities (4.4) can also be derived using the proposed framework, by considering solution subsets $Q_{k,k+1}^{\varnothing,\varnothing,V}$, $\widetilde{Q}_{k,k+1}^{\varnothing,\varnothing,V}(r)$, $r \in \{1,\ldots,|V|\}$ and associated variables $\widetilde{z} = \sum_{j=k}^{r-1} x_{j,k} - \sum_{k'=k+2}^{\min(r,K)} x_{r,k'}$. Recall that these inequalities break all-column-permutation symmetries in partition problems.

4.3 Scope extension

There is an exponential number of sub-symmetric subsets $Q_{c_1,c_2}^{S_1,S_2,R}$ thus in practice one must choose which subsets to consider. An interesting question is how to choose R once S_1 and S_2 are fixed. Indeed, sets R, S_1 and S_2 lead to $|R|$-symmetry-breaking inequalities for $Q_{c_1,c_2}^{S_1,S_2,R}$. One must find a trade-off between the size of Q and the size of R. Moreover, it is possible to apply scope extension as described in Sect. 2.4.

Cardinality of R For a given set R, one sub-symmetry-breaking inequality per $\widetilde{Q}_{c_1,c_2}^{S_1,S_2,R}(r)$, $r \in \{1,\ldots,|R|\}$, can be added, resulting in a set of $|R|$-symmetry-breaking inequalities. On the one hand, the larger R, the larger possible set of sub-symmetry-breaking inequalities. Note that the set R with maximum cardinality for fixed S_1 and S_2 is $R_{\max} = V\setminus[N(S_1)\cup N(S_2)\cup S_1\cup S_2]$.

On the other hand, a smaller subset $R' \subset R$ may lead to a larger subset $Q_{c_1,c_2}^{S_1,S_2,R'}$: indeed, if $N(R') \subseteq N(R)$, then $Q_{c_1,c_2}^{S_1,S_2,R} \subseteq Q_{c_1,c_2}^{S_1,S_2,R'}$. It means that the derived sub-symmetry-breaking inequalities have a larger scope.

In the experimental results presented in Sect. 4.5, we chose the following subsets $R = V\setminus[N(S_1)\cup N(S_2)\cup S_1\cup S_2]$, where S_1 and S_2 are singletons. The thing is that considering large sets R leads to a large corresponding set of $|R|$-symmetry-breaking inequalities. It was computationally more efficient to add only the corresponding σ-

symmetry-breaking inequalities, where σ was chosen in $\{1, 2, 3\}$. Then a large scope seems a better option than a large set of sub-symmetry-breaking inequalities.

Connected components of R Consider the subgraph G_R induced by R and its connected components $R_1, \ldots, R_k$. Suppose R_1 is the connected component containing $r_1 = \min(R)$. Note that the symmetry group $\mathcal{G}_{Q_{c_1,c_2}^{S_1,S_2,R_1}}$ contains the transposition π defined as $\pi(r_1, c_1) = (r_1, c_2)$. Moreover, $Q_{c_1,c_2}^{S_1,S_2,R} \subseteq Q_{c_1,c_2}^{S_1,S_2,R_1}$. Therefore, the scope of the sub-symmetry-breaking inequality associated to row r_1 can be extended to $Q_{c_1,c_2}^{S_1,S_2,R_1}$. This simplifies the expression of associated variable z as it only considers the neighbors of $R_1 \subseteq R$ instead of all neighbors of R. Applying such scope extension for each $r \in R$ is equivalent to use sub-symmetry-breaking inequalities corresponding to subsets $Q_{c_1,c_2}^{S_1,S_2,R_i}$, $i \in \{1, \ldots, k\}$ and associated tie-break sets.

For each tie-break set, scope extension can also be recursively applied to the corresponding sub-symmetry-breaking inequalities. For example, given $r_1, \ldots, r_{k_1}$ the vertex indices of R_1, the tie-break set $\widetilde{Q}_{c_1,c_2}^{S_1,S_2,R_1}(2, 2)$ associated to row r_2 is sub-symmetric with respect to $(\{r_2, \ldots, r_{k_1}\}, \{c_1, c_2\})$ (cf. Sect. 2.3). If the subgraph of G induced by $R_1 \backslash \{r_1\}$, i.e., $\{r_2, \ldots, r_{k_1}\}$, has multiple connected components $R'_1, \ldots, R'_{k'_1}$, then we can perform a scope extension by considering associated subsets $Q_{c_1,c_2}^{S_1,S_2,R'_k} \cap \{x_{r_1,c_1} = x_{r_1,c_2} = 0\}$, for each $k \in \{1, \ldots, k'_1\}$, instead of considering $\widetilde{Q}_{c_1,c_2}^{S_1,S_2,R_1}(2, 2)$.

4.4 Implementation description

Preliminary results lead us to choose the subset of sub-symmetry-breaking inequalities using the following parameters. Note that these parameters are customized automatically with respect to instance characteristics.

Vertex subsets Sets S_1 and S_2 are chosen to be singletons. For each pair of vertices $s_1 < s_2 \in V$, for each colors $c_1 < c_2$, we consider solution subset $Q_{c_1,c_2}^{\{s_1\},\{s_2\},R}$, where $R = V \backslash (\{s_1\} \cup \{s_2\} \cup N(\{s_1\}) \cup N(\{s_2\}))$, and $\min(R) > 1$ as column inequalities (4.4) already break symmetries on row 1.

Pairs of colors For each triplet (S_1, S_2, R), all pairs of consecutive colors are considered, except in three particular cases where all possible pairs are considered. The first case is when n is large, $n \geq 900$, and K is small, $K \leq 10$. We consider subsets $Q_{c_1,c_2}^{\{s_1\},\{s_2\},R}$ for each pair of vertices s_1, s_2 and for each pair of colors $c_1 < c_2$. Indeed, when n is large, there are more sub-symmetries to break, and a small K enables to consider all pairs of colors with barely no extra computational time. The second case is when the number of edges is small, $|E| < 300$. It proved beneficial to consider all pairs of colors. The third case is when the upper bound K on the chromatic number is large compared to n (but not too large in absolute): $\frac{n}{K} < 10$ and $K < 100$. It is also useful to consider all pairs of colors. Indeed, as a large K compared to n leads to many columns compared to rows, thus many symmetries on the columns arise. Therefore, the columns should be handled pairwise to break such symmetries, provided K is not too large.

Parameter σ For each triplet (S_1, S_2, R), we consider corresponding tie-break sets $\widetilde{Q}$ to obtain partial σ-symmetry-breaking inequalities, as defined in Sect. 2. Parameter $\sigma \in \{1, \ldots, |R|\}$ is chosen with respect to the number of vertices n. Indeed, subset R is potentially larger when n gets larger, thus there may be more sub-symmetries. Therefore we increase σ according to n as follows: $\sigma = 1$ when $n < 100$, $\sigma = 2$ when $100 \leq n \leq 900$, and $\sigma = 3$ when $n \geq 900$.

Number of variables in z To prevent too large a processing time, we consider in most cases sub-symmetry-breaking inequalities such that the number of variables needed to express z, referred to as size of z, is lower than or equal to 10. In this way, these inequalities are also likely to have a large scope. On the contrary, there are two cases where a larger size for z is considered. When the graph is relatively large and dense, i.e., $\frac{|E|}{n} > 10$ and $n > 200$ (resp. very dense, i.e., $\frac{|E|}{n} > 100$), variables z of size 20 (resp. 30) at most are needed to capture more sub-symmetries as subset R is likely to have many neighbors. Similarly, when the graph is smaller (i.e., $n < 200$ and $E < 1000$) but with large enough upper bound (i.e., $K > 5$) the sub-symmetries captured by variables z of size 20 seem to be quite helpful as well.

Connected components When K is not too large, and when the graph is dense, i.e. $\frac{|E|}{n} > 10$ and $K < 15$ (resp. very dense, i.e. $\frac{|E|}{n} > 50$ and $K < 100$), subset R is likely to be quite small as elements of R are chosen outside the neighborhood of s_1 and s_2. Therefore R may decompose into connected components. In this case it appears useful to perform the scope extension from Sect. 4.3, thus replacing R by its connected components.

Limit number on sub-symmetry-breaking inequalities We set the limit on the number of sub-symmetry-breaking inequalities to be added to 50.000, except for extremely symmetric instances (i.e., $K \geq 100$) where a (very) large number of sub-symmetry-breaking inequalities proves useful.

4.5 Experimental results

Experimental results are performed on DIMACS graph coloring benchmark instances [14]. These instances are classified according to their difficulty to be solved. In particular, class NP-s stands for instances which are solvable by the best known algorithm in less than a minute, class NP-m in less than an hour, class NP-h in less than a day, class NP-? means the instance is not solved or the time is not known. Note that the best known algorithm is unlikely to be formulation F using default Cplex.

All experiments are carried out using Cplex 12.8 C++ API on 28 threads of a cluster node with a 64 bit Intel Xeon CPU E5-2697 v3 processor running at 2.6 GHz, and 64 GB of RAM memory. Instances are solved until optimality, defined within 10^{-7} of relative optimality tolerance, or until the time limit of 7200 seconds is reached.

We compare the following symmetry-breaking techniques applied to the classical GCP formulation F, where the upper bound K on the number of colors is computed as a preprocessing step using DSATUR [7] algorithm :

F-Col formulation (F) with column inequalities (4.4)

F-Part formulation (F) with column inequalities (4.4), and shifted column inequalities (4.5)

F-Sub formulation (F) with column inequalities (4.4) and the subset of sub-symmetry-breaking inequalities described in Sect. 4.4.

We observed the best performance of shifted column inequalities on the small instances, i.e., $n < 200$ and $K < 20$. Therefore, we add these inequalities to F-Sub on such instances.

Note that column inequalities and the chosen sub-symmetry-breaking inequalities are initially added, whereas shifted column inequalities are separated using Cplex Generic Callback.

Since we use Cplex 12.8 C++ API with default setting, Cplex's internal symmetry-breaking techniques are turned on by default. To assess the impact of such techniques over the compared inequalities, we also include experiments where Cplex's internal symmetry-breaking techniques are turned off. In particular, we deactivate the latter techniques in formulations F and F-Sub, which are respectively denoted by F-S0 and F-Sub-S0. Since the performances of F and F-Col are similar, and since F-Col and F-Part handle the same symmetries, it does not appear useful to include F-Col-S0 and F-Part-S0 variants in the tables.

Tables 1, 2, 3 and 4 provide, for each DIMACS instance and for each symmetry-breaking technique:

n: number of vertices,
$|E|$: number of edges,
K: upper bound on the number of colors obtained with DSATUR,
UB: upper bound on the number of colors obtained at the end
LB: lower bound on the number of colors obtained at the end
#SSBI: number of sub-symmetry-breaking inequalities added
#Part: number of partitioning orbitope inequalities added
#nodes: number of nodes in the B&B tree, 0 meaning resolution at root node
CPU: CPU time in seconds, including the time spent to generate sub-symmetry-breaking inequalities

The number of column inequalities (4.4) is not indicated in the tables as it is in $O(nK)$.

To keep the focus on the most interesting instances, the results are presented for relatively hard instances only, i.e., instances for which Cplex needs at least 50 seconds to solve with formulation F. Moreover we report results as soon as a difference appears among some considered techniques with respect to either upper or lower bounds or CPU time.

In general, the performances of F-S0, F, F-Col are surprisingly similar, indicating that neither Cplex's internal symmetry-breaking techniques nor column inequalities lead to significant CPU time reduction. There are still some instances where F and F-Col slightly improve F-S0, for example "FullIns" instances (from NP-m and NP-?).

As for NP-s instances in Table 1, there is one instance (queen9-9) on which no sub-symmetry-breaking inequality is found. The time spent to search for sub-symmetry-breaking inequalities appears to be significant and could not be compensated for by any symmetry breaking. On 1-Insertions-4 instance, F-Part converges faster. On all other

Table 1 Experimental results on NP-s graph coloring DIMACS instances

	Method	n	E	K	UB	LB	#SSBI	#Part	#nodes	CPU
1-Insertions-4	F-S0	67	232	5	5	5	0	0	305,455	371.28
1-Insertions-4	F	67	232	5	5	5	0	0	305,455	372.78
1-Insertions-4	F-Col	67	232	5	5	5	0	0	305,455	373.51
1-Insertions-4	F-Part	67	232	5	5	5	0	1160	91,261	122.33
1-Insertions-4	F-Sub-S0	67	232	5	5	5	10,780	985	162,004	452.5
1-Insertions-4	F-Sub	67	232	5	5	5	10,780	660	73,288	218.14
DSJC125.1	F-S0	125	736	5	5	5	0	0	0	57.45
DSJC125.1	F	125	736	5	5	5	0	0	0	54.19
DSJC125.1	F-Col	125	736	5	5	5	0	0	0	59.94
DSJC125.1	F-Part	125	736	5	5	5	0	6	0	69.9
DSJC125.1	F-Sub-S0	125	736	5	5	5	7920	9	0	24.53
DSJC125.1	F-Sub	125	736	5	5	5	7920	9	0	25.82
queen9-9	F-S0	81	2112	10	10	10	0	0	130,119	5048.6
queen9-9	F	81	2112	10	10	10	0	0	130,119	5058.1
queen9-9	F-Col	81	2112	10	10	10	0	0	130,119	5036.87
queen9-9	F-Part	81	2112	10	10	10	0	63	104,943	4252.99
queen9-9	F-Sub-S0	81	2112	10	10	9	0	0	278,060	7200
queen9-9	F-Sub	81	2112	10	10	9	0	0	274,752	7200
r125.1c	F-S0	125	7501	46	46	46	0	0	0	969.82
r125.1c	F	125	7501	46	46	46	0	0	0	964.61
r125.1c	F-Col	125	7501	46	46	46	0	0	0	968.14
r125.1c	F-Part	125	7501	46	46	46	0	14,431	0	568.96
r125.1c	F-Sub-S0	125	7501	46	46	46	50,000	0	0	355.96
r125.1c	F-Sub	125	7501	46	46	46	50,000	0	0	349.94
school1	F-S0	385	19,095	14	14	14	0	0	0	2603.03
school1	F	385	19,095	14	14	14	0	0	0	2580.66
school1	F-Col	385	19,095	14	14	14	0	0	0	2490.22
school1	F-Part	385	19,095	14	14	14	0	195	0	6530.22
school1	F-Sub-S0	385	19,095	14	14	14	5772	0	0	1021.12
school1	F-Sub	385	19,095	14	14	14	5772	0	0	1007.35

instances, F-Sub is the most efficient technique. For example, on school1 instance, F-S0, F and F-Col (resp. F-Part) terminate in about 2500 seconds (resp. 6500 seconds), while F-Sub finishes in around 1000 seconds. Interestingly, F-Sub and F-Sub-S0 perform similarly except on 1-Insertions-4 instance.

Among NP-m instances in Table 2, there is one instance (ash608GPIA) where, surprisingly, F-S0 is the most efficient. On le450-15a instance, none of the techniques is able to reach optimality within time limit but F-Part seems slightly better as the upper bound found is tighter. On all other instances, F-sub or F-sub-S0 outper-

Table 2 Experimental results on NP-m graph coloring DIMACS instances (continued)

	Method	n	E	K	UB	LB	#SSBI	#Part	#nodes	CPU
ash608GPIA	F-S0	1216	7844	8	4	4	0	0	0	3260.55
ash608GPIA	F	1216	7844	8	4	4	0	0	0	3884.08
ash608GPIA	F-Col	1216	7844	8	4	4	0	0	0	3910.26
ash608GPIA	F-Part	1216	7844	8	4	4	0	2005	0	3853.68
ash608GPIA	F-Sub-S0	1216	7844	8	4	4	11,256	0	3760	7200
ash608GPIA	F-Sub	1216	7844	8	4	4	11,256	0	0	3473.38
school1-nsh	F-S0	352	14,612	21	20	14	0	0	0	7200
school1-nsh	F	352	14,612	21	20	14	0	0	0	7200
school1-nsh	F-Col	352	14,612	21	20	14	0	0	0	7200
school1-nsh	F-Part	352	14,612	21	20	14	0	71	0	7200
school1-nsh	F-Sub-S0	352	14,612	21	14	14	4640	0	0	1712.02
school1-nsh	F-Sub	352	14,612	21	14	14	4640	0	0	1707.78
2-FullIns-4	F-S0	212	1621	6	6	6	0	0	3174	339.7
2-FullIns-4	F	212	1621	6	6	6	0	0	691	143.2
2-FullIns-4	F-Col	212	1621	6	6	6	0	0	691	155.36
2-FullIns-4	F-Part	212	1621	6	6	6	0	942	444	163.57
2-FullIns-4	F-Sub-S0	212	1621	6	6	6	140	0	1140	231.02
2-FullIns-4	F-Sub	212	1621	6	6	6	140	0	857	127.03
4-Insertions-3	F-S0	79	156	4	4	4	0	0	837,188	1364.46
4-Insertions-3	F	79	156	4	4	4	0	0	837,188	1356.85
4-Insertions-3	F-Col	79	156	4	4	4	0	0	837,188	1363.89
4-Insertions-3	F-Part	79	156	4	4	4	0	347	71,877	112.68
4-Insertions-3	F-Sub-S0	79	156	4	4	4	4500	310	36,702	76.38
4-Insertions-3	F-Sub	79	156	4	4	4	4500	269	86,430	185.6
5-FullIns-3	F-S0	154	792	8	8	8	0	0	25,091	288.4
5-FullIns-3	F	154	792	8	8	8	0	0	18,129	221.15
5-FullIns-3	F-Col	154	792	8	8	8	0	0	18,129	218.51
5-FullIns-3	F-Part	154	792	8	8	8	0	2092	6832	153.91
5-FullIns-3	F-Sub-S0	154	792	8	8	8	14,924	6000	21,295	527.74
5-FullIns-3	F-Sub	154	792	8	8	8	14,924	2378	7565	169.94
le450-15a	F-S0	450	8168	19	17	15	0	0	132	7200
le450-15a	F	450	8168	19	17	15	0	0	125	7200
le450-15a	F-Col	450	8168	19	17	15	0	0	128	7200
le450-15a	F-Part	450	8168	19	16	15	0	7717	0	7200
le450-15a	F-Sub-S0	450	8168	19	17	15	2520	0	93	7200
le450-15a	F-Sub	450	8168	19	17	15	2520	0	96	7200

Table 2 continued

	Method	n	E	K	UB	LB	#SSBI	#Part	#nodes	CPU
myciel6	F-S0	95	755	7	7	5	0	0	849,532	7200
myciel6	F	95	755	7	7	5	0	0	850,427	7200
myciel6	F-Col	95	755	7	7	5	0	0	851,264	7200
myciel6	F-Part	95	755	7	7	5.2	0	10,924	369,313	7200
myciel6	F-Sub-S0	95	755	7	7	6	11,508	14,235	864,610	7200
myciel6	F-Sub	95	755	7	7	5.25	11,508	7030	263,566	7200
wap05a	F-S0	905	43,081	51	50	50	0	0	0	838.9
wap05a	F	905	43,081	51	50	50	0	0	0	822.08
wap05a	F-Col	905	43,081	51	50	50	0	0	0	836.3
wap05a	F-Part	905	43,081	51	50	50	0	9674	0	849.68
wap05a	F-Sub-S0	905	43,081	51	50	50	12,300	0	0	501.02
wap05a	F-Sub	905	43,081	51	50	50	12,300	0	0	495.89

forms the other techniques. For example on wap05a instance, other techniques need around 800 seconds to reach optimality while F-Sub and F-Sub-S0 need only 400 seconds. Similarly, on school1-nsh instance, the other techniques do not converge within 7200 seconds while F-Sub and F-Sub-S0 do in 1700 seconds. Interestingly, on 4-Insertions-3 instance, F-Sub-S0 is better than F-Sub, suggesting that Cplex's internal symmetry-breaking-techniques are computationally expensive compared to the amount of symmetries broken when sub-symmetry-breaking inequalities are used.

None of the techniques converged on NP-h instances in Table 3, but the lower bounds obtained after 7200 seconds are different depending on the technique used. On r250.5 instance, the best lower bound (61.33) is obtained by F, F-S0 and F-Col, while F-Part and F-Sub (resp. F-Sub-S0) only obtain a bound of respectively 58 and 60. On flat300-28-0 instance, F-Part provides the best lower bound (9.82062) while other techniques only obtain 9.81756. On DSJC125.5 instance, this is the other way around as F, F-S0, F-Col, F-Sub and F-Sub-S0 all obtain the same lower bound and F-Part does not as good. On remaining instances, F-Sub and F-Sub-S0 are able to obtain a better lower bound than the other techniques. It is particularly the case on DSJR500.5 instance, where F-Sub and F-Sub-S0 are able to reach a bound of 114, while other techniques provide a bound of 117.

For NP-? instances in Table 4, only two instances are solved to optimality. On some instances (2-Insertions-4, flat300-20-0 and flat300-26-0), F-Sub shows no improvement compared to the other techniques. On 3-FullIns4 instance, F-Col is the most efficient as it converges in 2598 seconds, while F-Sub (resp. F, F-Part) converges in 3769 seconds (resp. 4442 seconds, 5012 seconds). On all other instances, F-Sub performs better. For example, on 1-FullIns-5 instance, the other techniques do not terminate within 7200 seconds while F-Sub converges in 3700 seconds. Also, on r125.5 instance, F-Sub is able to converge faster (in 170 seconds) than the other techniques (from 259 to 1000 seconds).

Table 3 Experimental results on NP-h graph coloring DIMACS instances

	Method	n	E	K	UB	LB	#SSBI	#Part	#nodes	CPU
DSJC125.5	F-S0	125	3891	24	19	10.5254	0	0	1084	7200
DSJC125.5	F	125	3891	24	19	10.5254	0	0	1067	7200
DSJC125.5	F-Col	125	3891	24	19	10.5254	0	0	1112	7200
DSJC125.5	F-Part	125	3891	24	19	10.4365	0	35,743	116	7200
DSJC125.5	F-Sub-S0	125	3891	24	19	10.5254	0	0	1063	7200
DSJC125.5	F-Sub	125	3891	24	19	10.5254	0	0	1084	7200
DSJC125.9	F-S0	125	6961	57	45	35.6026	0	0	164	7200
DSJC125.9	F	125	6961	57	45	35.6026	0	0	160	7200
DSJC125.9	F-Col	125	6961	57	45	35.6026	0	0	164	7200
DSJC125.9	F-Part	125	6961	57	47	36.1708	0	49,330	142	7200
DSJC125.9	F-Sub-S0	125	6961	57	46	35.9338	19,152	0	97	7200
DSJC125.9	F-Sub	125	6961	57	46	35.9338	19,152	0	94	7200
DSJC250.9	F-S0	250	27,897	100	97	35.299	0	0	0	7200
DSJC250.9	F	250	27,897	100	97	35.299	0	0	0	7200
DSJC250.9	F-Col	250	27,897	100	97	35.299	0	0	0	7200
DSJC250.9	F-Part	250	27,897	100	97	32.7273	0	911	0	7200
DSJC250.9	F-Sub-S0	250	27,897	100	86	35.9854	12,276	0	0	7200
DSJC250.9	F-Sub	250	27,897	100	86	35.9854	12,276	0	0	7200
DSJR500.5	F-S0	500	58,862	134	134	114	0	0	0	7200
DSJR500.5	F	500	58,862	134	134	114	0	0	0	7200
DSJR500.5	F-Col	500	58,862	134	134	114	0	0	0	7200
DSJR500.5	F-Part	500	58,862	134	134	114	0	45,765	0	7200
DSJR500.5	F-Sub-S0	500	58,862	134	134	117	697,452	0	0	7200
DSJR500.5	F-Sub	500	58,862	134	134	117	697,452	0	0	7200
flat300-28-0	F-S0	300	21,695	48	39	9.81756	0	0	0	7200
flat300-28-0	F	300	21,695	48	39	9.81756	0	0	0	7200
flat300-28-0	F-Col	300	21,695	48	39	9.81756	0	0	0	7200
flat300-28-0	F-Part	300	21,695	48	39	9.82062	0	10	0	7200
flat300-28-0	F-Sub-S0	300	21,695	48	39	9.81756	0	0	0	7200
flat300-28-0	F-Sub	300	21,695	48	39	9.81756	0	0	0	7200
r250.5	F-S0	250	14,849	72	68	61.3333	0	0	0	7200
r250.5	F	250	14,849	72	68	61.3333	0	0	0	7200
r250.5	F-Col	250	14,849	72	68	61.3333	0	0	0	7200
r250.5	F-Part	250	14,849	72	70	58	0	8847	0	7200
r250.5	F-Sub-S0	250	14,849	72	70	60	50,000	0	0	7200
r250.5	F-Sub	250	14,849	72	70	60	50,000	0	0	7200

Table 4 Experimental results on NP-? graph coloring DIMACS instances

	Method	n	E	K	UB	LB	#SSBI	#Part	#nodes	CPU
2-Insertions-4	F-S0	149	541	5	5	4	0	0	1.10839e+06	7200
2-Insertions-4	F	149	541	5	5	4	0	0	1.10282e+06	7200
2-Insertions-4	F-Col	149	541	5	5	4	0	0	1.1005e+06	7200
2-Insertions-4	F-Part	149	541	5	5	4	0	5156	1.5963e+06	7200
2-Insertions-4	F-Sub-S0	149	541	5	5	3	20,160	2809	100,253	7200
2-Insertions-4	F-Sub	149	541	5	5	4	20,160	2179	662,877	7200
1-FullIns-5	F-S0	282	3247	6	6	4.14286	0	0	18,896	7200
1-FullIns-5	F	282	3247	6	6	4.33333	0	0	19,353	7200
1-FullIns-5	F-Col	282	3247	6	6	4.33333	0	0	19,304	7200
1-FullIns-5	F-Part	282	3247	6	6	4	0	3801	11,511	7200
1-FullIns-5	F-Sub-S0	282	3247	6	6	5	740	0	23,787	7200
1-FullIns-5	F-Sub	282	3247	6	6	6	740	0	26,320	7139.29
3-FullIns-4	F-S0	405	3524	7	7	7	0	0	8147	4442.78
3-FullIns-4	F	405	3524	7	7	7	0	0	2898	2601.7
3-FullIns-4	F-Col	405	3524	7	7	7	0	0	2898	2598.23
3-FullIns-4	F-Part	405	3524	7	7	7	0	10,622	14,146	5012.53
3-FullIns-4	F-Sub-S0	405	3524	7	7	7	96	0	9204	5312.42
3-FullIns-4	F-Sub	405	3524	7	7	7	96	0	3895	3769.1
myciel7	F-S0	191	2360	8	8	4.22826	0	0	60,242	7200
myciel7	F	191	2360	8	8	4.22826	0	0	60,234	7200
myciel7	F-Col	191	2360	8	8	4.24913	0	0	60,364	7200
myciel7	F-Part	191	2360	8	8	4	0	15,955	10,091	7200
myciel7	F-Sub-S0	191	2360	8	8	4	5488	17,608	10,775	7200
myciel7	F-Sub	191	2360	8	8	5	5488	18,330	16,524	7200
r125.5	F-S0	125	3838	36	36	36	0	0	0	259.37
r125.5	F	125	3838	36	36	36	0	0	0	259.02
r125.5	F-Col	125	3838	36	36	36	0	0	0	257.74
r125.5	F-Part	125	3838	36	36	36	0	9962	0	999.33
r125.5	F-Sub-S0	125	3838	36	36	36	50,000	0	0	170.66
r125.5	F-Sub	125	3838	36	36	36	50,000	0	0	170.93
flat300-20-0	F-S0	300	21,375	38	29	10.0961	0	0	0	7200
flat300-20-0	F	300	21,375	38	29	10.0961	0	0	0	7200
flat300-20-0	F-Col	300	21,375	38	29	10.0961	0	0	0	7200
flat300-20-0	F-Part	300	21,375	38	29	9.98414	0	1179	0	7200
flat300-20-0	F-Sub-S0	300	21,375	38	29	10.0961	0	0	0	7200
flat300-20-0	F-Sub	300	21,375	38	29	10.0961	0	0	0	7200
flat300-26-0	F-S0	300	21,633	39	37	10.0629	0	0	0	7200
flat300-26-0	F	300	21,633	39	37	10.0629	0	0	0	7200
flat300-26-0	F-Col	300	21,633	39	37	10.0629	0	0	0	7200
flat300-26-0	F-Part	300	21,633	39	37	10.0624	0	25	0	7200
flat300-26-0	F-Sub-S0	300	21,633	39	37	10.0629	0	0	0	7200
flat300-26-0	F-Sub	300	21,633	39	37	10.0629	0	0	0	7200

All in all on DIMACS instances, sub-symmetry breaking appears to significantly improve on only symmetry breaking.

5 Application to the unit commitment problem

The framework of Sect. 2 is now applied to the Unit Commitment Problem, which features many sub-symmetries undetected by symmetry group $\mathcal{G}$.

Given a discrete time horizon $\mathcal{T} = \{1, \ldots, T\}$, a demand for electric power D_t is to be met at each time period $t \in \mathcal{T}$. Power is provided by a set $\mathcal{N}$ of n production units. At each time period, unit $j \in \mathcal{N}$ is either down or up, and in the latter case, its production is within $[P^j_{min}, P^j_{max}]$. Each unit must satisfy minimum up-time (resp. down-time) constraints, i.e., it must remain up (resp. down) during at least L^j (resp. ℓ^j) periods after start up (resp. shut down). Each unit j also features three different costs: a fixed cost c^j_f, incurred each time period the unit is up; a start-up cost c^j_0, incurred each time the unit starts up; and a cost c^j_p proportional to its production. The Min-up/min-down Unit Commitment Problem (MUCP) is to find a production plan minimizing the total cost while satisfying the demand and the minimum up and down time constraints. The MUCP is strongly NP-hard [3].

In the real-world Unit Commitment Problem (UCP), some more technical constraints have also to be taken into account, such as ramp constraints or reserve requirement constraints, and the start-up costs are an exponential function of the unit downtime. From a combinatorial point of view, the MUCP is the core structure of the UCP. In this section, we study the MUCP with and without ramp constraints.

For each unit $j \in \mathcal{N}$ and time period $t \in \mathcal{T}$, let us consider three variables: $x_{t,j} \in \{0, 1\}$ indicates if unit j is up at time t; $u_{t,j} \in \{0, 1\}$ whether unit j starts up at time t; and $p_{t,j} \in \mathbb{R}$ is the quantity of power produced by unit j at time t. Without loss of generality we consider that $L^j, \ell^j \leq T$. Formulation $F(x, u)$ for the MUCP is as follows [2,31,35].

$$\min_{x,u,p} \quad \sum_{j=1}^{n} \sum_{t=1}^{T} c^j_f x_{t,j} + c^j_p p_{t,j} + c^j_0 u_{t,j}$$

$$\text{s. t.} \quad \sum_{t'=t-L^j+1}^{t} u_{t',j} \leq x_{t,j} \quad \forall j \in \mathcal{N}, \quad \forall t \in \{L^j, \ldots, T\} \tag{5.1}$$

$$\sum_{t'=t-\ell^j+1}^{t} u_{t',j} \leq 1 - x_{t-\ell^j,j} \quad \forall j \in \mathcal{N}, \quad \forall t \in \{\ell^j, \ldots, T\} \tag{5.2}$$

$$u_{t,j} \geq x_{t,j} - x_{t-1,j} \quad \forall j \in \mathcal{N}, \quad \forall t \in \{2, \ldots, T\} \tag{5.3}$$

$$P^j_{min} x_{t,j} \leq p_{t,j} \leq P^j_{max} x_{t,j} \quad \forall j \in \mathcal{N}, \quad \forall t \in \mathcal{T} \tag{5.4}$$

$$\sum_{j=1}^{n} p_{t,j} \geq D_t \quad \forall t \in \mathcal{T} \tag{5.5}$$

$$x_{t,j}, u_{t,j} \in \{0, 1\} \quad \forall j \in \mathcal{N}, \quad \forall t \in \mathcal{T} \tag{5.6}$$

For convenience, we will also use variable $w_{t,j} = x_{t-1,j} - x_{t,j} + u_{t,j}$, indicating whether unit j shuts down at time t.

5.1 Symmetries and sub-symmetries in the UCP

Symmetries in the MUCP (and in the UCP) arise from the existence of groups of identical units, i.e., units with identical characteristics (P_{min}, P_{max}, L, ℓ, c_f, c_0, c_p). The instance is partitioned into *types* $h \in \{1, \ldots, H\}$ of n_h identical units. The unit set of type h is denoted by $\mathcal{N}_h = \{j_1^h, \ldots, j_{n_h}^h\}$.

The solutions of the MUCP can be expressed as a series of binary matrices. For a given type h, we introduce matrix $x^h \in \mathcal{P}(T, n_h)$ such that entry $x_{t,k}^h$ corresponds to variable x_{t,j_k^h}, where j_k^h is the index of the k^{th} unit of type h, $k \in \{1, \ldots, n_h\}$. Column j of matrix x^h corresponds to the up/down plan relative to the j^{th} unit of type h. Similarly, we introduce matrices u^h and p^h.

The set of all feasible $x = (x_{t,j})_{t \in \mathcal{T}, j \in \mathcal{N}}$ is denoted by $\mathcal{X}_{MUCP}$. Note that any solution matrix x (resp. u, p) can be partitioned in H matrices x^h (resp. u^h, p^h). Since all units of type h are identical, their production plans can be permuted, provided that the same permutation is applied to matrices x^h, u^h and p^h. Thus, the symmetry group $\mathcal{G}$ contains the symmetric group $\mathfrak{S}_{n_h}$ acting on the columns of x^h, for each unit type h. Consequently, for each type h, feasible solutions x^h can be restricted to be in the $T \times n_h$ full orbitope. As binary variables u are uniquely determined by variables x, breaking the symmetry on x variables will break the symmetry on u variables. Note that this restriction to the $T \times n_h$ full orbitope for each type h can possibly be done using inequalities from Sect. 3 featuring z variables.

There are also other sources of symmetry, arising from the possibility of permuting some sub-columns of matrices x^h. For example, consider two identical units. Suppose at some time period t, these two units are down and ready to start up. Then their plans after t can be permuted, even if they do not have the same up/down plan before t.

More precisely, a unit $j \in \mathcal{N}$ is *ready to start up* at time $t \in \{1, \ldots, T\}$ if and only if $\forall t' \in \{t - \ell^j, \ldots, t-1\}$, $x_{t',j} = 0$. Similarly, a unit $j \in \mathcal{N}^k$ is *ready to shut down* at time $t \in \{1, \ldots, T\}$ if and only if $\forall t' \in \{t - L^j, \ldots, t-1\}$, $x_{t',j} = 1$.

5.2 Sub-symmetry-breaking inequalities for the MUCP

For each time period $t \in \{1, \ldots, \mathcal{T}\}$ and any two consecutive units j_k^h, j_{k+1}^h of type h, $k \in \{1, \ldots, n_h - 1\}$, consider the following subsets of $\mathcal{X}_{MUCP}$:

$$\begin{aligned}
\check{Q}_{k,h}^t = \big\{ & x \in \mathcal{X}_{MUCP} \mid x_{t',j} = 0, \quad \forall t' \in \{t - \ell^h, \ldots, t-1\}, \\
& t \geq \ell^h + 1, \quad \forall j \in \{j_k^h, j_{k+1}^h\} \big\} \\
\widehat{Q}_{k,h}^t = \big\{ & x \in \mathcal{X}_{MUCP} \mid x_{t',j} = 1, \quad \forall t' \in \{t - L^h, \ldots, t-1\}, \\
& t \geq L^h + 1, \quad \forall j \in \{j_k^h, j_{k+1}^h\} \big\}
\end{aligned}$$

where ℓ^h (resp. L^h) is the minimum down (resp. up) time of units of type h.

Note that $\check{Q}^t_{k,h}$ and $\widehat{Q}^t_{k,h}$ are different from subsets $Q_{i,j}$ defined in Sect. 3. Actually, $Q_{t,j^h_{k+1}} \subset \check{Q}^t_{k,h}$ and $Q_{t,j^h_{k+1}} \subset \widehat{Q}^t_{k,h}$.

Let $\mathcal{G}_{\check{Q}^t_{k,h}}$ and $\mathcal{G}_{\widehat{Q}^t_{k,h}}$ be the sub-symmetry groups associated to $\check{Q}^t_{k,h}$ and $\widehat{Q}^t_{k,h}$, $t \in \{1, \ldots, \mathcal{T}\}$, $h \in \{1, \ldots, H\}$, $k \in \{1, \ldots, n_h - 1\}$. The sub-symmetries in $\mathcal{G}_{\check{Q}^t_{k,h}}$ (resp. $\mathcal{G}_{\widehat{Q}^t_{k,h}}$) are called *start-up sub-symmetries* (resp. *shut-down sub-symmetries*). Most of these sub-symmetries are not detected in the symmetry group of the MUCP.

Groups $\mathcal{G}_{\check{Q}^t_{k,h}}$ and $\mathcal{G}_{\widehat{Q}^t_{k,h}}$ contain the sub-symmetric group associated to the submatrix defined by rows and columns $(\{t, \ldots, T\}, \{j^h_k, j^h_{k+1}\})$.

Applying results from Sect. 2, variables $\check{z}^t_{k,h}$ and $\widehat{z}^t_{k,h}$, indicating whether $x \in \check{Q}^t_{k,h}$ and $x \in \widehat{Q}^t_{k,h}$ respectively, can be directly derived from variables x and u:

$$\check{z}^t_{k,h} = x_{t-\ell^h,j'} + \sum_{t'=t-\ell^h+1}^{t-1} u_{t',j'} + x_{t-\ell^h,j} + \sum_{t'=t-\ell^h+1}^{t-1} u_{t',j} \quad t \geq \ell^h + 1$$

$$\widehat{z}^t_{k,h} = 1 - x_{t-L^h,j'} + \sum_{t'=t-L^h+1}^{t-1} w_{t',j'} + 1 - x_{t-L^h,j} + \sum_{t'=t-L^h+1}^{t-1} w_{t',j} \quad t \geq L^h + 1$$

where $j = j^h_k$ and $j' = j^h_{k+1}$ for sake of clarity.

Consider $\mathbb{S}_{MUCP} = \left\{\check{Q}^t_{k,h},\ \widehat{Q}^t_{k,h},\ t \in \{1, \ldots, \mathcal{T}\},\ h \in \{1, \ldots, H\},\ k \in \{1, \ldots, n_h - 1\}\right\}$. In this case, set $\mathbb{S}_{MUCP}$ directly satisfies condition $\mathcal{C}$. Note that $|\mathbb{S}_{MUCP}| = O(2Tn)$ thus leading to $O(2Tn)$ inequalities.

For each $h \in \{1, \ldots, H\}$, $k \in \{1, \ldots, n_h - 1\}$ and $t \in \{1, \ldots, \mathcal{T}\}$, inequalities $(\check{Q}^t_{k,h}(j, j'))$ and $(\widehat{Q}^t_{k,h}(j, j'))$, where $j = j^h_k$ and $j' = j^h_{k+1}$, are as follows.

$$\begin{aligned} x_{t,j'} \leq & \left[x_{t-\ell^h,j'} + \sum_{t'=t-\ell^h+1}^{t-1} u_{t',j'}\right] \\ & + \left[x_{t-\ell^h,j} + \sum_{t'=t-\ell^h+1}^{t-1} u_{t',j}\right] + x_{t,j} \quad t \geq \ell^h + 1 \\ x_{t,j'} \leq & \left[1 - x_{t-L^h,j'} + \sum_{t'=t-L^h+1}^{t-1} w_{t',j'}\right] \\ & + \left[1 - x_{t-L^h,j} + \sum_{t'=t-L^h+1}^{t-1} w_{t',j}\right] + x_{t,j} \quad t \geq L^h + 1 \end{aligned}$$

Strengthening symmetry-breaking inequalities Inequalities $(\check{Q}^t_{k,h}(j, j'))$ and $(\widehat{Q}^t_{k,h}(j, j'))$ can be further strengthened, using the relationship between variables x and u.

First note that by definition of variables w:

$$x_{t,j'} - \left[x_{t-\ell^h,j'} + \sum_{t'=t-\ell^h+1}^{t-1} u_{t',j'}\right] = u_{t,j'} - \sum_{t'=t-\ell^h+1}^{t} w_{t',j'} \quad t \geq \ell^h + 1$$

$$x_{t,j} + \left[1 - x_{t-L^h,j} + \sum_{t'=t-L^h+1}^{t-1} w_{t',j}\right] = -w_{t,j} + 1$$
$$+ \sum_{t'=t-L^h+1}^{t} u_{t',j} \quad t \geq L^h + 1$$

Note that if $u_{t,j'} = 1$ (resp. $w_{t,j} = 1$), then $\sum_{t'=t-\ell^h+1}^{t} w_{t',j'} = 0$ (resp. $\sum_{t'=t-L^h+1}^{t} u_{t',j} = 0$). Replacing the previous two equalities into inequalities ($\check{Q}^t_{k,h}(j, j')$) and ($\widehat{Q}^t_{k,h}(j, j')$) yields the following valid and stronger *Start-Up-Ready* and *Shut-Down-Ready* inequalities.

$$u_{t,j'} \leq \left[x_{t-\ell^h,j} + \sum_{t'=t-\ell^h+1}^{t-1} u_{t',j}\right] + x_{t,j} \quad t \geq \ell^h + 1 \tag{5.7}$$

$$w_{t,j} \leq \left[1 - x_{t-L^h,j'} + \sum_{t'=t-L^h+1}^{t-1} w_{t',j'}\right] + 1 - x_{t,j'} \quad t \geq L^h + 1 \tag{5.8}$$

Note that for any $h \in \{1, \dots, H\}$ and $k \in \{1, \dots, n_h - 1\}$, $\check{Q}^1_{k,h} = \widehat{Q}^1_{k,h} = \mathcal{X}_{MUCP}$. As condition (C) is satisfied by $\mathbb{S}_{MUCP}$, any $x = (x_1, \dots, x_H)$ satisfying inequalities (5.7) and (5.8) is such that x_h is in the $T \times n_h$ full orbitope, $h \in \{1, \dots, H\}$. Hence inequalities (5.7) and (5.8) ensure in particular that any solution x_h is in the full orbitope.

5.3 Sub-symmetry-breaking inequalities for the ramp-constrained MUCP

In the real-world UCP, each unit j must also feature *ramp-up* (resp. *ramp-down*) constraints, i.e., the maximum increase (resp. decrease) in generated power from time period t to time period $t+1$ is RU^j (resp. RD^j). Moreover, if unit i starts up at time t (resp. shuts down at time $t+1$), its production at time t cannot be higher than SU^j (resp. SD^j).

For each unit $j \in \mathcal{N}$ and time period $t \in \{2, \dots T\}$, ramp constraints can be formulated as follows:

$$p_{t,j} - p_{t-1,j} \leq RU^j x_{t-1,j} + SU^j u_{t,j} \tag{5.9}$$
$$p_{t-1,j} - p_{t,j} \leq RD^j x_{t,j} + SD^j w_{t,j} \tag{5.10}$$

The MUCP formulation including ramp constraints can be further strengthened with valid inequalities as proposed in [32,34]. As the aim of this article is to compare symmetry-breaking techniques, we will only consider the classical MUCP formulation (5.1) – (5.6) with ramp-constraints (5.9) – (5.10), as done in [21,31].

When ramp-constraints are considered, the symmetry group of set $\check{Q}^t_{k,h}$ still contains the sub-symmetric group associated to the submatrix defined by rows and columns $(\{t, \dots, T\}, \{j^h_k, j^h_{k+1}\})$. Therefore, inequalities (5.7) can still be used.

However the symmetry group of set $\widehat{Q}^t_{k,h}$ no longer contains the sub-symmetric group associated to the submatrix defined by rows and columns $(\{t, \dots, T\}, \{j^h_k, j^h_{k+1}\})$. Indeed, if two identical units have been up for at least L^h time periods at time $t-1$, they may produce distinct power values at time $t-1$ and thus, because of ramp constraints, their up/down trajectories from time t to T cannot be permuted. Therefore, inequalities (5.8) can no longer be used.

Note that when two identical ramp-constrained units are ready to shut down, there still exist some sub-symmetries that could be exploited. These sub-symmetries are more intricate because they depend, for example, on the quantity of power produced by both units, or on the time of their last start-up.

5.4 Experimental results

In this section, we compare various (sub-)symmetry-breaking techniques for the MUCP with or without ramp constraints. Some of these techniques operate during the branching process, while the others are compact or exponential symmetry-breaking MIP formulations.

5.4.1 Aggregated formulations for the UCP

In [22], the authors propose to break symmetries of the UCP by aggregating variables corresponding to identical units. This method is shown to outperform existing symmetry-breaking inequalities whenever the *integer decomposition property* holds [1], i.e., any integer solution of the aggregated formulation can be disaggregated into an integer solution of the disaggregated formulation.

• *Aggregated (x, u) formulation* In the case of the MUCP, variables x, u of formulation (5.1–5.6) are aggregated into variables $\widetilde{x}_{t,h} = \sum_{j \in \mathcal{N}_h} x_{t,j} \in \{0, \dots, n_h\}$ (resp. $\widetilde{u}_{t,h} = \sum_{j \in \mathcal{N}_h} u_{t,j} \in \{0, \dots, n_h\}$) indicating how many units of type h are up (resp. start up) at time t. Variables $\widetilde{p}_{t,h} = \sum_{j \in \mathcal{N}_h} p_{t,j} \in \mathbb{R}$ is the total amount of power produced at time t by units of type h. Aggregated (x, u) formulation, denoted by A-$(\widetilde{x}, \widetilde{u})$, is formulation F$(x, u)$ where variables (x, u, p) are replaced by $(\widetilde{x}, \widetilde{u}, \widetilde{p})$.

When aggregating variables corresponding to h identical units, one must ensure that the aggregated production plan can be disaggregated into h feasible production plans satisfying. Inequalities (5.1)–(5.4) have the integer decomposition property, i.e., any integer solution $(\widetilde{x}, \widetilde{u}, \widetilde{p})$ of aggregated (x, u) formulation can be disaggregated into an integer solution (x, u, p) of formulation (5.1)–(5.6). A disaggregation algorithm for the MUCP is proposed in [22].

When ramp constraints are considered in formulation (5.1)–(5.6), the integer decomposition property is lost. Examples of aggregated solutions which cannot be disaggregated are given in [22].

• *Aggregated interval formulation* As the integer decomposition property depends on the formulation considered, an interval-based formulation is introduced in [22] for the ramp-constrained MUCP. For each unit $j \in \mathcal{N}$, for each interval $\{t_0, \ldots, t_1 - 1\}$ of size $t_1 - t_0 \geq L^j$, variable $y_j^{t_0,t_1} = 1$ if and only if unit j starts up at time t_0, remains up on interval $\{t_0, \ldots, t_1 - 1\}$ and shuts down at time t_1. For each time period $t \in \mathcal{T}$, variable $p_{t,j}^{t_0,t_1}$ represents the quantity of power produced by unit j at time t if $y_j^{t_0,t_1} = 1$, and $p_{t,j}^{t_0,t_1} = 0$ otherwise. To each interval $\{t_0, \ldots, t_1 - 1\}$ is associated a production polytope giving the feasible domain of variable $p_{t,j}^{t_0,t_1}$.

The interval formulation consists in finding, for each unit i, a set of compatible intervals (i.e., non-overlapping intervals such that the minimum down time is satisfied) in order to satisfy the demand. Such a formulation has the integer decomposition property, thus variables $y_j^{t_0,t_1}$ (resp. $p_{t,j}^{t_0,t_1}$) can be replaced by aggregated variables $\widetilde{y}_h^{t_0,t_1} = \sum_{j \in \mathcal{N}_h} y_j^{t_0,t_1}$ and $\widetilde{p}_{t,h}^{t_0,t_1} = \sum j \in \mathcal{N}_h \, p_{t,j}^{t_0,t_1}$, leading to aggregated formulation $\mathrm{Int}(\widetilde{y})$.

5.4.2 Modified orbital branching

In [31] the authors present the Modified Orbital Branching (MOB) technique which operates at each node of the branching tree. The idea is to branch on a subset of symmetric variables instead of a single one. They apply MOB alongside with several complementary branching rules to break symmetries of the MUCP with additional technical constraints. Among the proposed branching rules, the most flexible one ensuring full-symmetry breaking is called Relaxed Minimum-Rank Index (RMRI). Note that sub-symmetries are not exploited in practice. Different approaches are compared experimentally: Default Cplex, Callback Cplex, OB (orbital branching), MOB with no branching rules enforced (Cplex is free to choose the next branching variable), and MOB with RMRI. It is shown that MOB with RMRI is more efficient than MOB, OB and Callback Cplex in terms of CPU time. The difference of (geometric) average CPU time speed-up between using MOB with RMRI and MOB alone is 1.098.

In this paper, we choose to compare our methods to MOB, even though MOB with RMRI is shown to perform slightly better than MOB with no branching rules [31]. The rationale behind is that its implementation is straightforward, thus leaving no room to interpretation.

5.4.3 Orbitopal fixing for the full orbitope

In [4], a variable fixing algorithm, called *Orbitopal fixing for the full (sub-)orbitope,* is proposed in order to enumerate only solutions in full (sub-)orbitopes from the B&B tree. A dynamic version of the orbitopal fixing algorithm is proposed, where the lexicographical order at node a is defined with respect to the branching decisions leading to a. Experimental results on MUCP instances show that the dynamic variant of the algorithm performs much better than the static variant. Moreover, it is clear that sub-symmetries greatly impair the solution process for MUCP instances, since dynamic orbitopal fixing for both full orbitope and full sub-orbitope performs even better than dynamic orbitopal fixing for the full orbitope. The experiments show also that the approach is competitive with commercial solvers like Cplex and state-of-the-art techniques like MOB.

5.4.4 Experimental settings

In this section, we compare various symmetry-breaking formulations for the MUCP with or without ramp constraints.

Each experiment is carried out using Cplex 12.8 C++ API on only 1 thread of a cluster node with a 64 bit Intel Xeon CPU E5-2697 v3 processor running at 2.6 GHz, and 64 GB of RAM memory. The UCP instances are solved until optimality (defined within 10^{-7} of relative optimality tolerance) or until the time limit of 3600 seconds is reached.

As shown in [31], neither Friedman inequalities (1.1) nor column inequalities (1.2) are competitive with respect to the classical UCP formulation when solved by Cplex.

On the opposite, the weaker form of Friedman inequality (1.3) has been shown in [25] to outperform Default Cplex.

Hence the following symmetry-breaking techniques are compared:

- $F(x, u)$ (x, u)-formulation (5.1)–(5.6)
- MOB Modified Orbital Branching
- Fixing Dynamic orbitopal fixing for the full (sub-)orbitope
- A-$(\widetilde{x}, \widetilde{u})$ Aggregated $(\widetilde{x}, \widetilde{u})$-formulation (only when disaggregation applies)
- Int$(\widetilde{y})$ Aggregated interval formulation
- $W(x, u)$ (x, u)-formulation (5.1)–(5.6) with weaker Friedman inequalities (1.3)
- $F(x, u, z)$ (x, u)-formulation (5.1)–(5.6) with variables z, inequalities (3.1a)–(3.1e) and sub-symmetry-breaking inequalities (3.3)
- $LF(x, u)$ (x, u)-formulation (5.1)–(5.6) with sub-symmetry-breaking inequalities (5.7)-(5.8).

Symmetry-breaking techniques MOB and Fixing are implemented within Cplex C++ API using the BranchCallback feature.

As for graph coloring experiments, we also include experiments where Cplex's internal symmetry-breaking techniques are turned off. We deactivate the latter techniques for $F(x, u)$, $W(x, u)$, $F(x, u, z)$ and $LF(x, u)$, which are respectively denoted by $F(x, u)$-S0, $W(x, u)$-S0, $F(x, u, z)$-S0 and $LF(x, u)$-S0. Due to the use of Cplex BranchCallback in MOB and Fixing, the Cplex's internal symmetry-breaking techniques are already turned off. For A-$(\widetilde{x}, \widetilde{u})$ and Int$(\widetilde{y})$, no change has been detected with or without the latter techniques. It does not appear useful to include their S0 variant in the tables.

Formulation $F(x, u, z)$ is obtained from the classical MUCP formulation $F(x, u)$ by a direct use of the inequalities given in Sect. 3. As seen in Sect. 5, taking into account sub-symmetries in the MUCP leads to formulation $LF(x, u)$ featuring lifted symmetry breaking-inequalities (5.7) and (5.8), namely Start-up-ready and Shut-down-ready inequalities, in place of inequalities (3.1a)–(3.3). Note that the start-up and shut-down sub-symmetries of the MUCP are not handled by formulations $F(x, u)$, $W(x, u)$ and $F(x, u, z)$.

Table 5 Instance characteristics

Size (n, T)	Sym. factor	Nb singl.	Nb groups	Av. group size	Group max. size
(20,48)	$F = 3$	1.25	4.90	3.96	5.75
	$F = 2$	0.75	3.20	6.45	8.75
(20,96)	$F = 3$	0.90	4.75	4.08	5.60
	$F = 2$	0.75	3.45	5.93	8.65
(30,48)	$F = 3$	1.10	5.35	5.51	9.45
	$F = 2$	0.25	3.85	8.30	12.60
(30,96)	$F = 3$	0.40	5.25	5.97	8.65
	$F = 2$	0.55	4.05	7.59	11.40
(60,48)	$F = 4$	0.80	7.70	7.86	13.20
	$F = 3$	0.55	5.80	10.90	17.80
	$F = 2$	0.20	4.75	13.90	23.80
(60,96)	$F = 4$	0.60	7.90	7.79	13.20
	$F = 3$	0.30	5.95	10.50	16.60
	$F = 2$	0.20	4.35	14.80	24.90

Formulations $F(x, u)$, $W(x, u)$, $F(x, u, z)$ and $LF(x, u)$ feature $O(nT)$ variables while formulation A-$(\widetilde{x}, \widetilde{u})$ (resp. Int$(\widetilde{y})$) features $O(HT)$ (resp. $O(T^2 H)$) variables, where H is the number of groups of identical units.

For the ramp-constrained MUCP, inequalities (5.9)–(5.10) enforcing ramp constraints are added to formulations $F(x, u)$, $W(x, u)$, $F(x, u, y)$ and $LF(x, u)$. Aggregated formulation A-$(\widetilde{x}, \widetilde{u})$ can no longer be used, as its solutions cannot be disaggregated [22]. Note also that in this context, Start-up-ready inequalities are adjoined to $LF(x, u)$, but Shut-down-ready inequalities cannot.

5.4.5 Instances

We generate MUCP instances as follows.

For each instance, we generate a "2-peak per day" type demand with a large variation between peak and off-peak values: during one day, the typical demand in energy has two peak periods, one in the morning and one in the evening. The amplitudes between peak and off-peak periods have similar characteristics to those in the dataset from [9].

We consider the parameters $(P_{min}, P_{max}, L, \ell, c_f, c_0, c_p)$ of each unit from the dataset presented in [9]. We draw a correlation matrix between these characteristics and define a possible range for each characteristic. In order to introduce symmetries in our instances, some units are randomly generated based on the parameters correlations and ranges. Each unit generated is duplicated d times, where d is randomly selected in $[1, \frac{n}{F}]$ in order to obtain a total of n units. The parameter F is called symmetry factor, and can vary from 2 to 4 depending on the value of n. Note that these instances are generated along the same lines as literature instances considered in [2], but with different F factors.

In order to determine which symmetry-breaking technique performs best with respect to the number of rows and columns of matrices in feasible set $\mathcal{X}$, we consider various instance sizes $n \in \{20, 30, 60\}$ and $T \in \{48, 60\}$, and various symmetry factors $F \in \{2, 3, 4\}$. For each size (n, T) and symmetry factor F, we generate a set of 20 instances. Symmetry factor $F = 4$ is not considered for instances with a small number n of units ($n = 20$ or 30), as it leads to very small sets of identical units.

Table 5 provides some statistics on the instances characteristics. For each instance, a group is a set of two or more units with the same characteristics. Each unit which has not been duplicated is a singleton. The first and second entries column-wise are the number of singletons and groups. The third entry is the average group size and the fourth entry is the maximum group size. Each entry row-wise corresponds to the average value obtained over 20 instances with same size (n, T) and same symmetry factor F.

The ramp-constrained MUCP instances considered are the same as in the non-ramp-constrained case, with additional ramp characteristics $RU^j = \frac{P^j_{max} - P^j_{min}}{3}$, $RD^j = \frac{P^j_{max} - P^j_{min}}{2}$ and $SU^j = SD^j = P^j_{min}$.

5.4.6 Results for the non-ramp-constrained MUCP

Tables 6 and 7 provide, for each formulation and each considered group of 20 instances:

- #opt: Number of instances solved to optimality,
- #nodes: Average number of nodes,
- gap: Average optimality gap,
- CPU time: Average CPU time in seconds.

Performance results reported in Tables 6 and 7 are relative to large instances, i.e., with $(n, T) = (60, 48)$ and $(n, T) = (60, 96)$. Smaller instances for the non-ramp-constrained MUCP can be solved quite efficiently, thus making the comparison of the different techniques in terms of performance not meaningful. The corresponding results are not reported in the enclosed tables.

From Tables 6 and 7, two extreme cases stand out of the comparison. On the one hand, aggregated (x, u) formulation A-$(\widetilde{x}, \widetilde{u})$ outperforms by far all the other techniques. This could be explained by the reduced size of aggregated formulation A-$(\widetilde{x}, \widetilde{u})$, but also by the good performance of Cplex on ILP featuring integer variables (with bounds greater than 1). This efficiency will certainly be preserved any time the integer decomposition property holds for an (x, u) formulation of the UCP. On the other hand, aggregated interval formulation Int$(\widetilde{y})$ is in average one or even two orders of magnitude slower than the other techniques, except $F(x, u)$-S0.

Obviously turning off Cplex's symmetry-breaking techniques in $F(x, u)$-S0 leads to poor performance compared to that obtained with $F(x, u)$. As no symmetries at all are handled in $F(x, u)$-S0, this highlights the impact symmetries can have on the difficulty to solve an MUCP instance. For other techniques, the "S0" variant does not seem worse than the original. On the contrary, on many instances, deactivating Cplex's internal symmetry-breaking techniques helps improving the computational time for many techniques. It is the case for example on $(n, T) = (60, 48)$, $F = 4$

Table 6 Performance indicators relative to the comparison of symmetry-breaking techniques for (non-ramp-constrained) MUCP instances with $(n, T) = (60, 48)$

Instances		Formulation	#opt	#nodes	Gap (%)	CPU time
(60,48)	$F = 2$	$F(x, u)$-S0	14	603,627	0.00022	1121.7
		$F(x, u)$	19	77,937	0.00003	270.3
		MOB	16	164,474	0.00026	739.1
		Fixing	20	36,028	0	198.2
		A-(x, u)	20	89	0	0.1
		A-Int$(\widetilde{y})$	2	857,402	0.00945	3279.0
		$W(x, u)$-S0	17	79,972	0.00005	754.1
		$W(x, u)$	17	124,718	0.00007	797.0
		$F(x, u, z)$-S0	20	7441	0	361.8
		$F(x, u, z)$	20	12,631	0	426.7
		$LF(x, u)$-S0	17	197,580	0.00014	593.1
		$LF(x, u)$	17	186,902	0.00014	606.6
	$F = 3$	$F(x, u)$-S0	14	951,079	0.00046	1122.4
		$F(x, u)$	17	412,333	0.00005	545.2
		MOB	16	150,251	0.00021	730.5
		Fixing	19	49,243	0.00030	279.3
		A-(x, u)	20	16	0	0.1
		A-Int$(\widetilde{y})$	5	469,685	0.01063	2758.6
		$W(x, u)$-S0	18	42,198	0.00007	380.9
		$W(x, u)$	18	45,432	0.00011	381.5
		$F(x, u, z)$-S0	19	24,024	0.00005	379.7
		$F(x, u, z)$	19	19,251	0.00004	330.5
		$LF(x, u)$-S0	20	15,810	0	72.6
		$LF(x, u)$	20	10,984	0	67.6
	$F = 4$	$F(x, u)$-S0	15	936,362	0.00014	1045.4
		$F(x, u)$	19	199,432	0.00002	248.5
		MOB	15	217,980	0.00024	909.4
		Fixing	19	50,095	0	202.2
		A-(x, u)	20	85	0	0.1
		A-Int$(\widetilde{y})$	8	628,345	0.00985	2343.6
		$W(x, u)$-S0	20	11,744	0	58.0
		$W(x, u)$	20	79,876	0	209.5
		$F(x, u, z)$-S0	19	53,702	0.00001	430.7
		$F(x, u, z)$	19	40,117	0.00003	424.9
		$LF(x, u)$-S0	20	3304	0	29.5
		$LF(x, u)$	20	25,381	0	61.3

Table 7 Performance indicators relative to the comparison of symmetry-breaking techniques for (non-ramp-constrained) MUCP instances with $(n, T) = (60, 96)$

Instances		Formulation	#opt	#nodes	Gap (%)	CPU time
(60,96)	$F = 2$	$F(x, u)$-S0	13	463,340	0.00016	1292.1
		$F(x, u)$	15	377,968	0.00015	1015.2
		MOB	7	303,871	0.00068	2453.6
		Fixing	9	242,482	0.00032	1992.3
		A-(x, u)	20	0	0	0.1
		A-Int$(\widetilde{y})$	2	253,361	10.01218	3321.5
		$W(x, u)$-S0	8	288,736	0.00040	2374.5
		$W(x, u)$	11	218,720	0.00035	1914.0
		$F(x, u, z)$-S0	7	126,002	0.00044	2478.8
		$F(x, u, z)$	8	133,994	0.00036	2421.9
		$LF(x, u)$-S0	17	28,853	0.00007	807.4
		$LF(x, u)$	18	17,334	0.00006	728.8
	$F = 3$	$F(x, u)$-S0	9	1,199,784	0.00061	2033.8
		$F(x, u)$	13	878,444	0.00049	1486.7
		MOB	3	388,267	0.00162	3134.9
		Fixing	6	316,278	0.00105	2558.1
		A-(x, u)	20	101	0	0.2
		A-Int$(\widetilde{y})$	2	280,725	15.01051	3453.1
		$W(x, u)$-S0	5	423,153	0.00099	2760.6
		$W(x, u)$	5	478,280	0.00084	2827.0
		$F(x, u, z)$-S0	6	143,152	0.00120	2732.3
		$F(x, u, z)$	5	192,726	0.00100	2775.8
		$LF(x, u)$-S0	19	47,209	0.00014	335.4
		$LF(x, u)$	19	39,414	0.00005	324.1
	$F = 4$	$F(x, u)$-S0	10	1,654,635	0.00028	1901.0
		$F(x, u)$	15	444,663	0.00022	969.2
		MOB	1	439,323	0.00171	3420.6
		Fixing	6	367,577	0.00083	2856.2
		A-(x, u)	20	54	0	0.3
		A-Int$(\widetilde{y})$	0	17,663	60.54249	3600.0
		$W(x, u)$-S0	6	612,729	0.00055	2602.7
		$W(x, u)$	4	645,343	0.00063	2903.6
		$F(x, u, z)$-S0	8	146,496	0.00081	2343.5
		$F(x, u, z)$	9	137,258	0.00071	2275.3
		$LF(x, u)$-S0	20	3249	0	83.3
		$LF(x, u)$	20	3640	0	90.7

Table 8 For each triplet (n, T, F), comparison of formulations $F(x, u)$ and $LF(x, u)$ on the subset of non-ramp-constrained MUCP instances that are solved within the time limit in both settings

Instances		# I	CPU $F(x, u)\|I$	CPU $LF(x, u)\|I$
(60,48)	$F = 2$	17	105.7	78.3
	$F = 3$	17	6.1	71
	$F = 4$	19	72.1	35.9
(60,96)	$F = 2$	15	153.6	256.9
	$F = 3$	13	348.8	71.6
	$F = 4$	15	92.3	34.9

instances, where $W(x, u)$-S0 (resp. $LF(x, u)$-S0) CPU time is 58 (resp. 29) seconds on average while $W(x, u)$'s (resp. $LF(x, u)$'s) is 209 (resp. 61) seconds. It shows that the impact of Cplex's internal symmetry-breaking techniques is limited compared to the time it takes to detect and handle symmetries in these cases. On the most symmetric instances $(n, T) = (60, 96)$, $F = 2$, however, $W(x, u)$-S0 performs not as well (2374 seconds) as $W(x, u)$ (1914 seconds). In this case, Cplex's internal symmetry-breaking techniques come in useful to compensate for $W(x, u)$ being only partial-symmetry-breaking. For the other techniques, the S0 variant does not seem to significantly impact the CPU time on these instances.

On very symmetric $(n, T) = (60, 48)$ instances, i.e., when $F = 2$, the second best technique after aggregation is Fixing, which solves all instances to optimality with the second best CPU time. On less symmetric $(n, T) = (60, 48)$ instances, i.e., when $F = 3$ (resp. $F = 4$), the second best technique after aggregation is $LF(x, u)$ (resp. $LF(x, u)$-S0), which solves all instances to optimality with the second best CPU time.

On $(n, T) = (60, 96)$ instances, the second best technique after aggregation is clearly $LF(x, u)$ as it solves many more instances to optimality compared to any other techniques.

In order to complete the results provided in Tables 6 and 7, Table 8 provides, for each considered size (n, T) and factor F, average results for subset I of instances on which both $F(x, u)$ and $LF(x, u)$ terminate within time limit:

$\#I$	cardinality of instance subset I
CPU $F(x, u)\|I$	average CPU time of $F(x, u)$, in seconds, on instance subset I
CPU $LF(x, u)\|I$	average CPU time of $LF(x, u)$, in seconds, on instance subset I

Interestingly, for $(n, T) = (60, 48)$ and $F = 3$, $F(x, u)$ converges faster on the subset of instances where both formulations finish, even if the average CPU time of $LF(x, u)$ (67 seconds) is better than that of $F(x, u)$ (545 seconds). Since $LF(x, u)$ solves 3 more instances to optimality than $F(x, u)$, this shows that even if $F(x, u)$ is on average slightly faster on the remaining 17 instances, $LF(x, u)$ performs really well on the 3 instances not solved at all by $F(x, u)$. The same applies to $(n, T) = (60, 96)$ and $F = 2$.

The techniques breaking sub-symmetries, i.e. Fixing, A-$(\widetilde{x}, \widetilde{u})$, and $LF(x, u)$, perform better on all instances than techniques breaking symmetries only, thus showing the impact of sub-symmetry breaking.

5.4.7 Results for the ramp-constrained MUCP

Recall that aggregated formulation A-($\widetilde{x}$, $\widetilde{u}$) can no longer be used in this context.

Tables 9, 10 and 11 provide, for each formulation and each group of 20 instances, the exact same column entries as those in Tables 6 and 7.

First note that the ramp constraints make the MUCP instances much harder to solve by Cplex in general, as the CPU times in Table 11 relative to $(n, T) = (60, 48)$ ramp-constrained MUCP instances are much larger than those in Table 6 relative to the corresponding non-constrained MUCP instances. For example, the integrality gap is in average more than 10 times larger for the ramp-constrained problem on $(n, T) = (60, 48)$ and $F = 2$ instances. Thus, smaller instances with $(n, T) = (20, 48)$, $(20, 96)$ and $(n, T) = (30, 48)$, $(30, 96)$, respectively, are also presented in Tables 9 and 10.

Formulation Int($\widetilde{y}$) is still the less efficient formulation. It does not solve to optimality any instance with $n > 20$ except one. Moreover, on $n = 30$ instances, and on $(n, T) = (60, 96)$ instances, the root node cannot be processed at all within the time limit for formulation Int($\widetilde{y}$); the number of nodes explored is 0 and the optimality gap is 100%.

On instances of size $(n, T) = (60, 48)$, formulation $F(x, u, z)$ is the most efficient, as it solves to optimality a larger number of instances than the other techniques do. Formulations $LF(x, u)$ and $W(x, u)$ performs also well in this context. For example, for $F = 3$, $F(x, u, z)$ (resp. $LF(x, u)$, resp. $W(x, u)$), solves 14 (resp. 8) instances to optimality, while formulations $F(x, u)$ and Int($\widetilde{y}$) only solve 1 to optimality. Interestingly on $(n, T) = (60, 48)$, $F = 4$ instances, even if $F(x, u, z)$ solves 17 instances to optimality while $LF(x, u)$ only solves 15, the best average CPU time (962 seconds) is still achieved by $LF(x, u)$, as average $F(x, u, z)$ CPU time is 1262 seconds.

On the other test sets, i.e., $(n, T) = (20, 48)$, $(20, 96)$ and $(n, T) = (30, 48)$, $(30, 96)$ and $(n, T) = (60, 96)$, $LF(x, u)$ is more efficient than all considered techniques. For example, on $(n, T) = (20, 96)$ and $F = 3$ instances, $LF(x, u)$ solves to optimality 13 instances with an average CPU time of 1679, while the second best technique on this instance set, $F(x, u, z)$, solves only 7 instances to optimality, and the average CPU time is 2549 seconds. On $(n, T) = (30, 96)$ and $F = 2$ instances, $LF(x, u)$ solves to optimality 9 instances and has an average CPU time of 2091, while $F(x, u, z)$, $F(x, u, z)$-S0, $LF(x, u)$-S0, and $W(x, u)$-S0 solve only 5 instances to optimality on this instance set, and the average CPU time is around 2800 seconds. Other techniques solve less than 5 instances to optimality on this test set, and their average CPU time is higher than 3000 seconds. Interestingly on these instances, $LF(x, u)$ performs much better than $LF(x, u)$-S0. It seems that sub-symmetry breaking enhances Cplex's internal symmetry-breaking techniques. A guess is that there remains some symmetries after applying sub-symmetry-breaking inequalities, for example "shut-down" sub-symmetries, that Cplex is able to partially break once "start-up" sub-symmetries are broken.

On the largest instances ($(n, T) = (60, 96)$), $LF(x, u)$ manages to solve to optimality two instances, while other formulations do not reach optimality on any of these instances. Note that the table corresponding to $(n, T) = (60, 96)$ is not included as too few instances terminate within time limit.

Table 9 Performance indicators relative to the comparison of symmetry-breaking techniques for ramp-constrained MUCP instances with $n = 20$

Instances		Formulation	#opt	#nodes	Gap (%)	CPU time
(20,48)	$F = 2$	$F(x, u)$-S0	6	1,119,029	0.01243	2600.0
		$F(x, u)$	11	779,156	0.00941	2063.4
		MOB	7	933,509	0.01857	2432.0
		Fixing	7	831,828	0.02043	2478.5
		A-Int($\widetilde{y}$)	11	14,193	0.02040	2394.1
		$W(x, u)$-S0	10	241,898	0.01192	1947.4
		$W(x, u)$	10	241,595	0.01190	1941.8
		$F(x, u, z)$-S0	13	143,469	0.00875	1678.9
		$F(x, u, z)$	13	140,331	0.01021	1710.1
		$LF(x, u)$-S0	16	209,873	0.00219	976.8
		$LF(x, u)$	16	190,531	0.00230	937.5
	$F = 3$	$F(x, u)$-S0	12	788,243	0.00711	1567.6
		$F(x, u)$	13	879,270	0.00479	1445.5
		MOB	8	935,221	0.01039	2184.5
		Fixing	11	720,463	0.00928	1851.2
		A-Int($\widetilde{y}$)	3	5640	0.04120	3281.5
		$W(x, u)$-S0	15	257,665	0.00369	1210.6
		$W(x, u)$	15	271,876	0.00410	1195.1
		$F(x, u, z)$-S0	18	110,324	0.00261	917.8
		$F(x, u, z)$	18	107,921	0.00262	889.8
		$LF(x, u)$-S0	20	15,559	0	70.1
		$LF(x, u)$	20	21,166	0	89.9
(20,96)	$F = 2$	$F(x, u)$-S0	3	651,409	0.02123	3069.6
		$F(x, u)$	4	541,930	0.01883	2912.2
		MOB	0	671,072	0.05151	3600.0
		Fixing	2	613,269	0.04830	3400.7
		A-Int($\widetilde{y}$)	0	0	100.00000	3600.0
		$W(x, u)$-S0	4	301,307	0.02409	3003.2
		$W(x, u)$	3	312,391	0.02426	3083.0
		$F(x, u, z)$-S0	4	107,589	0.02311	2948.4
		$F(x, u, z)$	4	115,529	0.02282	2955.6
		$LF(x, u)$-S0	5	214,206	0.00709	2725.3
		$LF(x, u)$	6	232,265	0.00652	2666.9
	$F = 3$	$F(x, u)$-S0	6	549,541	0.01305	2681.9
		$F(x, u)$	6	544,877	0.01132	2633.6
		MOB	1	756,515	0.02901	3519.7
		Fixing	3	624,378	0.03286	3200.1
		A-Int($\widetilde{y}$)	0	0	100.00000	3600.0
		$W(x, u)$-S0	5	264,221	0.01441	2707.6
		$W(x, u)$	5	270,094	0.01419	2708.1
		$F(x, u, z)$-S0	5	114,113	0.01552	2711.9
		$F(x, u, z)$	7	110,321	0.01512	2549.1
		$LF(x, u)$-S0	14	137,192	0.00503	1573.6
		$LF(x, u)$	13	144,603	0.00522	1679.5

Table 10 Performance indicators relative to the comparison of symmetry-breaking techniques for ramp-constrained MUCP instances with $n = 30$

Instances		Formulation	#opt	#nodes	Gap (%)	CPU time
(30,48)	$F = 2$	$F(x, u)$-S0	3	1,202,051	0.01074	3133.4
		$F(x, u)$	4	1,144,442	0.00719	2922.1
		MOB	1	1,010,438	0.01657	3480.5
		Fixing	7	692,926	0.01695	2672.5
		A-Int($\widetilde{y}$)	0	0	100.00000	3600.0
		$W(x, u)$-S0	7	279,210	0.01113	2542.4
		$W(x, u)$	6	276,914	0.01129	2612.0
		$F(x, u, z)$-S0	9	86,139	0.00569	2352.6
		$F(x, u, z)$	9	84,914	0.00569	2364.2
		$LF(x, u)$-S0	13	353,064	0.00287	1809.1
		$LF(x, u)$	14	346,568	0.00276	1768.3
	$F = 3$	$F(x, u)$-S0	5	1,556,647	0.00922	2876.7
		$F(x, u)$	8	1,082,060	0.00648	2365.5
		MOB	4	991,032	0.01232	3142.1
		Fixing	9	539,062	0.00998	2048.6
		A-Int($\widetilde{y}$)	0	0	100.00000	3600.0
		$W(x, u)$-S0	10	334,639	0.00701	2064.1
		$W(x, u)$	10	328,935	0.00646	2059.5
		$F(x, u, z)$-S0	11	122,580	0.00341	1761.1
		$F(x, u, z)$	11	122,133	0.00359	1758.4
		$LF(x, u)$-S0	14	338,088	0.00199	1318.0
		$LF(x, u)$	14	337,476	0.00231	1309.8
(30,96)	$F = 2$	$F(x, u)$-S0	2	583,630	0.00585	3253.3
		$F(x, u)$	4	564,244	0.00483	3009.7
		MOB	1	368,914	0.01423	3435.3
		Fixing	2	287,804	0.01583	3334.7
		A-Int($\widetilde{y}$)	0	0	100.00000	3600.0
		$W(x, u)$-S0	5	142,411	0.00535	2879.2
		$W(x, u)$	4	154,970	0.00509	3014.7
		$F(x, u, z)$-S0	5	76,181	0.00518	2846.3
		$F(x, u, z)$	5	82,723	0.00503	2891.4
		$LF(x, u)$-S0	5	510,958	0.00307	2739.5
		$LF(x, u)$	9	205,904	0.00317	2091.3
	$F = 3$	$F(x, u)$-S0	1	735,071	0.00391	3426.5
		$F(x, u)$	2	713,144	0.00335	3247.8
		MOB	0	432,731	0.01292	3600.0
		Fixing	0	445,503	0.01315	3600.0
		A-Int($\widetilde{y}$)	0	0	100.00000	3600.0
		$W(x, u)$-S0	5	228,263	0.00421	3007.6
		$W(x, u)$	4	285,665	0.00422	3062.9
		$F(x, u, z)$-S0	5	107,442	0.00328	2896.9
		$F(x, u, z)$	5	84,094	0.00336	2838.5
		$LF(x, u)$-S0	9	275,787	0.00119	2183.4
		$LF(x, u)$	9	272,120	0.00110	2164.7

Table 11 Performance indicators relative to the comparison of symmetry-breaking techniques for ramp-constrained MUCP instances with $n = 60$

Instances		Formulation	#opt	#nodes	Gap (%)	CPU time
(60,48)	$F = 2$	$F(x, u)$-S0	2	913,875	0.00448	3397.3
		$F(x, u)$	1	1,118,040	0.00287	3426.4
		MOB	1	363,483	0.00820	3421.6
		Fixing	3	287,440	0.00811	3123.5
		A-Int($\widetilde{y}$)	0	15,463	0.01482	3600.0
		$W(x, u)$-S0	4	225,614	0.00321	2928.9
		$W(x, u)$	4	252,508	0.00318	2906.1
		$F(x, u, z)$-S0	8	99,207	0.00388	2607.3
		$F(x, u, z)$	8	117,186	0.00382	2551.0
		$LF(x, u)$-S0	2	1,020,320	0.00294	3269.8
		$LF(x, u)$	4	814,321	0.00267	2941.8
	$F = 3$	$F(x, u)$-S0	1	1,092,890	0.00383	3440.5
		$F(x, u)$	1	1,190,022	0.00354	3421.3
		MOB	1	468,546	0.00740	3433.9
		Fixing	7	328,643	0.00502	2626.7
		A-Int($\widetilde{y}$)	1	11,905	0.01858	3540.0
		$W(x, u)$-S0	8	214,408	0.00223	2480.9
		$W(x, u)$	8	241,259	0.00217	2424.3
		$F(x, u, z)$-S0	12	109,288	0.00037	1989.0
		$F(x, u, z)$	14	101,805	0.00035	1751.0
		$LF(x, u)$-S0	8	1,041,436	0.00066	2623.6
		$LF(x, u)$	8	657,602	0.00078	2222.8
	$F = 4$	$F(x, u)$-S0	2	1,646,866	0.00350	3264.4
		$F(x, u)$	7	1,578,458	0.00263	2774.1
		MOB	3	558,513	0.00461	3179.3
		Fixing	9	392,990	0.00361	2243.5
		A-Int($\widetilde{y}$)	0	3130	0.02369	3600.0
		$W(x, u)$-S0	12	292,577	0.00068	1772.9
		$W(x, u)$	11	260,574	0.00105	1758.2
		$F(x, u, z)$-S0	16	43,731	0.00021	1249.3
		$F(x, u, z)$	17	53,574	0.00015	1262.1
		$LF(x, u)$-S0	12	911,378	0.00049	1619.7
		$LF(x, u)$	15	300,012	0.00050	962.9

Recall that $W(x, u)$ is only partial symmetry-breaking. Thus, when T is smaller, the number of feasible columns featuring a given number of 1-entries is also smaller. On the opposite, when $T = 96$, the number of one-entries is not a very discriminating indicator among symmetric columns. Therefore $W(x, u)$ is not able to break as much symmetries, and $LF(x, u)$ globally performs better. Similarly, when T is larger the

Table 12 For each triplet (n, T, F), comparison of formulations $F(x, u)$ and $LF(x, u)$ on the subset of ramp-constrained MUCP instances that are solved within the time limit in both settings

Instances		# I	CPU $F(x,u)\|I$	CPU $LF(x,u)\|I$
(20,48)	$F = 2$	11	806.2	65.6
	$F = 3$	13	285.3	41.4
(20,96)	$F = 2$	4	160.8	32.6
	$F = 3$	6	378.4	341.7
(30,48)	$F = 2$	4	210.4	171.8
	$F = 3$	8	513.7	32.7
(30,96)	$F = 2$	4	648.5	33.8
	$F = 3$	2	78	23
(60,48)	$F = 2$	1	128.1	6.4
	$F = 3$	1	25.6	14.7
	$F = 4$	7	1240.3	26.9

number of sub-symmetries also increases. As $F(x, u, z)$ only handles symmetries, it performs not as well in this context as $LF(x, u)$, which is able to handle both symmetries and sub-symmetries.

In order to complete the results provided in Tables 9, 10 and 11. Table 12 compares the CPU times of formulations $F(x, u)$ and $LF(x, u)$ for the subset of instances on which both formulations terminate within time limit, for each size (n, T) and factor F. Column labels of Table 12 are the same as in Table 8. The table shows that even when considering an instances subset where both formulations terminate, $LF(x, u)$ remains much faster than $F(x, u)$.

6 Conclusion and perspectives

We propose a framework to build sub-symmetry-breaking inequalities, in order to handle the symmetries arising from a collection of sub-symmetric solution subsets. These inequalities may require to introduce one additional integer variable z per solution subset considered. Depending on the subset structure, variable z could be a linear expression of variables x, and therefore would not need to be introduced in the model as an additional variable. The derived sub-symmetry-breaking inequalities are full symmetry-breaking under a mild condition. Even if this condition is not satisfied, a new collection of sub-symmetric subsets can be polynomially constructed such that the derived inequalities are full symmetry-breaking.

The framework is applied to two problems: the GCP and the MUCP with or without ramp constraints.

It is well known that the classical GCP formulation is rife with symmetries. Experimental results highlight that when sub-symmetries have a significant impact on the resolution process, such sub-symmetries can be handled using appropriate subsets of inequalities derived from the proposed framework. Perspectives are to find other types of sub-symmetries in the classical formulation of the GCP to derive new sub-symmetry-breaking inequalities from the proposed framework. Another perspective is to apply the framework to other GCP formulations.

Experimental results for the MUCP show that aggregation of the classical formulation is a very efficient technique to handle symmetries and sub-symmetries arising in the MUCP. When ramp constraints are taken into account in the MUCP, disaggregation is no longer possible. Sub-symmetry-breaking inequalities can still be used and are competitive with state-of-the-art symmetry-breaking techniques. In particular, sub-symmetry-breaking inequalities outperform all other techniques on instances with a large number of time steps, i.e., $T = 96$. One perspective is to use the proposed framework to derive new sub-symmetry-breaking inequalities for "shut down" sub-symmetries in the ramp-constrained MUCP.

Aggregation techniques appear to work well for cases when the decomposition property holds. Such cases are more the exceptions than the rule. Sub-symmetry-breaking inequalities are always applicable as the solution subsets considered can capture the specific conditions under which the symmetries hold. Experimental results on MUCP and GCP instances show that sub-symmetry-breaking significantly improves on symmetry-breaking only.

Thus, perspectives are to apply the proposed framework to other problems featuring sub-symmetric solution subsets such as covering problems, or bin packing variants where one item can be placed in multiple bins.

Last but not least, it would be useful to study how the presented framework could be automated, so that sub-symmetric subsets are automatically detected and variables z automatically constructed.

References

1. Baum, S., Trotter, L.E.: Integer rounding and polyhedral decomposition for totally unimodular systems. In: Optimization and Operations Research, pp. 15–23. Springer (1978)
2. Bendotti, P., Fouilhoux, P., Rottner, C.: The min-up/min-down unit commitment polytope. J. Comb. Optim. **36**(3), 1024–1058 (2018)
3. Bendotti, P., Fouilhoux, P., Rottner, C.: On the complexity of the unit commitment problem. Ann. Oper. Res. **274**, 119–130 (2018)
4. Bendotti, P., Fouilhoux, P., Rottner, C.: Orbitopal fixing for the full (sub-)orbitope and application to the unit commitment problem. Optimization Online (2018)
5. Bendotti, P., Fouilhoux, P., Rottner, C.: Sub-symmetry-breaking inequalities for ILP with structured symmetry. In: Proceedings of the 20th Conference on Integer Programming and Combinatorial Optimization (IPCO 2019), LNCS, vol. 11480, pp. 57–71 (2019)
6. Berthold, T., Pfetsch, M.E.: Detecting orbitopal symmetries. In: Operations Research Proceedings 2008, pp. 433–438. Springer (2009)
7. Brélaz, D.: New methods to color the vertices of a graph. Commun. ACM **22**(4), 251–256 (1979)
8. Burke, E., Mareček, J., Parkes, A., Rudová, H.: A supernodal formulation of vertex colouring with applications in course timetabling. Ann. Oper. Res. **79**, 105–130 (2010)
9. Carrion, M., Arroyo, J.M.: A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. IEEE Trans. Power Syst. **21**, 1371–1378 (2006)
10. Coll, P., Marenco, J., Díaz, I., Zabala, P.: Facets of the graph coloring polytope. Ann. Oper. Res. **116**, 79–90 (2002)

11. Figueiredo, R., Barbosa, V., Maculan, N., Souza, C.: Acyclic orientations with path constraints. RAIRO-Oper. Res. **42**, 455–467 (2008)
12. Fischetti, M., Lodi, A., Salvagnin, D.: Just MIP it! In: Matheuristics, pp. 39–70. Springer (2009)
13. Friedman, E.J.: Fundamental Domains for Integer Programs with Symmetries, pp. 146–153. Springer Berlin Heidelberg, Berlin (2007)
14. Graph coloring benchmark https://sites.google.com/site/graphcoloring/vertex-coloring
15. Gualandi, S., Malucelli, F.: Exact solution of graph coloring problems via constraint programming and column generation. INFORMS J. Comput. **24**(1), 81–100 (2012)
16. Hojny, C., Pfetsch, M.E.: Polytopes associated with symmetry handling. Math. Program. **2018**, 1–44 (2018)
17. Jans, R.: Solving lot-sizing problems on parallel identical machines using symmetry-breaking constraints. INFORMS J. Comput. **21**(1), 123–136 (2009)
18. Kaibel, V., Loos, A.: Branched polyhedral systems. In: Proceedings of the 14th International Conference on Integer Programming and Combinatorial Optimization (IPCO), LNCS vol. 6080, pp. 177–190. Springer (2010)
19. Kaibel, V., Peinhardt, M., Pfetsch, M.E.: Orbitopal fixing. In: Proceedings of the 12th International Conference on Integer Programming and Combinatorial Optimization (IPCO), LNCS vol. 4513, pp. 74–88. Springer (2007)
20. Kaibel, V., Pfetsch, M.E.: Packing and partitioning orbitopes. Math. Program. **114**(1), 1–36 (2008)
21. Knueven, B., Ostrowski, J., Wang, J.: The ramping polytope and cut generation for the unit commitment problem. INFORMS J. Comput. **30**(4), 739–749 (2018)
22. Knueven, B., Ostrowski, J., Watson, J.P.: Exploiting identical generators in unit commitment. IEEE Trans. Power Syst. **33**, 4496–4507 (2017)
23. Liberti, L.: Reformulations in mathematical programming: automatic symmetry detection and exploitation. Math. Program. **131**(1), 273–304 (2012)
24. Liberti, L., Ostrowski, J.: Stabilizer-based symmetry breaking constraints for mathematical programs. J. Global Optim. **60**(2), 183–194 (2014)
25. Lima, R.M., Novais, A.Q.: Symmetry breaking in MILP formulations for Unit Commitment problems. Comput. Chem. Eng. **85**, 162–176 (2016)
26. Loos, A.: Describing orbitopes by linear inequalities and projection based tools. Ph.D. thesis, Universität Magdeburg (2011)
27. Malaguti, E., Monaci, M., Toth, P.: An exact approach for the vertex coloring problem. Discrete Optim. **8**(2), 174–190 (2010)
28. Margot, F.: Exploiting orbits in symmetric ILP. Math. Program. **98**(1), 3–21 (2003)
29. Mehrotra, A., Trick, M.: A column generation approach for graph coloring. INFORMS J. Comput. **8**, 344–354 (1996)
30. Méndez-Díaz, I., Zabala, P.: A polyhedral approach for graph coloring. Electron. Notes Discrete Math. **7**(1), 14 (2001)
31. Ostrowski, J., Anjos, M., Vannelli, A.: Modified orbital branching for structured symmetry with an application to unit commitment. Math. Program. **150**(1), 99–129 (2015)
32. Ostrowski, J., Anjos, M.F., Vannelli, A.: Tight mixed integer linear programming formulations for the unit commitment problem. IEEE Trans. Power Syst. **27**, 39–46 (2012)
33. Ostrowski, J., Linderoth, J., Rossi, F., Smriglio, S.: Orbital branching. Math. Program. **126**(1), 147–178 (2011)
34. Pan, K., Guan, Y.: A polyhedral study of the integrated minimum-up/-down time and ramping polytope. Optimization Online (2016). http://www.optimization-online.org/DB_HTML/2015/08/5070.html
35. Rajan, D., Takriti, S.: Minimum up/down polytopes of the unit commitment problem with start-up costs. IBM Research Report (2005)