Orbital independence in symmetric mathematical programs

Gustavo Dias and Leo Liberti

CNRS LIX, Ecole Polytechnique, 91128 Palaiseau, France
{dias,liberti}@lix.polytechnique.fr

Abstract. It is well known that symmetric mathematical programs are harder to solve to global optimality using Branch-and-Bound type algorithms, since the solution symmetry is reflected in the size of the Branch-and-Bound tree. It is also well known that some of the solution symmetries are usually evident in the formulation, making it possible to attempt to deal with symmetries as a preprocessing step. One of the easiest approaches is to "break" symmetries by adjoining some symmetry-breaking constraints to the formulation, thereby removing some symmetric global optima, then solve the reformulation with a generic solver. Sets of such constraints can be generated from each orbit of the action of the symmetries on the variable index set. It is unclear, however, whether and how it is possible to choose two or more separate orbits to generate symmetry-breaking constraints which are compatible with each other (in the sense that they do not make all global optima infeasible). In this paper we discuss a new concept of orbit independence which clarifies this issue.

1 Introduction

In this paper we address an important issue which arises when breaking symmetries of Mathematical Programs (MP) in view of solving them using Branch-and-Bound (BB) type algorithms. Symmetry-breaking devices are usually derived from orbits of the action of the formulation group on the decision variables. However, one cannot simply use such devices for all orbits: some orbits depend on each other, in a very precise mathematical sense, and hence it may be impossible to use more than one orbit for symmetry-breaking purposes. Below, we discuss a notion of orbit independence which permits to break symmetries from different orbits concurrently.

2 Previous work and notation

2.1 Mathematical Programming

An MP is a formulation which formally describes an optimization problem in terms of known parameters (input), decision variables (output), an objective function, and some constraints. We consider MPs of the following general form:

$$\left.\begin{array}{r}\min_x \ f(x) \\ \forall i \le m \ g_i(x) \le 0 \\ x \in D.\end{array}\right\} \quad (1)$$

In Eq. (1), $f, g_i : \mathbb{R}^n \to \mathbb{R}$ are functions for which we have closed form expressions f, g_i for each $i \le m$. The expressions are written in terms of a formal language $\mathcal{L}$ based on an alphabet $\mathcal{A}$ consisting of a finite number of operators (e.g. sum, difference, product, fractions, powers, square roots, basic transcendental functions such as logarithm and exponentials, and possibly more complicated operators depending on the application at hand), a countable supply of variable symbols $\mathsf{x}_1, \ldots, \mathsf{x}_n$ representing the decision variables $x_1, \ldots, x_n$, and the rational numbers. The set D might contain non-functional constraints such as ranges $[x^L, x^U]$ for the decision variables, and/or integrality constraints, encoded as an index set $Z \subseteq X = \{1, \ldots, n\}$ such that $x_j \in \mathbb{Z}$ for each $j \in Z$. This modelling paradigm contains Linear Programming (LP), Nonlinear Programming (NLP), Mixed-Integer Linear Programming (MILP), Mixed-Integer Nonlinear Programming (MINLP) and Semidefinite Programming (SDP) if $x_1, \ldots, x_n$ are matrices.

2.2 Symmetry detection

We emphasize that Eq. (1) subsumes the description of *two* mathematical entities: the MP itself, denoted by P, and its formal description P in the language $\mathcal{L}$, which we obtain when replacing x, f, g by their representing symbols $\mathsf{x}, \mathsf{f}, \mathsf{g}$. It is well known that P can be parsed into a Directed Acyclic Graph (DAG) data structure T (an elementary graph contraction of the well-known *parsing tree*) using a fairly simple context-free grammar [2]. The leaf nodes of T are labelled by constants or decision variable symbols, whereas the other nodes of T are labelled by operator symbols. The incidence structure of T encodes the parent-child relationships between operators, variables and constants. In practice, we can write P using a modelling language such as AMPL [3] and use an unpublished but effective AMPL API to derive T [4]. Since T is a labelled graph, we know how to compute the group $\mathcal{G}$ of its label-invariant isomorphisms (which must also respect a few other properties, such as non-commutativity of certain operators) [15, 16]. In practice, we can use the software codes Nauty or Traces [16] to obtain $\mathcal{G}$ and the set Θ of the orbits of its action on the nodes $V(T)$ of the DAG.

2.3 Formulation and solution groups

It was shown in [8] that: (a) the action of $\mathcal{G}$ can be projected to the leaf nodes of $V(T)$, which represent decision variables; (b) this projection induces a group homomorphism ϕ mapping $\mathcal{G}$ to a certain group image G_P; (c) G_P is a group of permutations of the indices of the variable symbols $\mathsf{x}_1, \ldots, \mathsf{x}_n$; (d) G_P is precisely the group of variable permutations of P which keeps $f(x)$ and $\{g_i(x) \mid i \le m\}$

invariant. In other words, [8] provides a practical methodology for computing the *formulation group* of a MP given as in Eq. (1). Since it is not hard to show that G_P is a subgroup of the *solution group* of P, meant as the group of permutations which keeps the set $\mathscr{G}(P)$ of global optima of P invariant, this methodology can be used to extract symmetries from P prior to solving it.

2.4 Symmetry Breaking Constraints

So much for detecting (some) symmetries. Once these are known, their most efficient exploitation appears to be their usage within the BB algorithm itself [12, 13, 18, 17]. Such approaches are, unfortunately, difficult to implement, as each solver code must be addressed separately. Their simplest exploitation is *static symmetry breaking* [14, §8.2] which, simply put, consists in adjoining some Symmetry-Breaking Constraints (SBCs) to the original formulation Eq. (1) in the hope of making all but one of the symmetric global optima infeasible. Following the usual trade-off between efficiency and generality, approaches which offer provable guarantees of removing symmetric optima are limited to special structures [6], whereas approaches which hold for any MP in the large class Eq. (1) are mostly common-sense constraints designed to work in general [9]. The consensus seems to be that sets of SBCs are derived from each orbit of the action of G_P on X (though this is not the only possibility: SBCs can also be derived from cyclic subgroups of G_P or single permutations).

2.5 Orbits

We recall that an orbit is an equivalence class of the quotient set $X/\sim$, where $i \sim j$ if there is $g \in G_P$ such that $g(i) = j$. This way, G_P partitions X into a set Ω_{G_P} of orbits $\omega_1, \ldots, \omega_p$, each of which can be used to generate SBCs. The projection homomorphism ϕ defined above for $\mathcal{G}$ and the leaf nodes of the parsing tree can be restricted to act on G_P and generalized to project its action to any subset $Y \subseteq X$ as follows: for each $\pi \in G_P$ let $\phi(\pi)$ be the product of the cycles of π having all components in Y. We denote by ϕ_Y this generalized action projection homomorphism. The image of ϕ_Y, when Y is some orbit $\omega \in \Omega_{G_P}$, is a group $G_P[\omega]$ called the *transitive constituent* of ω (a group action is *transitive* on a set S if $s \sim t$ for each $s, t \in S$).

2.6 Strong and weak SBCs

We borrow the square bracket notation to localize vectors: if $x^* \in \mathscr{G}(P)$ is a global optimum of P, then $x^*[\omega]$ is a projection of x^* on the coordinates indexed by ω. If $G_P[\omega]$ is the full symmetric group $\mathsf{Sym}(\omega)$ on the orbit, it means that $\mathscr{G}(P)$ contains vectors which, when projected onto ω, yield every possible order of $x^*[\omega]$. This implies that we can arbitrarily choose one order, e.g.:

$$\forall \ell < |\omega| \quad x_{\omega(\ell)} \leq x_{\omega(\ell+1)}, \tag{2}$$

where $\omega(\ell)$ is the ℓ-th element of ω (stored as a list), enforce this order by means of SBCs, and still be sure that at least one global optimum remains feasible. The SBCs in Eq. (2) are called *strong SBCs*. If $G_P[\omega]$ has any other structure, we observe that, by transitivity of the transitive constituents, at least one permutation in $G_P[\omega]$ will map the component having minimum value in $x^*[\omega]$ to the first component (the choice if minimum value and first components are arbitrary — alternative SBC sets can occur by choosing maximum and/or any other component). This yields the *weak SBCs*:

$$\forall \ell \in \omega \smallsetminus \{\omega(1)\} \quad x_{\omega(1)} \leq x_{\omega(\ell)}. \tag{3}$$

Strong SBCs select one order out of $|\omega|!$ many, and hence are able to break all symmetries in $G_P[\omega]$. Weak SBCs are unlikely to be able to achieve that. We let $g(x[B]) \leq 0$ denote SBCs involving only variables x_j with j in a given set B.

2.7 Stabilizers

Let $Y \subseteq X$. We recall that the pointwise stabilizer of Y w.r.t. G_P (or any group G) is defined as the subgroup of elements of G_P fixing each element of Y, i.e., $G^Y = \{g \in G_P \mid \forall y \in Y \ (gy = y)\}$. The setwise stabilizer of Y w.r.t. G_P is the subgroup of those elements of G_P under which Y is invariant, i.e., $\mathsf{stab}(Y, G_P) = \{g \in G_P \mid \forall y \in Y \ (gy \in Y)\}$. By definition, if Y is an orbit of G_P, then G^Y is the kernel of ϕ_Y and $\mathsf{stab}(Y, G_P) = G_P$.

3 Orbital Independence Notions

In this section we introduce our main results regarding orbit independence (OI). First we illustrate how SBCs built from different orbits may cut global optima from a MP; then we recall the conditions of OI originally introduced in [8], and finally we present a new concept of OI based on pointwise stabilizers.

3.1 Incompatible SBCs

In general, one may only adjoin to P the SBCs from *one* orbit. Adjoining SBCs from two or more orbits chosen arbitrarily may result in all global optima being infeasible, as Example 1 shows.

Example 1. Let P be the following MILP:

$$\begin{array}{rl} \min\limits_{x \in \{0,1\}^4} & x_1 + x_2 + 2x_3 + 2x_4 \\ & \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \leq \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix}. \end{array}$$

It has formulation group $G_P = \langle(1\,2)(3\,4)\rangle$, optima $\mathcal{G}(P) = \{(0,1,1,0),(1,0,0,1)\}$ and orbits $\Omega_{G_P} = \{\omega_1, \omega_2\} = \{\{1,2\},\{3,4\}\}$. Valid SBCs for ω_1 (resp. ω_2) are $x_1 \leq x_2$ (resp. $x_3 \leq x_4$). Whereas adjoining either of the two SBCs yields a valid narrowing, adjoining both simultaneously leads to an infeasible problem.

Yet, breaking symmetries from only one orbit does not generally make a strong computational impact in MPs of the form Eq. (1). In what follows, we explore the concept of "orbital independence" meant as sufficient conditions to guarantee that SBCs from many orbits preserve at least one global optimum of P feasible.

3.2 Some existing OI conditions

In order to concurrently combine sets of SBCs generated by two orbits $\omega, \theta \in \Omega_{G_P}$ into a valid narrowing (i.e. a reformulation guaranteed to keep at least one global optimum [7]) of a MINLP, two sufficient conditions were provided in [8]:

- there is a subgroup $H \leq G_P[\omega \cup \theta]$ such that $H[\omega] \cong C_{|\omega|}$ and $H[\theta] \cong C_{|\theta|}$;
- $\mathsf{gcd}(|\omega|, |\theta|) = 1$.

Two orbits with these properties are called *coprime*. Coprime orbits occur relatively rarely in practice [8].

Another set of conditions for OI was hinted at in [11], by means of the following iterative procedure. Initially, one sets $G \leftarrow G_P$ and picks an orbit $\omega \in \Omega_{G_P}$; then adjoins SBCs for ω to P, and then replaces G by G^ω. Termination occurs when G is the trivial group. At each iteration, the SBCs from different orbits can be concurrently adjoined to P. On the other hand, the orbits refer to the action of different groups: G_P initially, then the groups in a normal chain of pointwise stabilizers. In the following, we expand on this idea.

3.3 New conditions for OI

Our goal now is to introduce the concept of *independent* set of orbits and provide conditions that will help us to identify such sets. These new necessary conditions for OI will be established based on pointwise stabilizers.

First, let $\omega, \theta \in \Omega_{G_P}$. We look at what happens to θ when ω is pointwise stabilized: either G^ω fixes θ, or a subset of θ, or it does not fix any element of θ at all. So three cases follow:

(a) for any subset $\sigma \subseteq \theta$, $\sigma \notin \Omega_{G^\omega}$;
(b) there is a subset $\sigma \subsetneq \theta$ such that $\sigma \in \Omega_{G^\omega}$;
(c) $\theta \in \Omega_{G^\omega}$.

We can thus state the following binary *dependence* relations on the set Ω_{G_P}.

Definition 1. *The orbit θ is dependent of ω, denoted by $\theta \to \omega$, if θ is stabilized when ω is stabilized (case (a) above).*

Definition 2. *The orbit θ is semi-dependent of ω, denoted by $\theta \rightsquigarrow \omega$, if θ splits when ω is stabilized (case (b) above).*

Definition 3. *The orbit θ is independent of ω, denoted by $\theta \Lsh \omega$, if θ is not stabilized when ω is stabilized (case (c) above).*

Next, let Γ^ω be the set of permutations of G_P which move elements of the orbit ω nontrivially. By definition, Γ^ω does not contain the identity permutation e of G_P and thus it is not itself a group. Moreover, the following properties hold: $G^\omega \cap \Gamma^\omega = \varnothing$, $\mathsf{stab}(\omega, G_P) = G^\omega \cup \Gamma^\omega = G_P$ and $\phi_\omega(\Gamma^\omega) = G_P[\omega] \setminus e$.

Theorem 1 establishes the dependence relation between two orbits $\omega, \theta \in \Omega_{G_P}$ by comparing the sets Γ^ω and Γ^θ.

Theorem 1. *The following statements are true:*

(1) If $\Gamma^\theta = \Gamma^\omega$ then $\theta \to \omega$ and $\omega \to \theta$;
(2) If $\Gamma^\theta \subset \Gamma^\omega$ then $\theta \to \omega$ and either $\omega \Lsh \theta$ or $\omega \rightsquigarrow \theta$;
(3) If $\Gamma^\theta \cap \Gamma^\omega \neq \varnothing$ then ($\theta \Lsh \omega$ or $\theta \rightsquigarrow \omega$) and ($\omega \Lsh \theta$ or $\omega \rightsquigarrow \theta$);
(4) If $\Gamma^\theta \cap \Gamma^\omega = \varnothing$ then $\theta \Lsh \omega$ and $\omega \Lsh \theta$.

Proof. (1) Assume $\Gamma^\theta = \Gamma^\omega$ and consider ω. Then $G^\omega = G_P \setminus \Gamma^\omega \Rightarrow G^\omega \cap \Gamma^\theta = \varnothing \Rightarrow \theta \notin \Omega_{G^\omega}$ and $\theta \to \omega$. Since the same argument holds if we consider θ, we also have $\omega \to \theta$.

(2) Assume $\Gamma^\theta \subset \Gamma^\omega$ and consider ω. Then $G^\omega = G_P \setminus \Gamma^\omega \Rightarrow G^\omega \cap \Gamma^\theta = \varnothing \Rightarrow \theta \notin \Omega_{G^\omega}$ and $\theta \to \omega$. Considering θ, we have that $G^\theta = G_P \setminus \Gamma^\theta \Rightarrow G^\theta \cap \Gamma^\omega \neq \varnothing$. If the action of G^θ is transitive on ω, we have $\omega \Lsh \theta$. Otherwise, we have $\omega \rightsquigarrow \theta$.

(3) Assume $\Gamma^\theta \cap \Gamma^\omega \neq \varnothing$ but neither set is wholly contained in the other, and consider ω. Then $G^\omega = G_P \setminus \Gamma^\omega \Rightarrow G^\omega \cap \Gamma^\theta \neq \varnothing$. If the action of G^ω is transitive on θ, we have $\theta \Lsh \omega$. Otherwise, we have $\theta \rightsquigarrow \omega$. The same argument holds if we consider θ.

(4) Assume $\Gamma^\theta \cap \Gamma^\omega = \varnothing$ and consider ω. Then $G^\omega = G_P \setminus \Gamma^\omega \Rightarrow G^\omega \supset \Gamma^\theta \Rightarrow \theta \in \Omega_{G^\omega}$ and $\theta \Lsh \omega$. The argument is similar if we consider θ, thus $\omega \Lsh \theta$. □

Lemma 1. *The premise $\Gamma^\theta \cap \Gamma^\omega = \varnothing$ to condition (4) in Theorem 1 never holds.*

Proof. Let Δ be the set of generators of G_P. If there is $g \in \Delta$ such that $g[\omega]$ and $g[\theta]$ are nontrivial, then $g \in \Gamma^\theta \cap \Gamma^\omega$. Otherwise, let $\Delta^\theta = \{g \in \Delta \mid g[\omega] = e\}$ and $\Delta^\omega = \{g \in \Delta \mid g[\theta] = e\}$. Because every element of G_P can be expressed as the combination (under the group operation) of finitely many elements of Δ, there is $g \in G_P$ such that $g = g_\omega g_\theta$ where $g_\omega \in \Delta^\omega$ and $g_\theta \in \Delta^\theta$. Thus $g \in \Gamma^\theta \cap \Gamma^\omega$. □

Based on the above definitions and results, the following lemmata hold.

Lemma 2. *The relation $\to$ is reflexive and the relations $\rightsquigarrow$ and $\Lsh$ are irreflexive.*

Lemma 3. *The relation $\to$ is symmetric iff $\Gamma^\theta = \Gamma^\omega$ and asymmetric iff $\Gamma^\theta \subset \Gamma^\omega$.*

Lemma 4. *The relation $\rightarrow$ is transitive.*

Proof. Let $\theta, \omega, \tau \in \Omega_{G_P}$ be distinct orbits satisfying $\theta \rightarrow \omega$ and $\omega \rightarrow \tau$. From Theorem 1, $\theta \rightarrow \omega$ implies that either $\Gamma^\theta = \Gamma^\omega$ or $\Gamma^\theta \subset \Gamma^\omega$. Similarly, $\omega \rightarrow \tau$ implies that either $\Gamma^\omega = \Gamma^\tau$ or $\Gamma^\omega \subset \Gamma^\tau$. Then:
(i) $\Gamma^\theta = \Gamma^\omega \wedge \Gamma^\omega = \Gamma^\tau \Rightarrow \Gamma^\theta = \Gamma^\tau \Rightarrow \theta \rightarrow \tau$;
(ii) $\Gamma^\theta = \Gamma^\omega \wedge \Gamma^\omega \subset \Gamma^\tau \Rightarrow \Gamma^\theta \subset \Gamma^\tau \Rightarrow \theta \rightarrow \tau$;
(iii) $\Gamma^\theta \subset \Gamma^\omega \wedge \Gamma^\omega = \Gamma^\tau \Rightarrow \Gamma^\theta \subset \Gamma^\tau \Rightarrow \theta \rightarrow \tau$;
(iv) $\Gamma^\theta \subset \Gamma^\omega \wedge \Gamma^\omega \subset \Gamma^\tau \Rightarrow \Gamma^\theta \subset \Gamma^\tau \Rightarrow \theta \rightarrow \tau$. □

Whenever the dependence relations are symmetric, we write $\omega \leftrightarrow \theta$ or $\omega \leftrightsquigarrow \theta$ or $\omega \Lsh\!\Rsh \theta$. Using this notation, we set forth that:

Definition 4. *Two orbits $\omega, \theta \in \Omega_{G_P}$ are dependent if $\omega \leftrightarrow \theta$, semi-dependent if $\omega \leftrightsquigarrow \theta$ and independent if $\omega \Lsh\!\Rsh \theta$.*

Following, we extend the dependence relations presented above to sets of orbits. In this sense, consider a set $\Omega \subseteq \Omega_{G_P}$ and let $\Omega^\omega = \Omega \setminus \omega$ for $\omega \in \Omega$. We look at what happens to ω when the set Ω^ω is pointwise stabilized, i.e., when all of the orbits in Ω^ω are (simultaneously) pointwise stabilized. Similar cases to (a)-(c) may occur and suitable definitions can be stated.

Definition 5. *The orbit ω is dependent of Ω^ω, denoted by $\omega \hookrightarrow \Omega^\omega$, if ω is stabilized when all orbits of Ω^ω are stabilized.*

Definition 6. *The orbit ω is semi-dependent of Ω^ω, denoted by $\omega \rightsquigarrow \Omega^\omega$, if ω splits when all orbits of Ω^ω are stabilized.*

Definition 7. *The orbit ω is independent of Ω^ω, denoted by $\omega \not\hookrightarrow \Omega^\omega$, if ω is not stabilized when all orbits of Ω^ω are stabilized.*

Lemma 5 establishes necessary conditions to have $\omega \not\hookrightarrow \Omega^\omega$. The pointwise stabilizer of a set Ω of orbits is denoted as G^Ω hereafter.

Lemma 5. *If $\omega \not\hookrightarrow \Omega^\omega$, then $\omega \Rsh \theta$ for all $\theta \in \Omega^\omega$.*

Proof. By definition, $\omega \not\hookrightarrow \Omega^\omega$ implies that the action of G^{Ω^ω} on ω is transitive. Since G^{Ω^ω} is a subgroup of G^θ for every $\theta \in \Omega^\omega$, G^θ also acts transitively on ω and thus $\omega \Rsh \theta$. □

Finally we can define an independent set of orbits. We remark that, although we do not state them explicitly, corresponding definitions can be laid down concerning the concepts of dependent and semi-dependent sets of orbits.

Definition 8. *A set Ω is said to be independent if $\omega \not\hookrightarrow \Omega^\omega$ for all $\omega \in \Omega$.*

Corollary 1 provides necessary conditions so as to a set Ω be independent.

Corollary 1. *If the set Ω is independent, then $\omega \Lsh\!\Rsh \theta$ for all $\omega, \theta \in \Omega$.*

Proof. By Definition 8 and Lemma 5. □

3.4 SBCs from independent sets

Let Ω_I denote an independent set of orbits. Similarly to the results presented in [8], the following propositions set appropriate conditions to build weak and strong SBCs, respectively, from independent sets of orbtis.

Proposition 1. *The constraints* (3) *are SBCs for* P *and* $G^{\Omega_I^\omega}$ *with respect to* $\omega \in \Omega_I$.

Proof. Let $y \in \mathscr{G}(P)$. Since $G^{\Omega_I^\omega}$ acts transitively on ω, there exists $\pi \in G^{\Omega_I^\omega}$ mapping $\min y[\omega]$ to $y_{\omega(1)}$. □

Proposition 2. *Provided that* $G^{\Omega_I^\omega}[\omega] = \mathsf{Sym}(\omega)$, *the constraints* (2) *are SBCs for* P *and* $G^{\Omega_I^\omega}$ *with respect to* $\omega \in \Omega_I$.

Proof. Let $y \in \mathscr{G}(P)$. Since $G^{\Omega_I^\omega}[\omega] = \mathsf{Sym}(\omega)$, there exists $\pi \in G^{\Omega_I^\omega}$ such that $(\pi y)[\omega]$ is ordered by $\leq$. Therefore πy is feasible w.r.t. the contraints (2). □

4 Orbital Independence Algorithm

In this section we describe the methodology used to find an independent set of orbits of a mathematical program. We present how to model and solve the problem of finding such a set by means of a classical combinatorial optimization problem. Moreover, we describe in details the algorithm proposed to build SBCs from all orbits contained in an independent set.

4.1 Independence graph

Our interest relies in finding the largest $\Omega_I \subseteq \Omega_{G_P}$. Nevertheless, so far we do not have theoretical results providing sufficient conditions to find such a set. Yet we can use the necessary conditions provided by Corollary 1 and search for the largest set $\Omega_K \subseteq \Omega_{G_p}$ whose elements are pairwise independent. Having obtained Ω_K, we can then search for the largest $\Omega_I \subseteq \Omega_K$.

Hence we propose to find Ω_K by encoding the independence relation between orbits of G_P as an undirected graph $G_I = (V, E)$, as of now called the *independence* graph of P, where $V = \Omega_{G_P}$ and E is the set of pairs of independent orbits in Ω_{G_P}. In this manner we reduce the problem of finding Ω_K to the problem of finding the maximum clique in G_I.

4.2 Orbital independence reformulations

We expect that the larger the number of SBCs adjoined to the original formulation, the stronger their computational impact. Particularly, the larger the number of strong SBCs, the better. We thus propose two different reformulations based on the concept of OI: the first prioritizing the total number of SBCs generated and the second prioritizing the total number of strong SBCs generated.

In this sense, we look for cliques in G_I that either involve large orbits or involve mostly orbits which may satisfy the conditions to build strong SBCs.

In order to find such cliques, we associate a weight function $d : V \to W$ to $G_I = (V, E, d)$ and solve the Maximum Weight Clique problem (MWCP) for G_I using the MP formulation described in [1]. In the first reformulation, which we call *orbital independence narrowing*, we have $W = \{|\omega_1|, \ldots, |\omega_{|V|}|\}$ and $d(\omega_i) = |\omega_i|$ for all $\omega_i \in V$. In the second, which we call *strong orbital independence narrowing*, $W = \{d_1, d_2\}$. It is worth pointing out that the strong orbital independence narrowing prioritizes cliques having mostly orbits which satisfy $G_P[\omega] = \mathsf{Sym}(\omega)$; this is a necessary (but not sufficient) condition to have $G^{\Omega_I^\omega}[\omega] = \mathsf{Sym}(\omega)$ since $G^{\Omega_I^\omega}[\omega]$ is a subgroup of $G_P[\omega]$ for every $\omega \in \Omega_I$.

4.3 Algorithm description

Algorithm 1 Orbital Independence SBC generator

Require: nontrivial G_P and reformulation strategy ς

Let $C = \varnothing$ and $\Omega_I = \varnothing$
Let $\Omega_{G_P} = \mathsf{computeOrbits}(G_P)$
if $|\Omega_{G_P}| > 1$ **then**
    for $\omega \in \Omega_{G_P}$ **do**
        Let $G^\omega = \mathsf{computePointStab}(\omega)$
        for $\theta \in \Omega_{G_P}$ such that $\mathsf{pos}(\theta) > \mathsf{pos}(\omega)$ **do**
            Let $G^\theta = \mathsf{computePointStab}(\theta)$
            if $\mathsf{isTransitiveAction}(G^\omega, \theta) \wedge \mathsf{isTransitiveAction}(G^\theta, \omega)$ **then**
                Let $E = E \cup \{\{\omega, \theta\}, \{\theta, \omega\}\}$
            end if
        end for
    end for
    if $|E| \geq 2$ **then**
        Let $G_I = \mathsf{buildGraph}(\Omega_{G_P}, E, \varsigma)$
        Let $\Omega_K = \Omega_I = \mathsf{solveMWCP}(G_I)$
        for $\omega \in \Omega_K$ **do**
            if not $\mathsf{isTransitiveAction}(G^{\Omega_I^\omega}, \omega)$ **then**
                Let $\Omega_I = \Omega_I \smallsetminus \omega$
            end if
        end for
        for $\omega \in \Omega_I$ **do**
            Let $g(x[\omega]) \leq 0$ be some SBCs for P and $G^{\Omega_I^\omega}$ w.r.t. ω
            Let $C = C \cup \{g(x[\omega]) \leq 0\}$
        end for
    end if
end if
return C

The Algorithm 1 generates a set C containing SBCs derived from the largest independent set of orbits of P. It takes as inputs a nontrivial formulation group (parameter G_P) and a reformulation strategy (parameter ς). Some functions simplify the pseudocode of Alg. 1: $\mathsf{computeOrbits}(G_P)$ returns the orbits of the group G_P; $\mathsf{computePointStab}(\omega)$ returns the pointwise stabilizer of ω; $\mathsf{pos}(\omega)$ returns the position of orbit ω in the list Ω_{G_P}; $\mathsf{isTransitiveAction}(G, \omega)$ returns true if the action of the group G is transitive on the orbit ω and false otherwise; $\mathsf{buildGraph}(V, E, \varsigma)$ returns a graph with vertices V, edges E and weights appropriate to the strategy ς; $\mathsf{solveMWCP}(G_I)$ returns a solution of the MWCP for the graph G_I.

If G_P has more than one orbit ($|\Omega_{G_P}| > 1$), the algorithm first iteratively looks for all pairs of independent orbits in order to build the set E. Because the Condition (3) in Theorem 1 is not sufficient to ascertain whether two orbits $\omega, \theta \in \Omega_{G_P}$ satisfy $\omega \looparrowleft\!\looparrowright \theta$, ultimately we must check if the action of the stabilizers G^ω and G^θ is transitive on θ and ω, respectively. Thus the algorithm does not compare the sets Γ^ω and Γ^θ but rather directly checks whether the actions are transitive. Following the first loop, if at least one pair of independent orbits is found ($|E| \geq 2$), the algorithm builds the independence graph G_I according to the reformulation strategy ς and calls a third party MILP solver to solve the MWCP for G_I. Once Ω_K is known, the algorithm converges to a set Ω_I by iteratively removing (from a copy of Ω_K stored as Ω_I) the orbits that do not satisfy $\omega \looparrowleft\!\!\rightarrow \Omega_I^\omega$. We remark that our approach here is not optimal in the sense that the resulting Ω_I may not be the largest one; evaluating all possible $\Omega_I \subseteq \Omega_k$ would most likely require a large computational effort due to many stabilizer computations. Then, for each orbit in the set Ω_I, the algorithm builds and adds SBCs to the set C. We remark that if $|\Omega_{G_P}| = 1$ (unique orbit) or $|E| = 0$ (no pair of independent orbits in Ω_{G_P}), no reformulation is carried out.

Theorem 2. *The constraint set* $C_{\Omega_I} = \{g(x[\omega_k]) \leq 0 \mid \omega_k \in \Omega_I\}$ *is an SBC system for* P.

Proof. If P is infeasible then adjoining the constraints in C_{Ω_I} to P does not change its infeasibility, so assume P is feasible. Since $g(x[\omega_k]) \leq 0$ are SBCs for P and $G^{\Omega_I^{\omega_k}}$ w.r.t. ω_k, there exist $y \in \mathscr{G}(P)$ and $\pi_{\omega_k} \in G^{\Omega_I^{\omega_k}}$ such that $\pi_{\omega_k} y$ satisfies $g((\pi_{\omega_k} y)[\omega_k]) \leq 0$. But $\pi_{\omega_k} \in G_P$ for all $\omega_k \in \Omega_I$ and, due to the closure of the group operation, there exists $\pi \in G_P$ such that $\pi = \prod \pi_{\omega_k}$. So $\pi y \in \mathscr{G}(P)$. But $\pi[\omega_k] = \pi_{\omega_k}[\omega_k]$ since $\pi_{\omega_{k'}}$ stabilizes ω_k pointwise for every $k' \neq k$ and thus $(\pi y)[\omega_k] = (\pi_{\omega_k} y)[\omega_k]$. Therefore πy satisfies $g((\pi y)[\omega_k]) \leq 0$ for all $\omega_k \in \Omega_I$. $\square$

5 Computational experiments

In this section we show the computational impact on the resolution of MILPs when adjoining SBCs arising from different orbits simultaneously. We describe the computational environment involved (machinery, solvers, instances) and analyze the results obtained from the conducted experiments.

5.1 Environment

Our test set consists of symmetric MPs taken from the library MIPLIB2010.The reformulations were obtained on a quad-CPU Intel Xeon at 2.66GHz with 24Gb RAM. Automatic group detection is carried out using the ROSE reformulator [10] and the Traces software [16]. Other group computations are carried out using GAP v. 4.7.4 [19]. The MP results were obtained on a 24-CPU Intel Xeon at 2.53GHz with 48Gb RAM. All problems were solved under the AMPL [3] environment using CPLEX 12.6 [5]. The execution time was limited to 1800 seconds of user cpu time. In order to try and provide a fair assessment of our methodology, we disabled the symmetry handling methods built into CPLEX. We also ran CPLEX in single thread mode to impose its sequential (and deterministic) behaviour and increase the chances of measuring performance differences.

5.2 Results

We first comment the results regarding the reformulation process. Table 1 reports, per instance, the number of variables (n) and orbits ($|\Omega_{G_P}|$) of the original formulation, and the total number of variables indexed by the orbits Ω_{G_P} (#svar); for each OI narrowing type, the table reports the size of the maximum clique ($|\Omega_K|$), the size of the largest independent set ($|\Omega_I|$), the total number of variables indexed by all of the orbits in Ω_I (#var), and the number of weak (#wea) and strong (#str) SBCs generated.

We would like to remark that both reformulation strategies yielded the same narrowings for the most part of the instances. In these cases, we do not present results regarding the strong orbital independence reformulation. Additionally, we also point out that the size of the maximum cliques is equal to the size of the largest independent sets for all instances.

Apart from the structure of the group G_P, intuitively, the ratio $\nu =(\#\text{svar}/n)$ may also indicate how symmetric a formulation P is. Similarly, the ratios $\rho = (|\Omega_I|/|\Omega_{G_P}|)$ and $\upsilon =(\#\text{var}/\#\text{svar})$ may indicate how extensively we have exploited the symmetries of P. All together, we expect SBCs to make a strong computational impact whenever the triplet (ν, ρ, υ) tends to $(1, 1, 1)$. Table 1 shows that the symmetric instances tested so far have, in general, two low ratios, which suggests that the impact of the SBCs may not be too significative.

Table 2 reports the optimization results. Per instance and for each formulation, the table exhibits the best solution found, the user cpu time (in seconds), the gap (%) and the solver status at termination (opt = optimum found, lim = time limit reached, inf = infeasible instance). Best values are emphasized in boldface. Some intances from Table 1 do not appear in Table 2 because no method performed better than the other.

As expected, we do not observe cases of infeasible narrowings due to the usage of SBCs derived from different orbits simultaneously. Moreover, we also observe consistent improvements in favor of the orbital independence narrowings. In 22 out of 48 instances, the SBCs slightly helped to improve the performance of the solver. On the other hand, in 14 cases the SBCs were harmful and, in 12 other

	Original			Oi-narrowing				
Instance	n	$\|\Omega_{G_P}\|$	#svar	$\|\Omega_K\|$	$\|\Omega_I\|$	#var	#wea	#str
bab5	21600	1936	3872	4	4	8	0	4
blp-ar98	16017	2	4	2	2	4	0	2
blp-ic97	8445	2	4	2	2	4	0	2
core4872-1529	24605	505	1046	46	46	96	0	50
gmu-35-40	842	40	111	4	4	13	0	9
gmu-35-50	1177	40	111	4	4	13	0	9
gmut-75-50	36164	64	242	6	6	19	0	13
gmut-77-40	13140	70	280	7	7	26	0	19
iis-bupa-cov	345	2	7	2	2	7	0	5
lectsched-4-obj	3513	267	557	17	17	36	0	19
map06	46015	107	245	10	10	20	0	10
map10	46015	107	245	10	10	20	0	10
map14	46015	107	245	10	10	20	0	10
map18	46015	107	245	10	10	20	0	10
map20	46015	107	245	10	10	20	0	10
mcsched	1669	45	90	15	15	30	0	15
mzzv11	10240	155	310	16	16	32	0	16
neos-1311124	1092	52	1092	4	4	84	0	80
neos-1426635	520	52	520	4	4	40	0	36
neos-1426662	832	52	832	4	4	64	0	60
neos-1436709	676	52	676	4	4	52	0	48
neos-1440460	468	52	468	4	4	36	0	32
neos-1442119	728	52	728	4	4	56	0	52
neos-1442657	624	52	624	4	4	48	0	44
neos-911880	888	259	888	7	7	24	0	17
neos-952987	31329	37	81	4	4	8	0	4
neos18	963	53	248	5	5	26	0	21
ns1631475	22696	105	210	11	11	22	0	11
ns2081729	661	300	600	3	3	6	0	3
ns2122603	18052	36	72	18	18	36	0	18
p2m2p1m1p0n100	100	25	92	3	3	12	0	9
protfold	1835	558	1800	2	2	4	0	2
rocII-4-11	3409	2	27	2	2	27	0	25
rococoC10-001000	2566	41	82	4	4	8	0	4
rvb-sub	33765	113	226	12	12	24	0	12
satellites1-25	9013	200	400	20	20	40	0	20
seymour-disj-10	1209	49	106	5	5	12	0	7
seymour	1255	55	156	5	5	41	29	7
swath	6404	21	163	2	2	8	0	6
transportmoment	9099	85	189	17	17	38	0	21
uc-case3	36921	2687	5374	2	2	4	0	2
uct-subprob	2236	136	306	7	7	14	0	7

	Original			Oi-narrowing					Soi-narrowing				
Instance	n	$\|\Omega_{G_P}\|$	#svar	$\|\Omega_K\|$	$\|\Omega_I\|$	#var	#wea	#str	$\|\Omega_K\|$	$\|\Omega_I\|$	#var	#wea	#str
core2536-691	15288	88	187	12	12	29	3	14	12	12	27	0	15
macrophage	2260	251	566	18	18	42	5	19	18	18	39	0	21
neos-555424	3815	132	3810	8	8	190	107	75	8	8	145	58	79
neos-826841	5516	156	5436	3	3	200	191	6	4	4	46	0	42
neos-849702	1737	128	1737	2	2	36	34	0	2	2	9	0	7
toll-like	2883	386	1091	26	26	91	44	21	26	26	59	0	33

Table 1. OI narrowings of symmetric instances from MIPLIB2010.

instances, they made no difference at all. Although they provided good results, the few soi-narrowings did not achieve outstanding performances. Interestingly, the SBCs were harmful to all instances of the family map#. We shall investigate why this happens in order to get more insights on the impact of SBCs.

Overall, we understand that the results are few and at most reasonable, but they support our motivation and encourage a more extensive experimental evaluation against a larger set of instances that exhibit nontrivial symmetries.

	Original formulation				Oi-narrowing			
Instance	Best	Time (s)	Gap (%)	St.	Best	Time (s)	Gap (%)	St.
bab5	-106412	1800.10	0.16	lim	-106412	849.51	0	**opt**
blp-ar98	6205.21	1293.84	0	**opt**	6205.6	1800.12	0.12	lim
blp-ic97	4032.94	1800.13	0.69	lim	4025.02	1800.08	**0.39**	lim
core4872-1529	1479	1800.10	3.07	lim	1472	1800.11	**2.56**	lim
gmu-35-40	-2406600	52.83	0	opt	-2406600	**52.51**	0	opt
gmut-75-50	-14176700	1800.48	0.03	lim	-14178800	1800.41	**0.01**	lim
gmut-77-40	-14166700	1800.29	0.04	lim	-14167400	1800.30	**0.03**	lim
iis-bupa-cov	36	1800.04	7.90	lim	36	1800.05	**7.51**	lim
lectsched-4-obj	4	9.05	0	opt	4	**8.15**	0	opt
map06	-289	**683.60**	0	opt	-289	808.28	0	opt
map10	-495	**581.88**	0	opt	-495	719.42	0	opt
map14	-674	**642.28**	0	opt	-674	678.25	0	opt
map18	-847	**271.84**	0	opt	-847	323.00	0	opt
map20	-922	**148.02**	0	opt	-922	167.72	0	opt
mcsched	211913	**320.23**	0	opt	211913	361.66	0	opt
mzzv11	-21718	**20.43**	0	opt	-21718	34.13	0	opt
neos-1426635	-176	1800.83	1.14	lim	-176	1800.19	**0.57**	lim
neos-1426662	-44	1800.41	14.74	lim	-44	1800.35	**13.59**	lim
neos-911880	54.76	7.57	0	opt	54.76	**7.05**	0	opt
neos18	13	25.24	0	opt	13	**15.97**	0	opt
ns2081729	9	**394.59**	0	opt	9	812.08	0	opt
protfold	-25	1800.02	46.34	lim	-27	1800.02	**35.14**	lim
rocII-4-11	-5.65564	**389.22**	0	opt	-5.65564	397.36	0	opt
rvb-sub	27.51	1800.36	58.90	lim	27.4683	1800.31	**58.83**	lim
satellites1-25	-5	**195.57**	0	opt	-5	422.41	0	opt
seymour-disj-10	288	1800.04	1.91	lim	288	1800.04	**1.85**	lim
seymour	307	1800.06	**2.00**	lim	307	1800.06	2.06	lim
swath	467.408	1800.34	11.95	lim	467.408	1800.53	**11.27**	lim
transportmoment	∞	2.69	∞	inf	∞	**2.52**	∞	inf
uc-case3	6931.73	1800.39	0.12	lim	6931.39	1800.40	**0.11**	lim
uct-subprob	315	1800.06	**4.87**	lim	317	1800.09	7.29	lim

	Original formulation				Oi-narrowing				Soi-narrowing			
Instance	Best	Time (s)	Gap (%)	St.	Best	Time (s)	Gap (%)	St.	Best	Time (s)	Gap (%)	St.
core2536-691	683	56.47	0	opt	683	64.99	0	opt	683	**50.04**	0	opt
macrophage	374	755.91	0	opt	374	372.96	0	opt	374	**224.31**	0	opt
neos-555424	1286800	**5.71**	0	opt	1286800	6.68	0	opt	1286800	6.52	0	opt
neos-849702	0	717.74	0	opt	0	**8.68**	0	opt	0	89.52	0	opt
toll-like	617	1800.03	21.81	lim	611	1800.04	**20.33**	lim	617	1800.04	20.38	lim

Table 2. MIPLIB2010 results obtained with CPLEX 12.6.

6 Conclusions

In this paper we discussed the notion of orbital independence by presenting theoretical results that establish sufficient conditions to break symmetries from different orbits of MPs concurrently. These conditions allowed us to design an algorithm that efficiently generates SBCs to the largest independent set of orbits of MPs. We evaluated the impact of our methodology by conducting experiments with symmetric instances taken from MIPLIB2010. The results were at most reasonable but encouraging; we aim to extend our computational tests to a larger set of symmetric instances, either taken from public libraries such as MINLPLIB2 or generated so as to contain formulation groups with specific structures.

Acknowledgments

The first author (GD) is financially supported by a CNPq Ph.D. thesis award.

References

1. Bomze, I., Budinich, M., Pardalos, P., Pelillo, M.: The maximum clique problem. In: Du, D.Z., Pardalos, P. (eds.) Handbook of Combinatorial Optimization, Supp. A, vol. supp. A, pp. 1–74. Kluwer Academic Publishers, Dordrecht (1998)
2. Costa, A., Hansen, P., Liberti, L.: Formulation symmetries in circle packing. In: Mahjoub, R. (ed.) Proceedings of the International Symposium on Combinatorial Optimization. Electronic Notes in Discrete Mathematics, vol. 36, pp. 1303–1310. Elsevier, Amsterdam (2010)
3. Fourer, R., Gay, D.: The AMPL Book. Duxbury Press, Pacific Grove (2002)
4. Galli, S.: Parsing AMPL internal format for linear and non-linear expressions (2004), B.Sc. dissertation, DEI, Politecnico di Milano, Italy
5. IBM: ILOG CPLEX 12.6 User's Manual. IBM (2014)
6. Kaibel, V., Pfetsch, M.: Packing and partitioning orbitopes. Mathematical Programming 114(1), 1–36 (2008)
7. Liberti, L.: Reformulations in mathematical programming: Definitions and systematics. RAIRO-RO 43(1), 55–86 (2009)
8. Liberti, L.: Reformulations in mathematical programming: Automatic symmetry detection and exploitation. Mathematical Programming A 131, 273–304 (2012)
9. Liberti, L.: Symmetry in mathematical programming. In: Lee, J., Leyffer, S. (eds.) Mixed Integer Nonlinear Programming, IMA, vol. 154, pp. 263–286. Springer, New York (2012)
10. Liberti, L., Cafieri, S., Savourey, D.: Reformulation optimization software engine. In: Fukuda, K., van der Hoeven, J., Joswig, M., Takayama, N. (eds.) Mathematical Software. LNCS, vol. 6327, pp. 303–314. Springer, New York (2010)
11. Liberti, L., Ostrowski, J.: Stabilizer-based symmetry breaking constraints for mathematical programs. Journal of Global Optimization 60, 183–194 (2014)
12. Margot, F.: Pruning by isomorphism in branch-and-cut. Mathematical Programming 94, 71–90 (2002)
13. Margot, F.: Exploiting orbits in symmetric ILP. Mathematical Programming B 98, 3–21 (2003)
14. Margot, F.: Symmetry in integer linear programming. In: Jünger, M., Liebling, T., Naddef, D., Nemhauser, G., Pulleyblank, W., Reinelt, G., Rinaldi, G., Wolsey, L. (eds.) 50 Years of Integer Programming, pp. 647–681. Springer, Berlin (2010)
15. McKay, B.: Practical graph isomorphism. Congressus Numerantium 30, 45–87 (1981)
16. McKay, B., Piperno, A.: Practical graph isomorphism, II. Journal of Symbolic Computation 60, 94–112 (2014)
17. Ostrowski, J., Linderoth, J., Rossi, F., Smriglio, S.: Constraint orbital branching. In: Lodi, A., Panconesi, A., Rinaldi, G. (eds.) IPCO. LNCS, vol. 5035, pp. 225–239. Springer (2008)
18. Ostrowski, J., Linderoth, J., Rossi, F., Smriglio, S.: Orbital branching. Mathematical Programming 126, 147–178 (2011)
19. The GAP Group: GAP - Groups, Algorithms and Programming. Version 4.7.4 (2014)