# Pareto Leap: An Algorithm for Biobjective Mixed-Integer Programs*

Philip J. de Castro†
Margaret M. Wiecek‡

**Abstract**

Many real-life optimization problems need to make decisions with discrete variables and multiple, conflicting objectives. Due to this need, the ability to solve such problems is an important and active area of research. We present a new algorithm, called Pareto Leap, for identifying the (weak) Pareto slices of biobjective mixed-integer programs (BOMIPs), even when Pareto slices intersect. The underlying theory holds for BOMIPs whose continuous relaxations are nonlinear, while the implementation of the algorithm is limited only by the capability of solvers that are used. The algorithm works by starting on a Pareto slice of the outcome set of a BOMIP and then "leaping" to another Pareto slice, which we demonstrate on numerical examples.

**Keywords:** multiobjective optimization, mixed-integer programming, tabu constraints, achievement scalarizing functions
**MSC Codes:** 90B50, 90C11, 90C26, 90C29, 90C30

## 1 Introduction

Many real-life optimization problems require discrete decisions to be made. For example, in a facility location problem, a decision maker (DM) must choose whether or not to open a facility in a specific location. This is a "yes" or "no" question, which is readily modeled using a binary variable. On the other hand, DMs often have multiple conflicting objectives that they desire to reconcile and make a decision that balances these conflicts. This aspect of decision making is modeled by multiobjective optimization problems (MOPs). Therefore, it is of particular interest to bring these two components of decision making together and to develop methods for solving multiobjective mixed-integer programs (MOMIPs), that is, methods for computing the solution set in the decision space and/or objective space.

There is a recent survey in [16] where multiobjecitve mixed-integer linear programs (MOMILPs) are discussed in-depth. In this work, however, we focus on the more general case where the underlying continuous problem of an MOMIP is nonlinear. Broadly speaking, the literature on solving such general MOMIPs can be categorized into two main approaches: approximation and exact methods, respectively. Interestingly, the majority of the literature is focused on approximations.

Since MOMIPs are computationally difficult problems, approximations methods allow for speedy computational handling of such problems by producing an approximation of the Pareto set, typically within some tolerance preset by the user; see e.g. [4, 10, 5, 9]. In [18], a new approximation method is presented which uses inner and outer approximations of Pareto sets of subproblems of an MOMIP. Many approximation methods use the well-established technique of extending branch-and-bound to the multiobjective case (e.g., [3, 20, 7, 13, 8, 12]).

Exact methods, on the other hand, are able to provide provably efficient solutions to MOMIPs. Some methods, such as [15], also use extensions of branch-and-bound to find exact solutions.

Besides branch-and-bound, exact methods extend classical multiobjective scalarizations to handle integer variables. In [10], the weighted-sum method is used to produce a subset of nondominated points of arbitrary

*DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. OPSEC9645

†School of Mathematical and Statistical Sciences, Clemson University, Clemson, SC 29634-0975. https://orcid.org/0009-0003-7306-9324

‡School of Mathematical and Statistical Sciences, Clemson University, Clemson, SC 29634-0975. https://orcid.org/0000-0001-6853-2626

precision as described by the coverage gap. Another technique, decomposition-based methods, utilize the observation that it is possible to formulate subproblems of an MOMIP by fixing the integer variables at feasible values and optimizing over the continuous variables. These subproblems go by various names such as "subproblems", "patches", or "slices"; see, e.g., [2] for an exact method and [14] for an approximation method. However, the literature [2, 13] has discussed that this is a difficult approach since up to this point, there has been no method which can identify only the slices which contribute to the Pareto set.

Particularly relevant for this work are decomposition methods which use so-called "tabu constraints" or "no-good constraints" [17]. In the context of mixed-integer programming, tabu constraints are introduced in [15] and [26] and are of the form $||z - \bar{z}|| \geq 1$, where $z$ is an integer-valued variable, $||\cdot||$ is a norm, usually the $\ell_1$-norm, and $\bar{z}$ is some integer reference point, typically representing the current Pareto slice which is to be considered "tabu". Although this is a nonconvex constraint, most modern solvers can reformulate it into a bilinear binary constraint, for which there are many suitable solution methods in the literature (e.g., [6]). The idea for including tabu constraints is to cut off possible integer assignments as the algorithm searches for efficient solutions. To the best of the authors' knowledge, tabu constraint methods in the literature are exclusively focused on MOMILPs. For example, [24, 25] use tabu constraints in a two-phase method for biobjective mixed-integer linear programs (BOMILPs), while [23] proposes a method for MOMILPs which iteratively adds tabu constraints as it searches the objective space using an approximation of the nadir point. In [1], tabu constraints are also iteratively added but this method uses extreme points of previously found slices to define cuts in the objective space to ensure that an efficient solution has been found. Also of note, [21] combines tabu constraints with the boxed-line method described in [22] to find the Pareto set of a BOMILP. It is worth noting that these tabu methods are able to not only idenitfy Pareto slices, but to also produce a representation of the Pareto set. However, this is due to the fact that the underlying continuous problem is *linear*.

The contribution of the present work is threefold.

1. We develop a two-phase method, which uses a reference point on a Pareto slice of a BOMIP and a tabu constraint to identify a new Pareto slice, even when the underlying continuous problem is nonlinear.

2. Necessary and sufficient conditions for two Pareto slices to intersect are proven, a result which has not been explicitly discussed in the existing literature.

3. Our theoretical results lead naturally to an algorithm, Pareto Leap, which exactly identifies all (weak) Pareto slices of a BOMIP. We test Pareto Leap on test problems from the literature.

Thus, Pareto Leap provides a solution to the problem noted in the literature of identifying only the slices which contribute to the Pareto set. Furthermore, Pareto Leap presents improvements to the class of tabu methods in three ways. First, we have filled the gap in the literature by developing a tabu method for nonlinear problems. Second, the implementation of Pareto Leap ensures that the constraint set of the two-phase method tends not to grow over the course of the algorithm, unlike the tabu methods described in [23, 1], meaning that each iteration has as few tabu constraints as possible. Finally, our method has the advantage of flexibility in its implementation. Pareto Leap can be implemented with mixed-integer and continuous solvers which are best suited for the mathematical properties of the problem under consideration.

The rest of the paper is organized as follows. In Section 2, essential definitions of multiobjective optimization are presented. Furthermore, we provide details about Achievement Scalarizing Functions (ASFs), a class of scalarizing functions for MOPs, which we use to test the efficiency of a point found over the course of the algorithm. Section 3 provides the theoretical groundwork for Pareto Leap. Following this, Section 4 presents the algorithm and demonstrates Pareto Leap on an example, which includes intersecting slices. Section 5 shows the results of performing numerical experiments with test problems from the literature. Finally we discuss further work in Section 6.

## 2 Preliminaries

We start with the following definitions.

**Definition 2.1.** Let $n_C, n_I \in \mathbb{N}$ and $X \subseteq \mathbb{R}^{n_C} \times \mathbb{Z}^{n_I}$ be a nonempty set. Let $z^* \in \mathbb{Z}^{n_I}$. We define the set $X(z^*) := \{(x, z) \in X \mid z = z^*\}$.

**Remark 2.2.** Observe that it is possible for $X(z^*) = \emptyset$.

**Definition 2.3.** Let $u = (u_1, \ldots, u_n), v = (v_1, \ldots, v_n) \in \mathbb{R}^n$ be $n$-dimensional vectors. Then

1. $u < v \Leftrightarrow u_j < v_j$ for all $1 \leq j \leq n$;
2. $u \leq v \Leftrightarrow u_j \leq v_j$ for all $1 \leq j \leq n$ and there exists $k$ such that $u_k < v_k$;
3. $u \leqq v \Leftrightarrow u_j \leq v_j$ for all $1 \leq j \leq n$.

We additionally define the closed pointed convex cone $\mathbb{R}^2_{\geqq} = \{x \in \mathbb{R}^2 \mid x \geqq 0\}$ and denote the set of nonnegative scalars by $\mathbb{R}_{\geq} = \{x \in \mathbb{R} \mid x \geq 0\}$.

**Definition 2.4.** Let $X \subseteq \mathbb{R}^{n_C} \times \mathbb{Z}^{n_I}$ be a nonempty set. A **sequence** is a subset of $X$ indexed by $n = 1, 2, \ldots$ and denoted by $\{(x_n, z_n)\}_{n=1}^{\infty}$.

We now consider a general biobjective mixed-integer optimization problem. Let $n_C, n_I \in \mathbb{N}$. For $i = 1, 2$ let $f_i : \mathbb{R}^{n_C} \times \mathbb{Z}^{n_I} \to \mathbb{R}$ be a continuous and convex function in $x$ for every $z \in \mathbb{Z}^{n_I}$. Let $X$ be a nonempty set such that if for some $z \in \mathbb{Z}^{n_I}$ $X(z)$ is nonempty, then $X(z)$ is also a compact set. The biobjective mixed-integer programming problem is given by the following.

$$\begin{aligned} \min \quad & f(x,z) = \begin{bmatrix} f_1(x,z) \\ f_2(x,z) \end{bmatrix} \\ \text{s.t.} \quad & (x,z) \in X \end{aligned} \tag{BOMIP}$$

We say that $X$ is the **feasible set** of (BOMIP) and the set $Y = \{f(x,z) \mid (x,z) \in X\}$ is the **outcome set** of (BOMIP).

Multiobjective mixed-integer optimization problems have properties which are not shared with multiobjective optimization problems in general. In particular, the BOMIP may be decomposed into so-called slice problems.

**Definition 2.5.** Let $X$ be the feasible set for (BOMIP) and let $z^* \in \mathbb{Z}^{n_I}$ be such that $X(z^*)$ is a nonempty set. Then the **slice set corresponding to $\mathbf{z}^*$** is $Y(z^*) := \{f(x, z^*) \mid (x, z^*) \in X(z^*)\}$. When it is clear in context, we refer to $Y(z^*)$ as a **slice**.

Let $z^* \in \mathbb{Z}^{n_I}$ be such that $X(z^*)$ is a nonempty compact set. Then the slice problem corresponding to $z^*$ is

$$\begin{aligned} \min \quad & \begin{bmatrix} f_1(x,z^*) \\ f_2(x,z^*) \end{bmatrix} \\ \text{s.t.} \quad & (x,z^*) \in X(z^*). \end{aligned} \tag{BOMIP($z^*$)}$$

**Remark 2.6.** Observe that (BOMIP($z^*$)) is a biobjective optimization problem with continuous variables.

In this work, we use Pareto efficiency for the notion of optimality.

**Definition 2.7.** Let $z^* \in \mathbb{Z}^{n_I}$ be fixed. Let $X$ and $X(z^*)$ be the feasible sets of (BOMIP) and (BOMIP($z^*$)), respectively.

i. Let $(x, z)$ be feasible for (BOMIP). We say that $(x, z)$ is a **(weakly) efficient solution** if there is no feasible $(x', z') \in X$ such that
$$f(x', z')(<) \leq f(x, z).$$
We denote the set of (weakly) efficient solutions by $\mathcal{E}_{(w)}$. If $(x, z)$ is a (weakly) efficient solution, then we say that $f(x, z)$ is a **(weak) Pareto point** and we denote the set of (weak) Pareto points by $\mathcal{P}_{(w)}$.

ii. Let $(x, z^*)$ be feasible for (BOMIP($z^*$)). We say that $(x, z^*)$ is a **(weakly) $z^*$-efficient solution** if there is no feasible $(x', z^*) \in X(z^*)$ such that
$$f(x', z^*)(<) \leq f(x, z^*),$$
and we denote the set of (weakly) efficient $z^*$-solutions by $\mathcal{E}_{(w)}(z^*)$. If $(x, z^*)$ is a (weakly) efficient $z^*$-solution, then we say that $f(x, z^*)$ is a **(weak) $z^*$-Pareto point** and we denote the set of (weakly) $z^*$-Pareto points by $\mathcal{P}_{(w)}(z^*)$.

Definition 2.7 distinguishes between "global" optimal solutions and values (efficient solutions and Pareto points) and "local" optimal solutions and values (efficient $z^*$-solutions and $z^*$-Pareto points).

It may be observed that for $z^* \in \mathbb{Z}^{n_I}$, the $z^*$-efficient solutions and $z^*$- Pareto outcomes of (BOMIP($z^*$)) may have nonempty intersections with the efficient solutions and Pareto outcomes of (BOMIP), respecitvely. Therefore, as discussed in the introduction, a naive approach to finding the efficient set of (BOMIP) could be to find the $z^*$-Pareto set of each slice problem, take the union of all $z^*$-Pareto outcomes for all slice problems, and apply a nondominance operator to yield the Pareto set for (BOMIP). This naive approach is, of course, computationally inefficient at best. To develop a better algorithm for solving (BOMIP), we define Pareto slices.

**Definition 2.8.** Let $z^* \in \mathbb{Z}^{n_I}$ such that $X(z^*) \neq \emptyset$. We say that $Y(z^*)$ is a **(weak) Pareto slice** if $\mathcal{P}_{(w)}(z^*) \cap \mathcal{P}_{(w)} \neq \emptyset$.

We next introduce achievement scalarizing functions [28, 29]. These functions will be useful in ensuring that Pareto Leap only selects slices which contribute to the Pareto set.

**Definition 2.9.** Let $\sigma : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ be a continuous function. Let $y, y', r \in \mathbb{R}^p$.

1. We say that $r$ is the **reference point** of $\sigma(\cdot, \cdot)$;
2. We say that $\sigma(\cdot, \cdot)$ is **strictly order preserving** if $y < y'$ implies that $\sigma(y, r) < \sigma(y', r)$;
3. We say that $\sigma(\cdot, \cdot)$ is **order representing** if
$$\{y \in \mathbb{R}^p \mid \sigma(y, r) < 0\} = r - \text{int}(\mathbb{R}^p_{\geqq}),$$
where $\text{int}(S)$ denotes the interior of a set $S$.
4. If $\sigma(\cdot, \cdot)$ is both strictly order preserving and order representing, then we say that $\sigma(\cdot, \cdot)$ is a strict achievement scalarizing function (ASF).

The property of ASFs most salient for this work, however, is the following theorem, which shows how a strict ASF can be used to define a separating set at a point.

**Theorem 2.10** ([19])**.** Let $\sigma : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ be a strict ASF. Let $y, r \in \mathbb{R}^2$.

i. $\sigma(y, r) < 0$ if and only if $y \in r - \mathbb{R}^2_{>}$;

ii. $\sigma(y, r) = 0$ if and only if $y \in \partial(r - \mathbb{R}^2_{\geqq})$;

iii. $\sigma(y, r) > 0$ if and only if $y \in (r - \mathbb{R}^2_{\geqq})^C$.

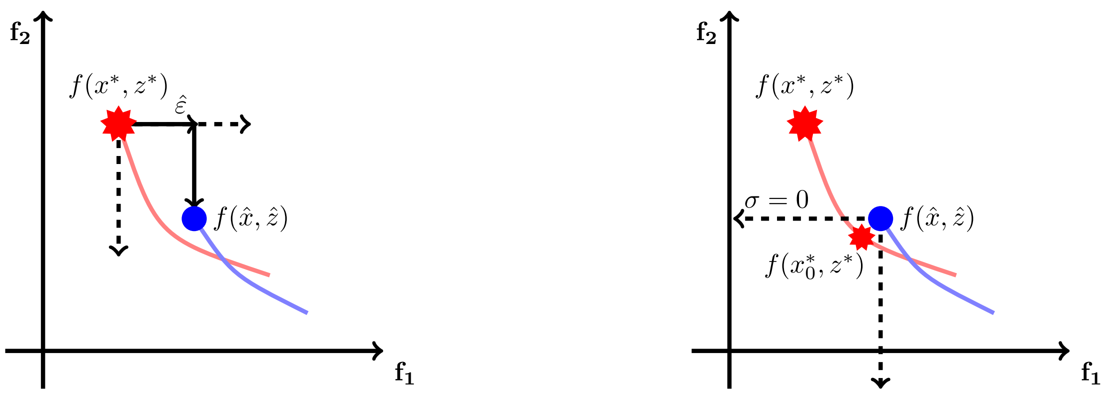
We refer the reader to [28, 29, 19] for more details on ASFs. With these preliminaries in place, we proceed to the main contribution of this work. In the next section, we develop the underlying theory for the Pareto Leap algorithm.

# 3 Theoretical Results

To find the (weak) Pareto slices of (BOMIP), we propose a two-phase method. Figure 1 shows geometrically the underlying intuition of both phases.

In the first phase, the so-called leap problem uses a (weak) Pareto point $f(x^*, z^*)$ and searches the cone $f(x^*, z^*) + \{(y_1, y_2) \in \mathbb{R}^2 \mid y_1 \geq 0, y_2 \leq 0\}$ for a new slice in the objective space of (BOMIP). Figure 1a depicts this by showing a "leap" away from the point $f(x^*, z^*)$ down to the point $f(\hat{x}, \hat{z})$. The leap is modeled by taking the smallest step away from the starting slice and moving as far down as possible. To ensure that a new slice is found, a tabu constraint is used to force a change in the integer variables.

During the second phase, in order to ensure that the algorithm has found another Pareto slice, an ASF is used by setting $f(\hat{x}, \hat{z})$ as the reference point and searching the cone $f(\hat{x}, \hat{z}) - \mathbb{R}^2_{\geqq}$, as shown in Figure 1b. If the starting slice $Y(z^*)$ is outside of the cone $f(\hat{x}, \hat{z}) - \mathbb{R}^2_{\geqq}$, then $f(\hat{x}, \hat{z})$ is a (weak) Pareto point, making $Y(\hat{z})$ a (weak) Pareto slice. If, on the other hand, at least a subset of the starting slice is inside $f(\hat{x}, \hat{z}) - \mathbb{R}^2_{\geqq}$, then $f(\hat{x}, \hat{z})$ is not a (weak) Pareto point. The algorithm returns to the first phase and leaps again.




(a) Leap problem. (b) ASF test.

Figure 1: Geometric intuition in the outcome space of a BOMIP for both phases of the Pareto Leap algorithm.

Let $(x^*, z^*)$ be a weakly efficient solution for (BOMIP). The leap problem is formulated as the following single-objective optimization problem.

$$
\begin{aligned}
\min_{x,z,\varepsilon} \quad & \varepsilon && (\text{LEAP}(x^*, z^*)) \\
\text{s.t.} \quad & f_1(x, z) = f_1(x^*, z^*) + \varepsilon && (1) \\
& f_2(x, z) \leq f_2(x^*, z^*) && (2) \\
& \sum_{i=1}^{n_I} |z_i - z_i^*| \geq 1 && (3) \\
& (x, z) \in X && (4) \\
& \varepsilon \geq 0 && (5)
\end{aligned}
$$

In constraint (1), we seek a new point $f_1(x, z)$ which is a step of size $\varepsilon$ away from the reference point $f_1(x^*, z^*)$. In constraint (2), we ensure that the new point found is below from where we started. Constraint (3) is the tabu constraint and ensures that we do indeed shift to a new slice. Finally, constraints (4) and (5) ensure feasibility. The following proposition shows that if $(\hat{x}, \hat{z}, \hat{\varepsilon})$ is an optimal solution for (LEAP$(x^*, z^*)$) then $(\hat{x}, \hat{z})$ is a weakly efficient solution for (BOMIP$(\hat{z})$).

**Proposition 3.1.** Let $(x^*, z^*)$ be a weakly efficient solution for (BOMIP) and let $(\hat{x}, \hat{z}, \hat{\varepsilon})$ be an optimal solution for (LEAP$(x^*, z^*)$). Then $(\hat{x}, \hat{z})$ is a weakly efficient solution for (BOMIP$(\hat{z})$).

*Proof.* Let $(x^*, z^*) \in \mathcal{E}_w$ and $(\hat{x}, \hat{z}, \hat{\varepsilon})$ be an optimal solution for (LEAP$(x^*, z^*)$). Towards a contradiction, suppose $(\hat{x}, \hat{z}) \notin \mathcal{E}_w(\hat{z})$. Then there exists $(x, \hat{z}) \in X(\hat{z})$ such that $f(x, \hat{z}) < f(\hat{x}, \hat{z})$. Consider the point $(x, \hat{z}, \varepsilon = f_1(x, \hat{z}) - f_1(x^*, z^*))$.

Observe that $f(x, \hat{z}) < f(\hat{x}, \hat{z})$ implies that $f_2(x, \hat{z}) < f_2(\hat{x}, \hat{z}) \leq f_2(x^*, z^*)$. Furthermore, since $(x^*, z^*) \in \mathcal{E}_w$ then it must be that $f_1(x, \hat{z}) \geq f_1(x^*, z^*)$ which implies that $\varepsilon \geq 0$. Finally, since $(\hat{x}, \hat{z}, \hat{\varepsilon})$ is feasible for (LEAP$(x^*, z^*)$), it must be that $\hat{z} \neq z^*$. Thus, $(x, \hat{z}, \varepsilon)$ is also feasible for (LEAP$(x^*, z^*)$). However, since $f_1(x, \hat{z}) < f_1(\hat{x}, \hat{z})$ this implies that $f_1(x, \hat{z}) - f_1(x^*, z^*) < f_1(\hat{x}, \hat{z}) - f_1(x^*, z^*)$ and so $\varepsilon < \hat{\varepsilon}$. But this contradicts the optimality of $\hat{\varepsilon}$. Thus, it must be that $(\hat{x}, \hat{z}) \in \mathcal{E}_w(\hat{z})$. □

**Remark 3.2.** Proposition 3.1 shows that an optimal solution to (LEAP$(x^*, z^*)$) is always weakly $\hat{z}$-efficient. In other words, (LEAP$(x^*, z^*)$) finds $\hat{z}$-weak Pareto points, but more is needed to determine weak Pareto points for (BOMIP).

The next proposition presents a sufficient condition for an optimal solution to the first phase to be weakly efficient for (BOMIP).

**Proposition 3.3.** Let $(x^*, z^*)$ be a weakly efficient solution for (BOMIP) and let $(\hat{x}, \hat{z}, \hat{\varepsilon})$ be an optimal solution for (LEAP$(x^*, z^*)$). If $\hat{\varepsilon} = 0$ then $(\hat{x}, \hat{z})$ is a weakly efficient solution for (BOMIP).

*Proof.* Let $(x^*, z^*) \in \mathcal{E}_w$ and $(\hat{x}, \hat{z}, \hat{\varepsilon})$ be an optimal solution for (LEAP$(x^*, z^*)$) such that $\hat{\varepsilon} = 0$. Towards a contradiction, suppose $(\hat{x}, \hat{z}) \notin \mathcal{E}_w$. Then there exists $(x, z) \in X$ such that $f(x, z) < f(\hat{x}, \hat{z})$. Consider the point $(x, z, \varepsilon = f_1(x, z) - f_1(x^*, z^*))$.

Observe that since $f(x, z) < f(\hat{x}, \hat{z})$ and $(\hat{x}, \hat{z}, \hat{\varepsilon})$ is feasible for (LEAP$(x^*, z^*)$) then $f_2(x, z) < f_2(\hat{x}, \hat{z}) \leq f_2(x^*, z^*)$. This implies that $f_2(x, z) < f_2(x^*, z^*)$. Furthermore, since $(x^*, z^*) \in \mathcal{E}_w$ then it must be that $f_1(x, z) \geq f_1(x^*, z^*)$, which implies that $\varepsilon \geq 0$. Next, suppose that $z = z^*$. Then by assumption $f(x, z^*) < f(\hat{x}, \hat{z})$ implies that $f_1(x, z^*) < f_1(\hat{x}, \hat{z})$ and since $\hat{\varepsilon} = 0$ then $f_1(x, z^*) < f_1(\hat{x}, \hat{z}) = f_1(x^*, z^*)$. And since $f_2(x, z^*) < f_2(x^*, z^*)$, we have that $f(x, z^*) < f(x^*, z^*)$, which contradicts the fact that $(x^*, z^*) \in \mathcal{E}_w$. Thus, it must be that $z \neq z^*$.

Therefore, $(x, z, \varepsilon)$ is feasible for (LEAP$(x^*, z^*)$). However, since $f_1(x, z) < f_1(\hat{x}, \hat{z})$ then $f_1(x, z) - f_1(x^*, z^*) < f_1(\hat{x}, \hat{z}) - f_1(x^*, z^*)$ and so $\varepsilon < \hat{\varepsilon} = 0$, which contradicts the optimality of $\hat{\varepsilon}$. □

Proposition 3.3 only applies in the case that $\hat{\varepsilon} = 0$. When $\hat{\varepsilon} > 0$, we propose a test problem which uses a strict ASF to determine if $(\hat{x}, \hat{z})$ is a weakly efficient solution of (BOMIP). Let $(\hat{x}, \hat{z}, \hat{\varepsilon})$ be an optimal solution for (LEAP$(x^*, z^*)$) and $\sigma : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ be an ASF. The test problem is the following.

$$\begin{aligned} \min_{x} \quad & \sigma(f(x, z^*), f(\hat{x}, \hat{z})) \\ \text{s.t.} \quad & x \in X(z^*) \end{aligned} \tag{ASFT$(z^*, \hat{x}, \hat{z})$}$$

Note that $f(\hat{x}, \hat{z}) = (f_1(\hat{x}, \hat{z}), f_2(\hat{x}, \hat{z}))$ is used as the reference point for $\sigma(\cdot, \cdot)$ and that optimization occurs over the slice set $X(z^*)$. In what follows, we present properties of $\sigma(\cdot, \cdot)$, which use the following assumptions.

**Assumption 3.4.**

1. $(x^*, z^*)$ be a weakly efficient solution for (BOMIP);
2. $(\hat{x}, \hat{z}, \hat{\varepsilon})$ be an optimal solution for (LEAP$(x^*, z^*)$);
3. $x_0^*$ be an optimal solution for (ASFT$(z^*, \hat{x}, \hat{z})$).

**Lemma 3.5.** Let $\sigma : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ be an order representing function and let Assumption 3.4 hold. If $\sigma(f(x_0^*, z^*), f(\hat{x}, \hat{z})) < 0$ then $f_1(x_0^*, z^*) \geq f_1(x^*, z^*)$ and $f_2(x_0^*, z^*) < f_2(x^*, z^*)$.

*Proof.* Suppose $\sigma(f(x_0^*, z^*), f(\hat{x}, \hat{z})) < 0$. Since $\sigma(\cdot, \cdot)$ is an order representing function, then $f(x_0^*, z^*) < f(\hat{x}, \hat{z})$. This implies that $f_2(x_0^*, z^*) < f_2(\hat{x}, \hat{z}) \leq f_2(x^*, z^*)$ and therefore $f_2(x_0^*, z^*) < f_2(x^*, z^*)$. However, since $(x^*, z^*) \in \mathcal{E}_w$, it must be that $f_1(x_0^*, z^*) \geq f_1(x^*, z^*)$. □

In the rest of this work, we make the following additional assumption.

**Assumption 3.6.** The function $\sigma : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ is a strict ASF.

In the next proposition we examine the relationship between the efficiency of an optimal solution for (LEAP$(x^*, z^*)$) and the optimal objective function value obtained in (ASFT$(z^*, \hat{x}, \hat{z})$).

**Proposition 3.7.** Let Assumptions 3.4 and 3.6 hold. Then the following two statements hold.

i. If $\sigma(f(x_0^*, z^*), f(\hat{x}, \hat{z})) \leq 0$ then $(x_0^*, z^*)$ is weakly efficient for (BOMIP).

ii. $(\hat{x}, \hat{z})$ is a weakly efficient solution for (BOMIP) if and only if

$$\sigma(f(x_0^*, z^*), f(\hat{x}, \hat{z})) \geq 0.$$

*Proof.* Let the above hold.

i. Towards a contradiction, suppose $(x_0^*, z^*) \notin \mathcal{E}_w$. Then there exists $(x, z) \in X$ such that $f(x, z) < f(x_0^*, z^*)$. We show that $(x, z, \varepsilon = f_1(x, z) - f_1(x^*, z^*))$ is feasible for (LEAP$(x^*, z^*)$). First, note that $\sigma(f(x_0^*, z^*), f(\hat{x}, \hat{z})) \leq 0$ implies that $f(x_0^*, z^*) \leqq f(\hat{x}, \hat{z})$. Thus, we have that $f_2(x, z) < f_2(x_0^*, z^*) \leq f_2(\hat{x}, \hat{z}) \leq f_2(x^*, z^*)$ and so $f_2(x, z) < f_2(x^*, z^*)$. Since $(x^*, z^*) \in \mathcal{E}_w$ then it must be that $f_1(x, z) \geq f_1(x^*, z^*)$ and so $\varepsilon \geq 0$. Next, suppose $z = z^*$. Then since $\sigma(\cdot, \cdot)$ is strictly order preserving, $f(x, z^*) < f(x_0^*, z^*)$ implies that $\sigma(f(x, z^*), f(\hat{x}, \hat{z})) < \sigma(f(x_0^*, z^*), f(\hat{x}, \hat{z}))$, which contradicts the optimality of $x_0^*$ for (ASFT$(z^*, \hat{x}, \hat{z})$). Thus, it must be that $z \neq z^*$. Therefore, $(x, z, \varepsilon)$ is feasible for (LEAP$(x^*, z^*)$). However, since $f(x, z) < f(x_0^*, z^*) \leqq f(\hat{x}, \hat{z})$ we have that $f_1(x, z) < f_1(\hat{x}, \hat{z})$. But this implies that $\varepsilon < \hat{\varepsilon}$, contradicting the optimality of $(\hat{x}, \hat{z}, \hat{\varepsilon})$ for (LEAP$(x^*, z^*)$). Therefore, $(x_0^*, z^*) \in \mathcal{E}_w$.

ii. Let $\sigma : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ be a strict ASF. Let $(x^*, z^*) \in \mathcal{E}_w$, $(\hat{x}, \hat{z}, \hat{\varepsilon})$ an optimal solution of (LEAP$(x^*, z^*)$), and $x_0^*$ an optimal solution of (ASFT$(z^*, \hat{x}, \hat{z})$). First, we show that if $(\hat{x}, \hat{z}) \in \mathcal{E}_w$ then $\sigma(f(x_0^*, z^*), f(\hat{x}, \hat{z})) \geq 0$ by the contrapostive. To that end, let $\sigma(f(x_0^*, z^*), f(\hat{x}, \hat{z})) < 0$. Then since $\sigma(\cdot, \cdot)$ is a strict ASF, it is order representing, which implies that $f(x_0^*, z^*) < f(\hat{x}, \hat{z})$. Thus, $(\hat{x}, \hat{z}) \notin \mathcal{E}_w$.

Conversely, assume $\sigma(f(x_0^*, z^*), (\hat{x}, \hat{z})) \geq 0$. Towards a contradiction, suppose $(\hat{x}, \hat{z}) \notin \mathcal{E}_w$. Then there exists $(x, z) \in X$ such that $f(x, z) < f(\hat{x}, \hat{z})$. We show that $(x, z, \varepsilon = f_1(x, z) - f_1(x^*, z^*))$ is feasible for (LEAP$(x^*, z^*)$). To that end, observe that $f_2(x, z) < f_2(\hat{x}, \hat{z}) \leq f_2(x^*, z^*)$. Furthermore, since $(x^*, z^*) \in \mathcal{E}_w$ it must be that $f_1(x, z) \geq f_1(x^*, z^*)$ and so $\varepsilon \geq 0$. Now suppose $z = z^*$. Then since by assumption $f(x, z^*) < f(\hat{x}, \hat{z})$ and $\sigma(\cdot, \cdot)$ is a strict ASF, we have that $\sigma(f(x, z^*), f(\hat{x}, \hat{z})) < \sigma(f(\hat{x}, \hat{z}), f(\hat{x}, \hat{z})) = 0$. But this contradicts the fact that $x_0^*$ is an optimal solution of (ASFT$(z^*, \hat{x}, \hat{z})$) with $\sigma(f(x_0^*, z^*), f(\hat{x}, \hat{z})) \geq 0$. Thus, $z \neq z^*$. So $(x, z, \varepsilon)$ is feasible for (LEAP$(x^*, z^*)$). However, by assumption, $f_1(x, z) < f_1(\hat{x}, \hat{z})$ implies that $\varepsilon < \hat{\varepsilon}$, contradicting the optimality of $(\hat{x}, \hat{z}, \hat{\varepsilon})$ for (LEAP$(x^*, z^*)$). Thus, $(\hat{x}, \hat{z}) \in \mathcal{E}_w$.

□

Figure 1b shows the result of the ASF test in the outcome space of a BOMIP. The dotted arrows show the level curve $\{y \in \mathbb{R}^2 \mid \sigma(y, f(\hat{x}, \hat{z})) = 0\}$. Furthermore, we see that Proposition 3.7 shows that the point $f(\hat{x}, \hat{z})$ is dominated by $f(x_0^*, z^*)$. Note, however, that Figure 1b also shows that even if $f(\hat{x}, \hat{z})$ is not identified as being a (weak) Pareto point, this does not imply that the entire *slice* $Y(\hat{z})$ is not a (weak) Pareto slice. This is especially important in the case of intersecting slices.

We now consider the question of determining when two weak Pareto slices intersect. To do so, we use sequences of feasible points

$$\{(x_n^*, z^*)\}_{n=1}^{\infty} \subseteq X \text{ and}$$
$$\{(\hat{x}_n, \hat{z}, \hat{\varepsilon}_n)\}_{n=1}^{\infty} \subseteq X \times \mathbb{R}_{\geq}$$

which satisfy the following properties.

**Assumption 3.8.** Let

$$\{(x_n^*, z^*)\}_{n=1}^{\infty} \subseteq X \text{ and}$$
$$\{(\hat{x}_n, \hat{z}, \hat{\varepsilon}_n)\}_{n=1}^{\infty} \subseteq X \times \mathbb{R}_{\geq}$$

be sequences satisfying the following.

1. $(x_n^*, z^*)$ is a weakly efficient solution for (BOMIP).
2. $(\hat{x}_n, \hat{z}, \hat{\varepsilon}_n)$ is an optimal solution of (LEAP$(x^*, z^*)$).
3. $x_{n+1}^*$ is an optimal solution of (ASFT$(z^*, \hat{x}, \hat{z})$) with

$$\sigma(f(x_{n+1}^*, z^*), f(\hat{x}_n, \hat{z})) < 0.$$

**Lemma 3.9.** If there exist sequences

$$\{(x_n^*, z^*)\}_{n=1}^{\infty} \subseteq X \text{ and}$$
$$\{(\hat{x}_n, \hat{z}, \hat{\varepsilon}_n)\}_{n=1}^{\infty} \subseteq X \times \mathbb{R}_{\geq}$$

satisfying Assumption 3.8 then for all $n$,

$$f_1(x_n^*, z^*) \leq f_1(x_{n+1}^*, z^*) < f_1(\hat{x}_n, \hat{z}) \leq f_1(\hat{x}_{n+1}, \hat{z})$$
$$f_2(\hat{x}_{n+1}, \hat{z}) \leq f_2(x_{n+1}^*, z^*) < f_2(\hat{x}_n, \hat{z}) \leq f_2(x_n^*, z^*).$$

*Proof.* Let $n \geq 1$. Since $\sigma(f(x_{n+1}^*, z^*), f(\hat{x}_n, \hat{z})) < 0$ then $f(x_{n+1}^*, z^*) < f(\hat{x}_n, \hat{z})$. Furthermore, since $(\hat{x}_n, \hat{z})$ is feasible for (LEAP$(x_n^*, z^*)$), $f_2(\hat{x}_n, \hat{z}) \leq f_2(x_n^*, z^*)$. Finally, since $(\hat{x}_{n+1}, \hat{z})$ is feasible for (LEAP$(x_{n+1}^*, z^*)$) then $f_2(\hat{x}_{n+1}, \hat{z}) \leq f_2(x_{n+1}^*, z^*$. Putting these inequalities together, we have

$$f_2(\hat{x}_{n+1}, \hat{z}) \leq f_2(x_{n+1}^*, z^*) < f_2(\hat{x}_n, \hat{z}) \leq f_2(x_n^*, z^*).$$

However, since $(x_n^*, z^*) \in \mathcal{E}_w$ it must be that $f_1(x_n^*, z^*) \leq f_1(x_{n+1}^*, z^*)$. Furthermore, since $(\hat{x}_{n+1}, \hat{z}) \in \mathcal{E}_w(\hat{z})$, $f_1(\hat{x}_n, \hat{z}) \leq f_1(\hat{x}_{n+1}, \hat{z})$. Therefore,

$$f_1(x_n^*, z^*) \leq f_1(x_{n+1}^*, z^*) < f_1(\hat{x}_n, \hat{z}) \leq f_1(\hat{x}_{n+1}, \hat{z})$$

□

Using such sequences as defined in Assumption 3.8, we now provide necessary and sufficient conditions for two weak Pareto slices of (BOMIP) to intersect. In the following proposition, $||\cdot||$ denotes the $\ell_2$-norm.

**Proposition 3.10.** There exist sequences

$$\{(x_n^*, z^*)\}_{n=1}^{\infty} \subseteq X \text{ and}$$
$$\{(\hat{x}_n, \hat{z}, \hat{\varepsilon}_n)\}_{n=1}^{\infty} \subseteq X \times \mathbb{R}_{\geq}$$

satisfying Assumption 3.8 such that

$$\lim_{n\to\infty} \hat{\varepsilon}_n = 0 \text{ and}$$
$$\lim_{n\to\infty} \sigma(f(x_{n+1}^*, z^*), f(\hat{x}_n, \hat{z})) = 0$$

if and only if there exists $y^*, \hat{y} \in \mathbb{R}^2$ such that

i. $y^* = \lim_{n\to\infty} f(x_n^*, z^*)$ and $\hat{y} = \lim_{n\to\infty} f(\hat{x}_n, \hat{z})$

ii. $y^* = \hat{y}$

iii. $y^* = \hat{y} \in \mathcal{P}_w$.

*Proof.* Let the conditions of the proposition hold. We first prove the necessary condition. To that end, suppose $\lim_{n\to\infty} \hat{\varepsilon}_n = 0$ and $\lim_{n\to\infty} \sigma(f(x^*_{n+1}, z^*), f(\hat{x}_n, \hat{z})) = 0$.

We first show that there exists $y^*, \hat{y} \in \mathbb{R}^2$ such that $\lim_{n\to\infty} f(x^*_n, z^*) = y^*$ and $\lim_{n\to\infty} f(\hat{x}_n, \hat{z}) = \hat{y}$. Observe that since $Y(z^*)$ and $Y(\hat{z})$ are compact sets, they are bounded. In particular, let $y^*_I = (y^*_{I,1}, y^*_{I,2}),\ \hat{y}_I = (\hat{y}_{I,1}, \hat{y}_{I,2}) \in \mathbb{R}^2$ be the ideal points of $Y(z^*)$ and $Y(\hat{z})$, respectively. Similarly, let $y^*_N = (y^*_{N,1}, y^*_{N,2}),\ \hat{y}_N = (\hat{y}_{N,1}, \hat{y}_{N,2}) \in \mathbb{R}^2$ be the nadir points of $Y(z^*)$ and $Y(\hat{z})$ respectively. By Lemma 3.9, we have that for all $n$,

$$\begin{aligned}
&y^*_{I,1} \le f_1(x^*_n, z^*) \le f_1(x^*_{n+1}, z^*) \le y^*_{N,1} \\
&y^*_{I,2} \le f_2(x^*_{n+1}, z^*) < f_2(x^*_n, z^*) \le y^*_{N,2} \\
&\hat{y}_{I,1} \le f_1(\hat{x}_n, \hat{z}) \le f_1(\hat{x}_{n+1}, \hat{z}) \le \hat{y}_{N,1} \\
&\hat{y}_{I,2} \le f_2(\hat{x}_{n+1}, \hat{z}) < f_2(\hat{x}_n, \hat{z}) \le \hat{y}_{N,2}
\end{aligned}$$

Thus, the sequences $\{f_1(x^*_n, z^*)\}_{n=1}^\infty, \{f_2(x^*_n, z^*)\}_{n=1}^\infty, \{f_1(\hat{x}_n, \hat{z})\}_{n=1}^\infty, \{f_2(\hat{x}_n, \hat{z})\}_{n=1}^\infty$ are each monotonic and bounded, so they each converge. Let $y^* = (y^*_1, y^*_2),\ \hat{y} = (\hat{y}_1, \hat{y}_2) \in \mathbb{R}^2$ be such that for $i = 1, 2$

$$\begin{aligned}
\lim_{n\to\infty} f_i(x^*_n, z^*) &= y^*_i \\
\lim_{n\to\infty} f_i(\hat{x}_n, \hat{z}) &= \hat{y}_i.
\end{aligned}$$

Thus,

$$\begin{aligned}
\lim_{n\to\infty} f(x^*_n, z^*) &= \lim_{n\to\infty} \begin{bmatrix} f_1(x^*_n, z^*) \\ f_2(x^*_n, z^*) \end{bmatrix} = \begin{bmatrix} y^*_1 \\ y^*_2 \end{bmatrix} = y^* \\
\lim_{n\to\infty} f(\hat{x}_n, \hat{z}) &= \lim_{n\to\infty} \begin{bmatrix} f_1(\hat{x}_n, \hat{z}) \\ f_2(\hat{x}_n, \hat{z}) \end{bmatrix} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} = \hat{y}.
\end{aligned}$$

However, by Lemma 3 in [27] $\mathcal{P}_w(z^*)$ and $\mathcal{P}_w(\hat{z})$ are closed in $\mathbb{R}^2$. This means that they must contain their limit points and so $y^* \in \mathcal{P}_w(z^*)$ and $\hat{y} \in \mathcal{P}_w(\hat{z})$.

We proceed to show that $y^* = \hat{y}$. First, by assumption, we see that

$$\begin{aligned}
\lim_{n\to\infty} \hat{\varepsilon}_n &= 0 \\
\lim_{n\to\infty} f_1(\hat{x}_n, \hat{z}) - f_1(x^*_n, z^*) &= 0 \\
\lim_{n\to\infty} f_1(\hat{x}_n, \hat{z}) &= \lim_{n\to\infty} f_1(x^*_n, z^*) \\
\hat{y}_1 &= y^*_1.
\end{aligned}$$

Next, by Lemma 3.9, for all $n$,

$$f_2(\hat{x}_{n+1}, \hat{z}) \le f_2(x^*_{n+1}, z^*) < f_2(\hat{x}_n, \hat{z}) \le f_2(x^*_n, z^*).$$

This implies that

$$\begin{aligned}
\lim_{n\to\infty} f_2(\hat{x}_{n+1}, \hat{z}) \le \lim_{n\to\infty} f_2(x^*_{n+1}, z^*) \le \lim_{n\to\infty} f_2(\hat{x}_n, \hat{z}) \le \lim_{n\to\infty} f_2(x^*_n, z^*) \\
\hat{y}_2 \le y^*_2 \le \hat{y}_2 \le y^*_2.
\end{aligned}$$

This means that $y^*_2 = \hat{y}_2$ and so $y^* = \hat{y}$.

It remains to show that $y^* = \hat{y} \in \mathcal{P}_w$. Let $(x^*, z^*) \in \mathcal{E}_w(z^*)$ be such that $y^* = f(x^*, z^*)$ and let $(\hat{x}, \hat{z}) \in \mathcal{E}_w(\hat{z})$ such that $\hat{y} = f(\hat{x}, \hat{z})$. Towards a contradiction, suppose $f(x^*, z^*) = f(\hat{x}, \hat{z}) \notin \mathcal{P}_w$. Then there exists $(x, z) \in X$ such that $f(x, z) < f(x^*, z^*) = f(\hat{x}, \hat{z})$. Since $f_1$ is continuous in the real variable, for $\delta = f_1(x^*, z^*) - f_1(x, z)$ there exists $N$ be such that for all $n > N$, $|f_1(x^*_n, z^*) - f_1(x^*, z^*)| < \delta$.

Furthermore, since $\{f_1(x_n^*, z^*)\}_{n=1}^{\infty}\}$ is an increasing sequence by Lemma 3.9 then $|f_1(x_n^*, z^*) - f_1(x^*, z^*)| = f_1(x^*, z^*) - f_1(x_n^*, z^*)$. Therefore, for all $n > N$,

$$\begin{aligned} |f_1(x_n^*, z^*) - f_1(x^*, z^*)| < \delta = f_1(x^*, z^*) - f_1(x, z) \\ f_1(x^*, z^*) - f_1(x_n^*, z^*) < f_1(x^*, z^*) - f_1(x, z) \\ f_1(x, z) < f_1(x_n^*, z^*). \end{aligned}$$

Furthermore, by assumption and Lemma 3.9 we have that $f_2(x, z) < f_2(x^*, z^*) \leq f_2(x_n^*, z^*)$. But this implies that $f(x, z) < f(x_n^*, z^*)$, which contradicts the fact that $(x_n^*, z^*) \in \mathcal{E}_w$. Therefore, it must be that $y^* = \hat{y} \in \mathcal{P}_w$.

Conversely, suppose $y^* = \lim_{n\to\infty} f(x_n^*, z^*) = \lim_{n\to\infty} f(\hat{x}_n, \hat{z}) = \hat{y}$ such that $y^* = \hat{y} \in \mathcal{P}_w$. We denote $y^* = \hat{y}$ by $\bar{y}$. Let $\delta > 0$ and let $N$ be such that for all $n > N$, $||f(x_n^*, z^*) - \bar{y}|| < \delta/2$ and $||f(\hat{x}_n, \hat{z}) - \bar{y}|| < \delta/2$. Observe that

$$\begin{aligned} |\hat{\varepsilon}_n| &= |f_1(\hat{x}_n, \hat{z}) - f_1(x_n^*, z^*)| \\ &= |f_1(\hat{x}_n, \hat{z}) - f_1(x_n^*, z^*) + \bar{y}_1 - \bar{y}_1| \\ &\leq |f_1(\hat{x}_n, \hat{z}) - \bar{y}_1| + |f_1(x_n^*, z^*) - \bar{y}_1| \\ &\leq ||f(\hat{x}_n, \hat{z}) - \bar{y}|| + ||f(x_n^*, z^*) - \bar{y}|| \\ &< \delta/2 + \delta/2 = \delta. \end{aligned}$$

Thus, $\lim_{n\to\infty} \hat{\varepsilon}_n = 0$. Finally, for every $\delta > 0$ there exists $N$ such that for all $n > N$, $\left|\left|\begin{bmatrix} f(x_{n+1}^*, z^*) \\ f(\hat{x}_n, \hat{z}) \end{bmatrix} - \begin{bmatrix} \bar{y} \\ \bar{y} \end{bmatrix}\right|\right| < \delta$. Since $\sigma(\cdot, \cdot)$ is continuous, then for any $\alpha > 0$, there exists a $\delta > 0$ and $N$ such that for all $n > N$, $\left|\left|\begin{bmatrix} f(x_{n+1}^*, z^*) \\ f(\hat{x}_n, \hat{z}) \end{bmatrix} - \begin{bmatrix} \bar{y} \\ \bar{y} \end{bmatrix}\right|\right| < \delta$ implies that $|\sigma(f(x_{n+1}^*, z^*), f(\hat{x}_n, \hat{z})) - \sigma(\bar{y}, \bar{y})| = |\sigma(f(x_{n+1}^*, z^*), f(\hat{x}_n, \hat{z}))| < \alpha$. Thus, $\lim_{n\to\infty} \sigma(f(x_{n+1}^*, z^*), f(\hat{x}_n, \hat{z})) = 0$. □

These theoretical results, which are summarized in Figure 2, allow the Pareto Leap Algorithm to work on BOMIPs where the underlying continuous problem is nonlinear. Note that (LEAP$(x^*, z^*)$) and (ASFT$(z^*, \hat{x}, \hat{z})$) provide pointwise information about the objective space of (BOMIP). We start with a weakly efficient solution $(x^*, z^*)$ for (BOMIP), whose image is used as the reference point in (LEAP$(x^*, z^*)$). Letting $(\hat{x}, \hat{z}, \hat{\varepsilon})$ be an optimal solution of (LEAP$(x^*, z^*)$), we have two cases. If $\hat{\varepsilon} = 0$, Proposition 3.3 guarantees that $(\hat{x}, \hat{z})$ is a weakly efficient solution for (BOMIP). On the other hand, if $\hat{\varepsilon} > 0$, we use the image of $(\hat{x}, \hat{z})$ as the reference point for a strict ASF in (ASFT$(z^*, \hat{x}, \hat{z})$). Letting $x_0^*$ be an optimal solution for (ASFT$(z^*, \hat{x}, \hat{z})$), we again have two cases. If $\sigma(f(x_0^*, z^*), f(\hat{x}, \hat{z})) < 0$, then since $\sigma(\cdot, \cdot)$ is a strict ASF, $f(x_0, z^*) < f(\hat{x}, \hat{z})$. Furthermore, Proposition 3.7 i. guarantees that $(x_0^*, z^*)$ is a weakly efficient solution for (BOMIP). On the other hand, if $\sigma(f(x_0^*, z^*), f(\hat{x}, \hat{z})) \geq 0$, then Proposition 3.7 shows that $(\hat{x}, \hat{z})$ is a weakly efficient solution for (BOMIP).

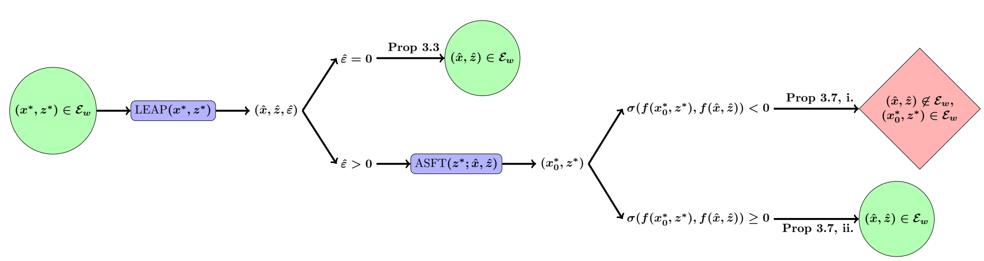



Figure 2: Summary of theoretical results from Section 3.

However, in order for the Pareto Leap Algorithm to find the weak Pareto set of (BOMIP), there must be a way to not only determine when a weak Pareto point is found but also when two weak Pareto slices intersect. To determine the intersection of two slices, the sequences described in Assumption 3.8 are needed. They arise from iteratively solving (LEAP$(x^*, z^*)$) and (ASFT$(z^*, \hat{x}, \hat{z})$), where the optimal solutions alternate between two slices, namely $\mathcal{P}_w(z^*)$ and $\mathcal{P}_w(\hat{z})$.When such sequences exist, Lemma 3.9 shows that in the objective space, the first component of the objective function yields an increasing sequence, while the second component yields a decreasing sequence. This result is important because it leads to the necessary and sufficient conditions in Proposition 3.10 for two (weak) Pareto slices to intersect. These conditions guarantee that the step size $\hat{\varepsilon}_n$ in the leap problem and the ASFT optimal objective value converge to zero if and only if the slices intersect, i.e., $\mathcal{P}_w(z^*) \cap \mathcal{P}_w(\hat{z}) \neq \emptyset$. While these conditions are theoretically valuable, they can also be easily integrated with the algorithm to detect the intersection of two slices.

As we iteratively solve both (LEAP$(x^*, z^*)$) and (ASFT$(z^*, \hat{x}, \hat{z})$), the results proven here guarantee that we will always find a new weakly efficient solution for (BOMIP), whether it is additionally $z^*$-weakly efficient or $\hat{z}$-weakly efficient. In the next section, we apply both (LEAP$(x^*, z^*)$) and (ASFT$(z^*, \hat{x}, \hat{z})$) in an algorithm for finding the weak Pareto slices of (BOMIP).

# 4 Pareto Leap Algorithm

The results of Section 3 provide the basis for the Pareto Leap algorithm to identify all slices which contribute to the global weak Pareto set. Up to this point, we have assumed that $f_1$ and $f_2$ are continuous in their real variables and that each slice set is nonempty and compact. However, for the algorithm, we make the following additional assumption about (BOMIP).

**Assumption 4.1.** There exists $\ell, u \in \mathbb{Z}^{n_I}$ such that $\ell \leqq z \leqq u$.

Observe that in Step 4, we initialize the algorithm by minimizing $f_1$ over the feasible set $X$ and obtain an optimal solution $(x^*, z^*)$. Since $f_1$ is minimized, we have that $Y(z^*)$ is a (weak) Pareto slice and thus we store the value of $z^*$. After this initialization step, we proceed to the main while loop.

Within the while loop, in Step 7, we attempt to find an optimal solution $(\hat{x}, \hat{z}, \hat{\varepsilon})$ for (LEAP$(x^*, z^*)$). If (LEAP$(x^*, z^*)$) is infeasible, then the algorithm terminates; all weak Pareto slices have been found. Otherwise, we then proceed to determine if $(\hat{x}, \hat{z})$ is weakly efficient for (BOMIP). Step 10 checks if Proposition 3.3 holds; if so, then $Y(\hat{z})$ is a weak Pareto slice, and we set $(x^*, z^*)$ to be $(\hat{x}, \hat{z})$. After this, the algorithm returns to Step 7.

If $\hat{\varepsilon} > 0$, then the algorithm proceeds to step 15 and performs the ASF test by solving (ASFT$(z^*, \hat{x}, \hat{z})$). Since $\sigma(\cdot, \cdot)$ is a strict ASF, Proposition 3.7 applies. Thus, in Step 16, if $\sigma(f(x', z^*), f(\hat{x}, \hat{z})) \geq 0$, then $(\hat{x}, \hat{z})$ is a weakly efficient solution for (BOMIP) and $Y(\hat{z})$ is a weak Pareto slice, so the algorithm changes its reference point to $(\hat{x}, \hat{z})$ and returns to Step 7. Otherwise, in Step 20, if $\sigma(f(x', z^*), f(\hat{x}, \hat{z})) < 0$ but $|\sigma(f(x', z^*), f(\hat{x}, \hat{z}))| > \texttt{tol}$, where `tol` denotes the tolerance set by the user, then the reference point is set to $(x', z^*)$ and the algorithm returns to Step 7. On the other hand, in Step 23, if $\hat{\varepsilon} < \texttt{tol}$ and $|\sigma(f(x', z^*), f(\hat{x}, \hat{z}))| < \texttt{tol}$ then Proposition 3.10 is applied and $Y(\hat{z})$ is a weak Pareto slice. The reference point is set to $(\hat{x}, \hat{z})$ and the algorithm returns to Step 7. Steps 23-26 are possible because Proposition 3.10 shows that if during the course of the Pareto Leap algorithm we find a sequence of step sizes $\hat{\varepsilon}_n$ which converge to zero and the ASF test values also converge to zero, then we must have two intersecting slices. Thus, for computational reasons, the tolerance level is used so that when $\varepsilon$ and $\sigma$ are less than the tolerance, we proceed to the next slice, recognizing that $\mathcal{P}_w(z^*)$ and $\mathcal{P}_w(\hat{z})$ have a non-empty intersection. Furthermore, in the actual implementation of Algorithm 1, an additional step to determine if $(\hat{x}, \hat{z})$ has already been found can help increase the speed of the algorithm. If $Y(\hat{z})$ has already been determined to be a weak Pareto slice, then $\hat{z}$ can be added as a tabu constraint to (LEAP$(x^*, z^*)$).

In the next section, we apply Algorithm 1 to a mathematical example.

**Algorithm 1** Pareto Leap Algorithm.

1: **input:** $f_1, f_2, X$,`tol`. {Input functions $f_1, f_2$, feasible set $X$, and a tolerance `tol`.}
2: Initialize $S = \emptyset$. {$S$ will store the integer values of the Pareto slices.}
3: Initialize flag $= 0$. {This will determine when to end the "while" loop.}
4: Minimize $f_1$ with optimal solution $(x^*, z^*)$.
5: Add $z^*$ to $S$.
6: **while** flag $= 0$ **do**
7: Solve (LEAP$(x^*, z^*)$) with optimal solution $(\hat{x}, \hat{z}, \hat{\varepsilon})$.
8: **if** (LEAP$(x^*, z^*)$) is infeasible **then**
9: flag $= 1$ {Nowhere else to leap to; all Pareto slices have been found.}
10: **else if** $\hat{\varepsilon} = 0$ **then**
11: Add $\hat{z}$ to $S$.
12: **if** $f(x^*, z^*) \neq f(\hat{x}, \hat{z})$ **then**
13: Set $(x^*, z^*) = (\hat{x}, \hat{z})$.
14: **else if** $f(x^*, z^*) = f(\hat{x}, \hat{z})$ **then**
15: Add $z \neq \hat{z}$ as a tabu constraint.
16: Go to Step 7.
17: **end if**
18: Return to Step 7.
19: **else if** $\hat{\varepsilon} > 0$ **then**
20: Solve (ASFT$(z^*, \hat{x}, \hat{z})$) with optimal solution $x'$.
21: **if** $\sigma(f(x', z^*), f(\hat{x}, \hat{z})) \geq 0$ **then**
22: Add $\hat{z}$ to $S$.
23: Set $(x^*, z^*) = (\hat{x}, \hat{z})$.
24: Go to Step 7.
25: **else if** $\sigma(f(x', z^*), f(\hat{x}, \hat{z})) < 0$ and $|\sigma(f(x', z^*), f(\hat{x}, \hat{z}))| >$ `tol` **then**
26: Set $(x^*, z^*) = (x', z^*)$.
27: Go to Step 7.
28: **else if** $|\hat{\varepsilon}| <$ `tol` and $|\sigma(f(x', z^*), f(\hat{x}, \hat{z}))| <$ `tol` **then**
29: Add $\hat{z}$ to $S$.
30: Set $(x^*, z^*) = (\hat{x}, \hat{z})$
31: Go to Step 7.
32: **end if**
33: **end if**
34: **end while**
35: **return** $S$, the collection of Pareto Slices.

## 4.1 Example

We consider the following biobjective mixed-integer program.

$$\begin{aligned} \min_{x,z} \quad & \begin{bmatrix} f_1(x,z) = \begin{bmatrix} x & z \end{bmatrix} \begin{bmatrix} 0.0586 & -0.1461 \\ -0.1461 & 0.7321 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} 0.3923 & 0.1543 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \\ f_2(x,z) = \begin{bmatrix} x & z \end{bmatrix} \begin{bmatrix} 0.2930 & 0.0395 \\ 0.0395 & 0.0221 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} -0.7347 & 0.0961 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \end{bmatrix} \\ \text{s.t.} \quad & x \in [-5, 5] \\ & z \in [-2, 1] \cap \mathbb{Z} \end{aligned} \tag{EX}$$

Observe that for fixed $z \in \mathbb{Z}$, the objective functions $f_1, f_2$ are strictly convex in $x$ and thus for each $z \in [-2,1] \cap \mathbb{Z}$, $\mathcal{P}_w(z) = \mathcal{P}(z)$. Figure 3a shows the Pareto sets of each slice. In this example, we use the well known strict ASF $\sigma(y,r) = \max_{i=1,2}\{y_i - r_i\}$ [19]. We work through the steps of the Pareto Leap Algorithm, following the rows outlined in Tables 1 and 2. Figure 3 visually represents the steps taken. In the discussion which follows, let $n$ denote the step of the algorithm.

| $n$ | $x_n^*$ | $z_n^*$, Tabu values | $\hat{x}_n$ | $\hat{z}_n$ | $\hat{\varepsilon}$ | $x_{n+1}^*$ | $z_{n+1}^*$ |
|---|---|---|---|---|---|---|---|
| 0 | −5 | −1 | – | – | – | – | – |
| 1 | −5 | −1 | −4.9038 | −2 | 0.6191 | −4.8372 | −1 |
| 2 | −4.8372 | −1 | −4.7420 | −2 | 0.6681 | −4.6578 | −1 |
| 3 | −4.6578 | −1 | −3.3473 | 0 | 0.6826 | −3.1518 | −1 |
| 4 | −3.1518 | −1 | −3.2190 | 0 | 0.3419 | −3.0358 | −1 |
| 5 | −3.0358 | −1 | −3.1013 | 0 | 0.3071 | −2.9298 | −1 |
| 6 | −2.9298 | −1 | −2.9935 | 0 | 0.2754 | −2.8331 | −1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| **14** | **−2.3710** | **−1** | **−2.4240** | **0** | **0.1091** | **−2.3290** | **−1** |
| 15 | −2.4240 | 0 | −2.1123 | −1 | 0 | – | – |
| 16 | −2.4240 | 0, −1 | −2.3014 | −2 | 1.2890 | −1.8162 | 0 |
| 17 | −1.8162 | 0, −1 | −0.8541 | 1 | 1.3629 | −0.0425 | 0 |
| 18 | −0.0425 | 0, −1 | 0.1407 | 1 | 0.9182 | 0.9659 | 0 |
| 19 | 0.9659 | 0, −1 | 0.4344 | −2 | 2.6215 | 1.2535 | 0 |
| 20 | 0.4344 | −2 | – | – | – | – | – |

Table 1: Optimal solutions for each step of Pareto Leap algorithm applied to (EX).

$n = 0$: Initialize the algorithm by minimizing $f_1(x,z)$ over the feasible set. (Figure 3b.)

$n = 1$: Formulate and solve (LEAP($x_0^*, z_0^*$)), which gives an optimal solution denoted by $(\hat{x}_1, \hat{z}_1, \hat{\varepsilon}_1)$. Observe that the algorithm lands in $\mathcal{P}(-2)$. Then (ASFT($z_0^*, \hat{x}_1, \hat{z}_1$)) is formulated and solved. The optimal value is negative.

$n = 2$: The algorithm leaps again by formulating and solving (LEAP($x_2^*, z_2^*$)), finding $(\hat{x}_2, \hat{z}_2, \hat{\varepsilon}_2)$. Once more, $\hat{z} = -2$. (ASFT($z_2^*, \hat{x}_2, \hat{z}_2$)) is formulated and solved. Again, $\sigma(f(x_3^*, z_3^*), f(\hat{x}_2, \hat{z}_2)) < 0$.

$n = 3$: Leap again using $f(x_3^*, z_3^*)$ as the reference point. However, a new slice is found, defined by $\hat{z}_3 = 0$. But $\sigma(f(x_4^*, z_4^*), f(\hat{x}_3, \hat{z}_3)) < 0$.

$n = 4 \ldots 13$: The algorithm continues leaping back and forth between $\mathcal{P}(-1)$ and $\mathcal{P}(0)$. Observe that the optimal values of the ASF test problem are decreasing: $\sigma(f(x_{n+1}^*, z_{n+1}^*), f(\hat{x}_n, \hat{z}_n)) > \sigma(f(x_n^*, z_n^*), f(\hat{x}_{n-1}, \hat{z}_{n-1}))$ for each $n$.

$n = 14$: The algorithm leaps to $\mathcal{P}(0)$, but $|\sigma(f(x_{15}^*, z_{15}^*), f(\hat{x}_{14}, \hat{z}_{14}))| \leq 0.1$, the tolerance. By Proposition 3.10, $\mathcal{P}(-1) \cap \mathcal{P}(0)$ intersect, so $\mathcal{P}(-1) \cap \mathcal{P}(0) \neq \emptyset$. The new reference point is set to $(x_{15}^*, z_{15}^*) = (\hat{x}_{14}, \hat{z}_{14})$.

| $n$ | $f(x^*_n, z^*_n)$ | $f(\hat{x}_n, \hat{z}_n)$ | $f(x^*_{n+1}, z^*_{n+1})$ | $\sigma^*_n$ |
|---|---|---|---|---|
| 0 | $(-1.3797, 11.3195)$ | – | – | – |
| 1 | $(-1.3797, 11.3195)$ | $(-0.7606, 11.3195)$ | $(-1.3621, 10.7179)$ | $-0.6016$ |
| 2 | $(-1.3621, 10.7179)$ | $(-0.6940, 10.7179)$ | $(-1.3391, 10.0728)$ | $-0.6451$ |
| 3 | $(-1.3391, 10.0728)$ | $(-0.6566, 5.7421)$ | $(-0.9975, 5.4012)$ | $-0.3409$ |
| 4 | $(-0.9975, 5.4012)$ | $(-0.6556, 5.4012)$ | $(-0.9602, 5.0966)$ | $-0.3045$ |
| 5 | $(-0.9602, 5.0966)$ | $(-0.6530, 5.0966)$ | $(-0.9246, 4.8250)$ | $-0.2716$ |
| 6 | $(-0.9246, 4.8250)$ | $(-0.6492, 4.8250)$ | $(-0.8911, 4.5831)$ | $-0.2419$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| **14** | $(\mathbf{-0.7157, 3.5025})$ | $(\mathbf{-0.6066, 3.5025})$ | $(\mathbf{-0.6986, 3.4105})$ | $\mathbf{-0.0919}$ |
| 15 | $(-0.6066, 3.5025)$ | $(-0.6066, 2.9521)$ | – | – |
| 16 | $(-0.6986, 3.4105)$ | $(0.6824, 3.5025)$ | $(-0.5192, 2.3009)$ | $-1.2028$ |
| 17 | $(-0.5192, 2.3009)$ | $(0.8437, 0.8920)$ | $(-0.0166, 0.0318)$ | $-0.8611$ |
| 18 | $(-0.0166, 0.0318)$ | $(0.9016, 0.0318)$ | $(0.4336, -0.4363)$ | $-0.4685$ |
| 19 | $(0.4336, -0.4363)$ | $(3.0551, -0.4363)$ | $(0.5838, -0.4606)$ | $-0.0255$ |
| 20 | $(3.0551, -0.4363)$ | – | – | – |

Table 2: Optimal values of each step of Pareto Leap algorithm applied to (EX).

$n = 15$: The algorithm leaps from $\mathcal{P}(0)$ and finds that $\hat{\varepsilon}_{15} = 0$ and $\hat{z}_{15} = -1$. Thus, $(\hat{x}_{15}, \hat{z}_{15})$ is a weakly efficient solution for (EX). But it is known that $Y(-1)$ is a weakly Pareto slice. $\hat{z}_{15} = -1$ is added as a tabu constraint.

$n = 16$: The algorithm leaps from $f(x^*_{15}, z^*_{15})$ but with the addition of a new tabu constraint, where $\hat{z}_{16} \neq 0$ and $\hat{z}_{16} \neq -1$. Here, the algorithm finds that $\hat{z}_{16} = -2$ but $\sigma(f(x^*_{17}, z^*_{17}), f(\hat{x}_{16}, \hat{z}_{16})) < 0$.

$n = 17 \ldots 19$: The algorithm leaps from $\mathcal{P}(0)$, but now finds $\mathcal{P}(1)$. However, the results from the ASF test problem show that $\mathcal{P}(1)$ is a dominated slice.

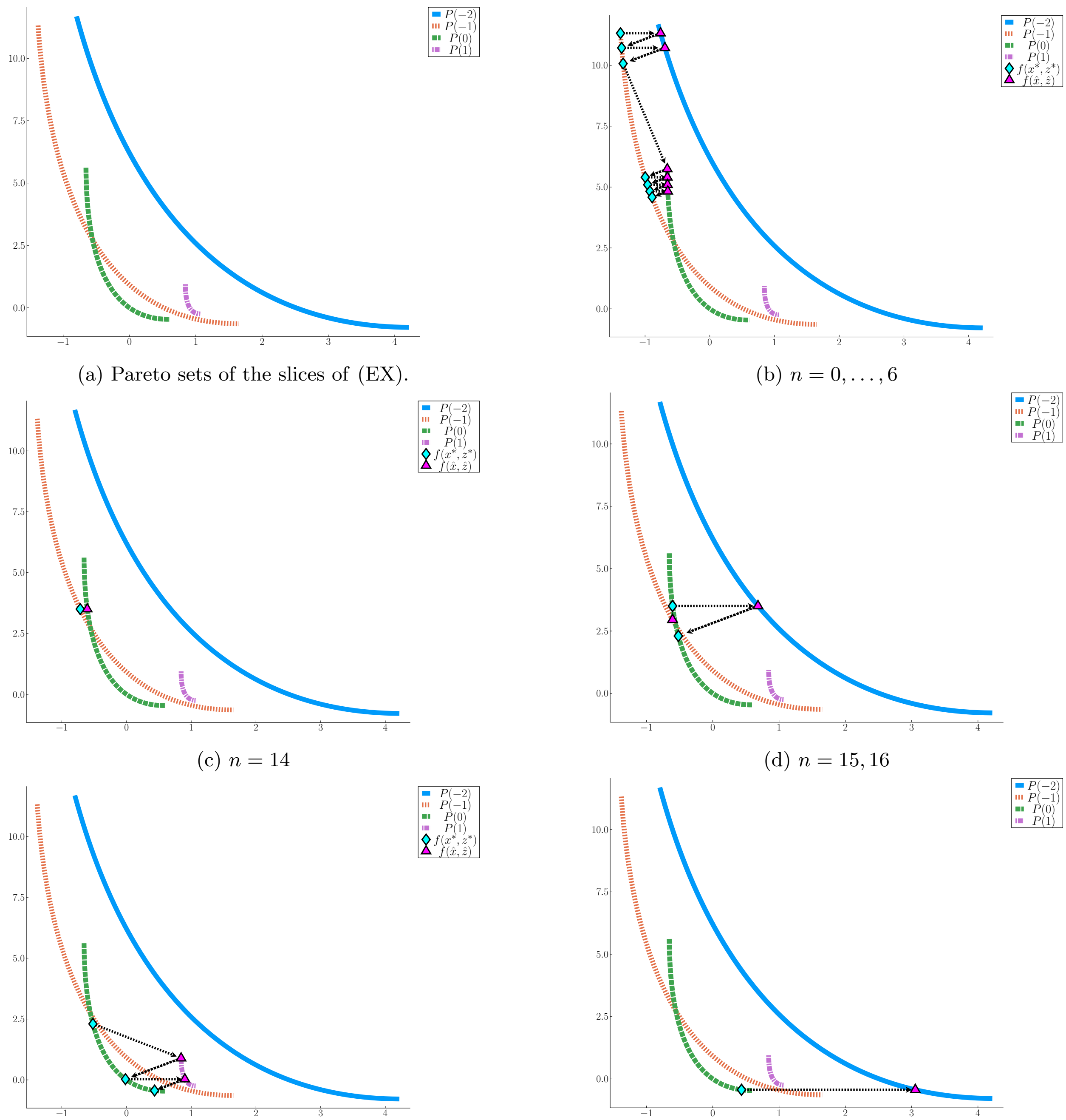
$n = 19$: Leaping from $\mathcal{P}(0)$, $\mathcal{P}(-2)$ is found and finally $|\sigma(f(x^*_{20}, z^*_{20}), f(\hat{x}_{19}, \hat{z}_{19}))| < 0.1$. The algorithm sets the reference point to $(x^*_{20}, z^*_{20}) = (\hat{x}_{19}, \hat{z}_{19})$.

$n = 20$: The algorithm formulates $(\text{LEAP}(x^*_{20}, z^*_{20}))$, but it is infeasible. The algorithm terminates.

The algorithm returns that the slices $\mathcal{P}(-1)$, $\mathcal{P}(0)$, and $\mathcal{P}(-2)$ are Pareto slices. Observe that when $n = 14$, by Propositioin 3.10, the algorithm detects that slices $\mathcal{P}(-1)$ and $\mathcal{P}(0)$ intersect. In particular, observe that for steps $n = 4, \ldots, 13$, $\hat{\varepsilon}_n$ is decreasing while $\sigma^*_n$ is increasing, as shown in Lemma 3.9. Thus, the algorithm concludes that $\mathcal{P}(-1) \cap \mathcal{P}(0) \neq \emptyset$ since $\hat{\varepsilon}_{14} < 0.1$ and $|\sigma^*_{14}| < 0.1$, where 0.1 is the tolerance used in this example. Although this tolerance is acceptable for this instance, in practice a much smaller value for the tolerance must be used in order to provide sufficient evidence that two slices intersect.

When $n = 15$, however, since the algorithm has already determined that $\mathcal{P}(-1)$ is a weak Pareto slice, it adds $z \neq -1$ as a tabu constraint to the leap problem. This tabu constraint remains until $n = 20$, when $\mathcal{P}(-2)$ is found. When the algorithm uses $f(x^*_{20}, z^*_{20})$ as the reference point in the leap problem, there is no need to keep the extra tabu constraint of $z \neq -1$ since the algorithm will use reference points from the new slice $\mathcal{P}(-2)$. Computationally, this is an important feature of the Pareto Leap Algorithm since the constraint set remains as small as possible through each iteration.

In the next section, we test the Pareto Leap Algorithm on test problems from the literature.




(a) Pareto sets of the slices of (EX).

(b) $n = 0, \ldots, 6$

(c) $n = 14$

(d) $n = 15, 16$

(e) $n = 17, 18$

(f) $n = 19, 20$

Figure 3: Graphical representations of notable steps from Pareto Leap Algorithm applied to (EX)

# 5 Numerical Results

We present results on applying Pareto Leap to four test instances from the literature. In particular, we use the test instance generator suggested in [11] and the test problems explored in [7]. For the experiments, we set a time limit of 3600 seconds. We implement Pareto Leap in `Julia 1.11.2` and we use `JuMP v1.23.5` to model each optimization problem and use `Gurobi 12.0.0` as the continuous and integer solver. All computations are done on a Linux machine with `11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz` processor and 32 `Gb` of memory.

Tables 3 and 4 present the results of test instances of BOMIPs from [7] and [11], respectively. In particular, the columns of Table 3 list the number of integer variables, the CPU time in seconds, and the number of times (LEAP($x^*, z^*$)) was solved over the course of instance. As the number of integer variables increases, Pareto Leap times out and is therefore unable to find all of the Pareto slices. However, this is due to the difficulty the solver faces as the test instances grow in the number of integer variables.

| $n_I$: # integer vars | CPU time (secs) | # leaps |
|---|---|---|
| Test Instance 4.2 | | |
| 1 | 0.130232546 | 12 |
| 2 | 0.276263542 | 6 |
| 4 | 1.417349512 | 6 |
| 8 | 22.792150244 | 6 |
| 10 | 210.478535816 | 6 |
| 16 | - | |
| 32 | - | |
| Test Instance 4.3 | | |
| 1 | 0.005424441 | 1 |
| 10 | 0.032031277 | 1 |
| 20 | 0.226425646 | 1 |
| 30 | 11.023565311 | 1 |
| 40 | 7.908380391 | 1 |
| 50 | 68.733359474 | 1 |
| Test Instance 4.4 (2 continuous variables used) | | |
| 1 | 0.882077267 | 5 |
| 2 | 0.382480336 | 101 |
| 4 | - | - |
| 8 | - | - |

Table 3: Results from testing Pareto Leap on test instances from [7].

The test instances from [11] provide the ability to observe this computational difficulty directly since these test instances have been specially constructed so that explicit computation of the number of integer values which correspond to Pareto slices, as well as the number of Pareto slices, is possible. Note that in Table 4, the first column lists the number of integer variables, while the third and fourth columns list the average CPU time in seconds and the average number times the leap problem was solved, respectively. The second column, however, list the parameters used in the construction of the test instances.

| # integer vars | $a \leq \|J\| \leq b$ | Avg CPU time (secs) | Avg # leaps |
|---|---|---|---|
| 2 | $a = 1, b = 1$ | 0.01149255 | 3 |
| 4 | $a = 1, b = 3$ | 0.134401771 | 30.33333333 |
| 8 | $a = 3, b = 7$ | 0.62530959125 | 126 |
| 16 | $a = 12, b = 15$ | 0.7292080355000001 | 124 |
| 32 | $a = 28, b = 31$ | 0.86698306625 | 121 |

Table 4: Results from testing Pareto Leap on test instance 4.13 from [11] with $\alpha_1 = \alpha_2 = 1/5$ and $n = 4$.

By the definition of Test Instance 4.13 in [11], the integer part of the mixed-integer program is constructed by a set $J$, which represents a strict subset of indices of the integer variables. In order to determine the number of integer values which correspond to a Pareto slice, [11] provides the formula $3^{n_I - |J|}$, where $n_I$ is the number of integer variables and $|J|$ is the size of the strict subset $J \subset \{1, 2, \ldots, n_I\}$. Similarly, the formula for the number of Pareto slices is $2(n_I - |J|) + 1$.

For instance, in the case of $n_I = 8$, if $|J| = 1$ then there are $3^{8-1} = 2187$ efficient integer assignments and $2(8-1)+1 = 15$ Pareto slices. One result of this is that there are on average $2187/15 = 145.8$ possible integer values per Pareto slice. As such, Pareto Leap adds a tabu constraint for each integer assignment corresponding to the same slice. In such cases, the algorithm times out. Nonetheless, when the algorithm does indeed find a new slice, the constraint set for the leap problem returns to only one tabu constraint since it uses a reference point from the new slice.

However, in the case that $n_I = 8$ and $|J| = 3$, then the number of integer values corresponding to a Pareto slice is $3^{8-3} = 243$ and the number of Pareto slices is $2(8-3)+1 = 11$, which results in an average of $243/11 = 22$ integer values per Pareto slice. This is much more manageable for the solver to find all Pareto slices before timing out. Indeed, similar conclusions may be inferred about the test instances from [7], although such explicit formulas about the efficient set and Pareto set are not known *a priori*.

Since Pareto Leap is the first method in the literature to focus on exactly identifying the (weak) Pareto slices of (BOMIP), we cannot make any comparisons to other MOMIP methods. Furthermore, since Pareto Leap is also the first tabu constraint algorithm to address the case of nonlinear BOMIPs, we cannot compare its performance to other tabu constraint methods either.

# 6 Conclusion

We have presented Pareto Leap, an algorithm which identifies all (weak) Pareto slices for a generic BOMIP. A special feature of this algorithm is the ability to recognize the existence of and to compute the intersection of the (weak) Pareto sets of two slices. In particular, we have provided necessary and sufficient conditions for (weak) efficiency and necessary and sufficient conditions for detecting and computing intersecting (weak) Pareto slices. This last result of detecting intersecting (weak) Pareto slices has previously not been possible in the literature. The requisite theoretical results we have proved lead naturally to the development of the algorithm. We then demonstrated the algorithm on an example which includes intersecting slices to show how Pareto Leap is able to handle such problem instances. Additionally, we tested the algorithm on test problems from the literature and concluded that Pareto Leap is a competitive algorithm for BOMIPs.

A benefit of the Pareto Leap Algorithm for solving (BOMIP) is that it provides information about the mapping between the Pareto set and the efficient set, namely which integer values correspond to which (weak) Pareto slices. Indeed, the results shown here enrich the class of tabu constraint methods because the Pareto Leap is able to handle BOMIPs whose underlying continuous problem is nonlinear.

As an additional benefit, Pareto Leap can be flexibly implemented. Since in general the (weak) Pareto set of (BOMIP) is disconnected, and Pareto Leap finds all values of the integer variables which correspond to each (weak) Pareto slice, the algorithm may be readily coupled with a solver that is well-suited for the underlying continuous problem. Thus, as Pareto Leap identifies (weak) Pareto slices, the continuous solver may be left with the task of finding a representation of each (weak) Pareto slice.

We see two primary directions for future work. First, improving the implementation of Pareto Leap for problem instances with a large number of integer variables. This direction is primarily a computational question and may depend on applying solvers that are specifically designed to take advantage of the properties of the particular problem instance. Secondly, it is interesting, and likely challenging, to consider in what ways the geometric intuition underlying Pareto Leap can be extended, or needs to be altered, to problems with more than two objective functions.

## Acknowledgements

This work was supported by Clemson University's Virtual Prototyping of Autonomy Enabled Ground Systems (VIPR-GS), under Cooperative Agreement W56HZV-21-2-0001 with the US Army DEVCOM Ground Vehicle Systems Center (GVSC). DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. OPSEC9645

# References


[1] Lavinia Amorosi and Marianna De Santis. On the accurate detection of the Pareto frontier for bi-objective mixed integer linear problems. `https://optimization-online.org/?p=26478`.

[2] Guillermo Cabrera-Geurrero, Matthias Ehrgott, Andrew J. Mason, and Andrea Raith. Bi-objective optimisation over a set of convex sub-problems. *Annals of Operations Research*, 2021.

[3] Valentina Cacchiani and Caludia D'Ambrosio. A branch-and-bound based heuristic algorithm for convex multi-objective MINLPs. *European Journal of Operational Research*, 260:920–933, 2017.

[4] Gökhan Ceyhan, Murat Köksalan, and Banu Lokman. Finding a representative nondominated set for multi-objective mixed integer programs. *European Journal of Operational Research*, 272:61–77, 2019.

[5] Gökhan Ceyhan, Murat Köksalan, and Banu Lokman. Finding a representative nondominated set for multi-objective mixed integer programs. *European Journal of Operational Research*, 272:61–77, 2019.

[6] Xin Cheng and Xiang Li. Discretization and global optimization for mixed integer bilinear programming. *Journal of Global Optimization*, 84, 2022.

[7] Marianna De Santis, Gabriele Eichfelder, Julia Niebling, and Stefan Rocktäschel. Solving multiobjective mixed integer convex optimization problems. *SIAM Journal of Optimization*, 30(4):3122–3145, 2020.

[8] Marianna De Santis, Gabriele Eichfelder, Daniele Patria, and Leo Warnow. Using dual relaxations in multiobjective mixed-integer convex quadratic programming. *Journal of Global Optimization*, 2024.

[9] Erik Diessel. An adaptive patch approximation algorithm for bicriteria convex mixed-integer problems. *Optimization*, 2021.

[10] Ilgın Doğan, Banu Lokman, and Murat Köksalan. Representing the nondominated set in multi-objective mixed-integer programs. *European Journal of Operational Research*, 296:804–818, 2022.

[11] Gabriele Eichfelder, Tobias Gerlach, and Leo Warnow. A test instance generator for multiobjective mixed-integer optimization. *Mathematical Methods of Operations Research*, 100:385–410, 2024.

[12] Gabriele Eichfelder and Oliver Stein. Limit sets in global multiobjective optimization. *Optimization*, 73(1):1–27, 2024.

[13] Gabriele Eichfelder, Oliver Stein, and Leo Warnow. A solver for multiobjective mixed-integer convex and nonconvex optimization. *Journal of Optimization Theory and Applications*, 203:1736–1766, 2024.

[14] Gabriele Eichfelder and Leo Warnow. A hybrid patch decomposition approach to compute an enclosure for multi-objective mixed-integer convex optimization problems. *Mathematical Methods of Operations Research*, 100:291–320, 2024.

[15] Matteo Fishetti and Andrea Lodi. Local branching. *Mathematical Programming*, 98:23–47, 2003.

[16] Pascal Halffmann, Luca E. Schäfer, Kerstin Dächert, Kathrin Klamroth, and Stefan Ruzika. Exact algorithms for multiobjective linear optimization problems with integer variables: A state of the art survey. *Journal of Multi-Criteria Decision Analysis*, 29:341–363, 2022.

[17] John Hooker. *Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction*. John Wiley & Sons, 2000.

[18] Ina Lammel, Karl-Heinz Küfer, and Philipp Süss. An approximation algorithm for multiobjective mixed-integer convex optimization. *Mathematical Methods of Operations Research*, 100, 2024.

[19] Kaisa Miettinen. *Nonlinear Multiobjective Optimization*. Springer, 1998.

[20] Julia Niebling and Gabriele Eichfelder. A branch-and-bound-based algorithm for nonconvex multiobjective optimization. *SIAM Journal on Optimization*, 29(1):794–821, 2019.

[21] Diego Pecin, Ian Herszterg, Tyler Perini, and Natashia Boland. A fast and robust algorithm for solving biobjective mixed integer programs. *Mathematical Methods of Operations Research*, 100, 2024.

[22] Tyler Perini, Natashia Boland, Diego Pecin, and Martin Savelsbergh. A criterion space method for biobjective mixed integer programming: The boxed line method. *INFORMS Journal on Computing*, 32(1):16–39, 2020.

[23] Seyyed Amir Babak Rasmi and Metin Türkay. GoNDEF: an exact method to generate all non-dominated points of multi-objective mixed-integer linear programs. *Optimization and Engineerings*, 20:89–117, 2019.

[24] Banu Soylu. The search-and-remove algorithm for biobjective mixed-integer linear programming problems. *European Journal of Operational Research*, 268:281–299, 2018.

[25] Banu Soylu and Gazi Bilal Yıldız. An exact algorithm for biobjective mixed integer linear programming. *Computers & Operations Research*, 72:204–213, 2016.

[26] Thomas Stidsen, Kim Allan Andersen, and Bernd Dammann. A branch and bound algorithm for a class of biobjective mixed integer programs. *Management Science*, 60(4):1009–1032, 2014.

[27] James Ward. Structure of efficient sets for convex objectives. *Mathematics of Operations Research*, 14(2):249–257, 1989.

[28] Andrzej P. Wierzbicki. A mathematical basis for satisficing decision making. *Mathematical Modelling*, 3(5):391–405, 1982. Special IIASA Issue.

[29] A.P. Wierzbicki. On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR Spektrum*, 8:73–87, 1986.