Industrial and
Systems Engineering

On the Relationship Between the Value Function and the Efficient Frontier of a Mixed Integer Linear Optimization Problem

Samira Fallah[1], Ted K. Ralphs[1], Natashia L. Boland[2], and Lawrence V. Snyder[1]

[1]Department of Industrial and Systems Engineering, Lehigh University, USA

Technical Report 22T-005-R1

LEHIGH UNIVERSITY

COR@L
COMPUTATIONAL OPTIMIZATION RESEARCH AT LEHIGH

On the Relationship Between the Value Function and the Efficient Frontier of a Mixed Integer Linear Optimization Problem

Samira Fallah*[1], Ted K. Ralphs†[1], Natashia L. Boland‡[2], and Lawrence V. Snyder§[1]

[1]Department of Industrial and Systems Engineering, Lehigh University, USA

Original Publication: April 15, 2022

Last Revised: October 15, 2022

Abstract

We study the relationship between the efficient frontier of a general multiobjective mixed integer linear optimization problem (MILP) and the so-called *restricted value function* (RVF) of a related MILP. We show that the efficient frontier lies on a subset of the boundary of the epigraph of the RVF and that methods of construction of the RVF can also be viewed as methods of constructing the efficient frontier. The relationship described is implicit in many applications in which analysis of either the value function or the efficient frontier is required as part of a larger algorithm, such as in bilevel optimization, Dantzig-Wolfe decomposition, Lagrangian relaxation, and dynamic programming. Methods of construction for the value function and the efficient frontier have so far been studied in largely separate bodies of work. However, we argue that algorithms intended for these separate tasks can be used interchangeably. By observing this relationship, we propose a generalized cutting plane algorithm for constructing the efficient frontier of a multiobjective MILP based on a generalization of an existing algorithm for constructing the classical value function. We show that the algorithm is finite under a standard boundedness assumption and comes with a performance guarantee if terminated early. Finally, we provide a Python package that shows the proposed algorithm works in practice.

1 Introduction

In this paper, we consider the relationship between a certain value function (VF) associated with a mixed integer linear optimization problem (MILP), which we refer to as the *restricted value function* (RVF), and the efficient frontier of a related multiobjective optimization problem. One of the main results is to show that the efficient frontier is a subset of the points on the boundary of the epigraph of the RVF. We also show that the efficient frontier provides all the critical information

*`saf418@lehigh.edu`

†`ted@lehigh.edu`

‡`natashia.boland@gmail.com`

§`lvs2@lehigh.edu`

needed to construct the RVF. This has broad-ranging implications, including the observation that any algorithm for constructing the efficient frontier can also be used to construct the RVF and vice versa. To illustrate this, we propose a generalized cutting plane algorithm for constructing both the RVF and the efficient frontier. The approach we suggest is finite and exploits the discrete structure of the RVF, providing a performance guarantee if terminated early. It is a modified version of an existing algorithm for the construction of the full value function, and to our knowledge, the approach is entirely different from existing algorithms for the construction of the efficient frontier. It is, furthermore, one of only two algorithms developed to date that addresses multiobjective MILPs in the presence of continuous variables, with any number of objectives, and it is easily analyzed to provide far better bounds on the number and size of subproblems that need to be solved to determine the discrete structure of the efficient frontier. These bounds are also comparable to existing algorithms for the pure integer case (without continuous variables).

Given that there are extensive, separate bodies of work addressing algorithms for each of these construction problems, the relationship between the RVF and the efficient frontier may seem surprising at first. However, the relationship is quite intuitive and easily understood. While we believe this is the first paper to formally and explicitly establish this relationship, it can be seen implicitly in the results of several works that have come before. The idea of the relationship between the efficient frontier and the value function is implicit in earlier works, e.g., by Trapp et al. [2013], Ralphs and Hassanzadeh [2014], and Bodur et al. [2016]. The so-called minimal tenders utilized in Trapp et al.'s algorithm for the construction of the value function of a pure integer program can be seen as the points on the efficient frontier of a related multiobjective problem. Ralphs and Hassanzadeh [2014] generalized this concept in their work on the structure of the value function of a general MILP. More recently, Bodur et al. [2016] observed that in block-structured problems, the solution of the column generation subproblem can be seen as equivalent to evaluating a certain value function, and solutions can thus be restricted only to so-called *nondominated points*. This is related to the fact that the Lagrangian relaxation of a given MILP can also be seen as a weighted sum subproblem in a multiobjective context and makes the connection to the multiobjective context more evident.

Multiobjective Optimization. Before going further, let us now set the stage by formally defining the multiobjective MILP that is our subject of study, as

$$\operatorname{vmin}_{(x_I,x_C)\in X_{\mathrm{MO}}} C_I x_I + C_C x_C, \tag{MO-MILP}$$

where

$$X_{\mathrm{MO}} = \{(x_I, x_C) \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+ : A_I x_I + A_C x_C = b\},$$

is the feasible region; $A \in \mathbb{Q}^{m\times n}$ is the matrix of coefficients of the constraints; $b \in \mathbb{Q}^m$ is the right-hand side (RHS) of the constraints; and the rows of matrix $C \in \mathbb{Q}^{(l+1)\times n}$ are the multiple objectives of the problem. The vmin operator indicates that this is a vector minimization (multi-objective) problem. A_I and C_I are the submatrices of A and C consisting of columns associated with the integer variables (indexed by set $I = \{0, \ldots, r-1\}$), while A_C and C_C are the submatrices corresponding to the columns associated with the continuous variables (indexed by set $C = \{r, \ldots, n-1\}$). We assume that X_{MO} is bounded.

The goal of multiobjective optimization is to examine the tradeoffs inherent in attempting to collectively optimize the multiple objectives. This analysis is most naturally done in the $(l+1)$-dimensional space known as the *criterion space*, which contains the vectors of objective values associated with points in the n-dimensional *decision space*, which is the space containing X_{MO}. Whereas "solving" an MILP with a single objective means to determine its unique optimal value, "solving" a multiobjective MILP means to generate the set of all vectors in criterion space associated with the so-called *efficient solutions*, those for which there is no other solution for which the objective value is at least as good for every objective and strictly better for at least one objective.

We briefly review some concepts in multiobjective optimization, referring the reader to [Ehrgott, 2005] for more details. A central concept is that of dominance. The point $z = (C_I x_I + C_C x_C) \in \mathbb{R}^{l+1}$ in criterion space associated with $(x_I, x_C) \in X_{\text{MO}}$ *dominates* $z' = (C_I x'_I + C_C x'_C) \in \mathbb{R}^{l+1}$ where $(x'_I, x'_C) \in X_{\text{MO}}$ if $C_I x_I + C_C x_C \lneqq C_I x'_I + C_C x'_C$, i.e., $(C_I x_I + C_C x_C)_j \leq (C_I x'_I + C_C x'_C)_j$ for all $j = \{0, 1, \ldots, l\}$ and $(C_I x_I + C_C x_C)_j < (C_I x'_I + C_C x'_C)_j$ for at least one index $j \in \{0, 1, \ldots, l\}$. A point z in criterion space that is not dominated by any other point is called a *nondominated point* (NDP). The "vmin" operator indicates that the goal of the problem is to generate the set of all NDPs, called the *efficient frontier*. A preimage of an NDP is a point in decision space and is referred to as an *efficient solution*. A point $(x_I, x_C) \in X_{\text{MO}}$ that is not necessarily efficient but for which there does not exist $(x'_I, x'_C) \in X_{\text{MO}}$ such that $C_I x'_I + C_C x'_C < C_I x_I + C_C x_C$ is called *weakly efficient* and the associated point $C_I x_I + C_C x_C$ in criterion space is called a *weakly nondominated point*.

The set of NDPs can be further divided into three subsets: supported, extreme supported, and unsupported. Supported NDPs are those that lie on the boundary of the convex hull of the efficient frontier, while extreme supported NDPs are the extreme points of the convex hull. Unsupported NDPs are those that lie in the interior of the convex hull. Figure 2 below illustrates these concepts.

Restricted Value Function. Another seemingly different way of analyzing the same trade-offs is by analyzing the *restricted value function* (RVF) associated with a related MILP. Consider the following MILP with a single objective obtained by imposing all but one of the objectives in (MO-MILP) as constraints.

$$\min_{(x_I, x_C) \in X} c_I^0 x_I + c_C^0 x_C, \tag{MILP}$$

where

$$X = \{(x_I, x_C) \in \mathbb{Z}_+^r \times \mathbb{R}_+^{n-r} : C_I^{1:l} x_I + C_C^{1:l} x_C \leq d, A_I x_I + A_C x_C = b\},$$

is the feasible region; c^0 is the first row of the matrix C; $C^{1:l}$ is the submatrix consisting of the remaining rows of C, and $d \in \mathbb{Q}^l$ is a vector that will be used as a parameter to obtain the aforementioned RVF.

We now define the RVF $z : \mathbb{R}^l \to \mathbb{R} \cup \{\pm\infty\}$ associated with (MILP) to be the function

$$z(\zeta) = \min_{(x_I, x_C) \in \mathcal{S}(\zeta)} c_I^0 x_I + c_C^0 x_C, \tag{RVF}$$

that returns the optimal solution value of (MILP) as a function of a RHS parameter ζ, where for each $\zeta \in \mathbb{R}^l$, we define

$$\mathcal{S}(\zeta) = \{(x_I, x_C) \in \mathbb{Z}_+^r \times \mathbb{R}_+^{n-r} : C_I^{1:l} x_I + C_C^{1:l} x_C \leq \zeta, A_I x_I + A_C x_C = b\}.$$

The function z is similar to the classical value function except that the right-hand side (RHS) $b \in \mathbb{Q}^m$ of some of the constraints are fixed. As usual, we let $z(\zeta) = \infty$ for $\zeta \notin \mathcal{C}$, where

$$\mathcal{C} = \{\zeta \in \mathbb{R}^l : \mathcal{S}(\zeta) \neq \emptyset\}.$$

The function z is always bounded from below because of our assumption that X_{MO} is bounded. Note that in practical applications, we may artificially constrain the finite domain of z to some bounded region of interest instead of constraining $\mathcal{C}$ in order to improve the efficiency of the algorithm.

Example 1. *Here, we illustrate the concepts discussed so far. Consider the following instance of* (RVF)*:*

$$
\begin{aligned}
z(\zeta) = \min \quad & 2x_1 + 0y_1 + 7y_2 + 10y_3 + 2y_4 + 10y_5 \\
& -x_1 + 10y_1 + 8y_2 + y_3 - 7y_4 + 6y_5 \leq \zeta \\
& -x_1 + 9y_1 + 3y_2 + 2y_3 + 6y_4 - 10y_5 = 4 \\
& x_1 \in \mathbb{Z}_+ \\
& y_j \in \mathbb{R}_+ \quad \forall j \in \{1, 2, \ldots, 5\}.
\end{aligned}
$$

Figure 1 below shows the value function for the MILP, while Figure 2 shows the efficient frontier for the associated multiobjective optimization problem. Note that the graph of the VF and the efficient frontier are identical.

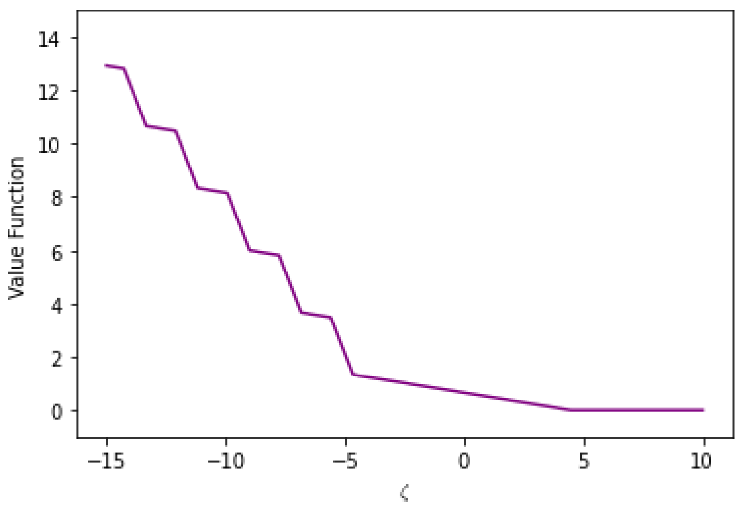


Figure 1: RVF associated with Example 1.

In the remainder of the paper, we formally show that the RVF and the efficient frontier capture the same information and that algorithms for the construction of the two are effectively interchangeable. Throughout the paper, we'll consider an instance (MILP) with its associated RVF and a corresponding instance of (MO-MILP). The paper is organized as follows. In Section 2 we review related work. In Section 3, we provide a characterization of the RVF in terms of a discrete set of integer parts of NDPs. In Section 4, we formalize the relationship between the RVF and the efficient frontier. Finally, in Section 5, we present our cutting plane algorithm for constructing both the efficient frontier and the value function. Both the value function representation and the cutting plane algorithm are finite under our assumption that X_{MO} is bounded. In Section 6, we provide a summary and concluding remarks, as well as directions for future work.

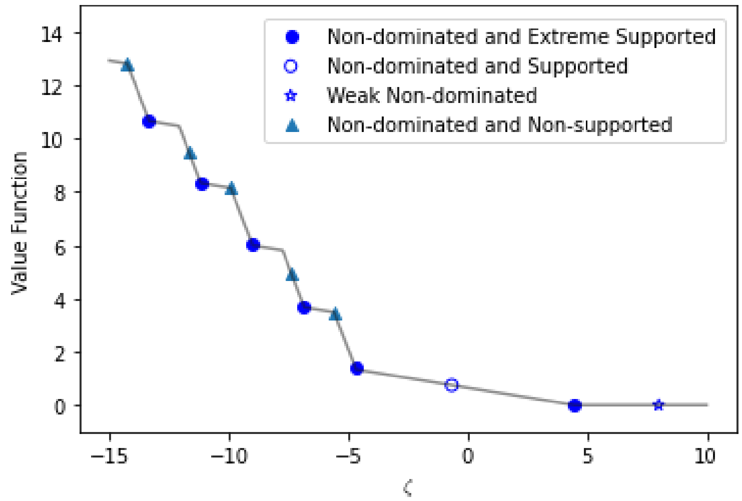


Figure 2: Efficient frontier associated with Example 1.

2 Related Work

Methods both for constructing the efficient frontier of a multiobjective optimization problem and for constructing the value function of an MILP have been extensively studied in the open literature. We review some of the important and related literature from each area in the following subsections.

2.1 Multiobjective Optimization

Multiobjective optimization is the study of optimization problems in which the goal is to analyze the tradeoff between multiple conflicting objective functions. Multiobjective optimization has applications in many fields, as most real-world problems arising in practice do have multiple objectives. In this study, we are interested in the algorithms that are capable of generating the exact efficient frontier for the MILP case.

The purpose of algorithms for multiobjective optimization is to generate the efficient frontier, i.e., in which there cannot be an improvement in one objective without sacrificing at least one other objective (see definitions in Section 1). The efficient frontier allows one to examine the tradeoffs between the conflicting objectives. One straightforward way of solving a multiobjective problem is by a technique known as *scalarization* in which the problem is reformulated as a single-objective problem. We refer the interested reader to Ehrgott [2006] for a comprehensive review of scalarization methods in the multiobjective area. Several scalarization methods that have been proposed include the *weighted sum* method [Zadeh, 1963], the *perpendicular search* method [Chalmet et al., 1986], the *weighted Tchebycheff* method [Bowman, 1976, Yu, 1973, Zeleny, 1973], the ϵ-*constraint* method [Haimes, 1971], the Hybrid method [Guddat et al., 1985], the Benson's method [Benson, 1978], and the Pacoletti-Serafini method [Pascoletti and Serafini, 1984]. All these methods follow the same basic approach. In each iteration of the algorithms, there is a list of NDPs (efficient solutions) and some unexplored criterion space (decision space). At each iteration, the procedure explores a new criterion space (decision space), i.e., looks for a new NDP (efficient solution), and expands the list of NDPs (efficient solutions). The procedure terminates when there are no more unexplored regions.

The most straightforward scalarization method is the weighted sum method. The single objective created by this method is a weighted sum of the objectives. When the weights are all positive, the solution to the weighted sum problem is guaranteed to be nondominated. On the other hand, not all nondominated solutions can be generated as a solution to some weighted sum problem—only the so-called *extreme supported NDPs* can be generated in this way. The NDPs that cannot be found via weighted sum based scalarization are called *unsupported* NDPs. The perpendicular search method is similar to the weighted sum, but it is adjusted in such a way that there is no possibility that one of the corner points in criterion space, which are the solutions to the different combinations of the lexicographic method of the exploring region, will be the next NDP.

Several methods have been proposed to address unsupported NDPs. One of them is the augmented weighted Tchebycheff method [Bowman, 1976]. It finds the NDPs in the exploring regions by minimizing the distance to the ideal point, which is defined as a point whose components are obtained by minimizing objective functions, by the assumption that we have a minimization multiobjective optimization problem over the feasible region. Ralphs et al. [2006] proposed a weighted Tchebycheff scalarization algorithm for constructing the efficient frontier of a biobjective integer programming problem. The ϵ-constraint method is another popular scalarization method, and all the NDPs can be found by using this method. The procedure includes minimizing a primary objective while restricting the other objectives in the form of inequality constraints. In other words, it can be considered as a parametric search on the RHS values of the objectives that are restricted in the constraint set. This parametric search is straightforward for biobjective problems; hence biobjective discrete optimization problems are often solved using the ϵ-constraint method.

For a detailed literature review on multiobjective optimization, we refer the interested reader to [Ehrgott and Gandibleux, 2000, Ehrgott and Wiecek, 2005, Ehrgott et al., 2016]. We also refer the interested reader to Halffmann et al. [2022] for the most recent survey and comprehensive overview of the algorithms for multiobjective MILPs.

2.2 Value Function

The classical value function of an MILP is well-studied, and understanding its structure is critically important for many applications. The VF is a core ingredient in the optimality conditions utilized in a wide range of algorithms for solving optimization problems. These same optimality conditions are also employed in formulating and solving important classes of multistage and multilevel optimization problems in which optimality conditions are embedded as constraints in a larger optimization problem. Finally, optimality conditions are also the basis for techniques used for warm-starting and sensitivity analysis, which are the areas in which the connection to multiobjective optimization is most obvious.

There have been several studies addressing the structure of the VF. Blair and Jeroslow [1977] and Blair and Jeroslow [1979] described fundamental properties of the value function, including that it is piecewise polyhedral. Blair and Jeroslow [1982] showed that the VF of a pure integer linear optimization problem (PILP) is a *Gomory function* that is the maximum of subadditive functions known as *Chvátal functions*. Blair and Jeroslow [1984] extended this result to general MILPs, showing that they are the maximum of Gomory functions. Finally, Blair [1995] identified what was then referred to as a "closed-form" representation of the MILP VF, the so-called *Jeroslow formula*, though this did not lead to what could be considered a practical representation. Güzelsoy and Ralphs [2007] further studied properties of the value function as it is related to methods of warm-starting and sensitivity analysis and also suggested a method of construction for the case of an MILP with a single constraint. Ralphs and Hassanzadeh [2014] built on this work by further detailing the structure and properties of the VF for a general MILP, as well as suggesting a practical representation.

Most methods of construction have focused on the case of a pure integer program, where the discrete structure is the most evident and finite representation is the easiest to achieve. Wolsey [1981] showed how to use a cutting-plane method to derive a sequence of Chvátal functions leading to the construction of the full value function for a PILP. Conti and Traverso [1991] used reduced Gröbner bases and modified the classical Buchberger's algorithm to solve PILPs. Later, Schultz et al. [1998] used Gröbner basis methods for solving two-stage stochastic programs with complete integer recourse and different RHSs. The authors identified a countable set called the candidate set of the first-stage variables in which the optimal solution is contained. Then Ahmed et al. [2004] developed a global optimization algorithm for solving general two-stage stochastic programs with integer recourse and discrete distributions by exploiting the structure of the second-stage integer problem VF. The authors showed that their algorithm avoids an enumeration of the search space. Kong et al. [2006] considered a two-stage pure IP and presented a superadditive dual formulation that exploits the VF in both stages, solving that reformulation by a global branch-and-bound or level-set approach. Trapp and Prokopyev [2015] proposed a constraint-aggregation based approach to alleviate the memory requirement for storing the VF. Zhang and Özaltın [2021] first generalized the complementary slackness theorem to bilevel IP (BIP) and showed that it can be an advantage for constructing the value functions of BIP. The authors also showed that the value functions can be constructed by bilevel minimal RHS vectors. Finally, the authors presented a dynamic programming algorithm for constructing the BIP VF. Brown et al. [2021] used a Gilmore-Gomory approach to construct the integer programming (IP) VF.

There have been relatively few algorithmic advances in finding the VF of a general MILP. Bank et al. [1982] studied the qualitative and quantitative stability properties of mixed integer multiobjective optimization problems, which also can be considered as a MILP VF. Guzelsoy and Ralphs [2006] proposed algorithms for constructing the value function of a MILP with a single constraint. Properties of the VF and a method of construction for the VF in the case of a general MILP were discussed in [Ralphs and Hassanzadeh, 2014]. In this paper, we generalize the work in [Ralphs and Hassanzadeh, 2014] to the multiobjective setting.

3 Structure of the RVF

Before presenting the main theoretical result of the paper regarding the relationship between the epigraph of the RVF and the efficient frontier, we describe some basic properties of the RVF. The first step in understanding its structure in the general MILP case is to examine the structure of the related RVF that arises when the underlying problem is a (continuous) linear optimization problem (LP). We refer to this special case of the RVF as the *restricted LP value function* (RLPVF) and define it as follows:

$$z_{\mathrm{LP}}(\zeta;\beta) = \min \quad \begin{array}{l} c_C^0 x_C \\ C_C^{1:l} x_C \le \zeta \\ A_C x_C = \beta \\ x_C \in \mathbb{R}_+^{n-r}. \end{array} \tag{RLPVF}$$

For now, $\beta \in \mathbb{Q}^m$ can be thought of as the part of the right-hand side that is fixed, in which case z_{LP} is the special case of (RVF) in which $r = 0$ (there are no integer variables). Later, however, we'll consider a parametric class of functions of this form, with different values of β arising from fixing the integer part of the solution in (RVF). This is why β is denoted as a secondary parameter here, in contrast with the right-hand side vector b in (RVF), which is fixed throughout the paper. As previously, we assume that $z_{\mathrm{LP}}(\cdot;\beta)$ is bounded from below for all $\beta \in \mathbb{Q}^m$.

Blair and Jeroslow [1977] studied the structure of the full LP VF (the case of $m = 0$ in which there are no equality constraints and the full RHS is allowed to vary). In this special case, it is well-known that z_{LP} is a convex polyhedral function whose epigraph is a polyhedral cone. In the more general case of the RLPVF, the function is instead a *slice* of the full LP VF, and its epigraph is hence the intersection of a hyperplane with a polyhedral cone.

To analyze the function's structure, we consider, for a fixed $\beta \in \mathbb{Q}^m$, the dual of the LP that arises in the evaluation of $z_{\mathrm{LP}}(\hat{\zeta};\beta)$ for some $\hat{\zeta} \in \mathbb{Q}^l$. The feasible region of this dual problem is the polyhedron

$$P_{\mathcal{D}} = \{(u, v) \in \mathbb{R}^l_- \times \mathbb{R}^m : {C_C^{1:l}}^\top u + A_C^\top v \leq c_C^0\},$$

and its objective is to maximize the function $\hat{\zeta}^\top u + \beta^\top v$. By assumption, $P_{\mathcal{D}}$ is non-empty. Let $\{(u^e, v^e)\}_{e\in\mathcal{E}}$ and $\{d^r, e^r\}_{r\in\mathcal{R}}$ be the sets of extreme points and extreme rays of $P_{\mathcal{D}}$, respectively. For a fixed β, let $\mathcal{C}_{\mathrm{LP}} = \{\zeta : \zeta^\top d^r + \beta^\top e^r \leq 0 \ \ \forall r \in \mathcal{R}\}$ be the values of ζ for which $z_{\mathrm{LP}}(\zeta;\beta)$ is finite. Note that the feasible region $P_{\mathcal{D}}$ does not depend on $\hat{\zeta}$; only the objective function depends on $\hat{\zeta}$.

Proposition 3.1. *For a given $\beta \in \mathbb{Q}^m$, $z_{\mathrm{LP}}(\hat{\zeta};\beta)$ is a convex polyhedral function over $\mathcal{C}_{\mathrm{LP}}$.*

Proof. From strong duality, we have that

$$\begin{aligned} z_{\mathrm{LP}}(\hat{\zeta};\beta) &:= \max_{(u,v)\in P_{\mathcal{D}}} \hat{\zeta}^\top u + \beta^\top v \\ &= \begin{cases} \max\limits_{e\in\mathcal{E}} \ \hat{\zeta}^\top u^e + \beta^\top v^e & \text{if } \hat{\zeta}^\top d^r + \beta^\top e^r \leq 0 \text{ for all } r \in \mathcal{R}, \\ \infty & \text{otherwise.} \end{cases} \end{aligned} \tag{D-RLPVF}$$

By (D-RLPVF), the epigraph of RLPVF over $\mathcal{C}_{\mathrm{LP}}$ is

$$\{(\zeta, z) \in \mathbb{R}^{l+1} : \zeta^\top u^e + \beta^\top v^e \leq z, \forall e \in \mathcal{E}\},$$

which is a polyhedron. Therefore, z_{LP} is a convex polyhedral function over $\mathcal{C}_{\mathrm{LP}}$. ∎

Regarding the differentiability of z_{LP}, we can generalize the result of Bazaraa et al. [1990], who characterized the differentiability of the full LP VF. Consider a fixed $\hat{\zeta} \in \mathcal{C}_{\mathrm{LP}}$ for which the optimal solution to (D-RLPVF) is unique. With a small perturbation in the RHS $\hat{\zeta}$, the optimal basis does not change, and it is easy to see that the function must therefore be locally affine. The differentiability of the RLPVF at $\hat{\zeta}$ follows. We formalize this as follows.

Proposition 3.2. *If, for given β, $z_{\mathrm{LP}}(\zeta,\beta)$ is differentiable at $\zeta = \hat{\zeta} \in \mathcal{C}_{\mathrm{LP}}$, then there exists a unique $\hat{u} \in \mathbb{R}^l_-$ such that $(\hat{u}, v) \in P_{\mathcal{D}}$, $z_{\mathrm{LP}}(\hat{\zeta};\beta) = \hat{\zeta}^\top \hat{u} + \beta^\top v$, and the gradient of z_{LP} at $\hat{\zeta}$ is $\hat{u}$. If $\hat{\zeta} \in \mathbb{R}^l$ is a point of non-differentiability of z_{LP}, then there exist $(u^1, v^1), (u^2, v^2), \ldots, (u^s, v^s) \in P_{\mathcal{D}}$ with $s > 1$ such that $z_{\mathrm{LP}}(\hat{\zeta};\beta) = \hat{\zeta}^\top u^1 + \beta^\top v^1 = \hat{\zeta}^\top u^2 + \beta^\top v^2 = \ldots = \hat{\zeta}^\top u^s + \beta^\top v^s$, and the directional derivatives $\nabla_d z(\hat{\zeta})$ in direction $d \in \mathbb{R}^l$, if it exists, is $d^\top u$, where $u \in \{u^1, u^2, \ldots, u^s\}$.*

Proof. From (D-RLPVF), we have that $z_{\text{LP}}(\hat{\zeta};\beta) = \max_{(u,v)\in P_{\mathcal{D}}} \hat{\zeta}^\top u + \beta^\top v$. Since the feasible solution $P_{\mathcal{D}}$ does not rely on $\hat{\zeta}$, the optimal solution to (D-RLPVF) lies on the basic feasible solutions of $P_{\mathcal{D}}$. Then two cases can happen. In one case, there is just one pair of $(\hat{u}, v)$ as the optimal solution to (D-RLPVF). In another case, there could be more than one pair, i.e., in the case of duplicate optimal solutions, then we have $z_{\text{LP}}(\hat{\zeta};\beta) = \hat{\zeta}^\top u^1 + \beta^\top v^1 = \hat{\zeta}^\top u^2 + \beta^\top v^2 = \ldots = \hat{\zeta}^\top u^s + \beta^\top v^s$. In the former case, the RVF is differentiable, and the gradient of z_{LP} at $\hat{\zeta}$ is $\hat{u}$. In the latter case, $\nabla_d z(\hat{\zeta}) = d^\top u$ where $d \in \mathbb{R}^l$ and $u \in \{u^1, u^2, \ldots, u^s\}$. ■

In other words, when z_{LP} is differentiable at $\hat{\zeta}$, its gradient is the (unique) optimum to (D-RLPVF). Similarly, when z_{LP} is non-differentiable, the directional derivatives $\nabla_d z(\hat{\zeta})$ in direction $d \in \mathbb{R}^l$, if it exists, is $d^\top u$, where u is one of the alternative optimal solutions to (D-RLPVF). It is easy to see that $\nabla_d z(\hat{\zeta}) \leq 0$ for all $d \in \mathbb{R}^l_+$ (and $\nabla_d z(\hat{\zeta}) \geq 0$ for all $d \in \mathbb{R}^l_-$), since $u \in \mathbb{R}^l_-$ for all $(u, v) \in P_{\mathcal{D}}$. It follows that z_{LP} is non-increasing.

Example 2. *In Figure 3, we plot an instance of* (RLPVF) *obtained from Example 1 by setting all the integer variables to zero. From Figure 3, it can be seen that* (RLPVF) *is a convex polyhedral function with some points that are non-differentiable at which the optimal basis changes. The optimal basis is constant on each piece of this piecewise linear function.*

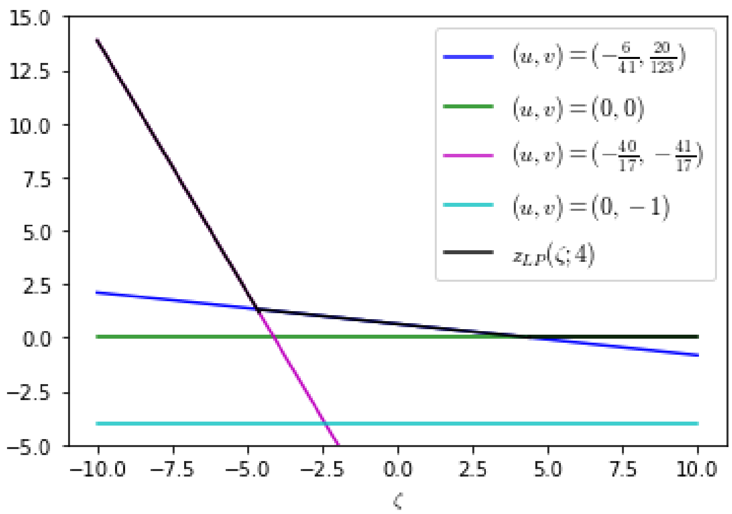


Figure 3: The RLPVF associated with Example 1 when $x_1 = 0$.

Here we discuss how the optimal basis changes and illustrate the relationship between directional derivatives and the optimal dual solutions of the corresponding (D-RLPVF) *explicitly with Figure 3. The dual problem of Example 1 when $x_1 = 0$ and for a fixed ζ is*

$$
\begin{aligned}
z(\zeta) = \max \quad & \zeta u + 4v \\
& 10u + 9v \leq 0 \\
& 8u + 3v \leq 7 \\
& 1u + 2v \leq 10 \\
& -7u + 6v \leq 2 \\
& 6u - 10v \leq 10 \\
& u \leq 0.
\end{aligned}
\tag{Ex1-D}
$$

The optimal solution to (Ex1-D) *lies on the basic feasible solutions of its feasible region, which does not rely on* ζ*. The dual problem can be rewritten based on the basic feasible solutions, which are* $[(-\frac{6}{41}, \frac{20}{123}), (0,0), (-\frac{40}{17}, -\frac{41}{17}), (0,-1)]$ *as*

$$z(\zeta) = \max(-\tfrac{6}{41}\zeta + \tfrac{20}{123} \times 4, 0\zeta + 0 \times 4, -\tfrac{40}{17}\zeta + -\tfrac{41}{17} \times 4, 0\zeta + -1 \times 4). \tag{1}$$

It can be observed from Figure 3 that the left part of the RLPVF associated with Example 1 when $x_1 = 0$ *is constructed by the line associated with* $(u^*, v^*) = (-\frac{40}{17}, -\frac{41}{17})$*, the mid part associated with* $(u^*, v^*) = (-\frac{6}{41}, \frac{20}{123})$*, and the right part associated with* $(u^*, v^*) = (0, 0)$*. Furthermore, the directional derivatives of the RLPVF for Example 1 when* $x_1 = 0$*, are exactly the optimal dual solutions of the associated* (D-RLPVF)*.*

Next, we characterize the structure of RVF. We observe that RVF is the minimum of a finite number of translations of functions of the form (RLPVF) for different values of the fixed RHS parameter β. Each of these translated functions defines a *stability region* over which we can fix the integer part of all solutions defining points on the graph of the RLPVF.

To develop our characterization, let us first define the following sets of integer parts of solutions by projecting $\mathcal{S}(\zeta)$ onto the space of the integer variables: we define

$$\mathcal{S}_I(\zeta) = \text{proj}_I\, \mathcal{S}(\zeta) = \{x_I \in \mathbb{Z}^r_+ : (x_I, x_C) \in \mathcal{S}(\zeta)\}, \text{ and}$$

$$\mathcal{S}_I = \bigcup_{\zeta \in \mathcal{C}} \mathcal{S}_I(\zeta).$$

Then $\mathcal{S}_I$ is the set of all integer parts of points in $\mathcal{S}(\zeta)$ for some $\zeta \in \mathcal{C}$. For a given $\hat{x}_I \in \mathcal{S}_I$, the *continuous restriction* (CR) with respect to $\hat{x}_I$ is the function

$$\bar{z}(\zeta; \hat{x}_I) = c^0_I \hat{x}_I + z_{\text{LP}}(\zeta - C^{1:l}_I \hat{x}_I; b - A_I \hat{x}_I). \tag{CR}$$

This is precisely a translation of a function of the form (RLPVF) for $\beta = b - A_I \hat{x}_I$. We can now provide our first characterization of the RVF as the minimum of a set of such translations.

$$z(\zeta) = \min_{x_I \in \mathcal{S}_I} \bar{z}(\zeta; x_I) = \min_{x_I \in \mathcal{S}_I} c^0_I x_I + z_{\text{LP}}(\zeta - C^{1:l}_I x_I; b - A_I x_I) \quad \forall \zeta \in \mathcal{C}. \tag{RVF-eq}$$

When X_{MO} is bounded, S_I is finite, so the number of such functions required is finite under that assumption.

Proposition 3.3. *For any* $\hat{x}_I \in \mathcal{S}_I$*,* $\bar{z}(\cdot; \hat{x}_I)$ *bounds* z *from above.*

Proof. For $\hat{x}_I \in \mathcal{S}_I$, we have

$$\begin{aligned}\bar{z}(\zeta; \hat{x}_I) = c^0_I \hat{x}_I + \quad & z_{\text{LP}}(\zeta - C^{1:l}_I \hat{x}_I; b - A_I \hat{x}_I) \geq \\ & \min_{x_I \in \mathcal{S}_I} c^0_I x_I + z_{\text{LP}}(\zeta - C^{1:l}_I x_I; b - A_I x_I) = z(\zeta) \quad \forall \zeta \in \mathcal{C}.\end{aligned}$$

■

Proposition 3.3 shows that any collection of points from $\mathcal{S}_I$ yields an upper approximation of z simply by taking the minimum of the associated set of bounding functions. The algorithm described in the next section constructs a subset of $\mathcal{S}_I$ that fully describes z by iteratively approximating it from above. In particular, we make use of the *stability region*, for a given $\hat{x}_I \in \mathcal{S}_I$, denoted by $\mathcal{C}(\hat{x}_I)$, and defined to be the set of ζ values for which the continuous restriction to $\hat{x}_I$ yields the value of the RVF:

$$\mathcal{C}(\hat{x}_I) = \{\zeta \in \mathcal{C} : z(\zeta) = \bar{z}(\zeta; \hat{x}_I)\}.$$

We discuss properties of stability regions shortly. First, we focus on characterizing z. Clearly, elements of $\mathcal{S}_I$ that have an empty stability region are not needed to describe z. The main result of this section is the following discrete characterization, which formally defines the minimal subset of $\mathcal{S}_I$ needed for a characterization.

Theorem 3.4. *Let $\mathcal{S}_{\min}$ be a minimal subset of $\mathcal{S}_I$ with the property that for any $\zeta \in \mathcal{C}$, there exist $x_I \in \mathcal{S}_{\min}$ and $x_C \in \mathbb{R}_+^{n-r}$ such that $C_I^{1:l} x_I + C_C^{1:l} x_C \leq \zeta$, $A_I x_I + A_C x_C = b$, and $c_I^0 x_I + c_C^0 x_C = z(\zeta)$. Then for $\zeta \in \mathcal{C}$ we have*

$$z(\zeta) = \min_{x_I \in \mathcal{S}_I} \bar{z}(\zeta; x_I) = \min_{x_I \in \mathcal{S}_{\min}} \bar{z}(\zeta; x_I).$$

Proof. It is clear from definitions that the value function domain, $\mathcal{C}$, is contained in, and in fact is given by, the union of all stability regions:

$$\bigcup_{x_I \in \mathcal{S}_I} \mathcal{C}(x_I) = \mathcal{C}.$$

$\mathcal{S}_{\min}$ is a subset of $\mathcal{S}_I$ that retains this property, so

$$\bigcup_{x_I \in \mathcal{S}_{\min}} \mathcal{C}(x_I) = \mathcal{C},$$

also, that is minimal, meaning that no element of $\mathcal{S}_{\min}$ may be removed without this property failing. As a consequence, for every $x_I \in \mathcal{S}_{\min}$, there exists $\zeta \in \mathcal{C}$ and x_C with (x_I, x_C) the *unique* point in $\mathcal{S}(\mathcal{C})$ such that $z(\zeta) = c_I^0 x_I + c_C^0 x_C$. ■

The stability region for a given element of $\mathcal{S}_{\min}$ need not be continuous. For example, consider the instance shown in Figure 13, with details expanded in Figures 14b, and 14a. Numerical details for the instance, as well as graphs of its four RLPVFs, are given in Appendix B. This instance has one element of $\mathcal{S}_{\min}$ with a stability region given by the projection onto the horizontal axis of the thick lines shown in red in the figure. This stability region consists of five connected components, one of which is an isolated point and the other four of which are (disjoint) intervals. This also illustrates that a stability region need not be convex, by virtue of not being connected.

In fact, even a connected component of a stability region may not be convex, nor need it be compact. We illustrate this in Example 3.

Example 3. *A stability region also need not be convex, nor need it even be compact. For example, consider the RVF instance given by*

$$\begin{aligned} z(\zeta) = \min \quad & x_2 \\ & x_3 \leq \zeta_1 \\ & x_4 \leq \zeta_2 \\ & 2(1 - x_1) \leq x_j \leq 2(1 - x_1) + 5x_1, \qquad j = 2, 3, 4 \\ & x_2 + x_3 \geq 5x_1 \\ & x_1 \leq 1, \quad x_1 \in \mathbb{Z}_+ \\ & x_j \leq 5, \quad x_j \in \mathbb{R}_+, \qquad j = 2, 3, 4. \end{aligned}$$

The value function for this instance can be written explicitly as

$$z(\zeta) = \begin{cases} 2, & \zeta_1 \in [2,3] \text{ and } \zeta_2 \geq 2 \\ 5 - \zeta_1, & \zeta_1 \leq 5 \text{ and } \zeta_2 < 2 \text{ or } \zeta_1 \in [0,2) \cup (3,5] \\ 0, & \text{otherwise} \end{cases}$$

for $\zeta = (\zeta_1, \zeta_2) \in \mathcal{C} = \mathbb{R}^2_+$. *The stability region for* $x_I = (x_1) = (1)$ *is*

$$\{\zeta \in \mathbb{R}^2_+ \ : \ \zeta_1 < 2 \text{ or } \zeta_1 \geq 3 \text{ or } \zeta_2 < 2\},$$

which is connected but is neither convex nor compact. Another example of the latter is found in Example 4. In this instance, there are four stability regions corresponding to the projection onto the horizontal axis of each of the linear pieces shown in Figure 6, the graph of its value function. Clearly, each of the four is a half-open interval and hence not compact. In Figure 4, we plot the stability regions for the instance in this example and illustrate the properties of the stability regions.

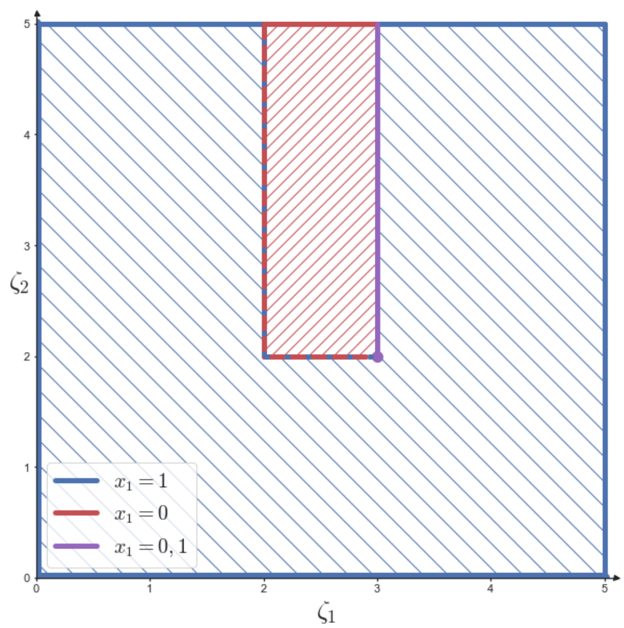


Figure 4: Stability regions of the instance in Example 3.

Note that the instance in Example 3 is not in the standard form as in RVF. We wanted to provide an instance for illustrating the properties of the stability regions, and with the standard form, the construction of the instance is not straightforward. Hence we preferred to put the instance in the format shown in Example 3.

The set $\mathcal{S}_{\min}$ in the above theorem is not unique, since we may have $x_I^1, x_I^2 \in \mathcal{S}_I$ with $x_I^1 \neq x_I^2$, but such that $\bar{z}(\cdot; x_I^1) = \bar{z}(\cdot; x_I^2)$, e.g., in the trivial case of duplicate variables. Nevertheless, the minimal set of CR functions required to fully represent the RVF *is* unique, and we have one such CR function associated with each stability region. As defined in Theorem 3.4, all minimal subsets of $\mathcal{S}_I$ must result in the same set of stability regions, as illustrated in Figure 5.

The RVF may lose continuity due to the integer variables as well as the fixed RHS constraints. This is illustrated in Example 4.

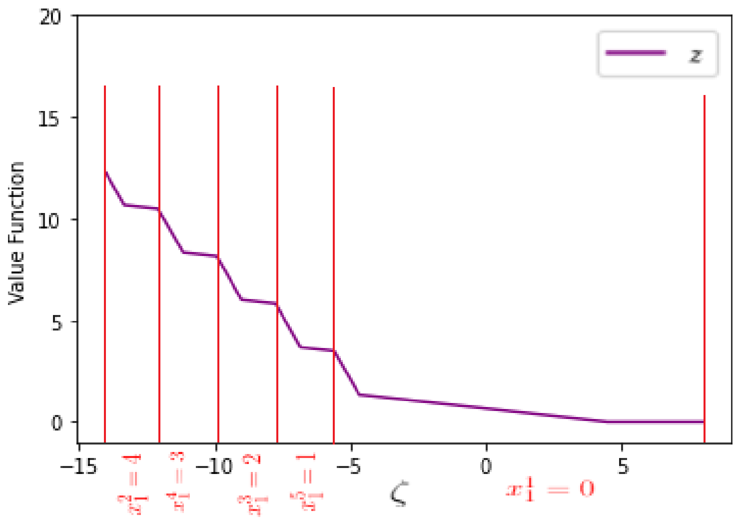


Figure 5: Stability regions and corresponding members of $\mathcal{S}_I$ in Example 1.

Example 4. *Consider the following instance of* (RVF)*:*

$$
\begin{aligned}
z(\zeta) = \min\ & x_1 + \frac{1}{4}x_2 + \frac{1}{2}y_1 - \frac{3}{4}y_2 \\
& \frac{4}{5}x_1 + \frac{1}{2}x_2 + \frac{1}{3}y_1 + 0y_2 \leq \zeta \\
& \frac{3}{5}x_1 + \frac{1}{3}x_2 + \frac{1}{4}y_1 - \frac{1}{5}y_2 = 4 \\
& x_i \in \mathbb{Z}_+ \quad \forall i \in \{1, 2\} \\
& y_j \in \mathbb{R}_+ \quad \forall j \in \{1, 2\}.
\end{aligned}
$$

Figure 6 shows the value function of the MILP in Example 4. The value function is piecewise linear, non-increasing, and lower semi-continuous. By non-increasing, we mean that if $\zeta_1 \geq \zeta_2$ for $\zeta_1, \zeta_2 \in \mathbb{R}^l$, then $z(\zeta_1) \leq z(\zeta_2)$. We formalize this in Proposition 3.5, which can be viewed as a generalization of the proposition by Nemhauser and Wolsey [1988] and Bank et al. [1982] for the full MILP VF.

Proposition 3.5. *The restricted value function* (RVF) *z is a lower semi-continuous, non-increasing, piecewise polyhedral function.*

Proof. By Equation (RVF-eq), the RVF is the minimum of a finite number of non-increasing convex polyhedral functions. It follows that the stability regions are polyhedral and that the RVF itself is piecewise polyhedral. Lower semi-continuity can be proven exactly as in [Nemhauser and Wolsey, 1988]. The non-increasing property follows from the fact that the function is the minimum of non-increasing functions and must therefore be non-increasing itself. ∎

Proposition 3.6. *If z is differentiable at $\hat{\zeta} \in \mathcal{C}(\hat{x}_I)\ \forall \hat{x}_I \in \mathcal{S}_{\min}$ (which is defined in Theorem 3.4) then there exists a unique $\hat{u} \in \mathbb{R}^l_-$ such that $(\hat{u}, v) \in P_{\mathcal{D}}$ and $z(\hat{\zeta}; \beta) = \hat{\zeta}^\top \hat{u} + \beta^\top v$ and the gradient of z at $\hat{\zeta}$ is $\hat{u}$. In the case of non-differentiability, two cases can happen. If $\hat{\zeta} \in \mathcal{C}(\hat{x}_I)$ is a point of non-differentiability of z and*

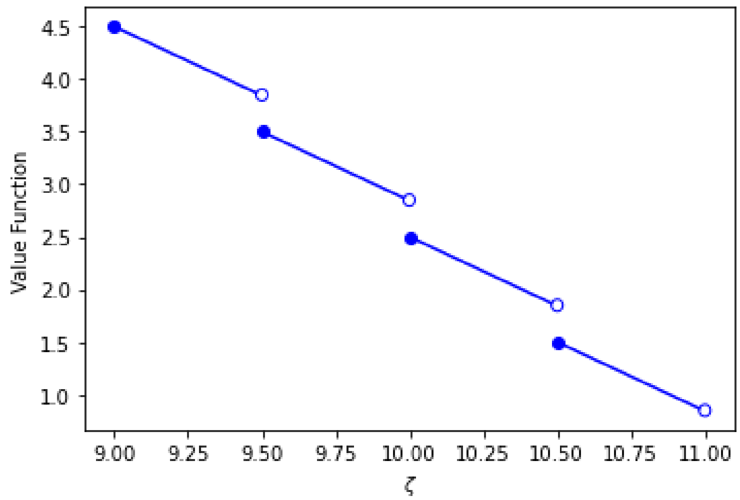


Figure 6: RVF associated with Example 4.

1. *if $\nabla_d z(\hat{\zeta})$ exists for all $d \in \mathbb{R}^l$, then there exist $(u^1, v^1), (u^2, v^2), \ldots, (u^s, v^s) \in P_{\mathcal{D}}$ with $s > 1$ such that $z(\hat{\zeta}; \beta) = \hat{\zeta}^\top u^1 + \beta^\top v^1 = \hat{\zeta}^\top u^2 + \beta^\top v^2 = \ldots = \hat{\zeta}^\top u^s + \beta^\top v^s$.*

2. *if $\nabla_d z(\hat{\zeta})$ does not exist for at least one of the directions $d \in \mathbb{R}^l$, then there do not exist dual variables related to that direction. For all the other directions $d \in \mathbb{R}^l$ that $\nabla_d z(\hat{\zeta})$ exists, there exist $(u^1, v^1), (u^2, v^2), \ldots, (u^s, v^s) \in P_{\mathcal{D}}$ with $s > 1$ such that $z(\hat{\zeta}; \beta) = \hat{\zeta}^\top u^1 + \beta^\top v^1 = \hat{\zeta}^\top u^2 + \beta^\top v^2 = \ldots = \hat{\zeta}^\top u^s + \beta^\top v^s$.*

Proof. From (RVF-eq), MILP VF, i.e., z, is a minimum of some (CR), i.e. $\bar{z}(\zeta; \hat{x}_I)$ for a fixed $\hat{x}_I$. Hence in the interior of a given stability region, (RVF) z coincide with (RLPVF) z_{LP} and the differentiability follows from z_{LP}. For the region that intersects the stability regions, the (RVF) z might be lower semi-continuous and non-differentiable as in case 2 above. ■

Here we illustrate Proposition 3.6 with Figures 5-6. Differentiability in (RVF) z works differently than (RLPVF). In the interior part of the stability regions, these two coincide as shown in Figure 5; we have five stability regions, each corresponding to a specific (RLPVF) with a fixed integer part. Two cases can happen as in (RLPVF). In one case, the (RVF) is differentiable with a unique directional derivative (interior of line segments in Figure 5). In another case, there can be more than one optimal dual solution (in the intersection of the line segments and the interior of the stability region). There could be another type of non-differentiability in (RVF) z in the case of lower semi-continuity, i.e., the intersections of the stability regions as shown in Figure 6.

4 The RVF and the Efficient Frontier

We are now ready to formalize the relationship between the RVF and the efficient frontier with the following theorem. Although a rigorous formulation of this theorem involves some technicalities that obscure intuition, we can informally interpret the theorem as saying that the boundary of the epigraph of the RVF and the efficient frontier effectively encode the same information. The difference is only that the boundary of the epigraph of the RVF may include some "flat" segments (regions over which the RVF has a zero gradient) that are not technically part of the efficient frontier because they consist of weak NDPs, which are *not* nondominated, or because there is no corresponding feasible point, meaning they are not objective space images of points in X_{MO}.

Theorem 4.2 formalizes this relationship. We first prove a lemma helpful to the second case of the theorem.

Lemma 4.1. *For any $\zeta \in \mathcal{C}$, there exists an efficient solution $(x_I, x_C) \in X_{\text{MO}}$ that yields $z(\zeta)$, i.e., for which $(x_I, x_C) \in \mathcal{S}(\zeta)$ and $c_I^0 x_I + c_C^0 x_C = z(\zeta)$.*

Proof. Let $\zeta \in \mathcal{C}$ and suppose that $(\tilde{x}_I, \tilde{x}_C)$ yields $z(\zeta)$, so $(\tilde{x}_I, \tilde{x}_C) \in \mathcal{S}(\zeta)$ and $c_I^0 \tilde{x}_I + c_C^0 \tilde{x}_C = z(\zeta)$. Now by well-known properties of multiobjective optimization either $(\tilde{x}_I, \tilde{x}_C)$ is an efficient solution or there exists $(x_I, x_C) \in X_{\text{MO}}$ an efficient solution that at least weakly dominates $(\tilde{x}_I, \tilde{x}_C)$, i.e., an efficient solution with $C_I x_I + C_C x_C \leq C_I \tilde{x}_I + C_C \tilde{x}_C$ must exist. Then $C_I^{1:\ell} x_I + C_C^{1:\ell} x_C \leq C_I^{1:\ell} \tilde{x}_I + C_C^{1:\ell} \tilde{x}_C \leq \zeta$, so $(x_I, x_C) \in \mathcal{S}(\zeta)$. Furthermore $c_I^0 x_I + c_C^0 x_C \leq c_I^0 \tilde{x}_I + c_C^0 \tilde{x}_C = z(\zeta)$, so in fact, (x_I, x_C) must also yield $z(\zeta)$. Since (x_I, x_C) is efficient, the result follows. ■

We now give the main theorem.

Theorem 4.2. *The efficient frontier associated with (MO-MILP) is a (possibly strict) subset of the boundary of the epigraph of the RVF z. In particular, the following statements hold for X_{MO} and the (RVF) z.*

1. *If $(x_I, x_C) \in X_{\text{MO}}$ is an efficient solution (equivalently, $C_I x_I + C_C x_C$ is an NDP), then $(\zeta, z(\zeta))$ is a point on the boundary of the epigraph of z for $\zeta = C_I^{1:l} x_I + C_C^{1:l} x_C$.*
2. *If $(\zeta, z(\zeta))$ is a point on the boundary of the epigraph of z, then $\nexists (x_I, x_C) \in X_{\text{MO}}$ for which $(\zeta, z(\zeta)) > C_I x_I + C_C x_C$. Furthermore, there are two cases, as follows, where $\mathcal{D}(\zeta)$ denotes the set of directions $d \in \mathbb{R}^l_-$, $d \neq 0$, for which the directional derivative $\nabla_d z(\zeta)$ exists.*
   (a) If for all $d \in \mathcal{D}(\zeta)$ the directional derivative is positive, so $\nabla_d z(\zeta) > 0$, then there exists an efficient solution $(x_I, x_C) \in X_{\text{MO}}$ that yields $z(\zeta)$ and satisfies $C_I^{1:l} x_I + C_C^{1:l} x_C = \zeta$.
   (b) If for some $d \in \mathcal{D}(\zeta)$ the directional derivative is zero, so $\nabla_d z(\zeta) = 0$, then there exists an efficient solution $(x_I, x_C) \in X_{\text{MO}}$ that yields $z(\zeta)$ and satisfies $C_I^{1:l} x_I + C_C^{1:l} x_C \lneqq \zeta$.

Proof.

1. To prove statement 1, let $(\hat{x}_I, \hat{x}_C) \in X_{\text{MO}}$ be a given efficient solution and let $\hat{\zeta} = C_I^{1:l} \hat{x}_I + C_C^{1:l} \hat{x}_C$. Now $(\hat{x}_I, \hat{x}_C) \in X_{\text{MO}}$ and $\hat{\zeta} = C_I^{1:l} \hat{x}_I + C_C^{1:l} \hat{x}_C$ implies, from definitions, that $(\hat{x}_I, \hat{x}_C) \in \mathcal{S}(\hat{\zeta})$. We want to show that $(\hat{\zeta}, c_I^0 \hat{x}_I + c_C^0 \hat{x}_C)$ is a point on the boundary of the epigraph of the RVF z. Since $z(\hat{\zeta}) = \min_{(x_I, x_C) \in \mathcal{S}(\hat{\zeta})} c_I^0 x_I + c_C^0 x_C$, by definition, and since $(\hat{x}_I, \hat{x}_C) \in \mathcal{S}(\hat{\zeta})$, it must be that $z(\hat{\zeta}) \leq c_I^0 \hat{x}_I + c_C^0 \hat{x}_C$. Assume, for the sake of contradiction, that $c_I^0 \hat{x}_I + c_C^0 \hat{x}_C \neq z(\hat{\zeta})$. Then it must be that $c_I^0 \hat{x}_I + c_C^0 \hat{x}_C > z(\hat{\zeta})$. Now, by the definition of $z(\hat{\zeta})$, there must exist $(x_I, x_C) \in \mathcal{S}(\hat{\zeta})$ with $z(\hat{\zeta}) = c_I^0 x_I + c_C^0 x_C$. Furthermore, $(x_I, x_C) \in \mathcal{S}(\hat{\zeta})$ implies $C_I^{1:l} x_I + C_C^{1:l} x_C \leq \hat{\zeta} = C_I^{1:l} \hat{x}_I + C_C^{1:l} \hat{x}_C$, while $c_I^0 x_I + c_C^0 x_C = z(\hat{\zeta}) < c_I^0 \hat{x}_I + c_C^0 \hat{x}_C$, so $C_I x_I + C_C x_C \lneqq C_I \hat{x}_I + C_C \hat{x}_C$. This contradicts the hypothesis that $(\hat{x}_I, \hat{x}_C)$ is an efficient solution, and the result follows.

2. To prove part 2 we begin by considering $(\hat{\zeta}, z(\hat{\zeta}))$ for some $\hat{\zeta} \in \mathcal{C}$, and suppose, for the sake of contradiction, that there does exist $(x_I, x_C) \in X_{\text{MO}}$ with $(\hat{\zeta}, z(\hat{\zeta})) > C_I x_I + C_C x_C$, so $\hat{\zeta} > C_I^{1:l} x_I + C_C^{1:l} x_C$ and $z(\hat{\zeta}) > c_I^0 x_I + c_C^0 x_C$. Since $C_I^{1:l} x_I + C_C^{1:l} x_C < \hat{\zeta}$, it must be that $(x_I, x_C) \in \mathcal{S}(\hat{\zeta})$. Thus by the definition of $z(\hat{\zeta})$, it follows that $z(\hat{\zeta}) \leq c_I^0 x_I + c_C^0 x_C$, giving a contradiction, and the result follows.

   We now prove the subparts. First, observe that, by the definition of $z(\hat{\zeta})$, there must exist $(\hat{x}_I, \hat{x}_C) \in \mathcal{S}(\hat{\zeta})$, so $(\hat{x}_I, \hat{x}_C) \in X_{\text{MO}}$ with $\hat{\zeta} \geq C_I^{1:l} \hat{x}_I + C_C^{1:l} \hat{x}_C$, and $z(\hat{\zeta}) = c_I^0 \hat{x}_I + c_C^0 \hat{x}_C$. Second, by Lemma 4.1, we may safely require that $(\hat{x}_I, \hat{x}_C)$ is an efficient solution.

   (a) To prove statement 2a, suppose that $\nabla_d z(\hat{\zeta}) > 0$ for all $d \in \mathcal{D}(\hat{\zeta})$. Since $(\hat{x}_I, \hat{x}_C)$ is an efficient solution, it only remains to show that we must in fact have that $\hat{\zeta} = C_I^{1:l} \hat{x}_I + C_C^{1:l} \hat{x}_C$. Assume instead that $\hat{\zeta} \gneqq C_I^{1:l} \hat{x}_I + C_C^{1:l} \hat{x}_C$ and let $\tilde{\zeta} = C_I^{1:l} \hat{x}_I + C_C^{1:l} \hat{x}_C$. Now $\mathcal{S}(\tilde{\zeta}) \subseteq \mathcal{S}(\zeta) \subseteq \mathcal{S}(\hat{\zeta})$ for all ζ with $\tilde{\zeta} \leq \zeta \leq \hat{\zeta}$. Thus, since $(\hat{x}_I, \hat{x}_C)$ minimizes the objective over $\mathcal{S}(\hat{\zeta})$ and $(\hat{x}_I, \hat{x}_C) \in \mathcal{S}(\tilde{\zeta})$, it must be that $z(\zeta) = c_I^0 \hat{x}_I + c_C^0 \hat{x}_C$ for all $\zeta \in [\tilde{\zeta}, \hat{\zeta}]$. So $z(\zeta) = z(\hat{\zeta})$ for all $\zeta \in [\tilde{\zeta}, \hat{\zeta}]$. Recalling that $\hat{\zeta} \gneqq \tilde{\zeta}$, it must thus be that $\hat{d} = \tilde{\zeta} - \hat{\zeta} \in \mathcal{D}(\hat{\zeta})$ and $\nabla_{\hat{d}} z(\hat{\zeta}) = 0$, which is a contradiction to the initial hypothesis. Therefore, $\hat{\zeta} = C_I^{1:l} \hat{x}_I + C_C^{1:l} \hat{x}_C$.

   (b) Finally, to prove statement 2b, suppose there exists $d \in \mathcal{D}(\hat{\zeta})$ such that $\nabla_d z(\hat{\zeta}) = 0$. Then, by Proposition 3.5, there must exist $\tilde{\zeta} \lneqq \hat{\zeta}$ with $z(\tilde{\zeta}) = z(\hat{\zeta})$. Now suppose $(\tilde{x}_I, \tilde{x}_C)$ is an efficient solution that yields $z(\tilde{\zeta})$, noting that one must exist by Lemma 4.1. So $(\tilde{x}_I, \tilde{x}_C) \in \mathcal{S}(\tilde{\zeta})$ and $c_I^0 \tilde{x}_I + c_C^0 \tilde{x}_C = z(\tilde{\zeta})$. Now the former condition implies $C_I^{1:\ell} \tilde{x}_I + C_C^{1:\ell} \tilde{x}_C \leq \tilde{\zeta} \lneqq \hat{\zeta}$ while the latter implies that $(\tilde{x}_I, \tilde{x}_C)$ yields $z(\hat{\zeta})$, since $z(\tilde{\zeta}) = z(\hat{\zeta})$. The result follows.

■

The formal statement of Theorem 4.2 makes clear that when all directional derivatives in the direction $d \in \mathbb{R}^l_-$ are strictly positive (the function is strictly decreasing everywhere), then the boundary of the epigraph of the RVF and the efficient frontier exactly coincide. The only situation in which they don't coincide is when there are directional derivatives that are zero. These conditions involving directional derivatives also have another interpretation that is possibly more intuitive. Recall from the previous section that when $\nabla_d z(\hat{\zeta})$ exists, we have that $\nabla_d z(\hat{\zeta}) = d^\top u$ for some alternative optimal dual solution to (D-RLPVF). This allows us to re-interpret the above conditions involving directional derivatives in terms of solutions to (D-RLPVF). In particular, the condition $\nabla_d z(\hat{\zeta}) > 0$ for all $d \in \mathbb{R}^l_-$ is equivalent to $u < 0$ for all alternative optimal solutions $(u, v) \in P_{\mathcal{D}}$ for (D-RLPVF) associated with $\hat{\zeta}$, while a zero directional derivative implies that the dual variable associated with one of the constraints is zero. This makes sense, as a zero dual value implies that the constraint can be tightened without changing the optimal solution, and this is exactly the condition that would indicate a given solution is not non-dominated in the multiobjective context.

These two cases are illustrated using Examples 1 from earlier and in 5 below. Example 1 considered an instance for which the efficient frontier exactly coincides with the boundary of the epigraph of the associated RVF, while Example 5 considers an instance for which the efficient frontier is a strict subset of the boundary of the epigraph of the associated RVF, which includes weak NDPs.

In Figure 7, we show the efficient frontier and the epigraph of the RVF associated with the Example 1. Indeed, the efficient frontier is the boundary of the epigraph of the associated RVF since, in this case, there are no weak NDPs. Example 5 shows a situation in which the RVF and

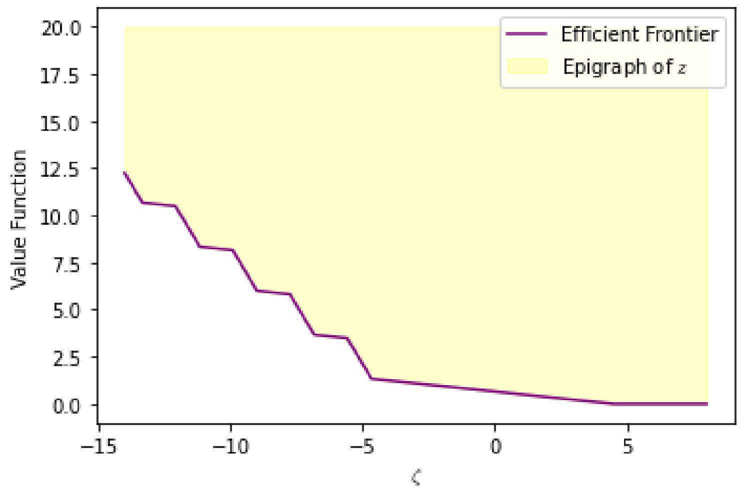


Figure 7: Efficient frontier and epigraph of the value function for Example 1.

the efficient frontier do not coincide.

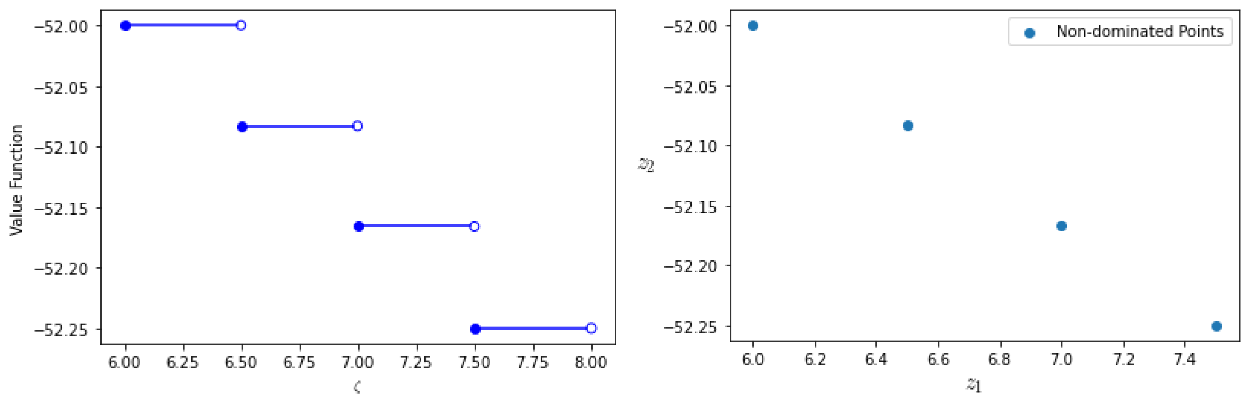
Example 5. *We consider the following instance of* (RVF)*:*

$$
\begin{aligned}
z(\zeta) = \inf\ & x_1 + \frac{1}{4}x_2 + \frac{1}{4}y_1 - \frac{3}{4}y_2 \\
& \frac{4}{5}x_1 + \frac{1}{2}x_2 + 0y_1 + 0y_2 \leq \zeta \\
& \frac{3}{5}x_1 + \frac{1}{3}x_2 + \frac{1}{4}y_1 - \frac{1}{5}y_2 = 4 \\
& x_i \in \mathbb{Z}_+ \quad \forall i \in \{1, 2\} \\
& y_j \in \mathbb{R}_+ \quad \forall j \in \{1, 2\}.
\end{aligned}
$$

Figure 8 illustrates that the efficient frontier of the multiobjective optimization problem associated with the RVF in Example 5 is a strict subset of the boundary of the epigraph of the RVF due to the presence of weak NDPs.

Note that it is possible to avoid the difficulty of the additional “flat pieces” of the RVF that aren’t part of the efficient frontier by changing from “$\leq$” to equality for the constraints associated with the objectives of the multiobjective version of the problem. However, in that case, a different difficulty is introduced—there may then be parts of the RVF that are *increasing* (strictly positive directional derivative in the direction $d \in \mathbb{R}^l_+$), and we then have that the boundary of the epigraph for *those* parts of the RVF is not part of the frontier. This approach does not make the statement of the theorem any cleaner.

It is worth mentioning that in the multiobjective optimization area, there is a term called *slice problem* [Belotti et al., 2013], which is defined by fixing the integer part of the solutions and



(a) Value function

(b) Efficient frontier

Figure 8: Value Function and efficient frontier of Example 5.

is equivalent to (RLPVF) in the value function area. There is also another term *slice*, which is equivalent to the stability region in the value function area.

Before closing this section, we have the following corollary to Theorem 3.4, which connects the representation of the RVF and the efficient frontier based on the result of Theorem 4.2.

Corollary 4.3. *Let $\mathcal{S}_{\min}$ be a minimal subset of $\mathcal{S}_I$, as described in Theorem 3.4. Then $\gamma \in \mathcal{F}$ if and only if $\exists x_I \in \mathcal{S}_{\min}, x_C \in \mathbb{R}^{n-r}_+$ such that $(x_I, x_C) \in X_{\mathrm{MO}}$ and $\gamma = C_I x_I + C_C x_C$ is an NDP.*

This corollary tells us that not only does a minimal subset $\mathcal{S}_{\min}$ give us a representation of the value function but also a representation of the efficient frontier. The stability regions equivalently represent regions of the efficient frontier over which the contribution $C_I x_I$ to the objective function from the integer part of solutions is a constant. Note that this representation gives us the ability to easily evaluate the RVF at any point in its domain by solving a sequence of LPs.

5 Finite Algorithm for Construction

In this section, we present our algorithm for constructing a discrete representation of both the RVF and the efficient frontier introduced in the previous section. We start with a discussion of the algorithm in the general case before discussing the pure integer case, which has particular properties.

5.1 General (Mixed Integer) Case

The goal is to construct a minimal subset $\mathcal{S}_{\min}$ of $\mathcal{S}_I$, the elements of which are in one-to-one correspondence with the stability regions of the RVF, as described in Theorem 3.4. The proposed algorithm is a generalized cutting-plane algorithm that iteratively improves an upper approximation until convergence to the true function. The "cuts" here are the convex bounding functions described in (RLPVF). The upper approximation in iteration k is given by

$$\bar{z}^k(\zeta) = \min\{\min_{x_I \in S^{k-1}} \bar{z}(\zeta; x_I), \bar{z}^0(\zeta)\} \quad \forall \zeta \in \mathcal{C}, \tag{2}$$

where $\mathcal{S}^{k-1}$ is the set of points in $\mathcal{S}_I$ identified so far. We set $\bar{z}^0(\zeta) = U$, where U is a global upper bound on the value function needed to ensure that the subproblem solved to discover new points remains bounded (recall that such a bound exists, since X_{MO} was assumed to be bounded). The most straightforward way of calculating such an upper bound is by solving the following optimization problem, although in some special cases, stronger bounds can be obtained by exploiting the problem structure.

$$U = \max_{(x_I, x_C) \in X_{\text{MO}}} c_I^0 x_I + c_C^0 x_C. \tag{UB$_z$}$$

Consider the optimal solution to (UB$_z$) as (x_I^*, x_C^*). Since it is not necessarily an efficient solution, we convert it to an efficient one with the method discussed in (11) and call it (x_I^0, x_C^0). We add the integer part (x_I^0) to our set $\mathcal{S}^0$.

The algorithm proceeds by identifying in iteration k the value $\zeta^* \in \mathcal{C}$ that maximizes the difference between the approximation $\bar{z}^k(\zeta^*)$ and the true value function $z(\zeta^*)$. This yields a new stability region associated with a new member of $\mathcal{S}_I$, which we add to obtain $\mathcal{S}^{k+1}$. We iterate until the approximation is exact. A high-level description of the algorithm is shown below. A few

---

RVF Algorithm : Algorithm for constructing the RVF and the associated efficient frontier

---

Input: X_{MO}, $C \in \mathbb{Q}^{(l+1)\times n}$.

Output: $\mathcal{S}^k (= \mathcal{S}_{\min})$ such that $z(\zeta) = \bar{z}^k(\zeta) = \min\limits_{x_I \in \mathcal{S}^k} \bar{z}(\zeta; x_I) \quad \forall \zeta \in \mathcal{C}$.

1 $\bar{z}^0(\zeta) = U$ for all $\zeta \in \mathbb{R}^l$, $k = 0$, $\mathcal{S}^0 = x_I^0$, $\theta^0 = \infty$.

while $\theta^k > 0$ **do**

2 Determine $(x_I^{k+1}, x_C^{k+1}) \in \underset{(x_I, x_C) \in X_{\text{MO}}}{\arg\max} \; \bar{z}^k \left(C_I^{1:l} x_I + C_C^{1:l} x_C\right) - c_I^0 x_I - c_C^0 x_C$.

Convert x_I^{k+1} associated with weakly NDP to integer parts associated with NDP using optimization problem (11) (if applicable).

Set $\mathcal{S}^{k+1} \leftarrow \mathcal{S}^k \cup \{x_I^{k+1}\}$.

Set $\theta^{k+1} = \bar{z}^k \left(C_I^{1:l} x_I^{k+1} + C_C^{1:l} x_C^{k+1}\right) - c_I^0 x_I^{k+1} - c_C^0 x_C^{k+1}$.

$\bar{z}^{k+1}(\zeta) = \min\{\bar{z}^k(\zeta), \bar{z}(\zeta; x_I^{k+1})\}$ for all $\zeta \in \mathcal{C}$.

$k \leftarrow k + 1$.

3 **end**

---

notes of explanation are in order. The RVF Algorithm produces a sequence of points in X_{MO}, but only the integer parts are stored, as this is all that is needed to construct $\mathcal{S}_{\min}$. The problem solved in Step 2 can be formulated as a mixed integer nonlinear optimization problem (MINLP) as follows. First, we write the optimization problem in Step 2 as a mathematical optimization problem involving $\bar{z}^k$.

$$\begin{aligned} \theta^k = \max \quad & \bar{z}^k(\zeta) - c_I^0 x_I - c_C^0 x_C \\ \text{subject to} \quad & C_I^{1:l} x_I + C_C^{1:l} x_C \leq \zeta \\ & (x_I, x_C) \in X_{\text{MO}} \\ & \zeta \in \mathbb{R}^l. \end{aligned}$$

We next introduce an auxiliary variable θ to move $\bar{z}$ into the constraints. Also, since $\bar{z}^k$ is non-increasing, smaller values for ζ result in larger values for $\bar{z}^k(\zeta)$. Since ζ is otherwise unconstrained, we can assume $\zeta = C_I^{1:l} x_I + C_C^{1:l} x_C$ and substitute out ζ (this is already implicitly done in the RVF Algorithm).

$$\begin{aligned} \theta^k = \max \quad & \theta \\ \text{subject to} \quad & \theta \leq \bar{z}^k(C_I^{1:l} x_I + C_C^{1:l} x_C) - c_I^0 x_I - c_C^0 x_C \\ & (x_I, x_C) \in X_{\text{MO}}. \end{aligned} \tag{3}$$

Next, we have that

$$\bar{z}^k(\zeta) = \min\{\min_{i=0,1,\dots,k-1} \bar{z}(\zeta; x_I^i), U\} \quad \forall \zeta \in \mathcal{C},$$

where U can be constructed in the same way as discussed in (UB_z). Therefore for the first constraint in (3), we have

$$\theta + c_I^0 x_I + c_C^0 x_C \le c_I^0 x_I^i + z_{\mathrm{LP}}(C_I^{1:l} x_I + C_C^{1:l} x_C - C_I^{1:l} x_I^i; b - A_I x_I^i) \quad i = 0, 1, \dots, k-1.$$

We can write the term involving z_{LP} as

$$\begin{aligned} &z_{\mathrm{LP}}(C_I^{1:l} x_I + C_C^{1:l} x_C - C_I^{1:l} x_I^i; b - A_I x_I^i) = \\ &\max_{(u^i, v^i) \in P_{\mathcal{D}}} (C_I^{1:l} x_I + C_C^{1:l} x_C - C_I^{1:l} x_I^i)^\top u^i + (b - A_I x_I^i)^\top v^i, \end{aligned} \tag{4}$$

by (D-RLPVF).

Then we need to cap the term $\theta + c_I^0 x_I + c_C^0 x_C$ by U, and add this constraint to the subproblem to be solved in each iteration. Finally, we have the overall explicit formulation as

$$\theta^k = \max \quad \theta \tag{5}$$

$$\text{subject to} \quad \theta + c_I^0 x_I + c_C^0 x_C \le c_I^0 x_I^i +$$

$$(C_I^{1:l} x_I + C_C^{1:l} x_C - C_I^{1:l} x_I^i)^\top u^i + (b - A_I x_I^i)^\top v^i \quad i = 0, 1, \dots, k-1 \tag{6}$$

$$\theta + c_I^0 x_I + c_C^0 x_C \le U \tag{7}$$

$$(u^i, v^i) \in P_{\mathcal{D}} \quad i = 0, 1, \dots, k-1 \tag{8}$$

$$(x_I, x_C) \in X_{\mathrm{MO}} \tag{9}$$

$$\theta \in \mathbb{R}. \tag{10}$$

Here, $z_{\mathrm{LP}}(C_I^{1:l} x_I + C_C^{1:l} x_C - C_I^{1:l} x_I^i; b - A_I x_I^i)$ is guaranteed to take value equal to the maximum of terms on the RHS of (4) because the objective is to maximizes θ on the left-hand side of the constraint (6). This means the RHS of the constraint (6) must take its maximum value. Note that the constraint (6) makes the problem nonlinear. However, the problem (5)–(10) can be solved either with an off-the-shelf non-convex quadratic solver or possibly with a customized algorithm.

As mentioned in Theorem 4.2, the RVF Algorithm might produce some integer parts associated with weakly NDPs. Hence we need to design a procedure within the RVF Algorithm to convert those integer parts associated with weakly NDPs to integer parts associated with NDPs. The integer part obtained at iteration k is associated with a NDP, if $u^k < \mathbf{0}$, i.e., $u_j^k < 0$ for all $j = \{1, 2, \dots, l\}$ (in the case of the RVF z is differentiable at ζ) or the alternative optimal dual vectors are strictly less than the zero vector (in the case of the RVF z is not differentiable at ζ). The integer part that is found in iteration k is associated with either a weakly NDP or an NDP, if $u_j^k = 0$ for some $j \in \{1, 2, \dots, l\}$ (in the case of the RVF z is differentiable at ζ) or at least one of the alternative optimal dual vectors has a component zero (in the case of the RVF z is not differentiable at ζ). In the latter case, a second optimization problem has to be solved. We solve the following optimization problem

$$\min_{(x_I, x_C) \in X_{\mathrm{MO}}} \{\mathbf{1}^\top (C_I x_I + C_C x_C) : (C_I x_I + C_C x_C) \le (C_I x_I^k + C_C x_C^k)\}, \tag{11}$$

to convert the integer parts associated with weakly NDP to integer parts associated with NDP. Note that x_C^k is the continuous part of the solution in decision space related to iteration k.

Here, with Theorem 5.1, we show the correctness of the RVF Algorithm by proving that the RVF Algorithm terminates finitely and returns the correct value function.

Theorem 5.1. *(Correctness of the Algorithm) At termination, we have that $z(\zeta) = \bar{z}^k(\zeta) \quad \forall \zeta \in \mathcal{C}$ and that $\mathcal{S}^k$ is a minimal set describing both the value function and the efficient frontier, as in Theorem 3.4. Furthermore, the RVF Algorithm terminates in finitely many iterations under the assumption that X_{MO} is bounded.*

Proof. When X_{MO} is bounded, $\mathcal{S}_I$ is finite. Since each iteration of the algorithm produces a new member of $\mathcal{S}_I$, the number of iterations must be finite. To show that $z = \bar{z}^k$ at termination, assume not for the sake of contradiction. Then there must exist $\zeta \in \mathbb{R}^l$ such that $z(\zeta) \neq \bar{z}^k(\zeta)$. But this is a contradiction to $\theta^k = 0$, so we must have $z(\zeta) = \bar{z}^k(\zeta)\ \forall \zeta \in \mathcal{C}$. ■

To illustrate, in Example 6, we show the steps of the RVF Algorithm applied to Example 1.

Example 6. *The steps are shown graphically in Figure 9 below. There, we see that the RVF Algorithm starts with the tuple $(\zeta, z(\zeta)) = (-14, 12.24)$, and updates the upper approximation to the blue convex function in iteration 1. Then in iteration 2, the model finds the point with the largest difference between the current upper approximation and the VF (the red piecewise linear function), and then the RVF Algorithm updates the upper approximation again and in the subsequent iteration, finds the points with the most difference between the upper approximation and the VF. The RVF Algorithm continues this procedure until there is no such point. Therefore, the RVF Algorithm terminates in iteration 5 with $\theta = 0 \quad \forall \zeta \in \mathcal{C}$ when the upper approximation and the VF are the same. Note that in the pseudocode of the RVF Algorithm we have $\bar{z}^0(\zeta) = U$ implicitly, but we have $\bar{z}^0_{approx}$ for the approximation in step 0 in the top left in Figure 9 below since we updated the approximation with the x^0_I.*

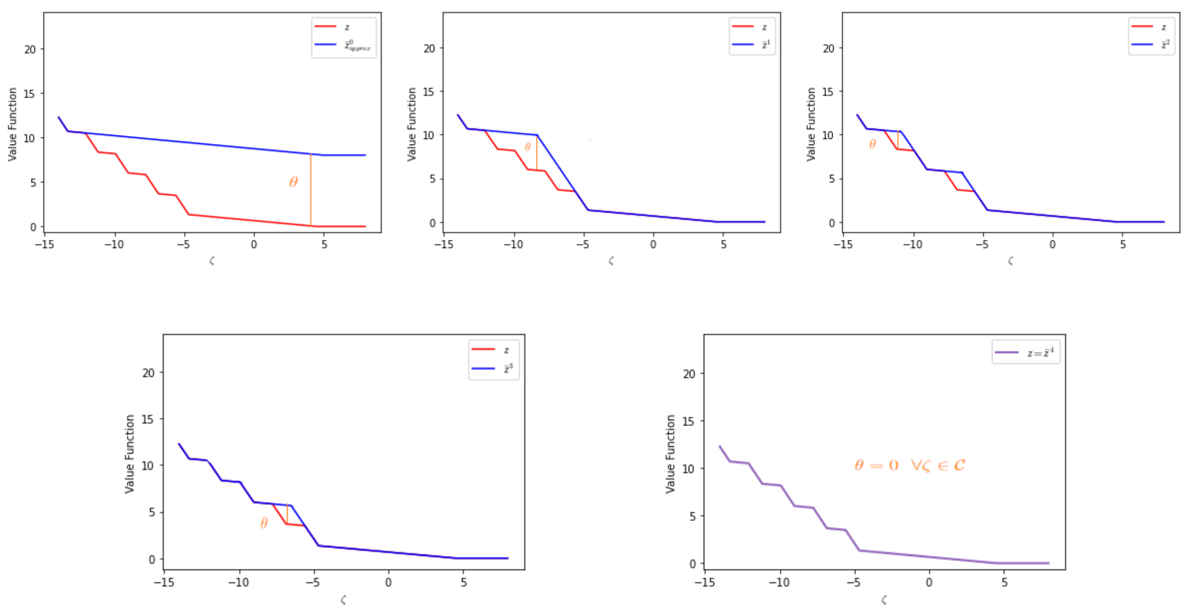


Figure 9: VF and upper approximation in iterations of the RVF Algorithm with Example 1.

With Example 7, we complete this section by showing how the normalized approximation gap decreases during the iterations.

Example 7. *Delorme et al. [2010] defines the biobjective set packing problem and provides a set of benchmark instances. We solve the instance* $2mis100\_300D$ *with the RVF Algorithm, and in Figure 10, we show the normalized gap* θ^k/θ^1 *versus the iteration. The instance file is provided in [BOSPP-benchmarks].*

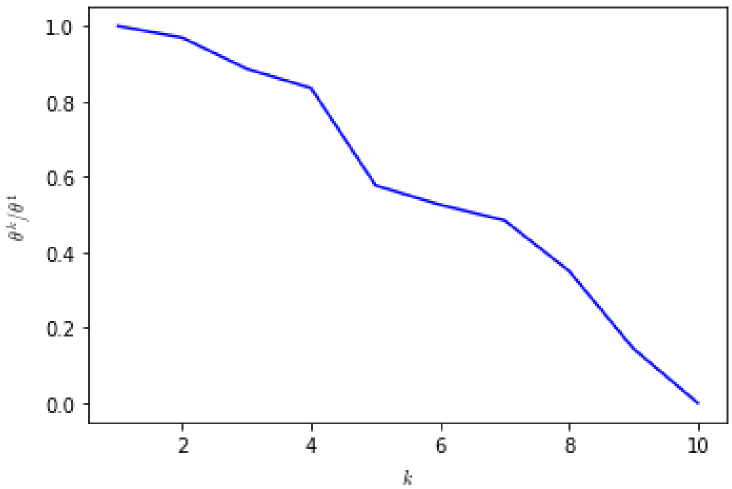


Figure 10: Normalized approximation gap vs. iteration number.

5.2 Pure Integer Case

The pure integer case, in which the problem has no continuous variables, is useful for illustrating connections to existing algorithms and providing insight into algorithm mechanisms. Here, we focus on the latter, leaving the former to Section 5.3.

In the pure integer case, the algorithm can be simplified substantially, as there is no continuous part. In this case, the primal LP parameterized by ζ and formed by fixing the integer variables to $\hat{x}_I$, for which $P_{\mathcal{D}}$ forms the dual LP feasible set, has no variables but can be thought of as the problem

$$z_{\text{LP}}(\zeta; \hat{x}_I) = \min\{0 \ : \ 0 \leq \zeta - C_I^{1:l}\hat{x}_I\}.$$

From strong duality, we have that

$$\begin{aligned} z_{\text{LP}}(\zeta; \hat{x}_I) &:= \max\{(\zeta - C_I^{1:l}\hat{x}_I)^T u \ : \ u \in \mathbb{R}^l_-\} \\ &= \begin{cases} 0 & \text{if } \zeta \geq C_I^{1:l}\hat{x}_I, \\ +\infty & \text{otherwise.} \end{cases} \end{aligned} \tag{12}$$

We may now rewrite problem (5)–(10) for the pure integer case explicitly as follows.

$$\theta^k = \max \quad \theta \tag{13}$$

$$\text{subject to} \quad \theta + c_I^0 x_I \leq c_I^0 x_I^i + (\zeta - C_I^{1:l}x_I^i)^\top u^i \quad i = 0, 1, \dots, k-1 \tag{14}$$

$$\zeta = C_I^{1:l} x_I \tag{15}$$

$$\theta + c_I^0 x_I \leq U \tag{16}$$

$$u^i \in \mathbb{R}^l_- \quad i = 0, 1, \ldots, k-1 \tag{17}$$
$$A_I x_I = b \tag{18}$$
$$x_I \in \mathbb{Z}^r_+ \tag{19}$$
$$\theta \in \mathbb{R}. \tag{20}$$

After making use of (12) to eliminate the u^i variables and (15) to eliminate ζ, we have

$$\theta^k = \max \quad \theta \tag{21}$$
$$\text{subject to} \quad \theta + c^0_I x_I \leq c^0_I x^i_I \quad i = 0, 1, \ldots, k-1, \text{ s.t. } C^{1:l}_I x_I \geq C^{1:l}_I x^i_I \tag{22}$$
$$(16),\ (18),\ (19) \text{ and } (20). \tag{23}$$

Note that the above formulation includes conditional constraints (22), and so cannot be directly solved by a typical MILP or MINLP solver. We include it to facilitate proofs of algorithm properties in what follows.

Note that in the pure integer case, the RVF is a step function (when it exists, the gradient is always zero). Because of this special structure, it can be observed that the RVF Algorithm generates all NDPs of the efficient frontier in non-decreasing (non-increasing) order of their minimum (maximum) $c^0_I x_I$ value. We formalize this in Theorem 5.5. We also linearize the problem to be solved in each iteration of the pure integer case, i.e., (21)–(23), in Appendix A.

The following result states that unless the optimal value of (13)–(20) is zero, the objective space image of its solution is new, meaning that it cannot already be in the objective space image of points in $\mathcal{S}^k$. Here we define the set of objective space images of points in $\mathcal{S}^k$ by

$$C(\mathcal{S}^k) = \{C_I x^i_I \ : \ i = 0, 1, \ldots, k-1\}. \tag{24}$$

Lemma 5.2. *Any optimal solution (x_I, θ) to (13)–(20) (equivalently (21)–(23)) with $\theta > 0$ has the property that $C_I x_I \notin C(\mathcal{S}^k)$.*

Proof. If $C_I x_I \in C(\mathcal{S}^k)$ then there exists $i \in \{0, 1, \ldots, k-1\}$ such that $C_I x_I = C_I x^i_I$. This means that $C^{1:l}_I x_I = C^{1:l}_I x^i_I$, so constraint (22) applies for this i, and $c^0_I x_I = c^0_I x^i_I$. But constraint (22) says $\theta + c^0_I x_I \leq c^0_I x^i_I$ so with $c^0_I x^i_I = c^0_I x_I$ it must be that $\theta \leq 0$. The results follow by contradiction. ■

We now consider the set of all NDPs of the (pure integer) multiobjective problem as

$$\text{vmin}\{C_I x_I : A_I x_I = b, x_I \in \mathbb{Z}^n_+\}.$$

Let $\mathcal{E}$ denote the set of all efficient solutions for the problem.

In proving the following result, we assume that any solution in the set $\mathcal{S}^k$ induces a (strong) NDP of the multiobjective problem. Here we assume that after solving the problem (13)–(20) to yield solution $\hat{x}_I$, an auxiliary problem, such as $\min\{\mathbf{1}^T C_I x_I \ : \ C_I x_I \leq C_I \hat{x}_I,\ A_I x_I = b, x_I \in \mathbb{Z}^n_+\}$ is solved to find a strongly efficient solution, x^k_I, which also solves the problem (13)–(20), to add to $\mathcal{S}^k$ to create $\mathcal{S}^{k+1}$. Here we verify this directly.

Lemma 5.3. *Let $(\hat{x}_I, \theta)$ solve (13)–(20) (equivalently (21)–(23)) and let y solve*

$$\min\{\mathbf{1}^T C_I x_I \ : \ C_I x_I \leq C_I \hat{x}_I,\ A_I x_I = b, x_I \in \mathbb{Z}^n_+\},$$

then $y \in \mathcal{E}$ and (y, θ) is also a solution of (13)–(20) (equivalently (21)–(23)).

Proof. That $y \in \mathcal{E}$ follows from well-known multiobjective optimization principles. Now if, for some $i \in \{0, 1, \ldots, k-1\}$, we have that $C_I^{1:l} y \geq C_I^{1:l} x_I^i$ then $C_I^{1:l} \hat{x}_I \geq C_I^{1:l} x_I^i$ also, since $C_I y \leq C_I \hat{x}_I$ by definition of y. Thus by (22), and since $c_I^0 y \leq c_I^0 \hat{x}_I$ by the definition of y, it must be that

$$\theta + c_I^0 y \leq \theta + c_I^0 \hat{x}_I \leq c_I^0 x_I^i.$$

Thus (y, θ) satisfies (22). Similarly,

$$\theta + c_I^0 y \leq \theta + c_I^0 \hat{x}_I \leq U,$$

so (y, θ) satisfies (16), and hence (23), and the result follows. ■

Thus we may assume that whenever we solve (13)–(20), the solution we return is in $\mathcal{E}$.

Proposition 5.4. *Suppose in the RVF Algorithm at iteration k, the elements of $\mathcal{S}^k$ are efficient solutions with the property that*

$$\min\{c_I^0 y \ : \ y \in \mathcal{E}, C_I y \notin C(\mathcal{S}^k)\} \geq \max\{c_I^0 y \ : \ y \in \mathcal{S}^k\}, \tag{25}$$

in other words, the points in $\mathcal{S}^k$ induce NDPs, which have the least value of the first objective. Then optimal solution (x_I, θ) to (13)–(20) having $x_I \in \mathcal{E}$ and $\theta > 0$ satisfies $c_I^0 x_I = \min\{c_I^0 y \ : \ y \in \mathcal{E},\ C_I y \notin C(\mathcal{S}^k)\}$, so giving an NDP with next-smallest first objective value.

Proof. Optimal solution (x_I, θ) to (13)–(20) having $x_I \in \mathcal{E}$ and $\theta > 0$ implies that $C_I x_I \notin C(\mathcal{S}^k)$ by Lemma 5.2. Now suppose that for some $i \in \{0, 1, \ldots, k-1\}$, we have that $C_I^{1:l} x_I \geq C_I^{1:l} x_I^i$. Then since $x_I \in \mathcal{E}$, it must be that $c_I^0 x_I < c_I^0 x_I^i$, since otherwise $C_I x_I^i$ would dominate $C_I x_I$, contradicting $x_I \in \mathcal{E}$. But $c_I^0 x_I < c_I^0 x_I^i$ contradicts equation (25). Thus it must be that $C_I^{1:l} x_I \not\geq C_I^{1:l} x_I^i$ for all $i = 1, \ldots, k-1$. Hence (x_I, θ) must solve

$$\begin{array}{ll} \max & \theta' \\ \text{subject to} & \theta' + c_I^0 x_I' \leq U \\ & x_I' \in \mathcal{E} \\ & C_I x_I' \notin C(\mathcal{S}^k). \end{array}$$

Now if any $x_I' \in \mathcal{E}$ with $C_I x_I' \notin C(\mathcal{S}^k)$ has $c_I^0 x_I' < c_I^0 x_I$, then (x_I', θ') is feasible for this problem, where

$$\theta' := U - c_I^0 x_I' > U - c_I^0 x_I \geq \theta,$$

which contradicts optimality of (x_I, θ). Thus it must be that

$$c_I^0 x_I = \min\{c_I^0 y \ : \ y \in \mathcal{E},\ C_I y \notin C(\mathcal{S}^k)\},$$

as required. ■

Theorem 5.5. *In the pure integer case, the RVF Algorithm generates NDPs in the non-decreasing (non-increasing) order of the first objective of minimization (maximization).*

Proof. We proceed by induction on k. In the case $k = 0$, the set $\mathcal{S}^k = x_I^0$, where x_I^0 is the efficient solution to the problem $\max\{c_I^0 x_I : A_I x_I = b, x_I \in \mathbb{Z}_+^n\}$, i.e., the worst value of $c_I^0 x_I$. The RVF Algorithm finds the maximum violation between the upper approximation and real value function by finding the point for which the approximation is farthest from the real value function. Hence for any solution of (13)–(20) for $k = 0$, let's call it x_I^1, we have that $x_I^1 = \arg\min\{c_I^0 x_I \ : \ A_I x_I = b, x_I \in \mathbb{Z}_+^n\}$. Any other solution for $k = 1$ should have a larger value in terms of the first objective. Thus the base step for the induction, with $k = 1$, is proved. The inductive step is proved by Proposition 5.4. ■

We illustrate Theorem 5.5 with Figures 11 and 12 which show the value function (efficient frontier) for a biobjective and a tri-objective Knapsack instance with c_I^0 being a maximization objective. According to Figures 11 and 12, the RVF Algorithm generates NDPs in the non-increasing order of the first maximization objective c_I^0.

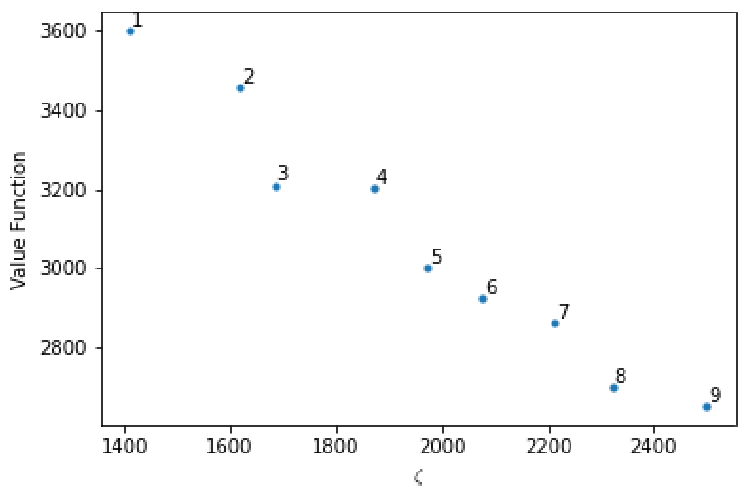


Figure 11: All NDPs for a biobjective maximization Knapsack instance.

We provide a Python package for enumerating all integer parts to construct the NDPs for instances of multiobjective integer and mixed integer programs with any arbitrary number of objective functions with the RVF Algorithm. In this package, we developed the RVF Algorithm, and to solve the problem at each iteration, we used the Couenne Belotti [2009] as the solver for an MINLP. Our package is available at `https://github.com/SamiraFallah/RestrictedValueFunction`.

5.3 Comparison to Other Algorithms

While many algorithms exist for solving multiobjective MILPs, and some for finding the VF of an MILP (see Section 2), the RVF Algorithm has some distinctive features. First, it is very general, applying across both pure and mixed integer cases and any number of objective functions. This is in contrast to multiobjective algorithms, which typically apply in either the pure or mixed cases, but not both, and which often apply only to a specific number of objectives. Second, the RVF Algorithm fills a gap in the multiobjective MILP field in its applicability to problems that include continuous variables and any number of objectives. This area of multiobjective MILP is still in its infancy, and the RVF Algorithm constitutes a significant advance in it. Thirdly, the RVF Algorithm

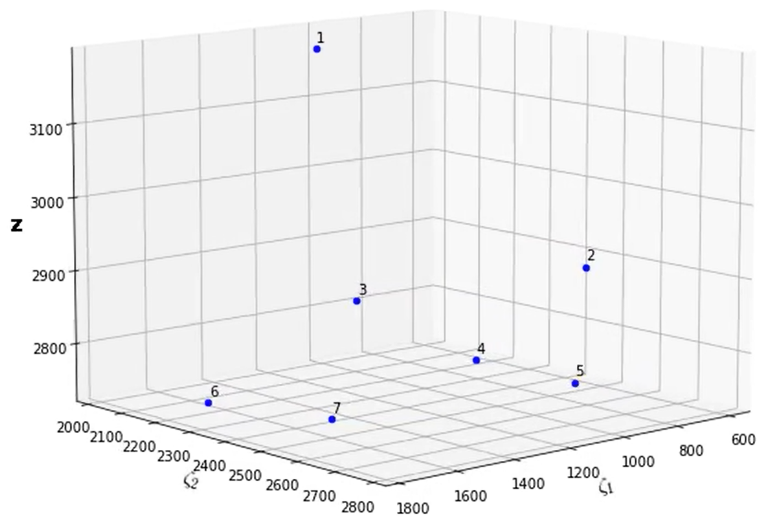


Figure 12: All NDPs for a tri-objective maximization Knapsack instance.

solves two optimization problems per stability region: one MINLP, which identifies a new stability region, and one "conversion" MILP, which converts a weakly nondominated solution to an NDP and is usually easily solved in practice. The size of the MINLP, in terms of its variables and constraints, grows linearly in the number of stability regions found: $l + m$ variables and $n - r + 1$ extra constraints (one including l bilinear terms and the rest linear) are added per new stability region. This is comparable to existing multiobjective algorithms in the pure integer case and offers advantages in the mixed case, even for small numbers of objectives. We provide more details for each of these cases below in comparison with well-known multiobjective algorithms.

5.3.1 Comparison to Algorithms for the Pure Integer Case

In the pure integer case, the stability regions are precisely the NDPs, and $\mathcal{S}_{\min}$ is unique, consisting of all NDPs. Thus if an instance has N NDPs, the RVF Algorithm solves N MINLPs and N conversion integer linear programmings (ILPs), with the size of the MINLPs increasing by l variables and only one constraint per NDP found, where the constraint added includes l bilinear terms. Note that none of these subproblems (MINLP or ILP) will be infeasible problems. In multiobjective algorithms, it is usually considered desirable to avoid infeasible ILPs, since these are often observed to be harder to solve in practice.

This compares reasonably well with some earlier algorithms for multiobjective (pure integer) ILP having any number of objectives, $p = l + 1$. For example, in theoretical analysis of their algorithm performance, following on from Lemma 4.4 of Özlen and Azizoğlu [2009] suggest their algorithm solves of the order of N choose $p-1$ ILPs. Similarly, Theorem 4 of Kirlik and Sayın [2014] suggests that their algorithm solves at most N^{p-1} "two-stage" ILPS, meaning an ILP minimizing one of the objectives over the ILP feasible set followed by a conversion ILP. The method of Sylva and Crema [2004] solves one ILP per NDP, but the ILP increases by p binary variables and $p + 1$ constraints per NDP found, which model the disjunctive requirement that ILP feasible points are not dominated by NDPs already found. The big-M values required in the model present challenges, in addition to its size. Improvements were made in Lokman and Köksalan [2013], who suggest a variant limiting the ILP size to have at most $N(p-1)$ variables and Np constraints (reducing the objective dimension by one), and who propose using the disjunctions to define a search tree. Although some practical improvements are suggested, the number of ILPs solved in the search procedure is given in Lokman and Köksalan [2013] to be of the order of N^{p-1} in the worst case.

Indeed, in the pure integer case, the RVF Algorithm can be directly related to the Lokman and Köksalan [2013] variant of the Sylva and Crema [2004] method by modeling its MINLP subproblem as a MILP. In Lokman and Köksalan [2013] variant, NDPs are found in non-decreasing order of one objective, more specifically the c^0x objective, and the size of the subproblem grows, similar to our algorithm as we showed in Theorem 5.5.

More recent algorithms for the pure integer case, such as the Disjunctive Constraints Method (DCM) of Boland et al. [2017] (given as Algorithm 1 in the paper), show good performance in practice, but the theoretical analysis in the paper gives a worst-case number of ILPs to be solved that is exponential in N. Furthermore, the ILPs increase in size through the need to model more disjunctions, and ILPs solved that do not yield a new NDP are infeasible. Hence the worst-case analysis suggests a number of infeasible ILPs solved can be exponential in N. The authors of Boland et al. [2017] observe in computational experiments that in practice, a small number of ILPs is solved per NDP and conjecture that the number is at most $pN+1$, with $(p-1)N+1$ of these an infeasible ILP. In fact, a closer study of DCM reveals that irrespective of p, at most $2N+1$ ILPs need to be solved, with at most $N+1$ infeasible ILPs, and that the number of disjunctions modeled in any ILP is N, in the worst case. (In practice, this number was found to be several orders of magnitude smaller than N.)

The point of this discussion is to show that the performance of the RVF Algorithm remains within contention of existing multiobjective algorithms for the pure integer case and any number of objectives, in theory. In practice, the state-of-the-art in MINLP solution versus ILP will play a part. The former is currently less well developed, but is an active research area. Aside from advances in MINLP, we anticipate that future research on the RVF Algorithm will extend the concept so as to improve practical performance, just as research on early versions of multiobjective algorithms let to marked improvements in practice.

5.3.2 Comparison to Algorithms with Continuous Variables

In the case of continuous variables, only the case of two objectives has existing algorithms with a theoretical performance analysis. These are two variants of the Boxed Line Method (BLM) Perini et al. [2020]. Importantly, the performance analysis of the BLM variants depends on the number of line segments in the efficient frontier, not on the number of stability regions. Let L denote the number of line segments in the frontier. Then the basic iterative variant of BLM solves at most $3L+2$ two-stage MILPs and at most $\frac{1}{2}L^2+\frac{5}{2}L-2$ scalarized (single-objective) MILPs. The recursive variant of BLM has better theoretical performance guarantees: the number of two-stage MILPs is the same, but the number of scalarized MILPs is at most $2L-1$. This gives a total of $5L+1$ MILPs plus $3L+2$ conversion MILPs as a worst-case bound. Note that the size of the MILPs does not change as the algorithm progresses. In comparison, the RVF Algorithm solves increasingly larger MINLPs, which can be expected to be harder to solve, but also may solve far fewer of them. If the number of stability regions found by the RVF Algorithm is denoted by N, so $N = |\mathcal{S}_{\min}|$, then $N \leq L$ and the number of MINLPs it solves is bounded above by $N \leq L < 5L+1$. However, each integer vector in $\mathcal{S}_{\min}$ (each stability region) could contribute *many* line segments, so RVF could need to solve far fewer subproblems.

An example to illustrate this difference is given in Figures 13 and 14, with numerical details in Appendix B. The example has two objectives, and its efficient frontier has 24 line segments but only 4 stability regions. Thus the RVF Algorithm can be expected to solve only 4 MINLPs, (with the first actually solvable as a MILP), while the best variant of the BLM method would require 121 – which is over *thirty times* more – MILPs to be solved.

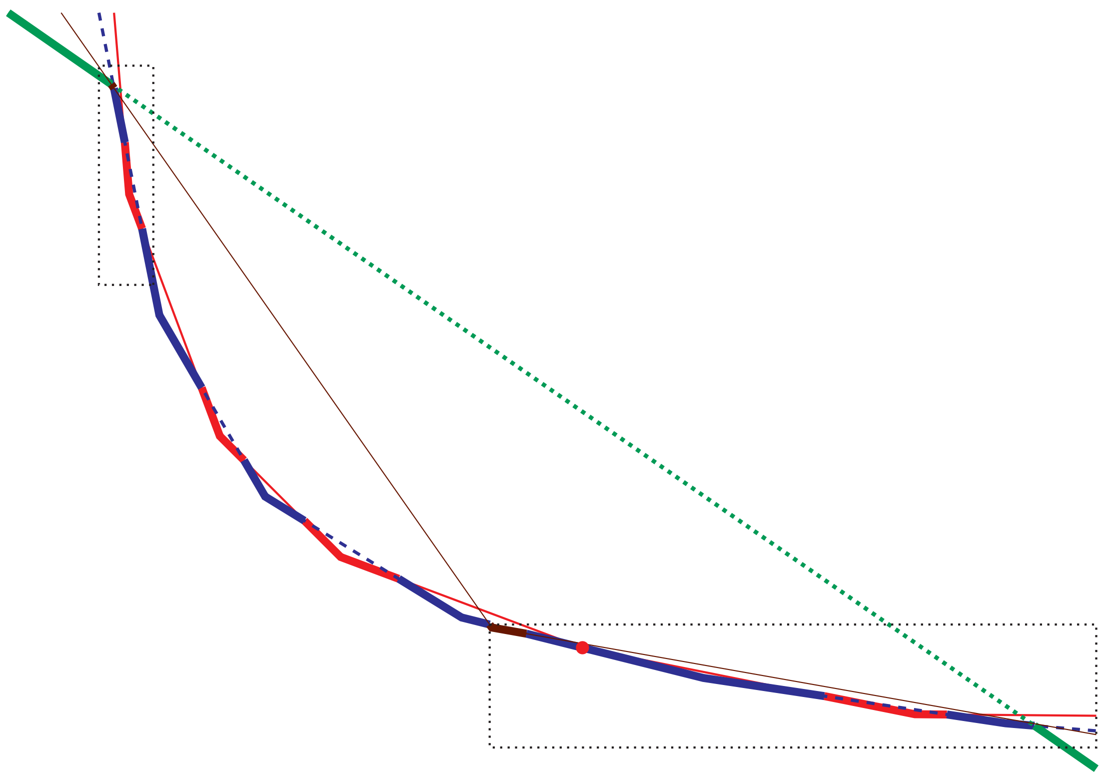
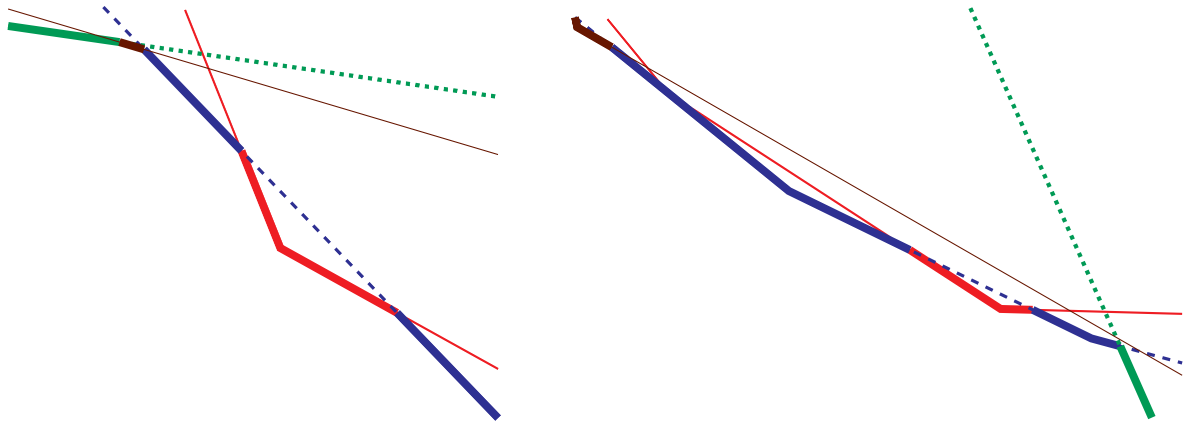


Figure 13: An example with two objectives, four elements of $\mathcal{S}_{\min}$, all with disconnected stability regions, having a total of 24 linear pieces in the efficient frontier, shown with thicker lines. The LP efficient frontiers for the four elements of $\mathcal{S}_{\min}$ are shown as red solid, blue dashed, sepia solid, and green dotted lines, respectively. Numerical details are given in Appendix B. Close-up of the areas in the top-left and bottom-right parts are given in Figure 14.

Recently, Rasmi et al. [2019] developed an algorithm to identify all the integer vectors needed for the frontier for a (MO-MILP) with an arbitrary number of objectives. They also give a method to identify all facets of a multiobjective linear programming (MOLP) efficient frontier, which contribute at least one point to the overall (MO-MILP) frontier. For all pairs of extreme points of such facets that are adjacent in the facet, Rasmi et al. [2019] show how to determine the parts of the line segment connecting the two points that are nondominated, if any. If only part of a facet of a MOLP efficient frontier appears in the (MO-MILP) frontier, Rasmi et al. [2019] do not attempt to describe the nondominated part explicitly. Their method for generating all integer vectors needed for the (MO-MILP) frontier combines "no-good" constraints to eliminate integer vectors already found and disjunctive constraints, similar to those of Sylva and Crema [2004], to eliminate parts of objective space dominated by individual NDPs found. Thus the size of the MILPs they solve grows in two ways as the algorithm progresses. No theoretical performance analysis is given. The



(a) Top-left

(b) Bottom right

Figure 14: Close-ups of the regions in the top-left and bottom-right ends of shown in Figure 13. Note that these have been asymmetrically scaled to allow better visibility of LP frontier intersections.

algorithm may solve MILPs that yield integer solutions that do not contribute to the frontier, and many MILPs solved may find the same integer solutions repeatedly. Both of these issues are avoided in the RVF algorithm. Computational results on instances with 2, 3, and 4 objectives are given. These show that even for only 2 objectives, the algorithm can solve more than 25 MILPs per integer solution, and for one class of instances with 3 objectives, on average more than 135 MILPs were solved per integer solution. No indication of how many disjunctive constraints are added to the MILPs in computation or how the size of these MILPs grows is provided. In another work by Pettersson and Ozlen [2019], an algorithm is developed to identify all supported and non-supported NDPs with an arbitrary number of objectives. First, they find a super-set (which contains the efficient frontier) via Benson's method. Then they modify the set of polytopes so that no two polytopes have a non-empty intersection. Finally, they refine the super-set to exclude the parts that are not part of the efficient frontier. The details of the algorithm are not available since it was a short proceedings article.

6 Conclusions and Future Research Directions

In this paper, we discussed the relationship between the RVF and the efficient frontier for a multiobjective problem. We have also shown that the frontier and the boundary of the epigraph of the restricted value function are equivalent. In this context, we proposed the RVF algorithm for finding all nondominated points in the efficient frontier for a multiobjective optimization problem. Our proposed algorithm is an alternative to existing algorithms for constructing the frontier that provides a convenient and precise performance guarantee if terminated early. More importantly, it highlights an important relationship and connects two parts of the literature that had been considered distinct. We showed that the RVF is the minimum of convex polyhedral functions associated with RLPVF and discussed the structure of the RVF, including continuity and being piecewise polyhedral. Finally, we showed that there is a discrete representation for both RVF and the frontier under the assumption that the X_{MO} is bounded. We also provide a Python package that shows the proposed algorithm works in practice.

The basic algorithm presented here can be improved in several ways, and computational efficiency can be enhanced. We have already implemented the RVF Algorithm and performed comprehensive tests. For future work, we plan to embed this algorithm into algorithms for related problems requiring parametric analysis or value function construction. We also could extend this research study as follows. Based on the structure of our proposed algorithm, as can be seen from its cutting-plane style, we add a cut at each iteration of the algorithm. Therefore, we can use the previous branch and bound tree as a warm start strategy. Another research direction would be customizing a branch and bound method based on the structure of our nonlinear problem. We used the Gurobi solver for solving the problem at each iteration, but we can customize a branch and bound with different branching, bounding, and searching strategies that could outperform the commercial solvers.

We hope that our new proposed algorithm encourages more researchers to study multiobjective programs in a new way. Exploring the relationship between the value function and the efficient frontier both theoretically and computationally could be valuable since there are several applications in which the value function (or, equivalently, the efficient frontier) arises.

References

Shabbir Ahmed, Mohit Tawarmalani, and Nikolaos V Sahinidis. A finite branch-and-bound algorithm for two-stage stochastic integer programs. *Mathematical Programming*, 100(2):355–377, 2004.

Bernd Bank, Jürgen Guddat, Diethard Klatte, Bernd Kummer, and Klaus Tammer. *Non-linear parametric optimization*. Springer, 1982.

Mokhtar S Bazaraa, John J Jarvis, and Hanif D Sherali. *Solutions Manual to Accompany Linear Programming and Network Flows*. Wiley, 1990.

Pietro Belotti. Couenne: a user's manual, 2009.

Pietro Belotti, Banu Soylu, and Margaret M Wiecek. A branch-and-bound algorithm for biobjective mixed-integer programs. *Optimization Online*, 2013.

Harold P Benson. Existence of efficient solutions for vector maximization problems. *Journal of Optimization Theory and Applications*, 26(4):569–580, 1978.

Charles Blair. A closed-form representation of mixed-integer program value functions. *Mathematical Programming*, 71(2):127–136, 1995.

Charles E Blair and Robert G Jeroslow. The value function of a mixed integer program: I. *Discrete Mathematics*, 19(2):121–138, 1977.

Charles E Blair and Robert G Jeroslow. The value function of an integer program. *Mathematical Programming*, 23(1):237–273, 1982.

Charles E Blair and Robert G Jeroslow. Constructive characterizations of the value-function of a mixed-integer program i. *Discrete Applied Mathematics*, 9(3):217–233, 1984.

Charles Eugene Blair and Robert G Jeroslow. The value function of a mixed integer program: Ii. *Discrete Mathematics*, 25(1):7–19, 1979.

Merve Bodur, Shabbir Ahmed, Natashia Boland, and George L Nemhauser. Decomposition of loosely coupled integer programs: A multiobjective perspective. *Optimization Online*, 2016.

Natashia Boland, Hadi Charkhgard, and Martin Savelsbergh. 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, 2017.

BOSPP-benchmarks. `https://www.emse.fr/~delorme/SetPacking.html#BOSPP`.

V Joseph Bowman. On the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives. In *Multiple criteria decision making*, pages 76–86. Springer, 1976.

Seth Brown, Wenxin Zhang, Temitayo Ajayi, and Andrew J Schaefer. A gilmore-gomory construction of integer programming value functions. *Operations Research Letters*, 2021.

LG Chalmet, L Lemonidis, and DJ Elzinga. An algorithm for the bi-criterion integer programming problem. *European Journal of Operational Research*, 25(2):292–300, 1986.

Pasqualina Conti and Carlo Traverso. Buchberger algorithm and integer programming. In *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, pages 130–139. Springer, 1991.

Xavier Delorme, Xavier Gandibleux, and Fabien Degoutin. Evolutionary, constructive and hybrid procedures for the bi-objective set packing problem. *European Journal of Operational Research*, 204(2):206–217, 2010.

Matthias Ehrgott. *Multicriteria optimization*, volume 491. Springer Science & Business Media, 2005.

Matthias Ehrgott. A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research*, 147(1):343–360, 2006.

Matthias Ehrgott and Xavier Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR-spektrum*, 22(4):425–460, 2000.

Matthias Ehrgott and Margaret M. Wiecek. *Mutiobjective Programming*, pages 667–708. Springer New York, New York, NY, 2005. ISBN 978-0-387-23081-8. doi: 10.1007/0-387-23081-5_17. URL `https://doi.org/10.1007/0-387-23081-5_17`.

Matthias Ehrgott, Xavier Gandibleux, and Anthony Przybylski. Exact methods for multi-objective combinatorial optimisation. In *Multiple criteria decision analysis*, pages 817–850. Springer, 2016.

Juergen Guddat, F Guerra Vasquez, Klaus Tammer, and Klaus Wendler. Multiobjective and stochastic optimization based on parametric optimization. *Mathematical Research*, 26, 1985.

M. Güzelsoy and T.K. Ralphs. Duality for Mixed-Integer Linear Programs. *International Journal of Operations Research*, 4:118–137, 2007. URL `http://coral.ie.lehigh.edu/~ted/files/papers/MILPD06.pdf`.

MENAL Guzelsoy and T Ralphs. The value function of a mixed-integer linear program with a single constraint. *To be submitted*, 2006.

Yacov Haimes. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE transactions on systems, man, and cybernetics*, 1(3):296–297, 1971.

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*, 2022.

Gokhan Kirlik and Serpil Sayın. A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. *European Journal of Operational Research*, 232 (3):479–488, 2014.

Nan Kong, Andrew J Schaefer, and Brady Hunsaker. Two-stage integer programs with stochastic right-hand sides: a superadditive dual approach. *Mathematical Programming*, 108(2):275–296, 2006.

Banu Lokman and Murat Köksalan. Finding all nondominated points of multi-objective integer programs. *Journal of Global Optimization*, 57(2):347–365, 2013.

George Nemhauser and Laurence Wolsey. The scope of integer and combinatorial optimization. *Integer and combinatorial optimization*, pages 1–26, 1988.

Melih Özlen and Meral Azizoğlu. Multi-objective integer programming: A general approach for generating all non-dominated solutions. *European Journal of Operational Research*, 199(1):25–35, 2009.

Adriano Pascoletti and Paolo Serafini. Scalarizing vector optimization problems. *Journal of Optimization Theory and Applications*, 42(4):499–524, 1984.

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.

William Pettersson and Melih Ozlen. Multi-objective mixed integer programming: An objective space algorithm. In *AIP Conference Proceedings*, volume 2070, page 020039. AIP Publishing LLC, 2019.

Ted K Ralphs and Anahita Hassanzadeh. On the value function of a mixed integer linear optimization problem and an algorithm for its construction. *COR@ L Technical Report 14T–004*, 2014.

Ted K Ralphs, Matthew J Saltzman, and Margaret M Wiecek. An improved algorithm for solving biobjective integer programs. *Annals of Operations Research*, 147(1):43–70, 2006.

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

Rüdiger Schultz, Leen Stougie, and Maarten H Van Der Vlerk. Solving stochastic programs with integer recourse by enumeration: A framework using gröbner basis. *Mathematical Programming*, 83(1):229–252, 1998.

John Sylva and Alejandro Crema. A method for finding the set of non-dominated vectors for multiple objective integer linear programs. *European Journal of Operational Research*, 158(1): 46–55, 2004.

Andrew C Trapp and Oleg A Prokopyev. A note on constraint aggregation and value functions for two-stage stochastic integer programs. *Discrete Optimization*, 15:37–45, 2015.

Andrew C Trapp, Oleg A Prokopyev, and Andrew J Schaefer. On a level-set characterization of the value function of an integer program and its application to stochastic programming. *Operations Research*, 61(2):498–511, 2013.

Laurence A Wolsey. Integer programming duality: Price functions and sensitivity analysis. *Mathematical Programming*, 20(1):173–195, 1981.

Po-Lung Yu. A class of solutions for group decision problems. *Management science*, 19(8):936–946, 1973.

Lofti Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE transactions on Automatic Control*, 8(1):59–60, 1963.

Milan Zeleny. Compromise programming. *Multiple criteria decision making*, 1973.

Junlong Zhang and Osman Y Özaltın. Bilevel integer programs with stochastic right-hand sides. *INFORMS Journal on Computing*, 2021.

Appendix A

We can linearize the problem to be solved in each iteration of the pure integer case ((21)–(23)) as follows.

$$
\begin{aligned}
\theta^k = \max \quad & \theta && (26)\\
\text{subject to} \quad & \theta + c_I^0 x_I \leq (1-\beta^i) c_I^0 x_I^i + \beta^i U \quad i = 0,1,\dots,k-1 && (27)\\
& (C_I^{1:l} x_I)_j - (C_I^{1:l} x_I^i)_j + \epsilon_j^i \leq \overline{M}_j^i (1-\alpha_j^i) \quad i = 0,1,\dots,k-1,\ j = 1,\dots,l && (28)\\
& (C_I^{1:l} x_I^i)_j - (C_I^{1:l} x_I)_j \leq \underline{M}_j^i \alpha_j^i \quad i = 0,1,\dots,k-1,\ j = 1,\dots,l && (29)\\
& \beta^i \geq \alpha_j^i \quad i = 0,1,\dots,k-1,\ j = 1,\dots,l && (30)\\
& \beta^i \leq \sum_{j=1}^{l} \alpha_j^i \quad i = 0,1,\dots,k-1 && (31)\\
& \alpha^i \in \{0,1\}^l \quad i = 0,1,\dots,k-1 && (32)\\
& \beta^i \in \{0,1\} \quad i = 0,1,\dots,k-1 && (33)\\
& (18),\ (19) \text{ and } (20), && (34)
\end{aligned}
$$

where for each i and j, $\underline{M}_j^i$ and ϵ_j^i are set to sufficiently large and small positive values, respectively, so that for any possible choice of x_I having $(C_I^{1:l} x_I)_j < (C_I^{1:l} x_I^i)_j$ it must be that $\epsilon_j^i \leq (C_I^{1:l} x_I^i)_j - (C_I^{1:l} x_I)_j \leq \underline{M}_j^i$, and $\overline{M}_j^i$ is set to a sufficiently large positive value so that for any possible choice of x_I having $(C_I^{1:l} x_I)_j \geq (C_I^{1:l} x_I^i)_j$ it must be that $(C_I^{1:l} x_I)_j - (C_I^{1:l} x_I^i)_j \leq \overline{M}_j^i - \epsilon_j^i$. Note that in the case that $C^{1:l}$ is entirely integer-valued, $\epsilon_j^i = 1$ for all i and j suffices. Furthermore, for each i, α^i is a vector of binary variables such that for all $j = \{1, 2, \dots, l\}$

$$
\alpha_j^i = \begin{cases} 0 & (C_I^{1:l} x_I)_j \geq (C_I^{1:l} x_I^i)_j, \\ 1 & (C_I^{1:l} x_I)_j < (C_I^{1:l} x_I^i)_j, \end{cases} \tag{35}
$$

if and only if (28) and (29) are satisfied. Note that we substituted out (22) and (16) in the problem set (21)–(23) with (27), (28), and (29).

The RVF algorithm with the problem set (26)–(34) would be equivalent to the method given in [Sylva and Crema, 2004], which enumerates all NDPs of a pure integer program using a sequence of progressively more constrained MILP that generates a new solution at each iteration. The method proposed in [Sylva and Crema, 2004] was improved slightly in [Lokman and Köksalan, 2013] using the disjunctions to define a search tree.

Appendix B

The MO-MILP illustrated in Figures 13 and 14, having four elements of $\mathcal{S}_{\min}$, has the LP efficient frontier for each formed as follows. The LP frontier shown in red in the figures, which has the right-most upper left corner point, is created by joining its extreme supported points given by

$$F_1 = \{(3\frac{1}{2}, 27), (4, 21), (7, 13), (11, 9), (19, 6), (30, 3\frac{4}{5}), (36, 3\frac{3}{4})\}.$$

The LP frontier shown in blue in the figures, which has the second right-most upper left corner point, is created by joining its extreme supported points given by

$$F_2 = \{(3, 27), (5, 17), (8\frac{1}{2}, 11), (15, 7), (23, 5), (33, 3\frac{1}{2}), (36, 3\frac{1}{4})\}.$$

The LP frontier shown in brown in the figures, which has the third right-most upper left corner point, is created by joining its extreme supported points given by

$$F_3 = \{(1\frac{3}{4}, 27), (16, 6\frac{2}{3}), (36, 3\frac{1}{8})\}.$$

The LP frontier shown in green in the figures, which has the left-most upper left corner point, is created by joining its extreme supported points given by

$$F_4 = \{(0, 27), (36, 2)\}.$$

The four individual LP frontiers are show in Figures 15–17, with the fourth LP frontier, shown in green, duplicated in each, to provide a common reference in addition to the axes.

A MO-MILP corresponding to these LP frontiers can be constructed in many ways, but the most direct uses four binary variables, one per LP frontier, to switch the corresponding LP constraints on or off, and two continuous variables, one for each of the two objectives.

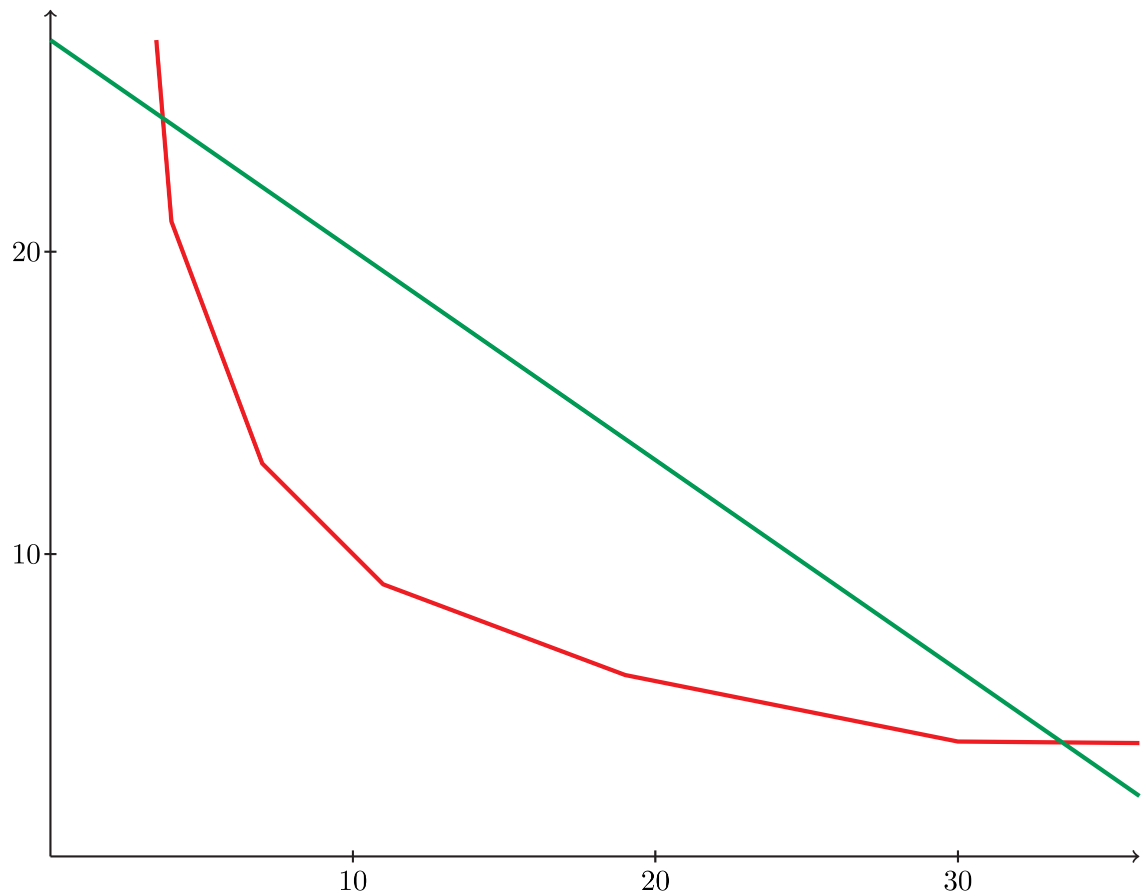


Figure 15: The complete LP frontier for the parts shown in red in Figure 13, together with the complete LP frontier for the part shown in green.

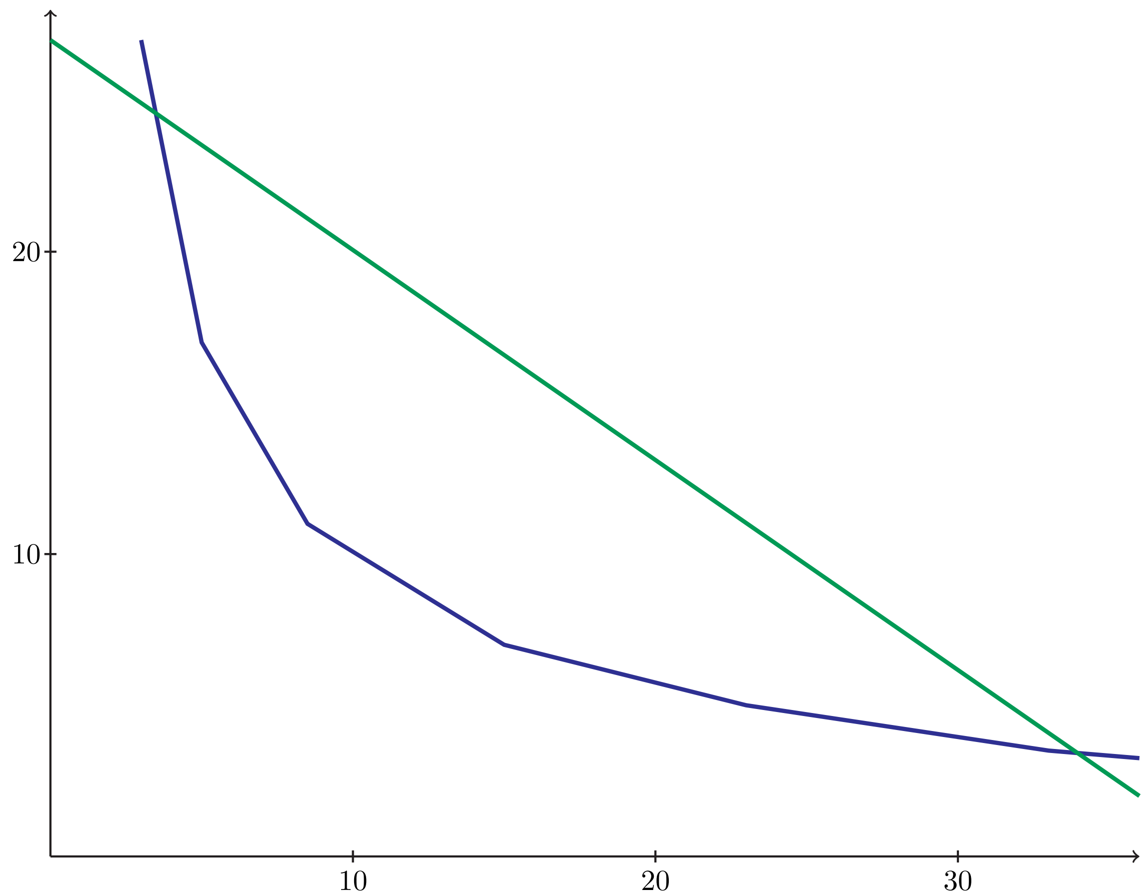


Figure 16: The complete LP frontier for the parts shown in blue in Figure 13, together with the complete LP frontier for the part shown in green.

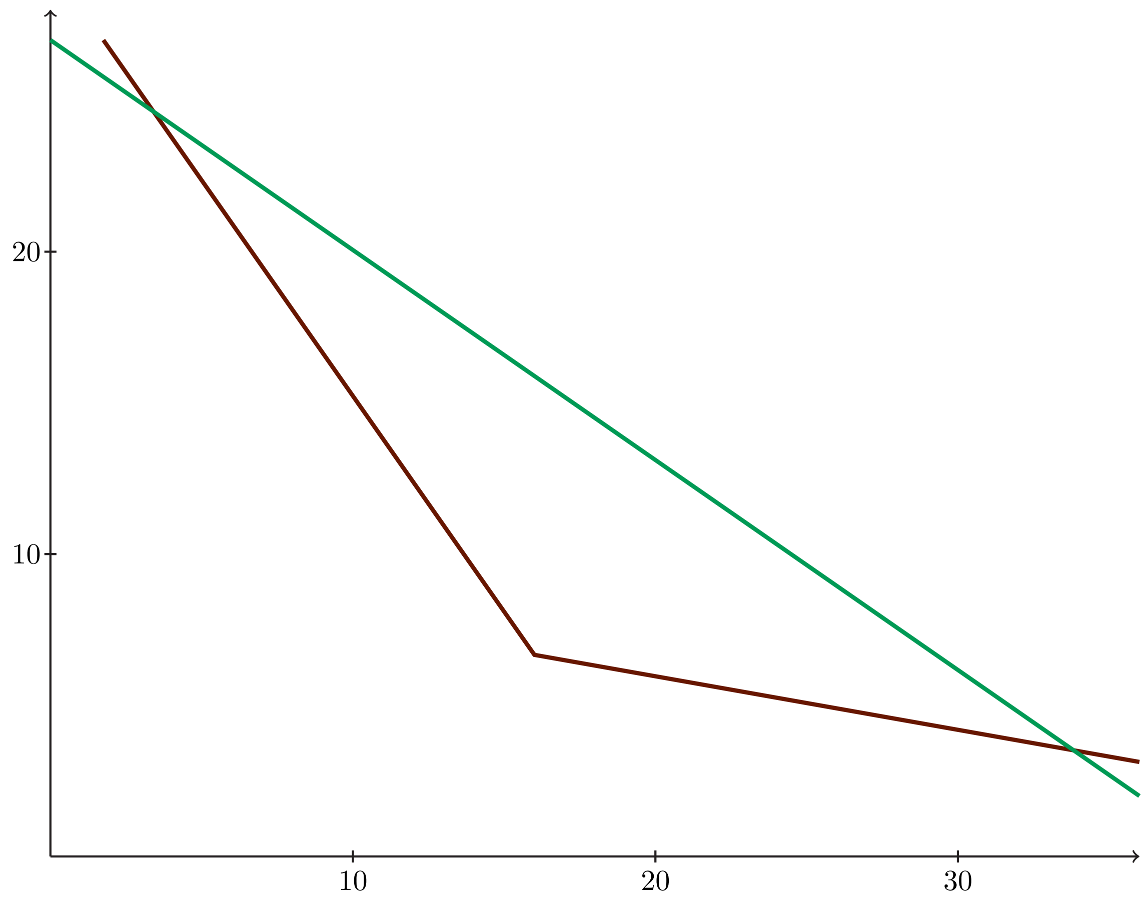


Figure 17: The complete LP frontier for the parts shown in brown in Figure 13, together with the complete LP frontier for the part shown in green.