Breaking structured symmetries and sub-symmetries in Integer Linear Programming

Cécile Rottner, Pascale Bendotti, Pierre Fouilhoux

Breaking structured symmetries and sub-symmetries in Integer Linear Programming

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

[1] EDF R&D, 7 boulevard Gaspard Monge, 91120 Palaiseau, France
{pascale.bendotti, cecile.rottner}@edf.fr
[2] Sorbonne Université, LIP6, 4 place Jussieu, 75005 Paris, France
{pierre.fouilhoux}@lip6.fr

Mots-clés : *Integer linear programming, (Sub)-symmetry breaking, Orbitopal fixing, Unit Commitment Problem.*

1 Introduction

It is well known that symmetries arising in integer linear programs (ILP) 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 ILP of the form

$$\min\left\{c(x) \mid x \in \mathcal{X}\right\}, \text{ with } \mathcal{X} \subseteq \mathcal{P}(m,n) \text{ and } c : \mathcal{P}(m,n) \to \mathbb{R} \tag{1}$$

where $\mathcal{P}(m,n)$ is the set of $m \times n$ binary matrices. A symmetry is defined as a permutation π of the variables $\{1, ..., n\}$ such that for any solution matrix $x \in \mathcal{X}$, matrix $\pi(x)$ is also solution and has same cost, *i.e.*, $\pi(x) \in \mathcal{X}$ and $c(x) = c(\pi(x))$. The *symmetry group* $\mathcal{G}$ of ILP (1) is the set of all such permutations. Symmetry group $\mathcal{G}$ 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.

In order to break symmetries, a general idea is to pick one solution, defined as the *representative*, in each orbit, and then restrict the solution set to the set of all representatives. 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. Moreover, a symmetry-breaking technique may introduce some specific branching rules that interfere with the B&B search. This can forbid exploiting a user-defined branching rule or, even, the default solver branching settings. 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. Such a technique can be based on specific branching and pruning rules during the B&B search, or on symmetry-breaking inequalities. Techniques based on symmetry-breaking inequalities are flexible, since they do not rely on a particular B&B

search. Efficient full-symmetry-breaking techniques are usually based on the pruning of the B&B tree (see survey [6]).

In this article, we focus on ILP featuring structured symmetries, *i.e.*, the symmetry group is the set of all column permutations of the solution matrix (or of a solution submatrix). 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 $i \in \{1, ..., m-1\}$ 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 $y \succeq z$ if y is equal to z or if y is lexicographically greater than z. A matrix $x \in \mathcal{P}(m,n)$ is chosen to be the representative of its orbit if 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 a *full orbitope* and is denoted by $\mathcal{P}_0(m,n)$. The solution set $\mathcal{X}$ of ILP (1) restricted to representatives set is then $\mathcal{P}_0(m,n) \cap \mathcal{X}$. No complete linear description of the full orbitope $\mathcal{P}_0(m,n)$ is known in the x space, and experiments conducted in [5] indicate that its facet defining inequalities are extremely complicated. Special cases of full orbitopes are *packing* and *partitioning orbitopes*, which are restrictions to matrices with at most (resp. exactly) one 1-entry in each row. If all matrices in $\mathcal{X}$ have at most (resp. exactly) one 1-entry in each row, then the solution set can be restricted to a packing (resp. partitioning) orbitope. A complete linear description of these polytopes is given in [3], alongside with a polynomial time separation algorithm. From this linear description, a symmetry-breaking algorithm, called *orbitopal fixing*, is derived in [3] in order to consider only the solutions included in the packing (resp. partitioning) orbitope during the B&B search.
There are many problems whose symmetry group is the symmetric group acting on the columns, or on a subset of the columns, but whose solution space cannot be restricted to a partitioning or a packing orbitope. Examples range from line planning problems in public transports [2] to scheduling problems with a discrete time horizon, like the Unit Commitment Problem. Dedicated techniques have been introduced in the literature to handle such structured symmetries, from symmetry-breaking inequalities [4] to *modified orbital branching* (MOB) [7]. These techniques are only partial symmetry-breaking. The full symmetry-breaking property can be obtained in MOB, by enforcement of a specific branching rule, and therefore at the expense of losing the flexible property.
In this article, we propose a linear time orbitopal fixing algorithm for the full orbitope (Section 2) which is a flexible full symmetry-breaking technique handling all-column-permutation structure symmetries. Moreover, it does not introduce any additional inequalities, thus not increasing the size of the LP solved at each node. We then propose to generalize the definition of symmetries and full orbitopes to account for symmetries arising from a collection of solution subsets, thus introducing sub-symmetries and full sub-orbitopes (Section 3). Such subsets appear in particular as underlying subproblems of a B&B search. The main motivation to look at sub-symmetries is that they are often undetected in the symmetry group $\mathcal{G}$ of the problem. We extend our orbitopal fixing algorithm to break structured sub-symmetries arising from a collection of solution subsets. Experimental results on UCP instances show the effectiveness of our approach (Section 4).

2 Orbitopal fixing for the full orbitope

Let $C^{(m,n)}$ be the (m,n)-dimensional 0/1-cube. Given an ILP of the form (1), consider a polytope $P \subset C^{(m,n)}$ such that the solution set of (1) is a subset of P. At a given node a of the B&B tree, some variables are already fixed, for example by previous branching decisions. Additional variable fixings can be performed on some of the remaining free variables. The idea is to fix to 0 (resp. 1) variables that would yield a solution outside P if fixed to 1 (resp. 0). Variable fixing methods, introduced in [3], are presented as follows.

A non-empty face F of $C^{(m,n)}$ is given by two index sets I_0, $I_1 \subset \{1, ..., m\} \times \{1, ..., n\}$:

$$F = \{x \in C^{(m,n)} \mid x_{i,j} = 0 \ \forall (i,j) \in I_0 \text{ and } x_{i,j} = 1 \ \forall (i,j) \in I_1\}.$$

For a polytope $P \subset C^{(m,n)}$ and a face F of $C^{(m,n)}$ defined by (I_0, I_1), the smallest face of $C^{(m,n)}$ that contains $P \cap F \cap \{0,1\}^{(m,n)}$ is denoted by $\mathrm{Fix}_F(P)$, *i.e.*, $\mathrm{Fix}_F(P)$ is the intersection of all faces of $C^{(m,n)}$ that contain $P \cap F \cap \{0,1\}^{(m,n)}$. If $\mathrm{Fix}_F(P)$ is a non-empty face of $C^{(m,n)}$, the index sets defining it will be denoted by $I_0^\star$ and $I_1^\star$. In general, the problem of computing $\mathrm{Fix}_F(P)$ is NP-hard.

When solving ILP (1) by B&B, to each node a corresponds a face $F(a)$ of $C^{(m,n)}$ defined by index sets I_0^a and I_1^a of variables already fixed to 0 and to 1 respectively. The aim of variable fixing is then to find, at each node a, sets $I_0^\star$ and $I_1^\star$ defining $\mathrm{Fix}_{F(a)}(P)$, where P is a given polytope containing the solution set. There are two cases. If $\mathrm{Fix}_{F(a)}(P) = \varnothing$ then $P \cap F(a) \cap \{0,1\}^{(m,n)} = \varnothing$ and node a can be pruned. If $\mathrm{Fix}_{F(a)}(P) \neq \varnothing$, then, any free variable in $I_0^\star$ (resp. $I_1^\star$) can be set to 0 (resp. 1). Any variable $x_{i,j}$ such that $(i,j) \notin I_0^\star \cup I_1^\star$ cannot be fixed, as it takes both values 0 and 1 in solution subset $P \cap F(a) \cap \{0,1\}^{(m,n)}$. It proves that the fixings occur as early as possible in the B&B tree.

Orbitopal fixing is variable fixing with polytope P being an orbitope. It corresponds to the case when the solution set $\mathcal{X}$ of ILP (1) is restricted to an orbitope P. The resulting solution set $\mathcal{X} \cap P$ is trivially included in P. Then variable fixing can be performed in order to restrict the solution set at each node a to be included in orbitope P.

We propose an orbitopal fixing algorithm for the case when P is the full orbitope. It is a linear time algorithm computing the index sets $I_0^\star$ and $I_1^\star$ defining $\mathrm{Fix}_F(P)$.

Theorem 1. *For any hypercube face F, sets $I_0^\star$ and $I_1^\star$ defining $Fix_F(P)$ can be computed in linear time when P is the full orbitope.*

The corresponding algorithm, as well as a proof of validity, can be found in the extended version of the article [1]. It relies on the construction of two binary matrices M_{min} and M_{max} in $P \cap F$ such that for any other binary matrix $x \in P \cap F$, the j^{th} column of x, denoted by $x(j)$, is lexicographically bounded as follows $M_{min}(j) \preceq x(j) \preceq M_{max}(j)$.

To illustrate, consider the cube face F defined by pair (I_0, I_1), with $I_0 = \{(4,1),(3,2),(5,2)\}$ and $I_1 = \{(2,1),(5,1),(4,2),(1,3),(2,3)\}$. Our fixing algorithm shows that for any binary matrix $M \in F \cap P$, the following inequalities hold column-wise :

$$M_{min} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \preceq \quad M \in \begin{bmatrix} * & * & 1 \\ 1 & * & 1 \\ * & 0 & * \\ 0 & 1 & * \\ 1 & 0 & * \end{bmatrix} \quad \preceq \quad M_{max} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Thus entries (1,1), (3,1), (1,2), (2,2) (resp. entry (3,3)) must be 1 (resp. 0) in M.

3 Sub-symmetries and full sub-orbitopes

3.1 Sub-symmetries

We propose to generalize symmetries and full orbitopes to a collection of solution subsets. Consider a subset $Q \subseteq \mathcal{X}$ of solutions of ILP (1). 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*.

For a given solution subset Q, the corresponding symmetry group $\mathcal{G}_Q$ is different from $\mathcal{G}$ and may contain symmetries undetected in $\mathcal{G}$. However, this observation is not exploited in practice by existing symmetry-breaking techniques, as this would imply to compute the problem's sub-symmetries at each node of the B&B tree, which is computationally prohibitive. However, in many applications, sub-symmetries can be easily obtained from the problem's structure, and therefore do not need to be computed at each node. In this section, we introduce a theoretical framework in order to simultaneously handle symmetries and sub-symmetries arising from a set of subproblems. In particular, we consider how to select one representative of each class of symmetrical solutions, when multiple symmetry groups are considered.

Let $\{Q_i \subset \mathcal{X},\ i \in \{1, ..., s\}\}$ be a collection of matrix subsets. To each Q_i, $i \in \{1, ..., s\}$, corresponds a sub-symmetry group $\mathcal{G}_{Q_i}$. The idea is that Q_i may contain sub-symmetries not detected in the symmetry group $\mathcal{G}$. Let O^i_k, $k \in \{1, ..., o_i\}$, be the orbits defined by $\mathcal{G}_{Q_i}$ on subset Q_i, $i \in \{1, ..., s\}$.

When considering only the symmetry group $\mathcal{G}$, the orbits of solutions form a partition of the solution set $\mathcal{X}$. However, the set $\mathcal{O} = \{O^i_k,\ k \in \{1, ..., o_i\},\ i \in \{1, ..., s\}\}$ of orbits defined by sub-symmetry groups $\mathcal{G}_{Q_i}$, $i \in \{1, ..., s\}$, does not form a partition of $\mathcal{X}$ anymore. Indeed, for given $i,\ j \in \{1, ..., s\}$, if $Q_i \cap Q_j \neq \varnothing$, then any $x \in Q_i \cap Q_j$ will appear in both the orbits of $\mathcal{G}_{Q_i}$ and the orbits of $\mathcal{G}_{Q_j}$. In order to break such sub-symmetries, removing all non-representatives of an orbit of $\mathcal{G}_{Q_i}$ may remove the representative of an orbit of $\mathcal{G}_{Q_j}$, thus leaving the latter unrepresented.

We therefore generalize the concept of orbit in order to define a new partition of $\mathcal{X}$ taking sub-symmetries into account. First, for given $X \in \mathcal{P}(m, n)$, let us define $\mathcal{G}(X)$, the set of all permutations π in $\bigcup_{i=1}^{s} \mathcal{G}_{Q_i}$ such that π can be applied to $X : \mathcal{G}(X) = \bigcup_{Q_i \ni X} \mathcal{G}_{Q_i}$.

We now define a relation $\mathcal{R}$ over the solution set $\mathcal{X}$. Matrix X' is said to be in relation with X, written $X' \mathcal{R} X$, if :

$$\exists r \in \mathbb{N},\ \exists \pi_1, ..., \pi_r \mid \pi_k \in \mathcal{G}(\pi_{k-1}...\pi_1(X)), \forall k \in \{1, ..., r\},\ \text{ and }\ X' = \pi_1\pi_2...\pi_r(X).$$

The *generalized orbit* $\mathbb{O}$ *of* X *with respect to* $\{Q_i, i \in \{1, ..., s\}\}$ is thus the set of all X' such that $X' \mathcal{R} X$. Roughly speaking, orbits that intersect one another are collected into generalized orbits. In other words, matrix X' is in the generalized orbit of X if X' can be obtained from X by composing permutations of groups $\mathcal{G}_{Q_i}$, ensuring that each permutation $\pi \in \mathcal{G}_{Q_i}$ is applied to an element of Q_i. Note that $\mathcal{R}$ is an equivalence relation, thus the set of all generalized orbits with respect to $\{Q_i, i \in \{1, ..., s\}\}$ is a partition of $\mathcal{X}$. Moreover, for a given $X \in \mathcal{X}$, each X' in the generalized orbit of X is such that $X' \in \mathcal{X}$ and $c(X') = c(X)$. By definition, for any generalized orbit $\mathbb{O}$, there exist orbits σ_1, ..., $\sigma_p \in \mathcal{O}$ such that $\mathbb{O} = \cup_{i=1}^{p} \sigma_i$.

While characterizing the representative of a simple orbit $\sigma \in \mathcal{O}$ may sometimes be easy, it may anyway be difficult to characterize the representative of a generalized orbit. In this case, one may want to choose a representative $r(\sigma) \in \sigma$ for each orbit $\sigma \in \mathcal{O}$, and then use a *sub-symmetry-breaking* technique to remove all elements $\sigma \backslash \{r(\sigma)\}$ from the search, for each orbit $\sigma \in \mathcal{O}$. As for given orbit σ, the set $\sigma \backslash \{r(\sigma)\}$ may contain the representative of another orbit σ', we need to ensure that there remains at least one element per generalized orbit after the removal of all elements $\cup_{\sigma \in \mathcal{O}}(\sigma \backslash r(\sigma))$. To this end the choice of the representatives $r(\sigma)$ must satisfy the following compatibility property.

Definition 1. *Representative set $\{r(\sigma), \sigma \in \mathcal{O}\}$ is orbit-compatible if for any generalized orbit $\mathbb{O} = \cup_{i=1}^{p} \sigma_i$, $\sigma_1, ..., \sigma_p \in \mathcal{O}$, there exists j such that $r(\sigma_j) = r(\sigma_i)$ for all $i \in \{1, ..., p\}$ such that $r(\sigma_j) \in \sigma_i$. Such a solution $r(\sigma_j)$ is said to be a generalized representative of $\mathbb{O}$.*

In other words, if $\{r(\sigma), \sigma \in \mathcal{O}\}$ is orbit-compatible then for each generalized orbit $\mathbb{O} = \cup_{i=1}^{p} \sigma_i$ there exists $i \in \{1, ..., p\}$ such that either $r(\sigma_i)$ is not contained in any other orbit $\sigma_j \in \mathcal{O}$, $j \neq i$, or $r(\sigma_i)$ is the representative of any orbit to which it belongs.

Note that there always exists a set of orbit-compatible representatives : start by choosing a representative $r(\sigma)$ for a given $\sigma \in \mathcal{O}$, and then choose $r(\sigma)$ as the representative of each orbit σ' in which $r(\sigma)$ is contained. Representatives of orbits not containing $r(\sigma)$ can be chosen arbitrarily.

There may exist several generalized representatives of a given generalized orbit.

The next lemma states that when representatives are orbit-compatible, there remains at least one element per generalized orbit even if all elements $\cup_{\sigma \in \mathcal{O}}(\sigma \backslash r(\sigma))$ have been removed.

Lemma 1. *For given orbit-compatible representatives $r(\sigma)$, $\sigma \in \mathcal{O}$, for any generalized orbit $\mathbb{O} = \cup_{i=1}^{p} \sigma_i$, $\sigma_1, ..., \sigma_p \in \mathcal{O}$, $\exists j \in \{1, ..., p\}$ such that $r(\sigma_j) \notin \cup_{i=1}^{p}(\sigma_i \backslash r(\sigma_i))$.*

Note that even if the set of representatives is orbit-compatible, it may happen that an entire orbit $\sigma \in \mathcal{O}$ is removed by a sub-symmetry-breaking technique. However, if orbit-compatibility is satisfied, there will always remain at least one element in the corresponding generalized orbit, with same cost as any solution in orbit σ.

3.2 Full sub-orbitopes

Given $X \in \mathcal{X}$ and sets $R \subset \{1, ..., m\}$ and $C \subset \{1, ..., n\}$, we consider sub-matrix (R, C) of X, denoted by $X(R, C)$, obtained by considering columns C of X on rows R only. A symmetry group is said to be 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).

In this section, we generalize the notion of full orbitope in order to account for sub-symmetries arising in sub-symmetric subsets of $\mathcal{X}$. We consider solutions subsets Q_i, $i \in \{1, ..., s\}$, such that for each i, Q_i is sub-symmetric with respect to (R_i, C_i), $R_i \subseteq \{1, ..., m\}$ and $C_i \subseteq \{1, ..., n\}$.

For each orbit O_k^i, $k \in \{1, ..., o_i\}$ of $\mathcal{G}_{Q_i}$, let its representative $X_k^i \in O_k^i$ be such that sub-matrix $X_k^i(R_i, C_i)$ is lexicographically maximal, *i.e.*, its columns are lexicographically non-increasing. We prove the following lemma in the extended version of the article [1] :

Lemma 2. *The set of representatives* $\{X^i_k,\ k \in \{1,...,o_i\},\ i \in \{1,...,s\}\}$ *is orbit-compatible.*

We define the *full sub-orbitope* $\mathcal{P}_{sub}$ with respect to subsets Q_i, $i \in \{1,...,s\}$ as the convex hull of binary matrices X such that for each $i \in \{1,...,s\}$, if $X \in Q_i$ then the columns of $X(R_i, C_i)$ are lexicographically non-increasing. In particular, $\mathcal{P}_{sub}$ contains the generalized representatives of each generalized orbit $\mathcal{O}$, but no other element of $\mathcal{O}$. Note that the full sub-orbitope generalizes the full orbitope, as for $s = 1$, $Q_1 = \mathcal{X}$, $\mathcal{G}_{Q_1} = \mathfrak{S}_n$ and $(R_1, C_1) = (\{1,...,m\}, \{1,...,n\})$, the associated full sub-orbitope is the full orbitope $\mathcal{P}_0(m, n)$.

Orbitopal fixing can be sequentially applied in order to restrict the feasible set $\mathcal{X}$ to the full sub-orbitope. It works as follows : at each node a, for each $i \in \{1,...,s\}$ such that all solutions X at node a are in Q_i, orbitopal fixing for the full orbitope is applied to sub-matrix $X(R_i, C_i)$.

4 Application to the Unit Commitment Problem

Given a discrete time horizon $\mathcal{T} = \{1,...,T\}$, 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 min-up (resp. min-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 min-up and down time constraints.

The classical formulation of the MUCP [8] features two sets of binary variables : variables $x_{t,j}$, indicating whether unit j is up at time t, and variables $u_{t,j}$, indicating whether unit j starts up at time t. The associated feasible set is denoted by $\mathcal{X}_{UCP}$.

In practical instances, there are H sets of n_h identical units, *i.e.*, units with identical characteristics, which induce symmetries. Indeed, assuming a solution is expressed as a matrix where column j corresponds to the up/down trajectory of unit j over the time horizon, then any permutation of columns corresponding to identical units leads to another solution with same cost.

Moreover, in some subproblems, there exist symmetries not contained in the symmetry group of the original problem, arising from the possibility of permuting some sub-columns of solution matrices. In particular, consider two identical units. Suppose at some time t, these two units are down (resp. up) and ready to start up (resp. shut down). Then their plans after t can be permuted, even if they do not have the same plan before t.

To capture such symmetries and sub-symmetries, for each time period $t \in \{1,...,\mathcal{T}\}$ and subset N of identical units, we consider the following subsets of $\mathcal{X}_{MUCP}$:

$$\overline{Q}^t_N = \{x \in \mathcal{X}_{MUCP} \mid x_{t',j} = 0,\ \forall t' \in \{t - \ell^j, ..., t-1\},\ \forall j \in N\}$$
$$\underline{Q}^t_N = \{x \in \mathcal{X}_{MUCP} \mid x_{t',j} = 1,\ \forall t' \in \{t - L^j, ..., t-1\},\ \forall j \in N\}$$

These subsets are sub-symmetric with respect to the sub-matrix defined by rows and columns $(\{t, ..., T\}, N)$. We apply orbitopal fixing (referred to as DOF-S) to the corresponding full sub-orbitope in a dynamic fashion, in the sense that the lexicographic order follows the branching decisions occurring along the B&B tree. Preliminary experiments have shown that dynamic versions of orbitopal fixing clearly outperform static versions, where the lexicographic order is based on the natural row-order $(1, ..., T)$.

We compare DOF-S to modified orbital branching (MOB) [7], which is a state-of-the-art pruning-based symmetry-breaking technique, specifically designed to handle all-column-permutation symmetries. We also compare to Default Cplex (version 12.6.1) and Callback Cplex (*i.e.*, Cplex with empty Branch and LazyConstraint Callbacks). We run our experiments on the instances described in [1] until optimality or until the time limit of 3600 seconds is reached.

Resolution methods are compared pairwise using a speed-up indicator. For given methods m_1 and m_2, the *speed-up* achieved by m_1 with respect to m_2 on a given instance is the ratio $\frac{CPU(m_2)}{CPU(m_1)}$. Table 1 presents, for each group of 20 instances of same size (n, T) and same symmetry factor F (higher F means less symmetries, as described in [1]), for each method m_1 (among MOB and DOF-S) and each method m_2 (among Default and Callback Cplex),

#opt : Number of instances solved to optimality by m_1,
opt_Δ : Difference in the number of instances solved to optimality by m_1 and m_2,
S_{CPU} : Geometric average of the speed-up of method m_1 w.r.t. m_2.

Instance				$m2$ = Default Cplex		m_2 = Callback Cplex	
(n, T)	Sym	m_1	#opt	opt_Δ	S_{CPU}	opt_Δ	S_{CPU}
(30,96)	$F = 4$	MOB	14	-6	0.0902	2	1.57
		DOF-S	20	0	0.725	8	12.6
	$F = 3$	MOB	12	-4	0.371	4	3.78
		DOF-S	16	0	1.05	8	10.7
	$F = 2$	MOB	12	-5	0.197	1	2.1
		DOF-S	17	0	0.716	6	7.62
(60,48)	$F = 4$	MOB	17	0	0.978	9	13.5
		DOF-S	17	0	1.92	9	26.5
	$F = 3$	MOB	13	0	0.94	6	8.6
		DOF-S	15	2	2.25	8	20.6
	$F = 2$	MOB	18	1	1.84	8	11.7
		DOF-S	19	2	2.6	9	16.5

TAB. 1 – Average speed-up for various instances compared to Default and Callback Cplex

In terms of CPU time, MOB and DOF-S greatly outperform Callback Cplex, but the improvement is even more significant with DOF-S. For example on instances $(n, T) = (60, 48)$, $F = 4$ (resp. $F = 3$, $F = 2$), MOB outperforms Callback Cplex by a factor 13.5 (resp. 8.6, 11.7) while DOF-S increases this factor to 26.5 (resp. 20.6, 16.5).

As observed in [7], there is a huge performance gap between Callback Cplex and Default Cplex. Thus, even if MOB and DOF-S substantially outperform Callback Cplex in each instance group, it is sometimes not enough to close the performance gap between Default and Callback Cplex, especially for instances such that n is small compared to T ($(n, T) = (30, 96)$). On the opposite, for $(n, T) = (60, 48)$ instances, in which symmetries are a major source of difficulty, DOF-S clearly outperforms Default Cplex.

When T is large compared to n ($(n, T) = (30, 96)$), it seems that non symmetry-related difficulties arise, and none of the compared methods catch up with Default Cplex. In this context, the cost of applying symmetry-breaking techniques (including the performance loss induced by the use of a Callback) seems too important compared to the impact of symmetries. The performance loss is less important with DOF-S than with MOB.

On the opposite, when n is large compared to T ($(n, T) = (60, 48)$), symmetry seems to be a major factor of computational difficulty. Indeed, DOF-S performs very well in this context and solves to optimality some instances Default Cplex cannot. For example, on instances $(n, T) = (60, 48)$, $F = 2$ (resp. $F = 3$), DOF-S solves two more instances to optimality than Default Cplex. MOB does not perform as well as DOF-S in this respect. Moreover, DOF-S outruns Default Cplex by a factor 2, while MOB is closer to a factor 1.

Perspectives

There is a wide range of problems featuring all column permutation symmetries and sub-symmetries, in particular many variants of the UCP, on which it would be desirable to analyze the effectiveness of our approach. Examples of such problems can be found among covering problems.

Another perspective would be to extend orbitopal fixing to full orbitopes under other group actions, for example the cyclic group. Another approach to handle symmetries related to the symmetric or the cyclic group would be to find a new set of representatives whose convex hull would be easier to describe than the full orbitope.

Références

[1] P. Bendotti, P. Fouilhoux, and C. Rottner. Orbitopal fixing for the full (sub-)orbitope and application to the unit commitment problem. *Optimization Online*, 2018. `http://www.optimization-online.org/DB_HTML/2017/10/6301.html`.

[2] R. Borndörfer, M. Grötschel, and M. E. Pfetsch. A column-generation approach to line planning in public transport. *Transportation Science*, 41(1) :123–132, 2007.

[3] V. Kaibel and M. Pfetsch. Packing and partitioning orbitopes. *Mathematical Programming*, 114(1) :1 – 36, 2008.

[4] R. M. Lima and A. Q. Novais. Symmetry breaking in MILP formulations for Unit Commitment problems. *Computers & Chemical Engineering*, 85 :162–176, 2016.

[5] A. Loos. *Describing Orbitopes by Linear Inequalities and Projection Based Tools.* PhD thesis, Universität Magdeburg, 2011.

[6] F. Margot. *Symmetry in Integer Linear Programming*, pages 647–686. Springer, Berlin, Heidelberg, 2010.

[7] J. Ostrowski, M.F. Anjos, and A. Vannelli. Modified orbital branching for structured symmetry with an application to unit commitment. *Mathematical Programming*, 2015.

[8] D. Rajan and S. Takriti. Minimum up/down polytopes of the unit commitment problem with start-up costs. *IBM Research Report*, 2005.