

Accepted Manuscript

Preprocessing and Cut Generation Techniques for Multi-objective Binary Programming

Natashia Boland, Hadi Charkhgard, Martin Savelsbergh

PII: S0377-2217(18)30887-7
DOI: <https://doi.org/10.1016/j.ejor.2018.10.034>
Reference: EOR 15426



To appear in: *European Journal of Operational Research*

Received date: 8 September 2017
Revised date: 6 August 2018
Accepted date: 17 October 2018

Please cite this article as: Natashia Boland, Hadi Charkhgard, Martin Savelsbergh, Preprocessing and Cut Generation Techniques for Multi-objective Binary Programming, *European Journal of Operational Research* (2018), doi: <https://doi.org/10.1016/j.ejor.2018.10.034>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Highlights

- We study conditions under which multi-objective binary programs have ideal points.
- We prove the computational complexity of identifying such conditions.
- We show that studying such conditions is useful even if there is no ideal point.
- We develop some enhancement techniques by studying such conditions.

ACCEPTED MANUSCRIPT

Preprocessing and Cut Generation Techniques for Multi-objective Binary Programming

Natashia Boland^b, Hadi Charkhgard^{a,*}, Martin Savelsbergh^b^a*Department of Industrial and Management Systems Engineering, University of South Florida, USA*^b*School of Industrial and Systems Engineering, Georgia Institute of Technology, USA***Abstract**

We present the theoretical foundations for a number of preprocessing and cut generation techniques for multi-objective binary programs. The techniques are based on a characterization of conditions under which the objective functions of a multi-objective binary program guarantee the existence of an ideal point in criterion space, i.e., the existence of a feasible solution that simultaneously minimizes all objectives. Even though few multi-objective binary programs of interest have objective functions satisfying these conditions, the conditions are likely to be satisfied for a subset of the objective functions and/or be satisfied when the objective functions are restricted to a subset of the variables. We show that recognizing whether or not this occurs is NP-hard, but can be done in pseudo-polynomial time. The preprocessing and cut generation techniques can be incorporated in any decision or criterion space search algorithm for multi-objective binary programs. Preliminary computational tests demonstrate their potential in practice.

Keywords: multi-objective binary program, ideal point, computational complexity, preprocessing, cut generation

1. Introduction

Multi-objective binary programs (MBPs) are optimization problems in which all decision variables are binary and there is a set of linear objective functions. The goal in multi-objective optimization is to identify the nondominated frontier: the set of vectors of objective function values associated with *efficient* solutions, i.e., feasible solutions in which it is impossible to improve the value of one objective without making the value of at least one other objective worse. For a general introduction to multi-objective optimization, see Ehrgott (2005) and Ehrgott and Gandibleux (2002). MBPs can be viewed as multi-objective combinatorial optimization problems; for an overview of this area, see Ehrgott and Gandibleux (2000, 2007).

Because many problems in practice can be formulated as MBPs, the development of efficient and reliable MBP solvers can have significant benefits. Consequently, research on algorithms for MBPs has increased substantially in the last two decades. For an introduction to algorithms for solving general MBPs as well as examples of recent developments, see Özpeynirci and Köksalan (2010); Lokman and Köksalan (2013); Stidsen et al. (2014); Kirlik and Sayın (2014); Parragh and Tricoire (2015); Belotti et al. (2016); Soylu and Yıldız (2016); Boland et al. (2016, 2017a), Dächert et al. (2017), and Soylu (2018).

Given the interest in algorithms for solving MBPs, it is surprising that little or no research has been published on preprocessing and cut generation for MBPs. Preprocessing and cut generation techniques are core components of commercial and open-source single-objective integer programming solvers, many of these specifically designed for 0-1 or binary integer programs. Preprocessing techniques aimed at fixing variables and cut generation techniques aimed at strengthening the linear programming relaxations have a long and continuing history in single-objective integer programming, e.g., Savelsbergh (1994); Gu et al. (1998); Atamtürk et al. (2000); Cornuéjols (2008), and Morrison et al. (2016). Therefore, a primary goal of this study is to develop the theoretical foundations for a number of preprocessing and cut generation techniques for MBPs.

*Corresponding author

Email address: hcharkhgard@usf.edu (Hadi Charkhgard)

In order to develop preprocessing and cut generation techniques for MBPs, we study properties of the objective functions. This is, in part, because recent studies have shown that properties of the objective functions can be used to deduce information about the set of nondominated points. For example, from properties of objective functions it may be possible to deduce that the number of nondominated points grows exponentially with the input size, or that there will only be supported nondominated points. Examples of results that have been obtained recently for MBPs, in part based on studying or characterizing objective functions (where N denotes the number of objective functions and n denotes the number of 0-1 variables) are:

- When the coefficients of $N - 1$ of the objective functions are binary, then the number of nondominated points is bounded by $(n + 1)^{N-1}$, and when the coefficients of $N - 1$ of the objective functions are integer and the largest coefficient in any of these $N - 1$ objective functions is polynomially bounded in n , then the number of nondominated points is also polynomially bounded (Figueira et al., 2017).
- When the set of feasible solutions is the set of bases of a matroid, $N = 2$, and the coefficients of one of the objective functions are binary, then all nondominated points are supported and can be computed in polynomial time (Gabow and Tarjan, 1984; Gorski, 2010).
- All nondominated points of a biobjective minimum spanning tree problem with objective coefficients in $\{0, 1, 2\}$ are supported, and the number of nondominated points is bounded by $2n - 1$ (Seipp, 2013).
- MBPs with binary coefficients in $N - 1$ of the objective functions can be solved in $\mathcal{O}(n \log n)$ for $N = 2$ and in $\mathcal{O}(n^2)$ for $N = 3$ (Gorski et al., 2012).

We refer the interested reader to Figueira et al. (2017) for more information on these recent developments.

In light of the above, the starting point for our investigation is studying conditions under which the objective functions of a MBP guarantee that the nondominated frontier consists of a single point. In this case, we say that the MBP *has an ideal point*¹. We say that a set of objective functions is *universally co-ideal* if any MBP with at least one feasible solution and this set of objective functions has an ideal point. However, the goal of the research we present here is not to recognize whether or not a MBP has an ideal solution, although our results do have something to offer in that respect. The goal is to discover properties that lead to a MBP having an ideal solution, and then to explore if and how these properties can be exploited in situations when a MBP does not have an ideal solution. For example, if a substructure of the problem has these properties, when looking at the substructure in isolation, can that knowledge help in tackling the overall problem? More specifically, can knowledge and understanding of properties that induce the existence of ideal solutions to MBPs be used to develop techniques that can enhance the solution of arbitrary MBPs? This paper is the first to explore such questions and provide answers to them.

The following core results concerning universal co-idealness are obtained. First, there are computational complexity results. These are significant not least because of how little is known about the complexity of MBPs. For example, it is not known if there exist algorithms that run in time polynomial in the length of the input and the output (so-called *output-sensitive algorithms*), even for the case in which the feasible set has *no* constraints and there are only two objectives (Böklér et al. (2017)). Our results provide sufficient conditions for an MBP to have an ideal point that can be checked in pseudopolynomial time, even if the single-objective MBP is strongly NP-Hard. Second, there are algorithm enhancement results. More specifically, we show how to exploit the case that the MBP does not have an ideal point but a subset of its objectives are universally co-ideal in order to reduce the number of objectives, and we show how to exploit the case that the set of objectives has the universally co-ideal property when restricted to a subset of the variables in order to derive valid inequalities. Note that this situation can be expected to occur quite often. For example, in problems with two objectives, any set of variables over which one of the objectives has identical coefficients gives rise to it, while in problems with any number of objectives, any pair of variables ranked the same way by all objectives yields a universally co-ideal substructure.

The results may be summarized as follows.

¹This is somewhat different to the common usage of the term, in which an ideal point is a vector lower bound on all nondominated points that may not, itself, be the vector of objective values of a feasible solution.

- Whether a set of objective functions is universally co-ideal can be determined in pseudo-polynomial time. This is the best possible (unless $NP=P$), because the problem of determining whether a set of objective functions is universally co-ideal is (weakly) NP-hard.
- When a subset of objectives of a MBP is universally co-ideal, this subset of objective functions can be replaced by a single objective function without changing the set of efficient solutions. (In fact, an even stronger statement can be made, see Section 5.1).
- When there exists a subset of variables with the property that the set of objective functions, when restricted to the variables in the subset, is universally co-ideal, then, given an efficient solution \bar{x} , a cut can be generated that is valid for all efficient solutions, x , that do not yield the same nondominated point as \bar{x} .

In the cut referred to above, the nondominated point corresponding to the efficient solution \bar{x} is an ideal point for the objective functions restricted to variables in the, so-called, *objective-aligning* set. This property can be generalized: *from* a given efficient solution, a subset of variables that is *not* necessarily objective-aligning, *per se*, but that nevertheless induces the corresponding nondominated point to be ideal, can be sought. Again, a cut can be derived, but in this case under weaker conditions.

In summary, our search for conditions on a set of objective functions of a MBP that guarantee the existence of an ideal point, i.e., our investigation of universal co-ideality, has resulted in further insights into the relationship between the characteristics of the objective functions of a MBP and the characteristics of its nondominated frontier. It has also lead, via its application to problem substructures, to the discovery of a number of techniques that have the potential to enhance the performance of algorithms for solving general MBPs.

The remainder of the paper is organized as follows. In Section 2, we introduce notation and a few basic results. In Section 3, we formally introduce the universally co-ideal concept and some basic theoretical results. In Section 4, we discuss the computational complexity of recognizing universal co-ideality. In Section 5, we introduce different ways to exploit the concept of universal co-ideality, so as to reduce the objective space dimension and/or deduce cuts. In Section 6, we indicate how the concept of universal co-ideality can be extended to make use of relaxations of the feasible set, which enables it to be even more powerful. Initial evidence of the potential benefits of these ideas for algorithms is given in the form of numerical examples, used throughout the paper, and via a small computational study, in Section 7. Finally, in Section 8, we give concluding remarks.

2. Preliminaries

A *Multi-objective Binary Program* can be formulated as follows:

$$\min_{x \in F} \{c^1 x, c^2 x, \dots, c^N x\} \quad (1)$$

where $c^1, c^2, \dots, c^N \in \mathbb{R}^n$ are row vectors that represent the objective functions (or criteria) and $F \subseteq \{0, 1\}^n$ represents the *feasible set in the decision space*, assumed to be a subset of the binary column vectors of dimension n . We denote the $N \times n$ matrix with the i th row given by c^i , for each $i = 1, \dots, N$, by C , which is called the *objective function matrix*. For a given *solution*, $x \in \{0, 1\}^n$, in decision space, its image in criterion space may be compactly written as Cx . Note that it is convenient to treat row and column vectors in criterion space as indistinguishable, and so $(c^1 x, \dots, c^N x)$ and Cx will be treated as identical. We use $Z := \{Cx : x \in \{0, 1\}^n\}$ to denote the set of all *points* in the criterion space that are the image of a solution in decision space and use $O \subseteq Z$, given by $O := \{Cx : x \in F\}$, to denote the *feasible set in the criterion space*. To aid in distinguishing the two spaces, we usually refer to elements of the decision space as solutions and elements of the criterion space as points. A MBP is defined by the pair (C, F) .

Definition 1. A solution $x^I \in F$ is ideal if $c^i x^I = \min_{x \in F} c^i x$ for all $i \in \{1, \dots, N\}$. We refer to $z^I := Cx^I$ as the ideal point.

Note that the term “ideal point” is widely used regardless of whether an ideal solution exists or not. In other words, the ideal point z^I is defined as $z_i^I = \min_{x \in F} c^i x$ for each $i = 1, \dots, N$, even when there does not exist $x \in F$ such that $z^I = Cx$. Here, by contrast, we use the term only if an ideal solution exists (i.e., we say a MBP has an ideal point only if $\bigcap_{i=1}^N \arg \min_{x \in F} c^i x \neq \emptyset$.)

Definition 2. A feasible solution $x \in F$ is called *efficient*, if there is no $x' \in F$ such that $Cx' \leq Cx$ and $Cx' \neq Cx$. If x is efficient, then Cx is called a *nondominated point*. The set of all efficient solutions is denoted by F_E . The set of all nondominated points, $Cx \in O$ for some $x \in F_E$, is denoted by O_N , and referred to as the *nondominated frontier*.

Note that we use the usual vector inequality notation, $p \leq q$, for a pair of vectors, $p, q \in \mathbb{R}^m$, for some $m \in \mathbb{Z}_+$, to indicate that $p_i \leq q_i$ for all $i = 1, \dots, m$; we do not assume, as is sometimes the case in multi-objective optimization, that $p \leq q$ implies that $p \neq q$.

Given a solution $x \in \{0, 1\}^n$, we denote the *support* of x by $\text{supp}(x)$, i.e., $\text{supp}(x) := \{j \in \{1, \dots, n\} : x_j = 1\}$. Moreover, given a set $S \subseteq \{1, \dots, n\}$, and a row vector $c \in \mathbb{R}^n$, we write $c(S)$ to denote $\sum_{j \in S} c_j$, and have that $c(\text{supp}(x)) = \sum_{j \in \text{supp}(x)} c_j = \sum_{j=1}^n c_j x_j = cx$.

A key definition in understanding ideal point properties of a MBP is the following. There are several different, equivalent, ways of stating this property, each of which is convenient, depending on the circumstance in which it is used.

Definition 3. A set of objectives, $c^1, c^2, \dots, c^N \in \mathbb{R}^n$, is *sortable-by-subsets* when, for every pair of sets $S, S' \subseteq \{1, \dots, n\}$, either

- (a) $c^i(S) \leq c^i(S')$ for all $i = 1, \dots, N$, or
- (b) $c^i(S) \geq c^i(S')$ for all $i = 1, \dots, N$.

Stating that c^1, c^2, \dots, c^N is sortable-by-subsets is equivalent to stating that the vector inequality relation, \leq , totally orders Z . To see this, assume that c^1, c^2, \dots, c^N is sortable-by-subsets. Then, if $p, p' \in Z$ are two arbitrarily chosen points in Z , say $p = Cx$ and $p' = Cx'$, for $x, x' \in \{0, 1\}^n$, either $p = p'$ or, for some i , $p_i < p'_i$ (without loss of generality). In the latter case, $p_i = c^i(\text{supp}(x)) < c^i(\text{supp}(x')) = p'_i$, so, by Definition 3 (a), $c^h(\text{supp}(x)) \leq c^h(\text{supp}(x'))$, for all $h = 1, \dots, N$. We know that $p_h = c^h(\text{supp}(x)) \leq c^h(\text{supp}(x')) = p'_h$, for all $h = 1, \dots, N$, so $p \leq p'$. This shows that \leq totally orders Z . The converse is similar.

Another, equivalent, way of stating the sortable-by-subsets property, is obtained from the definition below.

Definition 4. Two sets $S, S' \subseteq \{1, \dots, n\}$ are *mutually nondominated (MND)* for a set of objectives, $c^1, c^2, \dots, c^N \in \mathbb{R}^n$, if there exist $i, j \in \{1, \dots, N\}$ such that $c^i(S) < c^i(S')$ and $c^j(S) > c^j(S')$.

Stating that the set of objectives, $c^1, c^2, \dots, c^N \in \mathbb{R}^n$, is sortable-by-subsets is equivalent to stating that there does **not** exist a pair of subsets of $\{1, \dots, n\}$ that are MND for this set of objectives.

The following lemma and its corollary are also helpful. As the proof of each is straightforward, these proofs are omitted.

Lemma 1. Let $S, S' \subseteq \{1, \dots, n\}$. Then S and S' are MND if and only if $S \setminus S'$ and $S' \setminus S$ are MND.

Corollary 1. For a set of objectives, c^1, c^2, \dots, c^N , to be sortable-by-subsets, it suffices that either $c^i(S) \leq c^i(S')$ for all $i = 1, \dots, N$, or $c^i(S) \geq c^i(S')$ for all $i = 1, \dots, N$ holds for every pair of disjoint sets $S, S' \subseteq \{1, \dots, n\}$.

3. Universally co-ideal

Next, we introduce the concept that forms the foundation of this study. It defines a property of objectives that ensures that any (feasible) MBP with this set of objectives, irrespective of its feasible set, must have an ideal point.

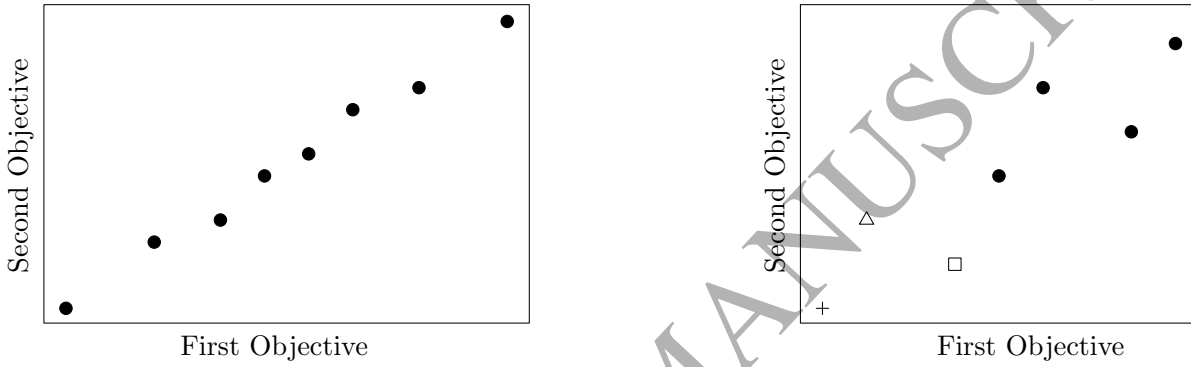
Definition 5. A set of objectives, $c^1, c^2, \dots, c^N \in \mathbb{R}^n$, is universally co-ideal if, for **any** nonempty feasible set, $F \subseteq \{0, 1\}^n$, the MBP

$$\min_{x \in F} \{c^1 x, c^2 x, \dots, c^N x\}$$

has an ideal point, i.e.,

$$\bigcap_{i=1}^N \arg \min_{x \in F} c^i x \neq \emptyset.$$

The following proposition shows that the sortable-by-subsets property (or, equivalently, the property that \leq totally orders Z) characterizes when a set of objectives is universally co-ideal. The correspondence between \leq totally ordering Z and the set of objectives being universally co-ideal is illustrated in Figure 1, which plots Z for two sets of (two) objectives. In Figure 1a, it is clear that Z is totally ordered by \leq , and that, irrespective of which subset of Z is feasible for a MBP, (which points constitute O), there must be a “bottom left” feasible point that dominates the others, and hence the MBP has an ideal point. By contrast, in Figure 1b, the pair of points plotted as an unfilled triangle and an unfilled square are not ordered by \leq , and clearly any feasible set that contains both those points and does not contain the bottom left point, plotted as a plus symbol, cannot have an ideal point.



(a) Objectives that are universally co-ideal:
 $c^1 = (4, 7, 9)$ and $c^2 = (3, 4, 6)$.

(b) Objectives that are not universally co-ideal:
 $c^1 = (1, 3, 4)$ and $c^2 = (2, 1, 3)$.

Figure 1: Illustration of sets of objectives that are, and are not, universally co-ideal. The figures show the set Z , for each set of objectives.

Proposition 1. A set of objectives, $c^1, c^2, \dots, c^N \in \mathbb{R}^n$, is universally co-ideal if and only if it is sortable-by-subsets.

PROOF. Suppose the set of objectives is not universally co-ideal, so there exists $F \subseteq \{0, 1\}^n$, $F \neq \emptyset$, with

$$\bigcap_{i=1}^N \arg \min_{x \in F} c^i x = \emptyset.$$

Then there must exist $k \in \{2, \dots, N\}$ with

$$G^{k-1} := \bigcap_{i=1}^{k-1} \arg \min_{x \in F} c^i x \neq \emptyset \quad \text{and} \quad \bigcap_{i=1}^k \arg \min_{x \in F} c^i x = \emptyset, \quad \text{i.e.,} \quad G^{k-1} \cap \arg \min_{x \in F} c^k x = \emptyset.$$

Now let $y \in G^{k-1}$ and $y' \in \arg \min_{x \in F} c^k x$. Then $y \notin \arg \min_{x \in F} c^k x$, so $c^k y > c^k y'$. Also, there must exist $i \in \{1, \dots, k-1\}$ with $y' \notin \arg \min_{x \in F} c^i x$. But $y \in G^{k-1} \subseteq \arg \min_{x \in F} c^i x$, so $c^i y < c^i y'$. Let $S = \text{supp}(y)$, and $S' = \text{supp}(y')$. Then

$$c^i(S) = c^i y < c^i y' = c^i(S')$$

while

$$c^k(S) = c^k y > c^k y' = c^k(S'),$$

which shows the objectives are not sortable-by-subsets.

Now suppose that the set of objectives is not sortable-by-subsets, so there exists $S, S' \subseteq \{0, 1\}^n$ and $i, k \in \{1, \dots, N\}$, $i \neq k$, with $c^i(S) < c^i(S')$ and $c^k(S) > c^k(S')$ (so obviously $S \neq S'$). Let $y \in \{0, 1\}^n$ be the indicator vector of S and $y' \in \{0, 1\}^n$ be the indicator vector of S' , (so $y \neq y'$), and set $F = \{y, y'\}$. Then $\arg \min_{x \in F} c^i x = \{y\}$ and $\arg \min_{x \in F} c^k x = \{y'\}$, and the result follows. \square

To decide that a set of objectives is sortable-by-subsets, direct use of the sortable-by-subsets definition would appear to require enumeration of $\mathcal{O}(2^n)$ sets. Here we give a necessary condition that is easy to check.

Definition 6. A set of objectives, $c^1, c^2, \dots, c^N \in \mathbb{R}^n$, is elementwise sortable if, for every $j, k \in \{1, \dots, n\}$, either $c_j^i \geq c_k^i$ for all $i = 1, \dots, N$, or $c_j^i \leq c_k^i$ for all $i = 1, \dots, N$. In this case, there exists a permutation, $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ of the cost vector indices such that

$$c_{\sigma(1)}^i \leq c_{\sigma(2)}^i \leq \dots \leq c_{\sigma(n)}^i, \quad \forall i = 1, \dots, N.$$

Example 1. Suppose that a MBP with four decision variables has the following two objective functions: $x_1 + 3x_2 + 6x_3 + 4x_4$ and $3x_1 + 5x_2 + 8x_3 + 7x_4$. In this case, $N = 2$ and we have that $c^1 = (1, 3, 6, 4)$ and $c^2 = (3, 5, 8, 7)$. By definition, c^1 and c^2 are elementwise sortable since we can set $\sigma : \{1, 2, 3, 4\} \rightarrow \{1, 2, 4, 3\}$. Intuitively, σ replaces all instances of x_3 by x_4 and all instances of x_4 by x_3 in the formulation of the MBP. (This essentially implies that we have renamed the decision variables and so the formulation has not changed.) Applying σ , we obtain $x_1 + 3x_2 + 4x_3 + 6x_4$ and $3x_1 + 5x_2 + 7x_3 + 8x_4$, i.e., $c^1 = (1, 3, 4, 6)$ and $c^2 = (3, 5, 7, 8)$, and we have $1 < 3 < 4 < 6$ and $3 < 5 < 7 < 8$.

Corollary 2 (to Proposition 1). If a set of objectives $c^1, c^2, \dots, c^N \in \mathbb{R}^n$ is not elementwise sortable then it is not universally co-ideal.

PROOF. Elementwise sortable is necessary for a set of objectives to be sortable-by-subsets, since it is precisely the same condition restricted to pairs of sets of cardinality 1. \square

Of course, if a set of objectives is sortable-by-subsets then it is sortable by subsets of cardinality not greater than a specified value, k , and checking this property can be done efficiently, in practice, for any small value of k ; elementwise sortable is simply the special case of $k = 1$.

We can also give a sufficient condition that is easy to check, in the case that all entries in all objectives have the same sign, and that each objective “increases” at a sufficiently fast rate. For example, it is not difficult to check that $c^1 = (1, 3, 5, 11)$, $c^2 = (1, 5, 7, 13)$ are universally co-ideal, since $3 \geq 1$, $5 \geq 1 + 3$ and $11 \geq 1 + 3 + 5$ and since $5 \geq 1$, $7 \geq 1 + 5$ and $13 \geq 1 + 5 + 7$. (Similarly, $c^1 = (-1, -3, -5, -11)$, $c^2 = (-1, -5, -7, -13)$ are universally co-ideal.) The sufficient condition may be stated as follows.

Corollary 3 (to Proposition 1). If a set of non-negative objectives, $c^1, c^2, \dots, c^N \in \mathbb{R}_+^n$, or non-positive objectives, $c^1, c^2, \dots, c^N \in \mathbb{R}_-^n$, satisfies

$$|c_j^i| \geq \sum_{k=1}^{j-1} |c_k^i|, \quad \forall j = 2, \dots, n, \quad \forall i = 1, \dots, N, \quad (2)$$

where $|\cdot|$ denotes the absolute value, then they are universally co-ideal.

PROOF. We only prove the case of non-negative objectives; the case of non-positive objectives can be proved similarly.

Suppose that (2) holds, but the set of objectives is not sortable-by-subsets. Then there exists $i, i' \in \{1, \dots, N\}$ and disjoint sets $S, S' \subseteq \{1, \dots, n\}$ with $c^i(S) > c^i(S')$ but $c^{i'}(S) < c^{i'}(S')$. Let j^* be the highest index of any element of S , i.e., $j^* = \arg \max\{j : j \in S\}$ and k^* be the highest index of any element of S' , i.e., $k^* = \arg \max\{k : k \in S'\}$. Since S and S' are disjoint, $j^* \neq k^*$. Since $c^i(S) > c^i(S')$, it must be that $j^* > k^*$, as, otherwise, $S \subseteq \{1, \dots, j^*\} \subseteq \{1, \dots, k^* - 1\}$ and $c^i(S') \geq c_{k^*}^i \geq \sum_{j=1}^{k^*-1} c_j^i \geq c^i(S)$, which is a contradiction. Similarly, it must be that $k^* < j^*$ since $c^{i'}(S) < c^{i'}(S')$. Since $j^* > k^*$ and $j^* < k^*$ cannot both hold, we obtain a contradiction, and the objectives must be sortable-by-subsets. The result follows by Proposition 1. \square

This sufficient condition is not necessary, as the example given in Figure 1a shows: the objectives $c^1 = (4, 7, 9), c^2 = (3, 4, 6)$ are universally co-ideal, but $9 \not\geq 4 + 7$. In fact, as we will show in the next section, there is no easily checked condition that is both necessary and sufficient, however the universally co-ideal property can be checked in time that is only polynomially dependent on n (and N); it can be checked in pseudo-polynomial time.

4. Computational complexity

We now consider the computational complexity of the decision problem **UNIVERSALLY CO-IDEAL**: given inputs $c^1, \dots, c^N \in \mathbb{Z}^n$, are these universally co-ideal? Note that for this analysis, we assume integer-valued objective vectors.

We first observe that **UNIVERSALLY CO-IDEAL** can be reduced to **2-UNIVERSALLY CO-IDEAL**: given instance (f, g) , where $f, g \in \mathbb{Z}^n$, is it true that for all $S, S' \subseteq \{1, \dots, n\}$, if $f(S) < f(S')$ then $g(S) \leq g(S')$ and if $f(S) > f(S')$ then $g(S) \geq g(S')$? By Proposition 1, this is equivalent to asking that (f, g) is universally co-ideal. We claim that an instance, (c^1, \dots, c^N) , of **UNIVERSALLY CO-IDEAL**, is a YES instance if and only if for all $i, k \in \{1, \dots, N\}$, the instance (c^i, c^k) of **2-UNIVERSALLY CO-IDEAL** is a YES instance. To see the “if” direction, we use Proposition 1, as follows. Let $S, S' \subseteq \{1, \dots, n\}$, and suppose, without loss of generality, that $c^1(S) < c^1(S')$. Then for all $i \in \{2, \dots, N\}$, since (c^1, c^i) is a YES instance of **2-UNIVERSALLY CO-IDEAL**, it must be that $c^i(S) \leq c^i(S')$. The “only if” direction is obvious. Thus an instance of **UNIVERSALLY CO-IDEAL** can be solved by solving $\mathcal{O}(N^2)$ instances of **2-UNIVERSALLY CO-IDEAL**, and hence the latter must be at least as hard.

We now show that **2-UNIVERSALLY CO-IDEAL** is pseudo-polynomially solvable, and hence so is **UNIVERSALLY CO-IDEAL**. Given an instance (f, g) , of **2-UNIVERSALLY CO-IDEAL**, where $f, g \in \mathbb{Z}^n$, we construct the following 0-1 knapsack optimization problem:

$$\begin{aligned} \psi(f, g) = \max \quad & fx - gy \\ \text{s.t.} \quad & gx - gy \leq -1, \\ & x, y \in \{0, 1\}^n. \end{aligned}$$

Then the instance (f, g) is a YES instance if and only if $\psi(f, g) \leq 0$. Since 0-1 knapsack optimization is pseudo-polynomial, the result follows.

Note that the 0-1 knapsack optimization problem above solves both the **2-UNIVERSALLY CO-IDEAL** instance and its complement: do there exist $S, S' \subseteq \{1, \dots, n\}$ with either $f(S) < f(S')$ and $g(S) > g(S')$ or $f(S) > f(S')$ and $g(S) < g(S')$? The answer is YES if and only if $\psi(f, g) > 0$, or, equivalently, $\psi(f, g) \geq 1$.

We now show that the complement of **2-UNIVERSALLY CO-IDEAL** is NP-complete.

Theorem 1. *The complement of 2-UNIVERSALLY CO-IDEAL is NP-complete.*

PROOF. Note first that the complement of **2-UNIVERSALLY CO-IDEAL** is obviously in NP.

We now show that **EQUAL-SUBSET-SUM** (Woeginger and Yu, 1992) is polynomially reducible to the complement of **2-UNIVERSALLY CO-IDEAL**, where **EQUAL-SUBSET-SUM** is defined as follows: given a set $\{a_1, \dots, a_n\}$ of positive integers, do there exist non-empty disjoint subsets, $S, S' \subseteq \{1, \dots, n\}$, with $\sum_{j \in S} a_j = \sum_{j \in S'} a_j$?

Given such an instance of **EQUAL-SUBSET-SUM**, we construct an instance of **2-UNIVERSALLY CO-IDEAL**, (f, g) , by

$$f_j = a_j - 2^{-j},$$

and

$$g_j = a_j + 2^{-j},$$

for all $j = 1, \dots, n$. Observe that for any $S \subseteq \{1, \dots, n\}$,

$$\lceil f(S) \rceil = \lfloor g(S) \rfloor = \sum_{j \in S} a_j, \quad (3)$$

since $0 \leq \sum_{j \in S} 2^{-j} < 1$ and hence

$$\lceil f(S) \rceil = \lceil \sum_{j \in S} a_j - \sum_{j \in S} 2^{-j} \rceil = \sum_{j \in S} a_j = \lfloor \sum_{j \in S} a_j + \sum_{j \in S} 2^{-j} \rfloor = \lfloor g(S) \rfloor.$$

Now suppose that this instance, (f, g) , of 2-UNIVERSALLY CO-IDEAL, is a NO instance, so there exist, (by Proposition 1 and Corollary 1), non-empty disjoint sets, $S, S' \subset \{1, \dots, n\}$, with $f(S) < f(S')$ and $g(S) > g(S')$. Then $f(S) < f(S')$ implies that

$$\lceil f(S) \rceil \leq \lceil f(S') \rceil = \lfloor g(S') \rfloor \leq \lfloor g(S) \rfloor,$$

by (3) and since $g(S') < g(S)$. But, also by (3), we have $\lceil f(S) \rceil = \lfloor g(S) \rfloor$, and hence, again by (3), it must be that

$$\sum_{j \in S} a_j = \lceil f(S) \rceil = \lfloor g(S) \rfloor = \lceil f(S') \rceil = \lfloor g(S') \rfloor = \sum_{j \in S'} a_j,$$

so EQUAL-SUBSET-SUM is a YES instance.

Finally, suppose that EQUAL-SUBSET-SUM is a YES instance, with non-empty, disjoint sets, $S, S' \subset \{1, \dots, n\}$, with $\sum_{j \in S} a_j = \sum_{j \in S'} a_j$. Let $m := \min\{j : j \in S \cup S'\}$ and, without loss of generality, assume that $m \in S$. Observe that

$$\sum_{j \in S} 2^{-j} \geq 2^{-m} > \sum_{j \in S'} 2^{-j}.$$

Hence

$$g(S) = \sum_{j \in S} a_j + \sum_{j \in S} 2^{-j} \geq \sum_{j \in S} a_j + 2^{-m} = \sum_{j \in S'} a_j + 2^{-m} > \sum_{j \in S'} a_j + \sum_{j \in S'} 2^{-j} = g(S').$$

Similarly,

$$f(S) = \sum_{j \in S} a_j - \sum_{j \in S} 2^{-j} \leq \sum_{j \in S} a_j - 2^{-m} = \sum_{j \in S'} a_j - 2^{-m} < \sum_{j \in S'} a_j - \sum_{j \in S'} 2^{-j} = f(S').$$

Thus the 2-UNIVERSALLY CO-IDEAL instance, (f, g) , is a NO instance. The result follows. \square

Corollary 4. 2-UNIVERSALLY CO-IDEAL, and hence UNIVERSALLY CO-IDEAL, is NP-hard.

5. Exploiting universal co-idealness

The nondominated points of a MBP are sometimes classified as *supported* and *unsupported* nondominated points, i.e., those that can be obtained by minimizing a positive weighted sum of the objective functions over the feasible set and those that cannot be obtained in that way. The existence of unsupported nondominated points is what makes MBPs especially hard to solve. Our results are valid for any MBP, regardless of whether its nondominated frontier contains unsupported nondominated points or not.

5.1. Reducing the number of objectives

In this section, we show how universal co-idealness can be exploited to reduce the number of objectives in a MBP. Reducing the number of objectives in a multi-objective optimization problem has, so far, been studied mostly in the context of multi-objective linear programming; interested readers are referred to Gal and Hanne (1999, 2006); Engau and Wiecek (2008); Lindroth et al. (2010); Malinowska and Torres (2008); Malinowska (2008), and Thoai (2012).

We begin by defining a notion of equivalence for MBPs. Note that if F_E is the efficient set of a MBP, $\min_{x \in F} \{c^1x, \dots, c^Nx\}$, with objective function matrix C , and O_N is the set of its nondominated points, so $O_N = \{Cx : x \in F_E\}$, then O_N induces a partition of F_E , as follows. For each $z \in O_N$, let $F_E(z)$ denote the set of efficient solutions that yield criterion space image, z , so $F_E(z) = \{x \in F_E : Cx = z\}$. Then the collection of sets $\{F_E(z) : z \in O_N\}$ partitions F_E .

Definition 7. Let $P^1 := \min_{x \in F} \{c^1x, c^2x, \dots, c^{k-1}x, c^kx, \dots, c^Nx\}$ with $c^i \in \mathbb{R}^n$ for $i = 1, \dots, N$ be a MBP and let $P^2 := \min_{x \in F} \{c^1x, c^2x, \dots, c^{k-1}x, \bar{c}^kx\}$ with $\bar{c}^k \in \mathbb{R}^n$ be another MBP. Furthermore, let F_E^1, O_N^1 and F_E^2, O_N^2 be the sets of efficient solutions and nondominated points of P^1 and P^2 , respectively. We say P^2 is equivalent to P^1 if $F_E^1 = F_E^2 =: F_E$ and the partition of F_E induced by O_N^2 is identical to that induced by O_N^1 .

If P^2 is equivalent to P^1 , then solving P^2 solves P^1 . To see this, observe that solving P^2 means finding O_N^2 and, for each $z \in O_N^2$, identifying at least one $x(z) \in F_E^2$ with $(c^1x, c^2x, \dots, c^{k-1}x, \bar{c}^kx) = z$, i.e., finding at least one member of each element of the partition of F_E^2 induced by O_N^2 . But $F_E^1 = F_E^2$ and the partition of F_E^1 induced by O_N^1 is identical to the partition induced by O_N^2 , thus, by solving P^2 , we have, in fact, identified at least one member of each element in the partition of F_E^1 induced by O_N^1 . By calculating the objective function, $(c^1x, c^2x, \dots, c^Nx)$, for each such member, x , we must obtain O_N^1 , as required.

We now show that if a MBP has a subset of objectives that is universally co-ideal, then replacing these objectives by any positive combination of them yields an equivalent MBP, having fewer objectives.

Proposition 2. Let $P^1 := \min_{x \in F} \{c^1x, c^2x, \dots, c^Nx\}$ be a MBP and let c^k, \dots, c^N for $k \geq 2$ be universally co-ideal. Then $P^2 := \min_{x \in F} \{c^1x, c^2x, \dots, c^{k-1}x, \sum_{i=k}^N \lambda_i c^i x\}$, with $\lambda_i > 0$ for $i = k, \dots, N$, is an equivalent MBP.

PROOF. Let F_E^1, O_N^1 and F_E^2, O_N^2 denote the sets of efficient solutions and nondominated points of P^1 and P^2 , respectively. Let C denote the $N \times n$ objective function matrix of P^1 , with i th row given by (row vector) c^i , for $i = 1, \dots, N$. Let \bar{C} denote the $k \times n$ objective function matrix of P^2 , with i th row given by c^i for $i = 1, \dots, k-1$ and k th row given by $\sum_{i=k}^N \lambda_i c^i$.

It is straightforward to show that $F_E^2 \subseteq F_E^1$. Let $x \in F_E^2$. Now suppose that $x \notin F_E^1$, so it must be that $Cx' \leq Cx$ and $Cx' \neq Cx$, for some $x' \in F$. Then $c^i x' \leq c^i x$ for all $i = 1, \dots, k-1$, and also $c^i x' \leq c^i x$ for all $i = k, \dots, N$, so, since $\lambda_i > 0$ for all $i = k, \dots, N$, it follows that $\sum_{i=k}^N \lambda_i c^i x' \leq \sum_{i=k}^N \lambda_i c^i x$. Thus $\bar{C}x' \leq \bar{C}x$ and, since $x \in F_E^2$, it must be that $\bar{C}x' = \bar{C}x$. Now since $Cx' \neq Cx$, it must be that $c^i x' < c^i x$ for some $i \in \{k, \dots, N\}$. But then $\sum_{i=k}^N \lambda_i c^i x' < \sum_{i=k}^N \lambda_i c^i x$, which contradicts $\bar{C}x' = \bar{C}x$. So it must be that $x \in F_E^1$.

To show that $F_E^1 \subseteq F_E^2$, we proceed as follows. Let $x \in F_E^1$. Now suppose that $x \notin F_E^2$, so there exists $x' \in F$ with $\bar{C}x' \leq \bar{C}x$ and $\bar{C}x' \neq \bar{C}x$. Hence $c^i x' \leq c^i x$ for all $i = 1, \dots, k-1$ and it must be that $c^i x' > c^i x$, for some $i \in \{k, \dots, N\}$, as, otherwise, $x \in F_E^2$ is contradicted. Also, $\bar{C}x' \leq \bar{C}x$ implies $\sum_{i=k}^N \lambda_i c^i x' \leq \sum_{i=k}^N \lambda_i c^i x$, so there must exist $h \in \{k, \dots, N\}$ with $c^h x' < c^h x$. Thus $S = \text{supp}(x)$ and $S' = \text{supp}(x')$ are mutually nondominated (MND) for c^k, \dots, c^N , so c^k, \dots, c^N are not sortable-by-subsets, and hence, by Proposition 1, cannot be universally co-ideal, yielding a contradiction.

Thus $F_E^1 = F_E^2$ and we may define $F_E := F_E^1 = F_E^2$.

Now consider distinct $x, x' \in F_E$ in the same element of the partition induced by O_N^1 , i.e., with $Cx = Cx'$ and $x \neq x'$. Then obviously $\bar{C}x = \bar{C}x'$, so x and x' are also in the same element of the partition induced by O_N^2 .

It remains to consider distinct $x, x' \in F_E$ in the same element of the partition induced by O_N^2 , i.e., with $\bar{C}x = \bar{C}x'$ and $x \neq x'$. Suppose they are in different elements of the partition induced by O_N^1 , i.e., suppose $Cx \neq Cx'$. Then, since $c^i x = c^i x'$ for all $i = 1, \dots, k-1$, (as $\bar{C}x = \bar{C}x'$), and since both x and x' are efficient for

P^1 , it must be that $c^i x' > c^i x$ and $c^h x' < c^h x$ for some $i, h \in \{k, \dots, N\}$. Again, this shows that $S = \text{supp}(x)$ and $S' = \text{supp}(x')$ are mutually nondominated (MND) for c^k, \dots, c^N , so c^k, \dots, c^N are not sortable-by-subsets, and hence, by Proposition 1, cannot be universally co-ideal, yielding a contradiction. We conclude that x and x' must also be in the same element of the partition of F_E induced by O_N^1 .

Thus the partitions of F_E induced by O_N^1 and O_N^2 are identical, as required. \square

The following example shows that solving the equivalent MBP obtained by aggregating universally co-ideal objectives may require less computational effort than solving the original MBP.

Example 2. Let P^1 be

$$\begin{aligned} \min \{ & 4x_1 + 2x_2 + x_3, \ x_1 + 2x_2 + 4x_3, \ 2x_1 + 3x_2 + 6x_3 \} \\ \text{s.t. } & x_1 + x_2 + x_3 \geq 2, \\ & x_1, x_2, x_3 \in \{0, 1\}. \end{aligned}$$

It is easy to verify that $F_E^1 = \{(1, 1, 0), (1, 0, 1), (0, 1, 1)\}$ and that the last two objectives are universally co-ideal. Let P^2 be the MPB obtained from P^1 by combining the last two objectives (with unit weights) into a single objective:

$$\begin{aligned} \min \{ & 4x_1 + 2x_2 + x_3, \ 3x_1 + 5x_2 + 10x_3 \} \\ \text{s.t. } & x_1 + x_2 + x_3 \geq 2, \\ & x_1, x_2, x_3 \in \{0, 1\}. \end{aligned}$$

Suppose we use the method proposed by Kirlik and Sayin (2014) to compute the nondominated frontier. First consider P^1 . In each iteration, the method first solves

$$x^i \in \arg \min \{ 2x_1 + 3x_2 + 6x_3 : x \in F, \ 4x_1 + 2x_2 + x_3 \leq u_1, \ x_1 + 2x_2 + 4x_3 \leq u_2 \},$$

for some upper bounds (right-hand sides) u_1 and u_2 , and then, if a solution exists, solves

$$x^n \in \arg \min \{ 5x_1 + 4x_2 + 5x_3 : x \in F, \ 2x_1 + 3x_2 + 6x_3 = 2x_1^i + 3x_2^i + 6x_3^i \}.$$

Next, consider P^2 . In each iteration, the method first solves

$$x^i \in \arg \min \{ 3x_1 + 5x_2 + 10x_3 : x \in F, \ 4x_1 + 2x_2 + x_3 \leq u \},$$

for some upper bound u , and then, if a solution exists, solves

$$x^n \in \arg \min \{ 4x_1 + 2x_2 + x_3 : x \in F, \ 3x_1 + 5x_2 + 10x_3 = 3x_1^i + 5x_2^i + 10x_3^i \}.$$

More specifically, to solve P^1 , the method explores the following bounds (u_1, u_2) : $(+\infty, +\infty)$, $(2, +\infty)$, $(+\infty, 5)$, $(4, 5)$, $(+\infty, 4)$, $(5, 4)$, and $(+\infty, 2)$. Consequently, 10 optimization problems have to be solved, 4 of which are infeasible. To solve P^2 , the method explores the following bounds u : $+\infty$, 14, 12, and 7. Consequently, 7 optimization problems have to be solved, only one of which is infeasible.

Since equivalence of MBPs, as in Definition 7, is transitive, aggregation of objectives that are universally co-ideal can be repeated for multiple subsets of universally co-ideal objectives. Specifically, we have the following corollary to Proposition 2.

Corollary 5 (to Proposition 2). Let $P^1 := \min_{x \in F} \{c^1 x, c^2 x, \dots, c^N x\}$ be a MBP and also let $Q_1, \dots, Q_D \subseteq \{1, \dots, N\}$ be disjoint sets such that the objectives c^i for $i \in Q_d$ are universally co-ideal, for each $d = 1, \dots, D$. Without loss of generality, say $\bigcup_{d=1}^D Q_d = \{k, \dots, N\}$ for some $k \in \{1, \dots, N\}$. Then

$$P^2 := \min_{x \in F} \{c^1 x, c^2 x, \dots, c^{k-1} x, \sum_{i \in Q_1} \lambda_i^1 c^i x, \sum_{i \in Q_2} \lambda_i^2 c^i x, \dots, \sum_{i \in Q_D} \lambda_i^D c^i x\},$$

with $\lambda_i^d > 0$ for all $i \in Q_d$ and $d = 1, \dots, D$, is an equivalent MBP.

This suggests the possibility of partitioning the objectives of a MBP into subsets, each of which is a universally co-ideal set of objectives, and seeking a partition of minimum cardinality. Aggregating the objectives in each universally co-ideal element of such a partition would yield an equivalent MBP, having minimum criterion space dimension. Recall that a set of objectives is universally co-ideal if and only if every pair in the set is universally co-ideal. So such a partition can be found by finding a minimum cardinality partition of the nodes of a graph into cliques, where the graph has a node for each objective and an edge for each universally co-ideal pair of objectives. By results in Section 4, the graph can be constructed in pseudo-polynomial time. Although this partitioning problem is equivalent to **PARTITION INTO CLIQUES**, and so is NP-hard (Garey and Johnson, 1979), for small values of N it may be practically tractable to solve to optimality; otherwise heuristic algorithms may suffice.

5.2. Deriving cuts

In this section, we introduce the concept of *objective-aligning* sets of variables and show how this concept can be used to derive a class of cuts for MBPs. These cuts are not inequalities that are valid for all feasible, or even all efficient, solutions of a MBP, but, rather, they are cuts that must be satisfied by any efficient solution that yields a nondominated point distinct from one already discovered. In this way, they may aid in the search for discovery of new nondominated points, during the solution of a MBP.

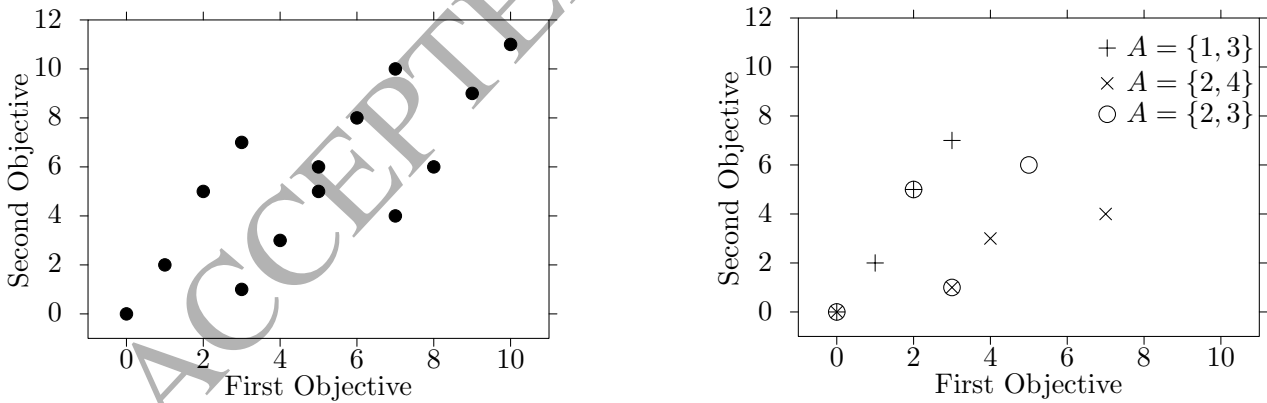
Definition 8. Given a set of objectives $c^1, c^2, \dots, c^N \in \mathbb{R}^n$, a set of variables $A \subseteq \{1, \dots, n\}$ is *objective-aligning* if the set of objectives $(c_j^1)_{j \in A}, (c_j^2)_{j \in A}, \dots, (c_j^N)_{j \in A}$ is universally co-ideal.

Before we show how objective-aligning sets of variables may be exploited, we first illustrate the concept with an example.

Example 3. Consider the two objective function vectors

$$c^1 = (1, 3, 2, 4) \quad \text{and} \quad c^2 = (2, 1, 5, 3).$$

It is obvious from Figure 2a, which is a plot of the points in Z , that c^1, c^2 are not universally co-ideal. For a given A , we can visualize $((c_j^1)_{j \in A}, (c_j^2)_{j \in A})$ by plotting $(\sum_{j \in A} c_j^1 x_j, \sum_{j \in A} c_j^2 x_j)$ for all $x \in \{0, 1\}^4$. Figure 2b shows the result for three alternative variable subsets, $A = \{1, 3\}$, $A = \{2, 4\}$ and $A = \{2, 3\}$. It is clear that the former two sets align the objective functions, since the points are simultaneously sorted in both objectives; $\{1, 3\}$ and $\{2, 4\}$ are both objective-aligning sets of variables for c^1, c^2 . The set $\{2, 3\}$ does not align the objectives, since, for example, the points $(2, 5)$ and $(3, 1)$ cannot be simultaneously sorted; $\{2, 3\}$ is not an objective-aligning set for c^1, c^2 .



(a) The points in Z for $c^1 = (1, 3, 2, 4), c^2 = (2, 1, 5, 3)$.

(b) Points $(\sum_{j \in A} c_j^1 x_j, \sum_{j \in A} c_j^2 x_j)$ for all $x \in \{0, 1\}^4$, for three different variable sets A .

Figure 2: Illustration of objective-aligning sets of variables, from Example 3.

Note that a single variable constitutes an objective-aligning set if and only if all its objective coefficients have the same sign (or are zero).

Proposition 3. *Let \bar{x} be an efficient solution for a MBP with objectives $c^1, c^2, \dots, c^N \subseteq \mathbb{R}^n$. If $A \subseteq \{1, \dots, n\}$ is an objective-aligning set of variables, then*

$$\sum_{j \notin A, \bar{x}_j=0} x_j + \sum_{j \notin A, \bar{x}_j=1} (1 - x_j) \geq 1 \quad (4)$$

is valid for all efficient solutions, x , that do not yield the same nondominated point as \bar{x} , i.e., it is satisfied by all efficient solutions, x , with $Cx \neq C\bar{x}$, where C is the objective function matrix.

PROOF. Suppose not, i.e., suppose there is an efficient solution, x , with $Cx \neq C\bar{x}$ and

$$\sum_{j \notin A, \bar{x}_j=0} x_j + \sum_{j \notin A, \bar{x}_j=1} (1 - x_j) = 0.$$

This implies that $x_j = \bar{x}_j$ for all $j \notin A$. Define $S := \text{supp}(x)$, $S' := \text{supp}(\bar{x})$, $S^1 := S \setminus S'$ and $S^2 := S' \setminus S$. Since $Cx \neq C\bar{x}$ and both x and \bar{x} are efficient solutions, S and S' must be MND. So, by Lemma 1, S^1 and S^2 are also MND. This, together with the fact that $S^1, S^2 \subseteq A$, (because $x_j = \bar{x}_j$ for all $j \notin A$), shows that $(c_j^1)_{j \in A}, (c_j^2)_{j \in A}, \dots, (c_j^N)_{j \in A}$ is not sortable-by-subsets. Then, by Proposition 1, the set of objectives $(c_j^1)_{j \in A}, (c_j^2)_{j \in A}, \dots, (c_j^N)_{j \in A}$ is not universally co-ideal, and hence is not objective-aligning. \square

Note that cuts of the form $\sum_{j: \bar{x}_j=0} x_j + \sum_{j: \bar{x}_j=1} (1 - x_j) \geq 1$ are known as *no-good constraints* (see Stidsen et al. (2014)). A cuts of type (4) can be viewed as a variant of a no-good constraint that is stronger than the traditional no-good constraint if $A \neq \emptyset$.

The example below shows the effect of adding cuts of type (4) to the LP relaxation of a MBP. In this example, the projection of the LP relaxation becomes smaller after adding each of the cuts. More importantly, the nondominated frontier of the LP relaxation improves, in each case. This demonstrates the potential utility of these inequalities in decision-space search methods, which (usually) fathom a node based on the nondominated frontier of the LP-relaxation at the node.

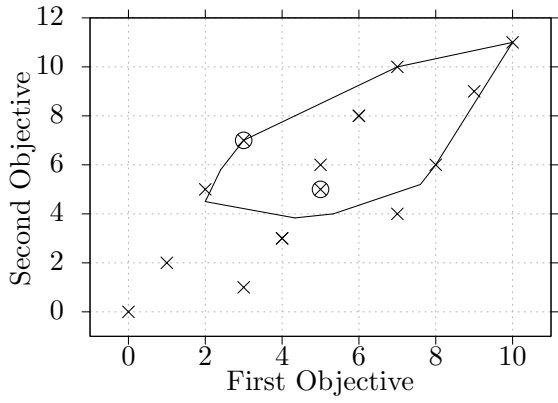
Example 4. Suppose that we want to compute all distinct nondominated points of

$$\begin{aligned} \min \quad & \{x_1 + 3x_2 + 2x_3 + 4x_4, 2x_1 + x_2 + 5x_3 + 3x_4\} \\ \text{s.t.} \quad & 5x_1 + 2x_2 + 6x_3 + 3x_4 \geq 8, \\ & x_1, x_2, x_3, x_4 \in \{0, 1\}. \end{aligned}$$

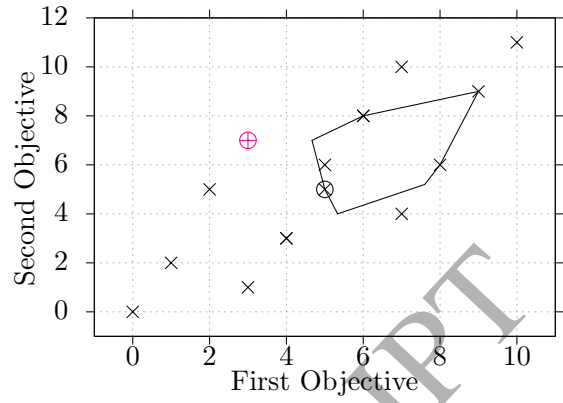
Note that this MBP has the same objective functions as are given in Example 3. It has two nondominated points, $(3, 7)$ and $(5, 5)$, corresponding to solutions $(1, 0, 1, 0)$ and $(1, 0, 0, 1)$, respectively. Projections of all $x \in \{0, 1\}^4$ and the LP-relaxation of this problem into the criterion space are shown in Figure 3a. To distinguish the nondominated points, we have drawn a circle around each in the figure. Note that in all figures in the remainder of the paper, we show the image of all $x \in \{0, 1\}^4$ in the criterion space. It is, of course, possible that $x \in \{0, 1\}^4$, but $x \notin F$, i.e., not all points in the criterion space are feasible. We show all of them, because they help clarify the concepts that we are introducing.

It is easy to verify that $\{2, 4\}$, $\{1, 3\}$, and $\{1, 4\}$ are objective-aligning sets (indeed, the first two of these were observed so from Figure 2b). Suppose that we have found one of the nondominated points, say $(3, 7)$. When searching for another nondominated point, we can use Proposition 3 to generate and add a cut of the form (4) for any objective-aligning set. The cuts corresponding to $\{2, 4\}$, $\{1, 3\}$ and $\{1, 4\}$ are

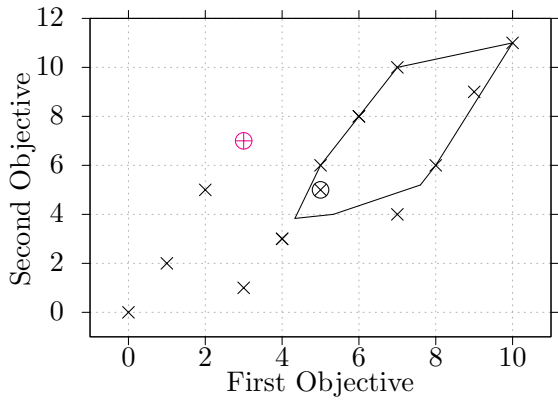
$$(1 - x_1) + (1 - x_3) \geq 1, \quad x_2 + x_4 \geq 1, \quad \text{and} \quad x_2 + (1 - x_3) \geq 1,$$



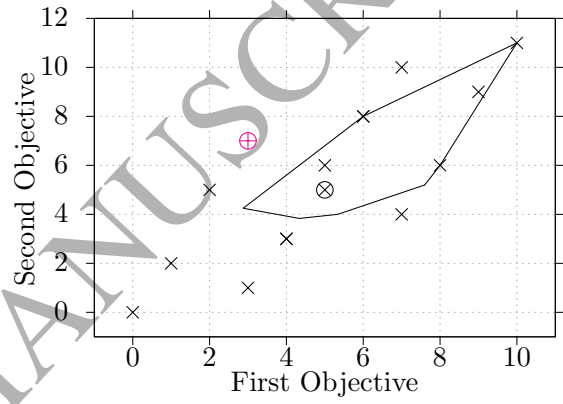
(a) No additional cut.



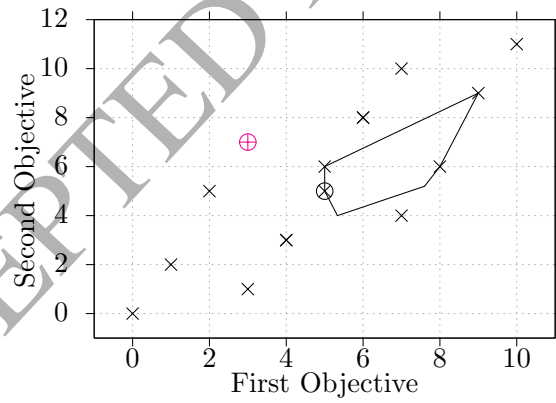
(b) After adding $(1 - x_1) + (1 - x_3) \geq 1$.



(c) After adding $x_2 + x_4 \geq 1$.



(d) After adding $x_2 + (1 - x_3) \geq 1$.



(e) After adding $(1 - x_1) + (1 - x_3) \geq 1$, $x_2 + x_4 \geq 1$, and $x_2 + (1 - x_3) \geq 1$.

Figure 3: Projections of $\{0, 1\}^4$ and the LP-relaxation of Example 4 into the criterion space.

respectively. The projection of the LP-relaxation after adding each of these three inequalities can be found in Figures 3b, 3c, and 3d, respectively. Figure 3e shows the projection of the LP-relaxation of the problem after adding all three. For convenience, we show the point $(3, 7)$, which was used to construct the cuts, by a red plus symbol.

Above, we derived a cut from an objective-aligning set of variables and a known, efficient solution. In this case, the corresponding nondominated point is an ideal point for the objective functions restricted to variables in the objective-aligning set. This property can be generalized: *from a given efficient solution, a subset of variables that is not necessarily objective-aligning, per se, but that nevertheless induces the corresponding nondominated point to be ideal, can be sought. Again, a cut is derived, but in this case under weaker conditions than those required for those discussed above. Thus a cut of the type we derive in this section may be available even when none of the type above can be found.*

Definition 9. The point $p \in \mathbb{R}^N$ is a *nexus point* for $c^1, c^2, \dots, c^N \in \mathbb{R}^n$, a set of objective functions, if, for any $x \in \{0, 1\}^n$, either $p_i \leq c^i x$ for all $i = 1, \dots, N$ or $p_i \geq c^i x$ for all $i = 1, \dots, N$. Equivalently, p is a *nexus point* if $Z \subseteq (\{p\} + \mathbb{R}_+^N) \cup (\{p\} - \mathbb{R}_+^N)$.

Figure 4 illustrates a nexus point: it shows Z for two objectives that are not universally co-ideal, but that have a nexus point. Recall Figure 1a: here we see Z in the case of two objectives that are universally co-ideal, in which case *every* point in Z is a nexus point.

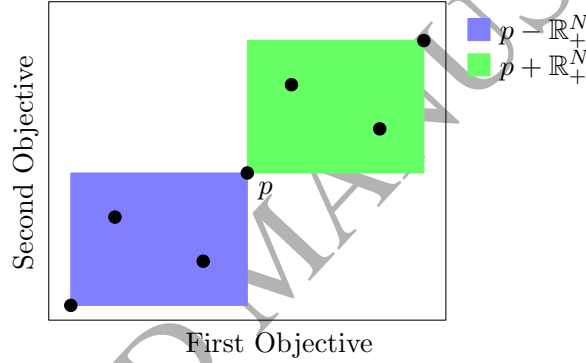


Figure 4: The points in Z for the objectives $c^1 = (1, 3, 4)$ and $c^2 = (2, 1, 3)$. These objectives are not universally co-ideal, but have $p = (4, 3)$ a nexus point.

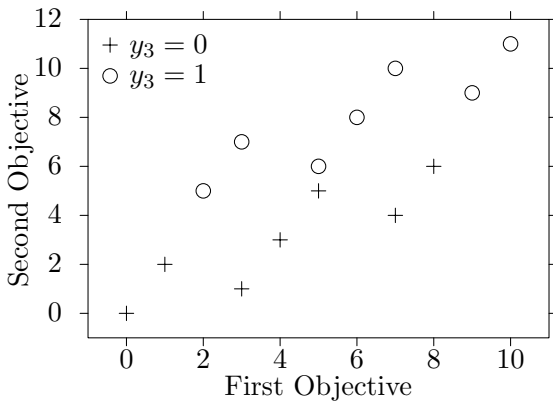
It is not difficult to see (for example, by considering Figure 4), that if p is a nexus point for c^1, c^2, \dots, c^N , then for any MBP with these objectives, we have that if p is a nondominated point, i.e., $p \in O_N$, then p is an ideal point. The following defines a special case of nexus point, for which the only condition needed to ensure that p is ideal is that the point p should be the image of a feasible solution (in the criterion space).

Definition 10. The point $p \in \mathbb{R}^N$ is a *lower bound point* for $c^1, c^2, \dots, c^N \in \mathbb{R}^n$, a set of objective functions, if, for any $x \in \{0, 1\}^n$, $p_i \leq c^i x$ for all $i = 1, \dots, N$. Equivalently, p is a *lower bound point* if $Z \subseteq \{p\} + \mathbb{R}_+^N$.

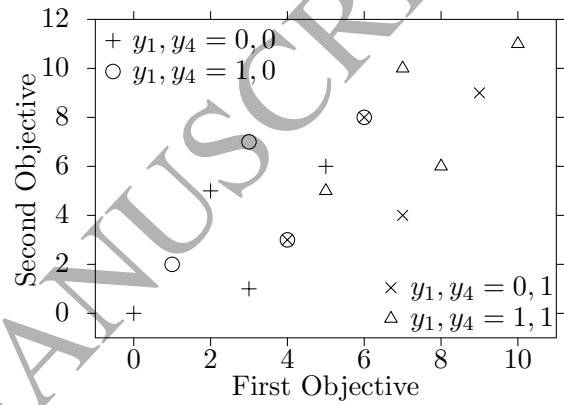
We generalize these definitions to consider optimization over only a subset of the variables, with variables outside this subset fixed so as to coincide with a given vector. Given $A \subseteq \{1, \dots, n\}$, $y \in \{0, 1\}^n$, we define $Z_{A,y} = \{Cx : x \in \{0, 1\}^n, x_j = y_j, \forall j \notin A\}$, where C is a given objective function matrix, to indicate the image, in criterion space, of every binary vector of dimension n that coincides with y for variables outside A . We call the pair (A, y) , with A nonempty, a *subspace-fixing*. Note that the values of y_j for $j \in A$ are irrelevant, and can safely be ignored; we take y to be in $\{0, 1\}^n$ purely for notational convenience.

Definition 11. Given a point $p \in \mathbb{R}^N$ and a set of objectives $c^1, c^2, \dots, c^N \in \mathbb{R}^n$, a subspace-fixing, (A, y) , is objective-aligning for p if p is a nexus point with respect to solutions fixed to y for variables not indexed by A . Equivalently, the subspace-fixing (A, y) is objective-aligning for p if $Z_{A,y} \subseteq (\{p\} + \mathbb{R}_+^N) \cup (\{p\} - \mathbb{R}_+^N)$. In the special case that $Z_{A,y} \subseteq \{p\} + \mathbb{R}_+^N$, we say that p is a lower bound point for the subspace-fixing (A, y) .

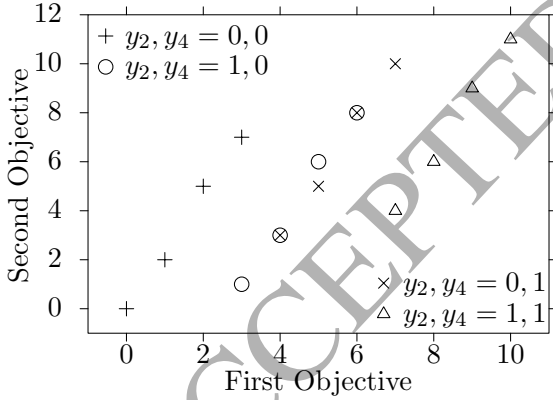
We illustrate the concept using the same objectives as in Examples 3 and 4. Recall that for these objectives, $c^1 = (1, 3, 2, 4)$ and $c^2 = (2, 1, 5, 3)$, the variables sets $\{1, 3\}$, $\{2, 4\}$ and $\{1, 4\}$ are objective-aligning. In fact, these are the *only* (non-trivial) objective-aligning sets for these objectives. So, for example, $\{2, 3\}$ and $\{1, 2, 4\}$ are not objective-aligning. In Figure 5a, we show $Z_{A,y}$ for $A = \{1, 2, 4\}$ and $y = (-, -, 0, -)$, (we do not specify y_j for $j \in A$, since these values are irrelevant), indicated with the plus symbol in the figure, and observe that $p = (4, 3)$ is a nexus point: the subspace-fixing $(\{1, 2, 4\}, (-, -, 0, -))$ is objective-aligning for $(4, 3)$. This Figure 5a also shows $Z_{A,y}$ for $A = \{1, 2, 4\}$ and $y = (-, -, 1, -)$, indicated with the unfilled circle in the figure, and observe that $p = (6, 8)$ is a nexus point: the subspace-fixing $(\{1, 2, 4\}, (-, -, 1, -))$ is objective-aligning for $(6, 8)$. In Figure 5b we show $Z_{A,y}$ for $A = \{2, 3\}$ and $y = (1, -, -, 1)$, indicated with the triangle symbol, and observe that $p = (5, 5)$ is a lower bound point for the subspace-fixing $(\{2, 3\}, (1, -, -, 1))$.



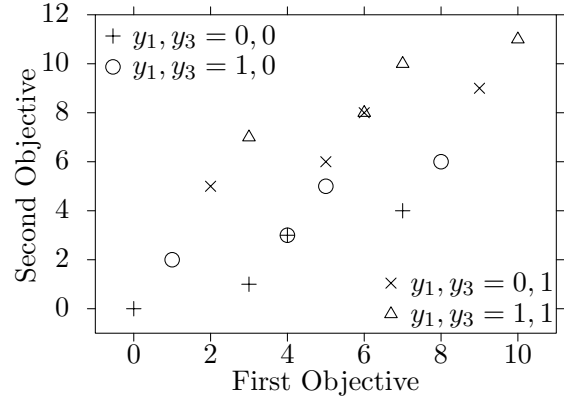
(a) All (both) subspace-fixings (A, y) with $A = \{1, 2, 4\}$ ($y_3 = 0$ and $y_3 = 1$).



(b) All (four) subspace-fixings (A, y) with $A = \{2, 3\}$ $((y_1, y_4) = (0, 0), (1, 0), (0, 1)$ and $(1, 1))$.



(c) All (four) subspace-fixings (A, y) with $A = \{1, 3\}$ $((y_2, y_4) = (0, 0), (1, 0), (0, 1)$ and $(1, 1))$.



(d) All (four) subspace-fixings (A, y) with $A = \{2, 4\}$ $((y_1, y_3) = (0, 0), (1, 0), (0, 1)$ and $(1, 1))$.

Figure 5: Illustration of subspace-fixings, showing $Z_{A,y}$ for $c^1 = (1, 3, 2, 4)$ and $c^2 = (2, 1, 5, 3)$, for different values of A and y .

To explain the relationship between sets of variables that are objective-aligning, *per se*, (i.e., satisfy Definition 8), and subspace-fixings that are objective-aligning for some point, we show, in Figures 5c and 5d, $Z_{A,y}$ for *every* possible y , for the objective-aligning sets $A = \{1, 3\}$ and $A = \{2, 4\}$, respectively. We see that for A an objective-aligning

set of variables, the set $Z_{A,y}$, for any y , is totally ordered, and hence for every $p \in Z_{A,y}$, the subspace-fixing (A, y) is objective-aligning for p . In fact, this property follows from the observation below, which can be seen to hold in all subfigures of Figure 5.

Observation 1. For any y , $Z_{A,y}$ is a translation of $Z_{A,0}$, where 0 is the zero vector. Specifically, $Z_{A,y} = \{p + CI_{\bar{A}}y : p \in Z_{A,0}\}$, where $I_{\bar{A}}$ denotes the matrix with diagonal entries given by the indicator vector of \bar{A} and all other entries zero, and $\bar{A} = \{1, \dots, n\} \setminus A$ denotes the complement of A . Note also that $Z_{A,0}$ is the image of the objectives restricted to A , $((c_j^1)_{j \in A}, (c_j^2)_{j \in A}, \dots, (c_j^N)_{j \in A})$, i.e.,

$$Z_{A,0} = \{((c_j^1)_{j \in A} \hat{x}, (c_j^2)_{j \in A} \hat{x}, \dots, (c_j^N)_{j \in A} \hat{x}) : \hat{x} \in \{0, 1\}^{|A|}\}.$$

The latter part of this observation can be illustrated by comparing Figures 4 and 5a: the former consists precisely of the points in the latter marked with plus symbol, i.e., those for the case $y_3 = 0$. (Note that Figure 4 is scaled so that the set of points shown fills the region.)

If A is a set of variables that is objective-aligning, then the subspace-fixing $(A, 0)$ is objective-aligning for every $p \in Z_{A,0}$, since this set is totally ordered. Thus for every $y \in \{0, 1\}^n$ the subspace-fixing (A, y) is objective-aligning for every $p \in Z_{A,y}$, since this set is simply a translation of $Z_{A,0}$. However, (A, y) objective-aligning for some p does *not* imply that A is objective-aligning, *per se* (as we saw in Figures 5a and 5b). Thus it may be that, for a given p , subspace-fixings that are objective-aligning for p can be found, and hence cuts are available, having the form given in the following proposition, that could not be deduced from objective-aligning sets.

Proposition 4. If \bar{x} is an efficient solution of the MBP (C, F) and the subspace-fixing (A, y) is objective-aligning for $p = C\bar{x}$, then

$$\sum_{j \notin A, y_j=1} (1 - x_j) + \sum_{j \notin A, y_j=0} x_j \geq 1 \quad (5)$$

is valid for all efficient solutions, x , that do not yield the same nondominated point as \bar{x} , i.e., it is satisfied by all $x \in F_E$ with $Cx \neq C\bar{x}$. If $p = C\bar{x}$ is a lower bound point for the subspace-fixing (A, y) , then, to ensure validity of (5), it suffices to require that \bar{x} is a feasible solution of the MBP, not necessarily efficient.

PROOF. Let $\bar{x} \in F_E$ and (A, y) be a subspace-fixing that is objective-aligning for $p = C\bar{x}$. \bar{x} an efficient solution implies that $O_N \cap (\{p\} + \mathbb{R}_+^N) \cup (\{p\} - \mathbb{R}_+^N) = \{p\}$ and (A, y) objective-aligning for p is equivalent to $Z_{A,y} \subseteq (\{p\} + \mathbb{R}_+^N) \cup (\{p\} - \mathbb{R}_+^N)$. So $O_N \cap Z_{A,y} \subseteq \{p\}$.

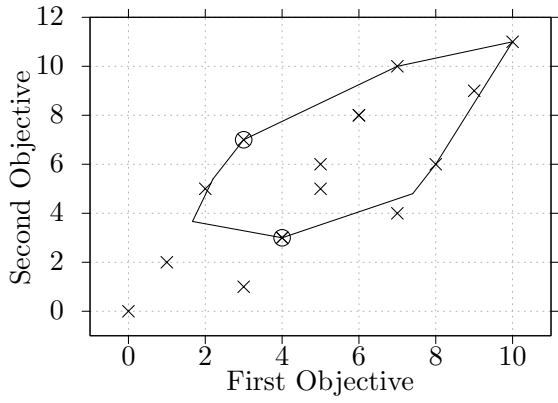
Now suppose $x \in F_E$ with $Cx \neq C\bar{x}$ and $\sum_{j \notin A, y_j=1} (1 - x_j) + \sum_{j \notin A, y_j=0} x_j < 1$, i.e., since x is binary, $x_j = 1$ for all $j \notin A$ with $y_j = 1$ and $x_j = 0$ for all $j \notin A$ with $y_j = 0$. Then $Cx \in O_N$, $Cx \neq p$, $x_j = y_j$ for all $j \notin A$, and hence $Cx \in Z_{A,y}$. But then $Cx \in O_N \cap Z_{A,y} \subseteq \{p\}$ and $Cx \neq p$, which is a contradiction.

We conclude that if x is an efficient solution yielding a nondominated point distinct from p , then x must satisfy (5).

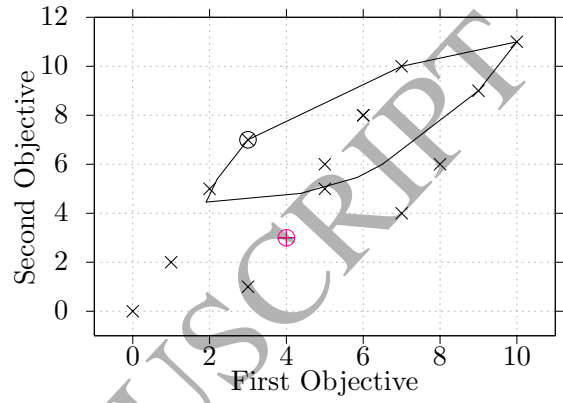
To complete the proof, let $\bar{x} \in F$, not necessarily efficient, and let (A, y) be a subspace-fixing for which $p = C\bar{x}$ is a lower bound point, so $Z_{A,y} \subseteq \{p\} + \mathbb{R}_+^N$. Again, suppose $x \in F_E$ with $Cx \neq C\bar{x}$ and $\sum_{j \notin A, y_j=1} (1 - x_j) + \sum_{j \notin A, y_j=0} x_j < 1$, i.e., $x_j = y_j$ for all $j \notin A$. Then $Cx \in O_N$, $Cx \neq p$, and $Cx \in Z_{A,y}$. But then $Cx \in O_N \cap (\{p\} + \mathbb{R}_+^N)$ and $Cx \neq p$, which is a contradiction, since for $p \in O$, the only nondominated point in $\{p\} + \mathbb{R}_+^N$ must be p itself. \square

In the case that A is a set of variables that is objective-aligning and $p = C\bar{x}$ for $\bar{x} \in F_E$, cuts of *both* types (4) and (5) may be applied. Taking $y = \bar{x}$ ensures that $p \in Z_{A,y}$, so in this case the two inequalities coincide. Otherwise neither dominates the other.

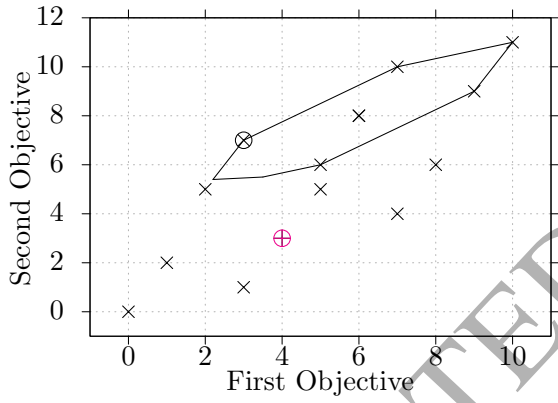
The following example illustrates the effect on the LP relaxation of a cut of type (5).



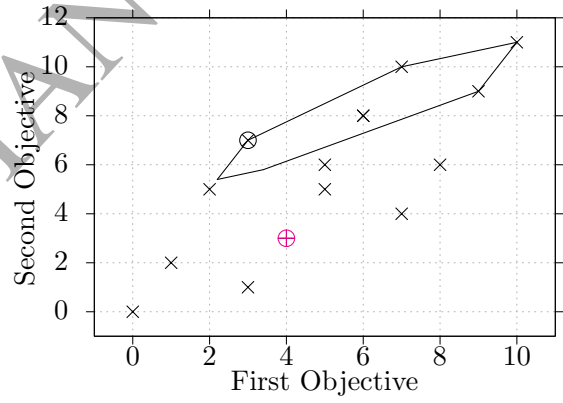
(a) No additional cut.



(b) After adding $x_4 \ge x_2$, $x_3 \ge x_2$, and $x_3 \ge x_1$.



(c) After adding $x_3 \ge 1$.



(d) After adding $x_4 \ge x_2$, $x_3 \ge x_2$, $x_3 \ge x_1$, and $x_3 \ge 1$.

Figure 6: The effect of cuts derived from objective-aligning sets and a subspace-fixing, in conjunction with the knowledge of the nondominated point, $(4, 3)$, on the LP relaxation, for Example 5.

Example 5. Consider the MBP

$$\begin{aligned} \min \quad & \{x_1 + 3x_2 + 2x_3 + 4x_4, 2x_1 + x_2 + 5x_3 + 3x_4\} \\ \text{s.t.} \quad & 5x_1 + 2x_2 + 6x_3 + 3x_4 \geq 7, \\ & x_1, x_2, x_3, x_4 \in \{0, 1\}. \end{aligned}$$

Note that this has the same objectives as Example 4, but has a slightly different (enlarged) feasible set. The points in Z , and the image of the LP relaxation in criterion space, are shown in Figure 6a.

This problem has two nondominated points, $(4, 3)$ and $(3, 7)$, corresponding to solutions $(1, 1, 0, 0)$ and $(1, 0, 1, 0)$, respectively. Suppose we have discovered the former, so $\bar{x} = (1, 1, 0, 0) \in F_E$, and use the three, maximal, objective-aligning sets of variables $\{1, 3\}$, $\{1, 4\}$, $\{2, 4\}$ to derive cuts of type (4). We obtain the cuts $x_4 \geq x_2$, $x_3 \geq x_2$, and $x_3 \geq x_1$, respectively. The effect of adding these cuts on the LP relaxation is shown in Figure 6b.

Furthermore, $p = C\bar{x} = (4, 3)$ is a nexus point for the subspace-fixing $(\{1, 2, 4\}, (-, -, 0, -))$: the corresponding cut of type (5) is $x_3 \geq 1$. Figure 6c shows the effect of this cut on the LP relaxation, while Figure 6d shows the effect of all four cuts, in combination.

5.2.1. Complexity.

Naturally, the questions of how difficult it is to (i) recognize that a set of variables, A , is objective-aligning, (ii) decide if a given point, p , is a nexus point for a subspace-fixing (A, μ) , and (iii) decide if a given point, p , is a lower bound point for a subspace-fixing (A, μ) , arise. The answer to (i) follows immediately from the results in Section 4: recognizing that the objectives restricted to A are universally co-ideal is NP-hard, but can be solved in pseudo-polynomial time. The answer to (ii) is the same. It suffices to check that p is a nexus point for every pair of objectives, say $c^i, c^{i'}$ with $i \neq i'$, for the subspace fixing (A, μ) , and this can be done using the 0-1 knapsack optimization problem, $\psi(c^i, c^{i'})$, given in Section 4, simply by replacing the term $c^i y$ by p_i and the term $c^{i'} y$ by $p_{i'}$, removing the y variables, and setting $x_j = \mu_j$ for all $j \notin A$. The remaining problem is still a 0-1 knapsack problem, in $|A|$ variables, and hence can be solved in pseudo-polynomial time. The problem can be proved to be NP-hard using ideas similar to those in Section 4; proof is given in Appendix A. The answer to (iii) is that the problem is trivial: p is a lower bound point for the subspace-fixing (A, μ) if and only if, for every objective, c^i , $i = 1, \dots, N$, $\min\{c^i x : x \in \{0, 1\}^n, x_j = \mu_j, \forall j \notin A\} = \sum_{j \notin A} c_j^i \mu_j + \sum_{j \in A} \min\{c_j^i, 0\} \geq p_i$, which can be checked in $\mathcal{O}(Nn)$ time.

5.2.2. Variable fixing.

Subspace-fixings that induce nexus points may be used in preprocessing, to fix variables. Given an MPB (C, F) and given $x \in F_E$, if $p = Cx$ is a nexus point for the subspace-fixing $(\{1, \dots, n\} \setminus \{j\}, y)$, for some $j \in \{1, \dots, n\}$, where $y_j = 0$, then (5) has the effect of fixing $x_j = 1$. Indeed, the cut derived from the subspace-fixing $(\{1, 2, 4\}, (-, -, 0, -))$ in Example 5, $x_3 \geq 1$, has exactly this effect. Alternatively, if $p = Cx$ is a nexus point for the subspace-fixing $(\{1, \dots, n\} \setminus \{j\}, y)$, for some $j \in \{1, \dots, n\}$, where $y_j = 1$, then (5) has the effect of fixing $x_j = 0$. To illustrate this, note that, in Example 5, the nondominated point $(4, 3)$ is a nexus point for the subspace-fixing $(\{1, 3, 4\}, (-, 1, -, -))$, giving cut $x_2 \leq 0$.

Although checking that p is a nexus point for such a subspace-fixing requires pseudo-polynomial time in the worst case, in practice it may still be tractable. Furthermore, the 0-1 knapsack problems to be solved do not necessarily need to be solved to optimality; they can be halted as soon as their dual (upper) bound does not exceed zero, or whenever their primal (lower) bound does exceed zero.

As discussed in the previous section, subspace-fixings that induce lower bounds can be recognized efficiently, so these may be a particularly attractive special case of nexus point to check. Furthermore, variable fixing can be deduced from $p = Cx$ for x a feasible solution, $(x \in F)$, not necessarily efficient. An illustration of this is given in Example 6.

Example 6. Consider the MBP

$$\begin{aligned} \min \quad & \{-3x_1 + 4x_2 + 5x_3 - 2x_4, -2x_1 + 4x_2 + 2x_3 - 3x_4\} \\ \text{s.t.} \quad & -3x_1 + 11x_2 + 2x_3 + 8x_4 \geq 8, \\ & x_1, x_2, x_3, x_4 \in \{0, 1\}. \end{aligned}$$

This problem has an ideal point of $(-2, -3)$ corresponding to solution $(0, 0, 0, 1)$. Projections of all $x \in \{0, 1\}^4$ and the LP-relaxation of this problem into the criterion space are shown in Figure 7a. Suppose that we know that $(-1, -1)$ is a point in criterion space (corresponding to feasible solution $(1, 1, 0, 1)$). Then $(-1, -1)$ is a lower bound point for the subspace-fixing $(\{1, 3, 4\}, (-, 1, -, -))$, and hence $x_2 = 0$ must be satisfied by any efficient solution yielding a nondominated point distinct from $(-1, -1)$. Figure 7b shows the projection of the LP-relaxation of the problem after fixing $x_2 = 0$.

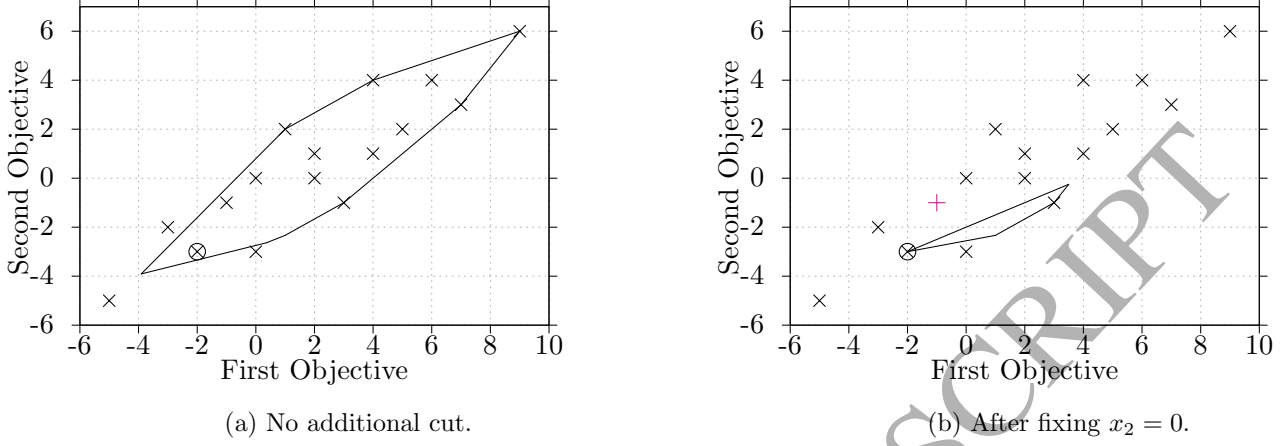


Figure 7: Projections of all $x \in \{0, 1\}^4$ and the LP relaxation, into the criterion space, for Example 6.

6. Relaxations

A useful extension of the concept of universally co-ideal is given in Definition 12.

Definition 12. A set of objectives, $c^1, c^2, \dots, c^N \in \mathbb{R}^n$, is universally co-ideal with respect to $R \subseteq \{0, 1\}^n$ if, for **any** nonempty feasible set contained in R , $F \subseteq R$, the MBP

$$\min_{x \in F} \{c^1 x, c^2 x, \dots, c^N x\}$$

has an ideal point, i.e.,

$$\bigcap_{i=1}^N \arg \min_{x \in F} c^i x \neq \emptyset.$$

We observe that the above results concerning universal co-idealness can be extended easily to universal co-idealness with respect to R . For example, when we consider feasible sets known to have bounded cardinality.

Definition 13. A set of objectives, $c^1, c^2, \dots, c^N \in \mathbb{R}^n$, is sortable-by- K - L -sets if for every pair of sets $S, S' \subseteq \{1, \dots, n\}$, with $K \leq |S|, |S'| \leq L$ either

- (a) $c^i(S) \leq c^i(S')$ for all $i = 1, \dots, N$, or
- (b) $c^i(S) \geq c^i(S')$ for all $i = 1, \dots, N$.

The following result is easy to prove.

Proposition 5. A set of objectives, $c^1, c^2, \dots, c^N \in \mathbb{R}^n$, is universally co-ideal with respect to $R = \{x \in \{0, 1\}^n : K \leq \sum_{j=1}^n x_j \leq L\}$ if and only if it is sortable-by- K - L -sets.

Note that in the case of feasible sets $F \subseteq \{x \in \{0, 1\}^n : K \leq \sum_{j=1}^n x_j \leq L\}$, the sortable-by- K - L -sets property can also be tested in pseudo-polynomial time: the 0-1 cardinality-constrained knapsack problem

$$\begin{aligned} \psi_{K,L}(f, g) = \max \quad & fx - fy \\ \text{s.t.} \quad & gx - gy \leq -1, \\ & K \leq \sum_{j=1}^n x_j \leq L \\ & K \leq \sum_{j=1}^n y_j \leq L \\ & x, y \in \{0, 1\}^n \end{aligned}$$

is also pseudo-polynomially solvable. This observation extends to any set of cardinality constraints on disjoint sets of variables.

All the concepts discussed so far, including objective aligning sets of variables, subspace-fixings, nexus points, and lower bound points, as well as the cuts derived from them, can readily be extended to make use of a relaxation of the feasible set, $R \subseteq \{0, 1\}^n$, simply by replacing Z by $\{Cx : x \in R\}$ and $Z_{A,y}$ by $\{Cx : x \in R, x_j = y_j, \forall j \notin A\}$. The following example illustrates the value of such relaxations.

Example 7. Consider the MBP

$$\begin{aligned} \min \quad & \{-2x_1 + 3x_2 + 5x_3 - 2x_4 - x_5, -x_1 + 5x_2 + 3x_3 - 3x_4 - x_5\} \\ \text{s.t.} \quad & -3x_1 + 11x_2 + 3x_3 + 8x_4 + 11x_5 \geq 8, \\ & x_4 + x_5 \leq 1, \\ & x_1, x_2, x_3, x_4, x_5 \in \{0, 1\}. \end{aligned}$$

The points in Z and the projection of the LP-relaxation into the criterion space, for this MBP, are shown in Figure 8a. The problem has two nondominated points, $(-3, -2)$ and $(-2, -3)$, corresponding to solutions $(1, 0, 0, 0, 1)$ and $(0, 0, 0, 1, 0)$, respectively.

Suppose that during the search, the point $(-1, -1)$, corresponding to feasible solution $(0, 0, 0, 0, 1)$, is found. Taking R to be

$$R := \{x \in \{0, 1\}^5 : x_4 + x_5 \leq 1\}$$

we find that $(-1, -1)$ is a lower bound point for the subspace-fixing $(\{1, 3, 4, 5\}, (-, 1, -, -, -))$, and so we obtain the cut $x_2 \leq 0$, and so may fix the variable $x_2 = 0$. Figure 8b shows the projection of the LP relaxation of the MBP after adding $x_2 = 0$.

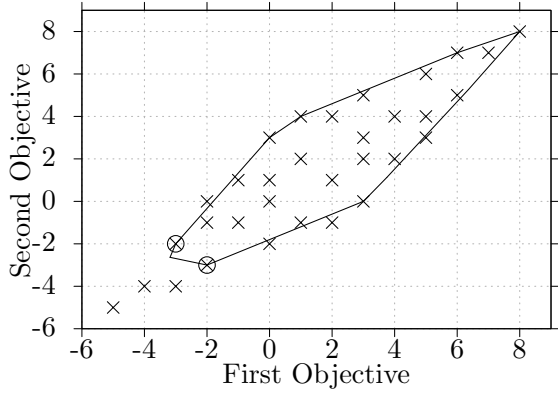
Also, $(-1, -1)$ is readily observed to be a lower bound for the subspace-fixing $(\{2, 3, 5\}, (0, -, -, 0, -))$. Thus, since it is the image of a feasible solution, we can derive the cut $x_1 + x_4 \geq 1$. Figure 8c shows the projection of the LP-relaxation of the problem after adding $x_2 = 0$ and $x_1 + x_4 \geq 1$.

Now suppose that the search continues after adding $x_2 = 0$ and $x_1 + x_4 \geq 1$, finding the nondominated point $(-3, -2)$, corresponding to efficient solution $(1, 0, 0, 0, 1)$. It is easy to verify that $\{1, 3, 5\}$ and $\{3, 4, 5\}$ are both objective-aligning sets with respect to the relaxation

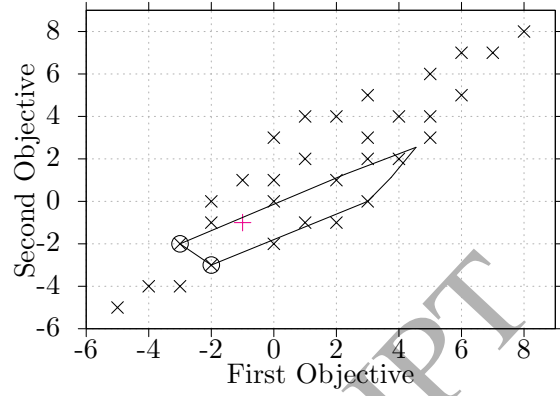
$$R = \{x \in \{0, 1\}^5 : x_4 + x_5 \leq 1\}.$$

Thus we may derive the cuts $x_2 + x_4 \geq 1$ and $(1 - x_1) + x_2 \geq 1$. Figure 8d shows the projection of the LP relaxation of the problem after adding $x_2 = 0$, $x_1 + x_4 \geq 1$, and $x_2 + x_4 \geq 1$. Also, Figure 8e shows the projection of the LP relaxation of the problem after adding $x_2 = 0$, $x_1 + x_4 \geq 1$, $x_2 + x_4 \geq 1$, and $(1 - x_1) + x_2 \geq 1$.

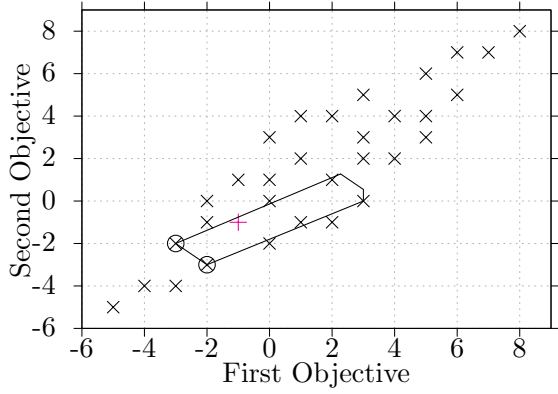
Finally, we observe that the nondominated point $(-3, -2)$ is a nexus point (in fact, it is a lower bound point) for the subspace-fixing $(\{1, 2, 4, 5\}, (-, -, 1, -, -))$, and hence we can derive the variable fixing $x_3 = 0$. Figure 8f shows the projection of the LP relaxation of the problem after adding all cuts (and variable fixings) derived so far. The LP relaxation is now a single point, the nondominated point $(-2, -3)$. We have shown the point $(-2, -3)$ by a blue star symbol in the figure.



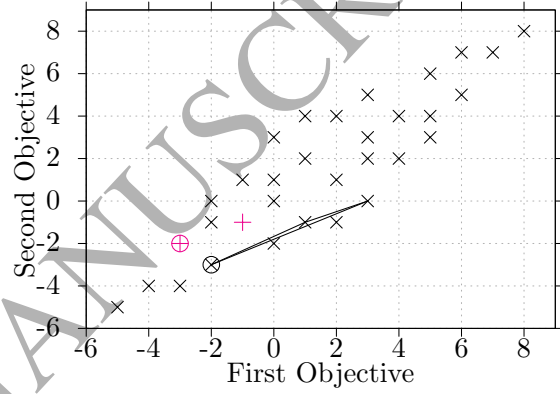
(a) No additional cut.



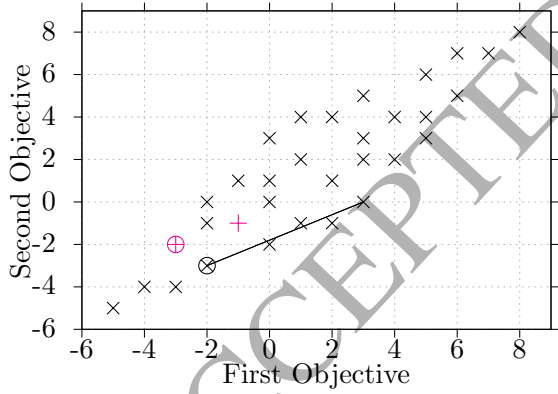
(b) After adding $x_2 = 0$.



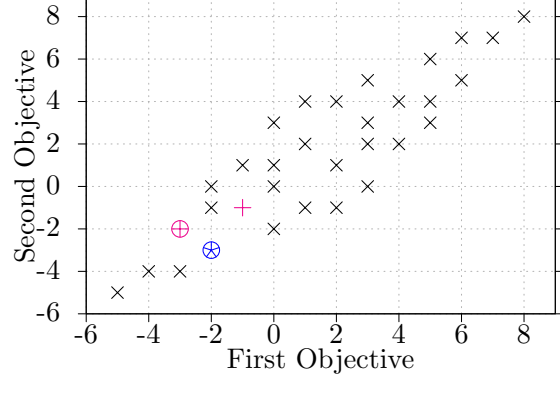
(c) After adding $x_2 = 0$ and $x_1 + x_4 \geq 1$.



(d) After adding $x_2 = 0$, $x_1 + x_4 \geq 1$, and $x_2 + x_4 \geq 1$.



(e) After adding $x_2 = 0$, $x_1 + x_4 \geq 1$, $x_2 + x_4 \geq 1$, and $(1 - x_1) + x_2 \geq 1$.



(f) After adding $x_2 = 0$, $x_1 + x_4 \geq 1$, $x_2 + x_4 \geq 1$, $(1 - x_1) + x_2 \geq 1$, and $x_3 = 0$. (The point $(-2, -3)$ is the projection of the LP-relaxation.)

Figure 8: Projections of all $x \in \{0, 1\}^5$ and the LP-relaxation of the final example into the criterion space.

7. A Computational study

To provide further evidence of the potential benefits of the ideas and concepts introduced above, we have conducted a small computational study. The sole purpose of our computational study is to provide (some) numerical evidence that (some of) the proposed techniques may improve the performance of multi-objective binary programming solvers.

We limit ourselves to demonstrating the value of the cuts of the form (4). First, we incorporate the use of these inequalities in a rudimentary *decision space search algorithm* for instances with two objectives. Second, we incorporate the use of these inequalities in a sophisticated *criterion space search algorithm* for instances with three objectives. Both algorithms have been implemented in C++ and use CPLEX 12.7 as the integer programming solver. All computational experiments have been carried out on a Dell PowerEdge R630 with two Intel Xeon E5-2650 2.2 GHz 12-Core Processors, 128GB RAM, and the RedHat Enterprise Linux 6.8 operating system, and using a single thread.

7.1. Enhancing a rudimentary decision space search algorithm

We start by introducing a rudimentary decision space search algorithm for solving 0-1 integer programs with two objectives (which also forms the basis for the algorithm of Stidsen et al. (2014)). The algorithm relies on three observations. First, a nondominated point x^* can be obtained by solving the following optimization problem

$$x^* \in \arg \min \{c^1 x + c^2 x : x \in F\},$$

where F represents the set of feasible solutions. Second, adding the constraint

$$\sum_{j:\bar{x}_j=0} x_j + \sum_{j:\bar{x}_j=1} (1 - x_j) \geq 1, \quad (6)$$

a so-called “no-good” constraint, eliminates \bar{x} from the set of feasible solutions. Finally, if z^1 and z^2 are two nondominated points with $z_1^1 < z_1^2$, then any other nondominated point \bar{z} with $\bar{z}_1 \in [z_1^1, z_1^2]$ and $\bar{z}_2 \in [z_2^1, z_2^2]$ will be such that $\bar{z} \leq u$ with $u = (z_1^2, z_2^1)$. We refer to such a point u as a local upper bound. Any set of nondominated points $\{z^1, \dots, z^K\}$ induces a set of local upper bounds: order z^1, \dots, z^K by increasing z_1^k values and take $\{(z_1^{k+1}, z_2^k) : k = 1, \dots, K-1\}$.

The algorithm maintains a list of nondominated points found, denoted by L , and its corresponding set of local upper bound points, denoted by U . The list is initialized with the two end-points of nondominated frontier, denoted by z^{UL} and z^{BR} , found in the usual way by lexicographic minimization.

In each iteration, the algorithm first solves the following single-objective optimization problem,

$$x^* \in \arg \min \{c^1 x + c^2 x : x \in F, \sum_{j:\bar{x}_j=0} x_j + \sum_{j:\bar{x}_j=1} (1 - x_j) \geq 1, \forall \bar{x} \in \bar{F}\},$$

which is equivalent to

$$x^* \in \arg \min \{c^1 x + c^2 x : x \in F \setminus \bar{F}\},$$

where \bar{F} is the set of feasible solutions found in the previous iterations. The algorithm then computes

$$v^* := \max \{u_1 + u_2 : u \in U\}.$$

It is not hard to see that if $x^* = \text{null}$ (the optimization problem was infeasible) or $c^1 x^* + c^2 x^* \geq v^*$, then all nondominated points have been found and the algorithm terminates. Otherwise, the algorithm updates the set \bar{F} of feasible solutions encountered. Also, if $z^* = (c^1 x^*, c^2 x^*)$ and there is no $z \in L$ with $z \leq z^*$, then z^* is a new nondominated point and is added to L . Algorithm 1 shows a precise description of this simple cutting plane algorithm for solving 0-1 integer programs with two objective functions.

Algorithm 1: A cutting-plane algorithm for solving a 2-objective 0-1 integer program.

```

List.create(L), List.create(U)
List.add(L, zUL), List.add(L, zBR)
List.add(U, (z1BR, z2UL))
 $\bar{F} \leftarrow \emptyset$ 
done  $\leftarrow$  false
while not done do
     $x^* \leftarrow \arg \min \{c^1x + c^2x : x \in F, \sum_{j:\bar{x}_j=0} x_j + \sum_{j:\bar{x}_j=1} (1 - x_j) \geq 1, \forall \bar{x} \in \bar{F}\}$ 
     $v^* \leftarrow \max \{u_1 + u_2 : u \in U\}$ 
    if  $x^* \neq \text{null}$   $\&\&$   $c^1x^* + c^2x^* < v^*$  then
         $\bar{F} \leftarrow \bar{F} \cup \{x^*\}$ 
         $z^* \leftarrow (c^1x^*, c^2x^*)$ 
        if there exists no  $z \in L$  with  $z \leq z^*$  then
            List.add(L, z*)
            List.update(U, z*)
        else
            done  $\leftarrow$  true
return L

```

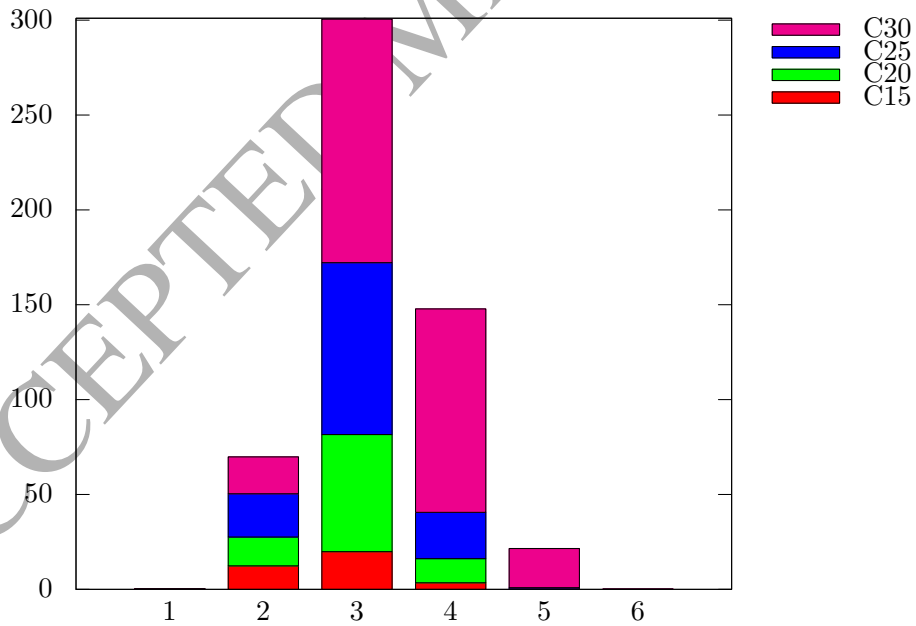


Figure 9: The (average) number of maximal objective aligning sets of a particular size for each set of instances.

Table 1: Performance of Algorithm 1 with and without cuts derived from randomly selected maximal objective-aligning subsets.

Set	#NDPs	No OA cuts		With OA cuts		%Improvement	
		Time (Sec.)	#IP	Time (Sec.)	#IP	Time	#IP
C15	11	0.33	33	0.31	33	6.1	0.0
	8	2.04	85	1.72	83	15.7	2.4
	3	3.10	104	2.60	102	16.1	1.9
	9	0.63	46	0.49	45	22.2	2.2
	6	0.19	20	0.13	18	31.6	10.0
	23	247.04	758	210.39	739	14.8	2.5
	7	0.65	47	0.56	43	13.8	8.5
	19	62.07	420	52.68	409	15.1	2.6
	11	5.74	142	4.56	137	20.6	3.5
C20	8	2.72	97	2.26	91	16.9	6.2
	6	1.96	81	1.70	79	13.3	2.5
	4	0.74	53	0.89	51	-20.3	3.8
	9	15.25	191	12.88	189	15.5	1.0
	3	0.16	18	0.14	18	12.5	0.0
	6	4.94	130	4.22	127	14.6	2.3
	9	0.60	42	0.54	40	10.0	4.8
	11	1,178.71	1,401	1,000.49	1,379	15.1	1.6
	11	6.26	139	4.72	134	24.6	3.6
C25	10	9.56	164	8.35	162	12.7	1.2
	10	585.29	987	575.99	976	1.6	1.1
	16	2,609.51	1,454	2,622.44	1,438	-0.5	1.1
	12	336.53	738	324.97	733	3.4	0.7
	20	4,004.12	1,651	4,251.95	1,632	-6.2	1.2
C30	12	60.70	361	57.40	354	5.4	1.9
	18	80.07	392	74.72	381	6.7	2.8
	19	8,004.21	1,886	7,716.56	1,879	3.6	0.4
	12	1,294.96	990	1,211.61	982	6.4	0.8
	10	2.43	82	2.73	81	-12.3	1.2
	13	322.43	669	322.77	664	-0.1	0.7
	14	295.55	606	276.46	598	6.5	1.3
	14	1.52	65	1.55	64	-2.0	1.5
Avg	11.10	617.42	446.84	604.80	440.68	9.14	2.43

We have generated four sets of instances of bi-objective binary knapsack problem, i.e., of the form

$$\begin{aligned} \max \quad & \left\{ \sum_{j=1}^n c_j^1 x_j, \sum_{j=1}^n c_j^2 x_j \right\} \\ \text{s.t.} \quad & \sum_{j=1}^n w_j x_j \leq b \\ & x \in \{0, 1\}^n \end{aligned}$$

where $c^1, c^2, w \in \mathbb{Z}^n$, with 15, 20, 25, and 30 variables, each containing 10 instances, and denoted by $C15$, $C20$, $C25$, and $C30$, respectively. The instances are generated following the scheme proposed in Kirlik and Sayın (2014), i.e., c_j^1, x_j^2, w_j are drawn randomly from the interval $[1, 1000]$ for each $j = 1, \dots, n$, and $b = \lfloor \frac{\sum_{j=1}^n w_j}{2} \rfloor$. We imposed a run time limit of 3 hours for the solution of each instance.

For each instance, we run Algorithm 1 as well as a variation of Algorithm 1 in which after finding an efficient solution, a maximal objective-aligning subset of variables is randomly selected and the corresponding cut (4) is added instead of the no-good constraint (because the corresponding cut (4) dominates the no-good constraint).

Table 2: Performance of Algorithm 1 with and without cuts derived from more intelligently selected maximal objective-aligning subsets.

Set	#NDPs	No OA cuts		With OA cuts		%Improvement	
		Time (Sec.)	#IP	Time (Sec.)	#IP	Time	#IP
C15	11	0.33	33	0.24	31	27.3	6.1
	8	2.04	85	1.53	79	25.0	7.1
	3	3.10	104	2.63	100	15.2	3.8
	9	0.63	46	0.52	44	17.5	4.3
	6	0.19	20	0.14	18	26.3	10.0
	23	247.04	758	202.09	727	18.2	4.1
	7	0.65	47	0.64	46	1.5	2.1
	19	62.07	420	48.68	406	21.6	3.3
	11	5.74	142	4.99	133	13.1	6.3
	8	2.72	97	2.26	93	16.9	4.1
C20	6	1.96	81	1.51	75	23.0	7.4
	4	0.74	53	0.77	48	-4.1	9.4
	9	15.25	191	12.67	186	16.9	2.6
	3	0.16	18	0.13	17	18.8	5.6
	6	4.94	130	3.76	126	23.9	3.1
	9	0.60	42	0.44	37	26.7	11.9
	11	1,178.71	1,401	1,002.54	1,388	14.9	0.9
	11	6.26	139	5.16	134	17.6	3.6
	10	9.56	164	8.23	157	13.9	4.3
	10	585.29	987	545.21	970	6.8	1.7
C25	16	2,609.51	1,454	2,545.14	1,434	2.5	1.4
	12	336.53	738	320.43	728	4.8	1.4
	20	4,004.12	1,651	4,178.42	1,628	-4.4	1.4
	12	60.70	361	58.41	354	3.8	1.9
	18	80.07	392	72.83	376	9.0	4.1
C30	19	8,004.21	1,886	7,597.94	1,860	5.1	1.4
	12	1,294.96	990	1,262.03	976	2.5	1.4
	10	2.43	82	2.35	81	3.3	1.2
	13	322.43	669	334.76	661	-3.8	1.2
	14	295.55	606	258.64	576	12.5	5.0
	14	1.52	65	1.38	61	9.2	6.2
	14	1.52	65	1.38	61	9.2	6.2
Avg	11.10	617.42	446.84	596.02	437.10	12.43	4.14

As mentioned above, we assume that Algorithm 1 has access to all maximal objective-aligning subsets of variables. Therefore, before presenting the results, we provide, in Figure 9, the number of maximal objective-aligning subsets of a particular size for the instances used in the study (averaged over the instances in a set).

As expected, we see that the number of maximal objective-aligning subsets grows as the number of variables grows, and that, for these instances, the size of the maximal objective-aligning subsets tends to be relatively small (most have size three and four).

The results of the computational experiment can be found in Table 1, where we report the number of nondominated points (#NDP), the solution time (Time (sec.)), and the number of single-objective integer programs solved (#IP). The algorithm was unable to solve 9 out of 40 instances within the imposed time limit of three hours for both settings; no results for these instances are reported. We observe that adding a small number of cuts derived from objective-aligning subsets of variables – one for each efficient solution found – is beneficial and improves the solution time, on average, by about 9.1%, and reduces the number of single-objective optimization problems solved, on average, by about 2.4%.

The encouraging results reported in Table 1 were obtained by adding a cut derived from a randomly selected maximal objective-aligning subset each time an efficient solution was found. In Table 2, we report the results obtained when a more intelligent scheme is used to select a maximal objective-aligning subset. After finding an efficient point and updating \bar{F} , we solve the LP relaxation of the new formulation to get LP solution x^{LP} , and choose the objective-aligning subset that minimizes the left-hand side of (4) when x^{LP} is substituted into it. The average solution time improvement goes up from about 9.1% to about 12.4% and the average reduction in number of IPs solved goes up from 2.4% to 4.1%.

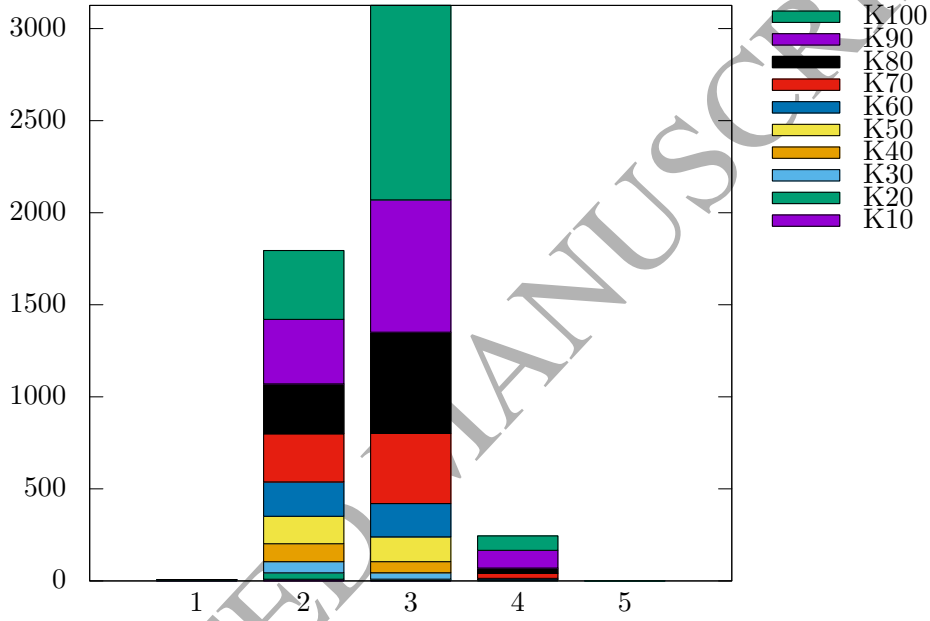


Figure 10: The (average) number of maximal objective aligning sets of a particular size for K10-K100.

7.2. Enhancing a sophisticated criterion space search algorithm

A criterion space search algorithm (typically) solves a large number of single-objective optimization problems when solving a MBP. Therefore, by reducing the time it takes to solve the individual single-objective optimization problems, we can enhance the performance of a criterion space search algorithm. Next, we show how the cuts of the form (4) can be used to that.

The Quadrant Shrinking Method (QSM) is one the fastest algorithms for solving 0-1 integer programs with three objectives (Boland et al., 2017b). In each iteration, when solving a single-objective 0-1 integer program, QSM exploits the fact that some of the efficient solutions produced in previous iterations are feasible. QSM chooses the best one among them and provides it as an initial solution to the integer programming solver to warm-start the search. Here, we take this a step further and add a cut of the form (4) for each of the other feasible efficient solutions. Specifically, for each of the other feasible efficient solutions, we randomly select a maximal objective-aligning subset and add the associated cut of the form (4). Because QSM maintains a single efficient solution for

each nondominated point that it discovers, none of the cuts added will be redundant and none of the cuts added will cut off the best feasible efficient solution provided as a warm start. Adding these cuts may improve the dual bound, which, in turn, may result in faster solution times.

To show the effectiveness of this idea (and thus, at the same time, the benefit of using cuts of the form (4)), we use 100 3-objective knapsack instances created by Kirlik and Sayın (2014). There are 10 sets of instances, each with 10 instances, denoted by $K10, K20, \dots, K100$, where the number following the K indicates the number of variables in the instances in that subset. As before, we first provide, in Figure 10, information regarding the number of maximal objective-aligning subsets of a particular size for the instances used in the study (averaged over the instances in a set). We see that the number of maximal objective-aligning subsets grows as the number of variables grows, but that the size of the maximal objective-aligning subsets remains relatively small (most have size two and three). Note that for instances in the sets $K90$ and $K100$, there are maximal objective-aligning subsets with size 5, but the number of them is very small, less than four.

To ensure that the single-objective 0-1 integer programs do not grow too much, we add no more than $\lceil \ln(n) \rceil$ cuts, i.e., we consider no more than $\lceil \ln(n) \rceil$ other feasible efficient solutions, where we give higher priority to higher quality feasible efficient solutions. Furthermore, when adding cuts, setting “MIP Emphasis” to “Emphasize Optimality Over Feasibility” in the integer programming solver appears to be slightly more efficient.

Figure 11 shows the ratio of the solution time of QSM with cuts to the solution time of QSM without cuts for three different cut generation schemes. More specifically, in Figure 11a standard no-good cuts of the form (6) are added, in Figure 11b, cuts of the form (4) derived from randomly selected maximal objective aligning subsets are added, and in Figure 11c, cuts of the form (4) derived from randomly selected minimum-cardinality maximal objective aligning subsets are added.

We see that adding standard no-good cuts has little effect; the average ratio is close to one. On the other hand, adding cuts derived from randomly selected maximal objective-aligning sets shows an improvement; the average ratio is about 0.95, i.e., a 5% improvement in solution time (on average). For the smaller instances, e.g., in the sets $K30$ and $K40$, improvements in solution time of 20% or more can be seen. However, for the larger instances, e.g., in the sets $K90$ and $K100$, deteriorations in solution time of up to 15% can be seen. Using only maximal objective-aligning subset of minimum cardinality leads to more stable and better performance; the performance gets better as the size of the instances grows. For the larger instances, e.g., in the sets $K90$ and $K100$, we see improvements in solution time of more than 20%.

These results demonstrate the potential of cuts of type (4) to reduce solutions times, even in state-of-the-art methods, but also that further algorithm engineering is needed to fully exploit their potential.

8. Final remarks

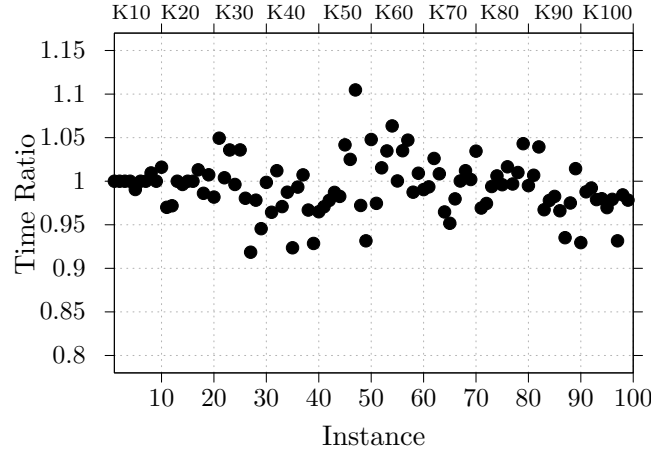
It goes without saying that there is scope and need for further research on algorithm engineering to exploit the cuts derived from maximal objective-aligning subsets most effectively and efficiently, e.g., investigation of cut generation, cut selection, and cut separation strategies.

As well as enabling reduction of the objective space dimension and providing cuts that can be useful in solving MBPs, the properties discussed in this paper can be used to deduce bounds on the number of efficient solutions, and nondominated points, of a MBP. We give two examples.

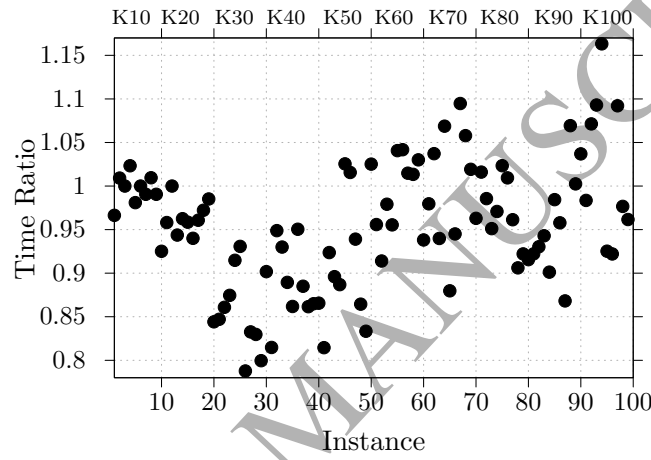
If $A \subseteq \{1, \dots, n\}$ is objective-aligning for a set of objectives, $c^1, c^2, \dots, c^N \in \mathbb{R}^n$, then for any binary vector $y \in \{0, 1\}^n$, the subspace-fixing (A, y) yields $Z_{A,y}$ totally ordered, and so either $Z_{A,y} \cap O$ is empty ($Z_{A,y}$ does not contain the image of any feasible solution), or $Z_{A,y} \cap O_N$ contains a single point. Since entries, y_j , in y , for $j \in A$, are irrelevant to the description of $Z_{A,y}$, there can be at most $2^{n-|A|}$ distinct sets $Z_{A,y}$, over all binary vectors y . Since each contains at most one nondominated point, it must be that $|O_N| \leq 2^{n-|A|}$.

If a set of objectives, $c^1, c^2, \dots, c^N \in \mathbb{R}^n$, is sortable-by- K - L -sets, then, for any MBP with this set of objectives, we have, by Proposition 5, that

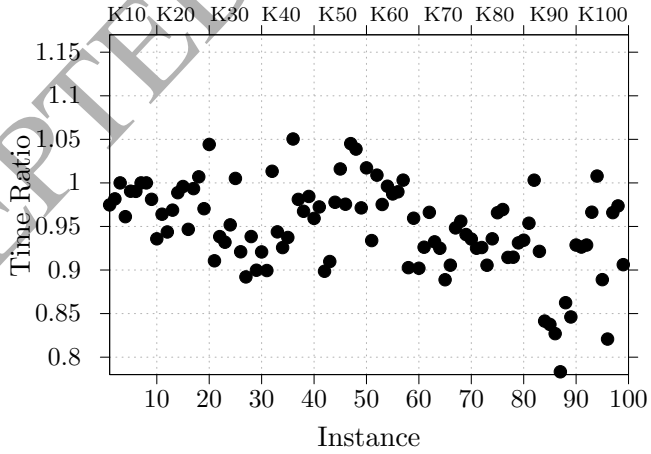
$$|F_E|, |O_N| \leq \sum_{i=0}^{K-1} \binom{n}{i} + \sum_{i=L+1}^n \binom{n}{i} + 1.$$



(a) Cuts of the form (6), i.e., no good cuts.



(b) Cuts of the form (4) derived from randomly selected maximal objective-aligning subsets.



(c) Cuts of the form (4) derived from randomly selected minimum-cardinality maximal objective-aligning subsets.

Figure 11: The ratio of the solution time of QSM and the solution time of QSM without such cuts for different cut generation schemes.

This suggests that if K is close to zero and L is close to n , we may be able to use an enumerative approach to compute all efficient solutions or nondominated points of the MBP.

References

- Atamtürk, A., Nemhauser, G. L., Savelsbergh, M. W. P., 2000. Conflict graphs in solving integer programming problems. *European Journal of Operational Research* 121 (1), 40 – 55.
- Belotti, P., Soylu, B., Wiecek, M. M., 2016. Fathoming rules for biobjective mixed integer linear programs: Review and extensions. *Discrete Optimization* 22, 341 – 363.
- Böckler, F., Ehrgott, M., Morris, C., Mutzel, P., 2017. Output-sensitive complexity of multiobjective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis* 24 (1-2), 25–36.
- Boland, N., Charkhgard, H., Savelsbergh, M., 2016. The L-shape search method for triobjective integer programming. *Mathematical Programming Computation* 8 (2), 217–251.
- Boland, N., Charkhgard, H., Savelsbergh, M., 2017a. A new method for optimizing a linear function over the efficient set of a multiobjective integer program. *European Journal of Operational Research* 260 (3), 904 – 919.
- Boland, N., Charkhgard, H., Savelsbergh, M., 2017b. The quadrant shrinking method: A simple and efficient algorithm for solving tri-objective integer programs. *European Journal of Operational Research* 260 (3), 873 – 885.
- Cornuéjols, G., Mar 2008. Valid inequalities for mixed integer linear programs. *Mathematical Programming* 112 (1), 3–44.
- Dächert, K., Klamroth, K., Lacour, R., Vanderpooten, D., 2017. Efficient computation of the search region in multi-objective optimization. *European Journal of Operational Research* 260 (3), 841 – 855.
- Ehrgott, M., 2005. *Multicriteria optimization*, 2nd Edition. Springer, New York.
- Ehrgott, M., Gandibleux, X., 2000. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR-Spektrum* 22 (4), 425–460.
- Ehrgott, M., Gandibleux, X., 2002. *Multiple criteria optimization: state of the art annotated bibliographic surveys*. Kluwer Academic Publishers.
- Ehrgott, M., Gandibleux, X., 2007. Bound sets for biobjective combinatorial optimization problems. *Computers & Operations Research* 34 (9), 2674–2694.
- Engau, A., Wiecek, M. M., 2008. Interactive coordination of objective decompositions in multiobjective programming. *Management Science* 54 (7), 1350–1363.
- Figueira, J. R., Fonseca, C. M., Halffmann, P., Klamroth, K., Paquete, L., Ruzika, S., Schulze, B., Stiglmayr, M., Willems, D., 2017. Easy to say they are hard, but hard to see they are easy towards a categorization of tractable multiobjective combinatorial optimization problems. *Journal of Multi-Criteria Decision Analysis* 24 (1-2), 82–98.
- Gabow, H. N., Tarjan, R. E., 1984. Efficient algorithms for a family of matroid intersection problems. *Journal of Algorithms* 5 (1), 80 – 131.
- Gal, T., Hanne, T., 1999. Consequences of dropping nonessential objectives for the application of MCDM methods. *European Journal of Operational Research* 119 (2), 373 – 378.
- Gal, T., Hanne, T., 2006. Nonessential objectives within network approaches for MCDM. *European Journal of Operational Research* 168 (2), 584 – 592, feature Cluster on Mathematical Finance and Risk Management.

- Garey, M. R., Johnson, D. S., 1979. Computers and Intractability. Bell Telephone Laboratories, Incorporated.
- Gorski, J., 2010. Multiple objective optimization and implications for single objective optimization. Ph.D. thesis, Bergische Universität Wuppertal, shaker Verlag, Aachen, Germany. <http://elpub.bib.uni-wuppertal.de/servlets/DerivateServlet/Derivate-1847/dc1018.pdf>.
- Gorski, J., Paquete, L., Pedrosa, F., 2012. Greedy algorithms for a class of knapsack problems with binary weights. *Computers & Operations Research* 39 (3), 498 – 511.
- Gu, Z., Nemhauser, G. L., Savelsbergh, M. W. P., 1998. Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS Journal on Computing* 10 (4), 427–437.
- Kirlik, G., Sayın, S., 2014. A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. *European Journal of Operational Research* 232 (3), 479 – 488.
- Lindroth, P., Patriksson, M., Strömberg, A.-B., 2010. Approximating the pareto optimal set using a reduced set of objective functions. *European Journal of Operational Research* 207 (3), 1519 – 1534.
- Lokman, B., Köksalan, M., 2013. Finding all nondominated points of multi-objective integer programs. *Journal of Global Optimization* 57 (2), 347–365.
- Malinowska, A., Torres, D., 2008. Computational approach to essential and nonessential objective functions in linear multicriteria optimization. *Journal of Optimization Theory and Applications* 139 (3), 577–590.
- Malinowska, A. B., 2008. Weakly and properly nonessential objectives in multiobjective optimization problems. *Operations Research Letters* 36 (5), 647 – 650.
- Morrison, D. R., Jacobson, S. H., Sauppe, J. J., Sewell, E. C., 2016. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization* 19, 79 – 102.
- Özpeynirci, Ö., Köksalan, M., 2010. An exact algorithm for finding extreme supported nondominated points of multiobjective mixed integer programs. *Management Science* 56 (12), 2302–2315.
- Parragh, S. N., Tricoire, F., 2015. Branch-and-bound for bi-objective integer programming. *Optimization Online* http://www.optimization-online.org/DB_FILE/2015/07/4444.pdf.
- Savelsbergh, M. W. P., 1994. Preprocessing and probing techniques for mixed integer programming problems. *ORSA Journal on Computing* 6 (4), 445–454.
- Seipp, F., 2013. On adjacency, cardinality, and partial dominance in discrete multiple objective optimization. Ph.D. thesis, TU Kaiserslautern, germany. <http://www.gbv.de/dms/tib-ub-hannover/773490043.pdf>.
- Soylu, B., 2018. The search-and-remove algorithm for biobjective mixed-integer linear programming problems. *European Journal of Operational Research* 268 (1), 281 – 299.
- Soylu, B., Yıldız, G. B., 2016. An exact algorithm for biobjective mixed integer linear programming problems. *Computers & Operations Research* 72, 204–213.
- Stidsen, T., Andersen, K. A., Dammann, B., 2014. A branch and bound algorithm for a class of biobjective mixed integer programs. *Management Science* 60 (4), 1009–1032.
- Thoi, N., 2012. Criteria and dimension reduction of linear multiple criteria optimization problems. *Journal of Global Optimization* 52 (3), 499–508.
- Woeginger, G. J., Yu, Z., 1992. On the equal-subset-sum problem. *Information Processing Letters* 42 (6), 299 – 302.

Appendix A: Complexity of recognizing a nexus point

We show that recognizing that a given point, $p \in \mathbb{R}^N$, is a nexus point for the subspace-fixing (A, y) is NP-hard. It suffices to show that the special case with $N = 2$ and $A = \{1, \dots, n\}$ (in which case y is irrelevant), is NP-hard. We thus define the decision problem 2-OBJ-NEXUS-POINT, as follows: given inputs consisting of the pair of objectives $f, g \in \mathbb{Z}^n$ and a point $p \in \mathbb{Z}^2$, is p a nexus point for f, g ? Note that for this analysis, we assume integer-valued inputs. We first show that the complement of 2-OBJ-NEXUS-POINT is NP-complete.

Theorem 2. *The complement of 2-OBJ-NEXUS-POINT is NP-complete.*

PROOF. Our proof is similar to the proof of Theorem 1 but we use PARTITION PROBLEM instead of EQUAL-SUBSET-SUM. Note first that the complement of 2-OBJ-NEXUS-POINT is obviously in NP.

We will now show that PARTITION PROBLEM is polynomially reducible to the complement of 2-OBJ-NEXUS-POINT, where PARTITION PROBLEM is defined as follows: given a set $\{a_1, \dots, a_n\}$ of positive integers, does there exist a non-empty subset, $S \subset \{1, \dots, n\}$, with $\sum_{j \in \{1, \dots, n\} \setminus S} a_j = \sum_{j \in S} a_j$? In this proof, to remove trivial cases, we assume that $\frac{\sum_{j \in \{1, \dots, n\}} a_j}{2} \in \mathbb{Z}$.

Given such an instance of PARTITION PROBLEM, we construct an instance of 2-OBJ-NEXUS-POINT, $((f, g), p)$, by

$$p = \left(\frac{\sum_{j \in \{1, \dots, n\}} a_j}{2}, \frac{\sum_{j \in \{1, \dots, n\}} a_j}{2} \right),$$

$$f_j = a_j - 2^{-j},$$

and

$$g_j = a_j + 2^{-j},$$

for all $j = 1, \dots, n$. Observe that for any $S \subseteq \{1, \dots, n\}$,

$$\lceil f(S) \rceil = \lfloor g(S) \rfloor = \sum_{j \in S} a_j, \quad (7)$$

since $0 \leq \sum_{j \in S} 2^{-j} < 1$.

First suppose that this instance, $((f, g), p)$, of 2-OBJ-NEXUS-POINT, is a NO instance, so there exist $S \subset \{1, \dots, n\}$, with $p_1 < f(S)$ and $p_2 > g(S)$. Then

$$\lceil p_1 \rceil \leq \lceil f(S) \rceil = \lfloor g(S) \rfloor \leq \lfloor p_2 \rfloor,$$

by (7). But, since $\frac{\sum_{j \in \{1, \dots, n\}} a_j}{2} \in \mathbb{Z}$, we have $\lceil p_1 \rceil = \lfloor p_2 \rfloor$, and hence, it must be that

$$\frac{\sum_{j \in \{1, \dots, n\}} a_j}{2} = \lceil p_1 \rceil = \lfloor p_2 \rfloor = \lceil f(S) \rceil = \lfloor g(S) \rfloor = \sum_{j \in S} a_j,$$

so PARTITION PROBLEM is a YES instance.

Finally, suppose that PARTITION PROBLEM is a YES instance, with non-empty $S \subset \{1, \dots, n\}$, with $\sum_{j \in \{1, \dots, n\} \setminus S} a_j = \sum_{j \in S} a_j = \frac{\sum_{j \in \{1, \dots, n\}} a_j}{2}$. Hence,

$$p_2 = \frac{\sum_{j \in \{1, \dots, n\}} a_j}{2} < \frac{\sum_{j \in \{1, \dots, n\}} a_j}{2} + \sum_{j \in S} 2^{-j} = \sum_{j \in S} a_j + \sum_{j \in S} 2^{-j} = g(S).$$

Similarly,

$$p_1 = \frac{\sum_{j \in \{1, \dots, n\}} a_j}{2} > \frac{\sum_{j \in \{1, \dots, n\}} a_j}{2} - \sum_{j \in S} 2^{-j} = \sum_{j \in S} a_j - \sum_{j \in S} 2^{-j} = f(S).$$

Thus the 2-OBJ-NEXUS-POINT instance, $((f, g), p)$, is a NO instance. The result follows. \square

Corollary 6. *2-OBJ-NEXUS-POINT is NP-hard.*