The Frontier Partitioner Algorithm

Marianna De Santis*, Giorgio Grani*, Laura Palagi*

April 18, 2018

Abstract

It is not unknown that a great amount of problems in real-world applications manages with more than one objective function. Although a lot of work has been done for the case where all the variables are continuous, when we take into account also integer variables is far to be sufficiently investigated. In our work we present an effective pure integer algorithm suitable for biobjective programs. The algorithm is more than competitive with respect to all the other known algorithms for linear integer problems. On the other side its crucial property is that it can manage also convex nonlinear pure integer problems. The main idea is to create a self constructing partition of the original frontier. In other words it uses the knowledge of having a Pareto point to split the feasible region, adding cuts separating efficient solutions. The computation ends in an exact number of iteration if the frontier has a finite number of points. The algorithmic framework is lean both to understand and implement.

1 Literature Review

Lets make a scheme:

- Theoretical approach:
  - [Belotti et al., 2013] and [Belotti et al., 2016] B&B algorithm for biobjective mixed-integer problems. They focus on the idea of find the complete Pareto frontier for a relaxed subproblem. This information is used to derive practical fathoming rules for the B&B. The results seems to be effective but the general scheme is quite complex.
  - [Büsing et al., 2017] links between reference points and approximation algorithms. The main result is to define the substantial and polynomial equivalence between approximating reference point solutions, approximating compromise solutions and approximating the Pareto set. Then they solve the reference point problem for some known combinatorial problems.
  - [Mavrotas, 2009] discussion around the implementation of the ϵ-constraints method, a known scalarization technique.
  - [Gabbani and Magazine, 1986] an heuristic which uses an interactive technique.
  - [Martin et al., 2017] a study of constraints propagation under a multi-objective Branch and Bound in a nonlinear context.

*Sapienza University of Rome
{marianna.desantis@uniroma1.it}
{g.grani@uniroma1.it}
{laura.palagi@uniroma1.it}

- [Mavrotas and Diakoulaki, 1998] the basic (widely enumerative) B& B method for 01MOMILP.
- [Mavrotas and Diakoulaki, 2005] an improvement of the previous work. This is the version we used as benchmark in the computational experience.
- [Cacchiani and DAmbrosio, 2017] this is an example of valid heuristic for Convex MINLPs.
- [Przybylski and Gandibleux, 2017] one of the atest surveys on the argument.
- [Alves and Clímaco, 2007] a survey on interactive methods.
- [Gutjahr and Pichler, 2016] an interesting survey which investigates non-scalarizing methods for stochastic problems.
- [Ralphs et al., 2006] algorithm for BOMILP.
- [Ramesh et al., 1986] an old paper focusing on interactive methods.
- [Stidsen et al., 2014] a B& B algorithm for a specified class of biobjective problems.
- [Villarreal and Karwan, 1981] here they present a recursive and dynamic programming approach to the problem.

- Similar Approaches:
  - [Boland et al., 2017a]
  - [Boland et al., 2017b] they present the Quadrant Shrinking method, a generalization for triobjective problems of the Split algorithm.
  - [Boland et al., 2016] really similar to our algorithm, it is very efficient and works iteratively.
  - [Kirlik and Sayın, 2014] they improve the Split algorithm.
  - [Lokman and Köksalan, 2013] they improve the Split algorithm.
  - [Sylva and Crema, 2004] the basic method which generates a sequence of problems (harder on each iteration) taking into account the barrier already defined. They called it Split algorithm.

- Applications
  - [Sedeño-Noda and González-Martın, 2001] for the Minimum Cost flow problem.
  - [Rezaee et al., 2017] for a green supply chain network design with stochastic demand and carbon price.
  - [Ralphs et al., 2004] applied to the network routing problem.
  - [Raith and Ehrgott, 2009] for the Minimum Cost flow problem.
  - [Moradi et al., 2015] biobjective multi-commodity minimum cost flow problem.
  - [Che et al., 2017] for the stable robotic flow shop scheduling.
  - [Przybylski et al., 2010] assignement problem with three objectives.

2 Concepts

The Multiobjective optimization problem is to determine a Pareto solution (weak, local or global) of an optimization problem with more than one objective.
We define the following Multiobjective Integer Model

$$\mathcal{P} = \min_{x\in\mathcal{X}} f(x) \tag{1}$$

Where $\mathcal{X} = \{x \in \mathbb{Z}^n \ : \ h(x) \leq \mathbf{0}\} \subseteq \mathbb{Z}^n$ is the feasible set. The functions $f(\cdot) : \mathbb{R}^n \to \mathbb{R}^2$ and $h(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ are supposed to be continuous.
Given two points $z,\, w \in \mathbb{R}^2$ we say z *dominates* w and we write $z \preceq w$ if $z_1 \leq w_1$ and $z_2 \leq w_2$ and $\exists i \in \{1, 2\} \ : \ z_i < w_i$.
Given two points $z,\, w \in \mathbb{R}^2$ we say z *strictly dominates* w and we write $z \prec w$ if $z_1 < w_1$ and $z_2 < w_2$.
We say $x \in \mathcal{X}$ is a solution of problem (1) if $\nexists\ \hat{x} \in \mathcal{X} \ : \ f(\hat{x}) \preceq f(x)$. We call x the *efficient solution* and $f(x)$ the *Pareto point.*

3 Notation

From problem (1), at node k the subproblem is given by

$$\mathcal{P}_k = \min_{x\in\mathcal{X}^k} f(x) \tag{2}$$

Where $\mathcal{X}^k = \left\{x \in \mathbb{Z}^n \ : \ h(x) \leq \mathbf{0},\ g_k(x) \leq v^k\right\} \subseteq \mathbb{Z}^n$ and the block of constraints $g_k(x) \leq v^k$ is the one of added constraints.

4 The Frontier Partitioner Algorithm for Problems with only discrete variables

Assumption 4.1. *$\mathcal{X} \subseteq \mathbb{R}^n$ is a closed and convex set.*

This is the basic case and the easier, where the Clever Frontier finds naturally its way.
Its known that a valid method to find a Pareto point of Multiobjective problems is by using scalarization techniques. There a lot of different methods, let us resume some of them:

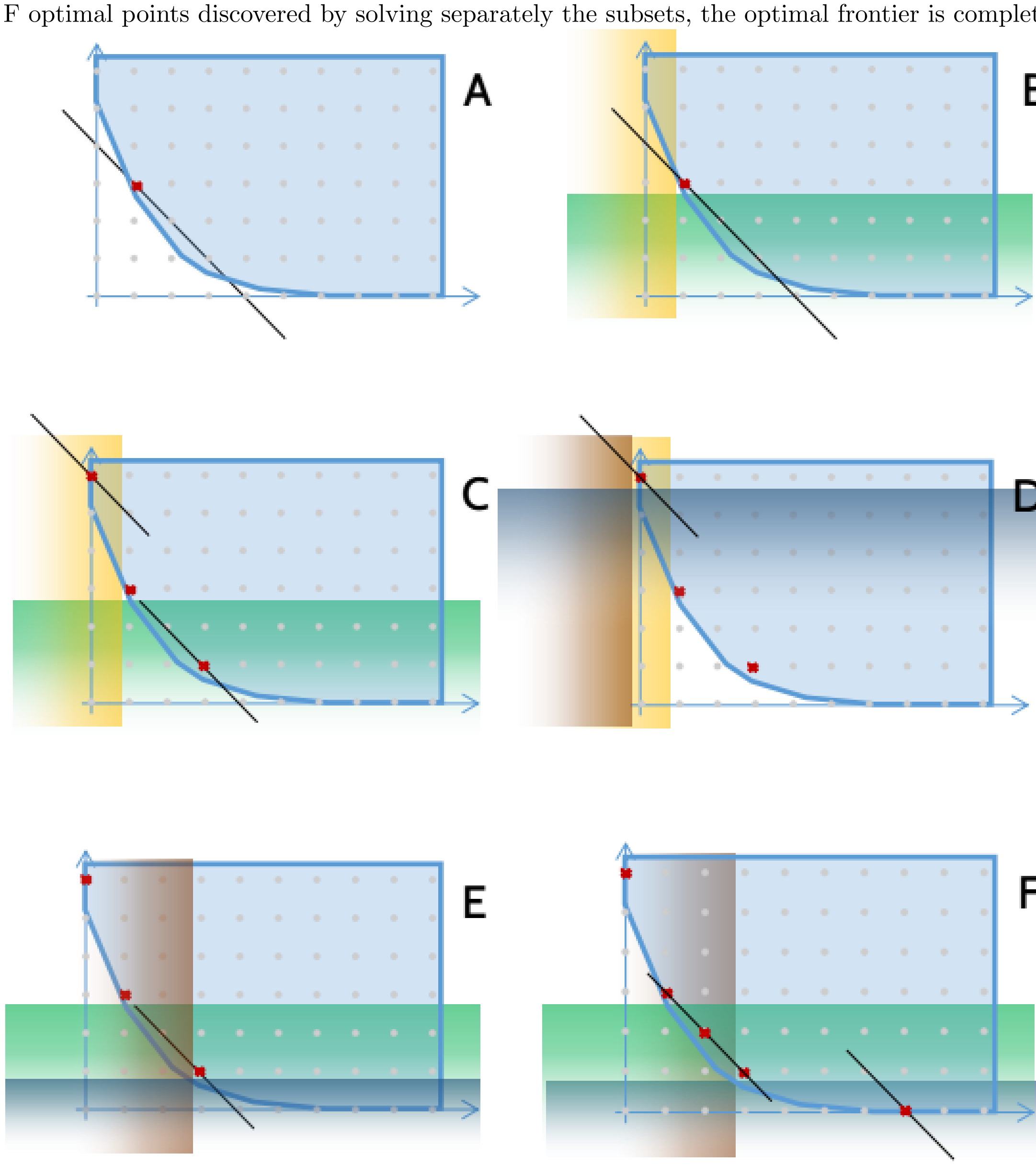
- *Without Preferences.*
  - GOAL methods, which are differentiated by the norm used:
    - Norm 1 $\min_{x\in\mathcal{X}^k} \sum_{i=1}^{n} |f_i(x) - z^I_{ik}|$ equivalent to $\min_{x\in\mathcal{X}^k} \sum_{i=1}^{n} f_i(x)$
    - Norm 2 $\min_{x\in\mathcal{X}^k} ||f(x) - z^I_k||_2$ equivalent to $\min_{x\in\mathcal{X}^k} ||f(x) - z^I_k||_2^2$
    - Chebyshev Norm $\min_{x\in\mathcal{X}^k} \min_{i=1,\ldots,n} |f_i(x) - z^I_{ik}|$ equivalent to $\min_{x\in\mathcal{X}^k} \min_{i=1,\ldots,n} f_i(x)$
  - Lexicographic methods write everything
- *With Preferences.* These are the most used in practice:
  - Weights $\min_{x\in\mathcal{X}^k} \sum_{i=1}^{n} w_i f_i(x)$ where $\sum_{i=1}^{n} w_i = 1,\ w_i \geq 0.$

– ϵ-Constraints $\min_{x \in \hat{\mathcal{X}}^k} \sum_{i=1}^{n} f_i(x)$ where $\hat{\mathcal{X}}^k = \left\{ x \in \mathcal{X}^k \ : \ f_i(x) \leq \epsilon_i, \ i = 1, \ldots, n \right\}$

- *Interactive Methods.* They are in a certain sense a merge of the previous algorithms so we will not take into consideration at all.

The central idea is the fact that these methods has interesting properties about the solution they found. In fact GOAL methods with finite norm always return a Pareto optimal point. On the other hand under certain and reasonable assumptions also methods with preferences can give a Pareto point. We use $s(\mathcal{P})$ to define the scalarized problem derived from the Multiobjective $\mathcal{P}$, supposing the scalarization technique adopted verify the condition under which the solution is a Pareto point.
Knowing this we can easily derive an algorithm which investigate iteratively the optimal frontier. We make the strong assumption that the number of Pareto points is finite. Taking a look among algorithms for multiobjective problems (even continuous) this is a basic assumption.
The main idea is to find a Pareto point of the frontier at each node of the exploration tree. To make this possible we solve a scalarized problem using one of the techniques seen before, ensuring to find a Pareto point of the optimal frontier.
Once the integer optimal point is found, we create two problems,each one with a personal cut based on the solution found before. We repeat recursively this procedure and a node is fathomed if its subproblem has no feasible solutions. The cuts are inherited by its sons, and shared with no other previous node in the tree, in thiis way each subset is a perfect partition in the space of objectives functions.

Figure 1: A first solution found by Weights method. B partitioning of the set of objectives. C optimal points discovered by solving separately the subsets. D applying FPA to the top-left point, both the subproblems generated are empty. E applying FPA to the bottom-right point. F optimal points discovered by solving separately the subsets, the optimal frontier is complete



Frontier Partitioner Algorithm

Initialization $k = 0$, $\mathcal{P}_0 = \mathcal{P}$, $\mathcal{D} = \mathcal{P}_0$, $\mathcal{Y} = \emptyset$.

While $\mathcal{D} \neq \emptyset$

  Take $\hat{\mathcal{P}} \in \mathcal{D}$, called $\hat{\mathcal{X}}$ its feasible set.

  Solve $s(\hat{\mathcal{P}})$.

  If $s(\hat{\mathcal{P}})$ is not feasible, then $\mathcal{D} \setminus \left\{ \hat{\mathcal{P}} \right\}$.

  Else

    Take one of its solutions $\hat{x} \in \arg s(\hat{\mathcal{P}})$ and call $\hat{f} = f(\hat{x})$.

    Add Cuts

      For $h = 1, 2$

$$\mathcal{X}_{k+h} = \mathcal{X}_k \cap \left\{ x \in \mathbb{Z}^n : f_h(x) \leq \hat{f}_h - \epsilon_h \right\}$$

$$\mathcal{P}_{k+h} = \min_{x \in \hat{\mathcal{X}}_{k+h}} f(x)$$

      End For

      $k \to k + 2$

    $\mathcal{Y} \cup \left\{ \hat{f} \right\}$

    $\mathcal{D} \setminus \left\{ \hat{\mathcal{P}} \right\}$

End While

Return $\mathcal{Y}$

The main algorithm will work in a setting of global optimality of the Pareto front, since the basic technique used to perform those points always return a Pareto point.
So if we suppose Convex Problems we will have no local Pareto points, then globally our idea is applicable.
The basic problem is

$$\begin{cases} \min & f(x) \\ s.t. & x \in \mathcal{X} \end{cases}$$

The difficulty here is to determine the value of the ϵ_h values. The hard problem of find such a value can be rewritten as

$$\begin{cases} \max & f(\hat{x}) - f(x) \\ s.t. & f_h(x) < f_h(\hat{x}) \\ & x \in \mathcal{X} \end{cases}$$

Where $f(x) : \mathbb{Z}^n \to \mathbb{R}^m$ is a multidimensional convex function and $\mathcal{X} \subseteq \mathbb{Z}^n$ is a convex set.
We do not need to discuss that this problem is NP-Hard, and its relaxation have not a solution.
This lead to the definition of properties the problem has to verify to guarantee the exactness of FPA.
In the following we shows the fundamental properties fo ϵ_h.

Definition 1 (Valid value)**.** *ϵ_h is said to be **valid** if $\epsilon_h \in (0, \gamma_h]$ where $\gamma_h \in \mathbb{R} : |f_h(x) - f_h(y)| \geq \gamma_h > 0, \quad \forall\ x, y \in \mathcal{X} : f_h(x) \neq f_h(y)$*

In some special cases we can easily verify the previous property.

Proposition 4.2. *If the objective function is such that $f_h(x) : \mathbb{Z}^n \to \mathbb{Z}$ then $\epsilon_h = 1$ is valid.*

Proof. Since $f(\mathcal{X}) = \mathbb{Z}$ then $|f_h(x) - f_h(y)| \geq 1, \ \ \forall\, x, y \in \mathcal{X} : f_h(x) \neq f_h(y)$ and we can apply proposition 1. ∎

Given a quadratic function $f_h(x) = x^\intercal Q x + d^\intercal x$, if $Q \succcurlyeq 0$, $Q \in \mathbb{Z}^{n\times n}$ and $d \in \mathbb{Z}$ then proposition 4.2 is verified.

Proposition 4.3. *Given a quadratic function $f_h(x) = x^\intercal Q x + d^\intercal x$, if $Q \succcurlyeq 0$, $Q \in \mathbb{Q}^{n\times n}$ and $d \in \mathbb{Q}$ then $\exists n \in \mathbb{N} : \epsilon_h = \dfrac{1}{n}$ is valid.*

Proof. Since $Q \in \mathbb{Q}^{n\times n}$ and $d \in \mathbb{Q}$ then $\exists n \in \mathbb{N} : nQ \in \mathbb{Z}^{n\times n}$ and $nd \in \mathbb{Z}$.
Consider now the function $g_h(x) = x^\intercal nQx + nd^\intercal x = nf_h(x)$, it verifies proposition 4.2 and then we can write

$$\begin{gathered}
|g_h(x) - g_h(y)| \geq 1, \ \ \forall\, x, y \in \mathcal{X} : g_h(x) \neq g_h(y) \\
\Updownarrow \\
|f_h(x) - f_h(y)| \geq \tfrac{1}{n}, \ \ \forall\, x, y \in \mathcal{X} : f_h(x) \neq f_h(y)
\end{gathered}$$

∎

We can make the same considerations considering linear functions, which are special cases with $\mathcal{Q} : q_{ij} = 0$.

Definition 2 (Limited Frontier)**.** *A problem $\mathcal{P}$ is said to have a limited Pareto frontier if the Pareto frontier is a limited set.*

We want to remark the following property: Given a, b, $c \in \mathbb{R}$, if $a \leq b \leq c$ then $|b| \leq \max\{|a|, |c|\}$.[1]

Proposition 4.4 (Sufficient condition of Limited Pareto Frontier)**.** *The Pareto Frontier is limited if $\omega^h \in \mathbb{R}^m$, for $h = 1, 2$.*
Where $\omega^h = f(\hat{x}^h)$ and $\hat{x}^h \in \arg\min_{x\in\mathcal{X}} f_h(x)$.

Before proving this proposition observe that by this definition $\omega^h_h = z^I_h$, said the ideal value for the h-function.

Proof. We call $\mathcal{S}$ the set of Pareto points in the space of the objective functions. Knowing the definition of ω^h we have that for a given point $s \in \mathcal{S}$ the following holds

$$s_h \geq \omega^h_h, \quad h = 1, 2$$

On the other hand, since s is a point of the Pareto frontier, then its other component must be at least better than the other one of the ω-points, in formulas

$$s_k \leq w^h_k, \quad h = 1, 2, \ k = 1, 2, \ k \neq h$$

Then we obtain

$$\begin{aligned}
w^h_h \leq s_h \leq w^k_h, &\qquad h = 1, 2, \ k = 1, 2, \ k \neq h \\
\Downarrow & \\
|s_h| \leq \max\left\{|w^h_h|, |w^k_h|\right\} = val_h, &\qquad h = 1, 2, \ k = 1, 2, \ k \neq h \\
\Downarrow & \\
|s_h|^2 \leq val_h^2, &\qquad h = 1, 2, \ k = 1, 2, \ k \neq h \\
\Downarrow & \\
||s||_2 = \sqrt{\sum_h |s_h|^2} \leq \sqrt{\sum_h val_h^2} = val &
\end{aligned}$$

[1] In fact, if $b \geq 0$ then $c \geq b \geq 0 \Rightarrow |c| \geq |b|$.
If $b \leq 0$ then $a \leq b \leq 0 \Rightarrow -a \geq -b \geq 0 \Rightarrow |a| \geq |b|$.

Thus $\mathcal{S}$ is a limited set. $\square$

In the following we show that the limitedness of the Pareto Frontier under certain hypothesis implies its finiteness.

Proposition 4.5 (Sufficient condition for the finiteness of the Pareto Frontier)**.** *If the Pareto frontier is limited, the feasible set $\mathcal{X}$ is a subset of $\mathbb{Z}^n$, the objective functions are continuous sempre definite over $\mathbb{R}^n$ and a valid value ϵ_h (see definition 1) exists for each objective function, then the Pareto Frontier is finite.*

Proof. Call $\mathcal{S}$ the Pareto Frontier.
Since the frontier is limited we have $\exists\ M > 0, M \in \mathbb{R}\ :\ ||f(x)||_2 < M,\ \forall f(x) \in \mathcal{S}$.
Since the objectives are continuous over $\mathbb{R}^n$, we can define the following quantities

$$M_h = \max\left\{f_h(x)\ :\ ||f(x)||_2 \leq M,\ x \in \mathbb{R}^n\right\} < \infty$$
$$m_h = \min\left\{f_h(x)\ :\ ||f(x)||_2 \leq M,\ x \in \mathbb{R}^n\right\} > -\infty$$

Call $\epsilon = \min\limits_{h=1,\ldots,m} \{\epsilon_h\}$ with the property

$$||p_1 - p_2||_2 \geq \epsilon$$

Then each objective $f_h(x)$ can at most assume $\dfrac{|M_h - m_h|}{\epsilon}$ values. So the maximum number of possible points in the frontier is given by $\dfrac{1}{\epsilon^m}\prod_{h=1}^{m}|M_h - m_h| < \infty$. $\square$

Proposition 4.6 (Sufficient condition for the finiteness of the Pareto Frontier)**.** *If the feasible set $\mathcal{X}$ is finite, then the Pareto Frontier is finite.*

Proof. If $\zeta = |\mathcal{X}| < \infty$ then $|E| \leq \zeta < \infty$, so the Pareto Frontier has no more points than ζ. $\square$

The fundamental result of FPA is it is possible to prove that in the end all the Pareto frontier will be analyzed without overlapping between subproblems.

Theorem 4.7. *Given a valid value for ϵ_1 and ϵ_2 and a finite frontier then the FPA returns all the Pareto points.*

Proof Suppose now that $\mathcal{X}^k$ is such that $\hat{x}_k \in args_k\left(f(x)\right)$ is a point of the Pareto front.
The new problems $k+1$ and $k+2$ are

$$\mathcal{X}^{k+1} = \left\{x \in \mathcal{X}^k\ :\ f_1(x) \leq f_1(\hat{x}_k) - \epsilon_1\right\}$$

$$\mathcal{X}^{k+2} = \left\{x \in \mathcal{X}^k\ :\ f_2(x) \leq f_2(\hat{x}_k) - \epsilon_2\right\}$$

We have
$$\mathcal{X}^{k+1} \cap \mathcal{X}^{k+2} = \left\{x \in \mathcal{X}^k\ :\ f_k(x) \leq f_k(\hat{x}_k) - \epsilon_k,\ k = 1, 2\right\}\ =\ \emptyset$$

In fact if $\exists\ \tilde{x} \in \mathcal{X}^{k+1} \cap \mathcal{X}^{k+2}\ \Rightarrow\ \tilde{x} \preceq \hat{x}_k$ and definitely this implies $\hat{x}_k$ is not part of the Pareto front, on opposite to our assumption about $args_k\left(f(x)\right)$.
Let now define the set of all points dominated by $\hat{x}_k$

$$\tilde{\mathcal{X}}^k = \left\{x \in \mathcal{X}^k\ :\ f_k(x) \geq f_k(\hat{x}_k),\ k = 1, 2\right\}\ =\ \emptyset$$

Its easy to see the family $\tilde{\mathcal{X}}^k,\ \mathcal{X}^{k+1},\ \mathcal{X}^{k+2}$ is a partition[2] of $\mathcal{X}^k$.
For a fixed $i \in \{1,2\}$ one of the two sentences is true

1. $\mathcal{X}^{k+i}\ =\ \emptyset$
2. $\hat{x}_{k+i} \in \arg s_{k+i}\left(f(x)\right)$ is on the frontier

If $\mathcal{X}^{k+i}$ is empty there will be no points and neither anyone on the frontier.

To prove the second part we need to prove that if $\hat{x}_{k+i} \in E_{k+i}$ then $\hat{x}_{k+i} \in E_k$.
Suppose now by contradiction that $\hat{x}_{k+i} \notin E_k$, then $\exists \bar{x} \in \mathcal{X}_k$ such that $\bar{x} \preceq \hat{x}_{k+i}$. This implies that

$$\bar{x}\ :\ \begin{cases} f_{-i}(\bar{x}) \leq f_{-i}(\hat{x}_{k+i}) \\ f_i(\hat{x}) \leq f_i(\hat{x}_{k+i}) \end{cases}$$

with one of the two inequalities strict.

We also know that $\hat{x}_{k+i} \in \mathcal{X}_{k+i}$ so the second inequality becomes $f_i(\hat{x}) \leq f_i(\hat{x}_{k+i}) \leq f_i(\hat{x}_k) - \epsilon_i$ thus $\bar{x} \in \mathcal{X}_{k+i}$. This is in contradiction with the fact that $\hat{x}_{k+i} \in E_{k+i}$.

So by the end we have $\hat{x}_{k+i} \in E_k$ for all k and i.

By induction we have that if $\hat{x}_{k+i} \in E_k$ then $\hat{x}_{k+i} \in E_{\hat{k}}$, where $\hat{k}$ is the iteration generating k-th problem, indeed $\hat{k}\ :\ \exists\ i \in \{1,2\} : \hat{k} + i = k$.
There is only one generator node $E_0 = E$ for all the subproblems so we can conclude that $\hat{x}_{k+i} \in E$.

Since our hypothesis on the scalarization algorithm used, when $\mathcal{X}^{k+i}\ \neq\ \emptyset$ we have that

$$\hat{x}_{k+i} \in \arg s_{k+i}\left(f(x)\right) \subseteq E_{k+i}\ \Rightarrow\ \hat{x}_{k+i} \in E$$

We now must show that we will find all the points of the Frontier.
To continue with the proof let us show some points scritto male

1. A composition of partitions is a partition.

   In other words suppose to have a set A and a partition $\{B_j\}$ with $j = 1, \ldots, n$. Suppose now to have for a certain set B_j a partition $\{C_h\}$ with $h = 1, \ldots, m$. Then it is trivial to show that the family $\{B_1, B_2, \ldots, B_{j-1}, C_1, \ldots, C_m, B_{j+1}, \ldots, B_n\}$ is a partition of A.

2. Every time in the algorithm we split the k-th set, we have shown we create a partition of the set. In formulas $\left\{\tilde{\mathcal{X}}^k,\ \mathcal{X}^{k+1},\ \mathcal{X}^{k+2}\right\}$ is a partition of $\mathcal{X}_k$.

3. At the end of the algorithm we have a family of sets of the form $\left\{\tilde{\mathcal{X}}_k\right\}$ with $k \in \mathcal{K} = \{k \in \mathbb{N}\ :\ \exists\ \hat{x}_k \in \arg s_k(f(x))\}$. In fact if a set of the form $\mathcal{X}_{k+i}$ is nonempty, then we can find another point $\hat{x}_{k+i}$ and split again nad so the algorithm can continue.

4. For what we have just shown every time we find a new point it is on the frontier. Since the frontier is finite we cannot find more points than the ones on the frontier. So by the end we have a bound on the cardinality $\left|\left\{\tilde{\mathcal{X}}_k\right\}\right| < \infty$.

[2] Given a set $\mathcal{A} \subseteq \mathbb{R}^n$ a family $\mathcal{B}_1,\ \mathcal{B}_1, \ldots,\ \mathcal{B}_q$ is a partition if

$$\begin{array}{l} \mathcal{B}_i \subseteq \mathcal{A},\ i = 1, \ldots, q \\ \mathcal{B}_i \cap \mathcal{B}_j = \emptyset,\ i = 1, \ldots, q,\ i = 1, \ldots, q,\ i \neq j \\ \bigcup_{i=1}^{q} \mathcal{B}_i = \mathcal{A} \end{array}$$

Suppose now that exists a point on the frontier that our algorithm is not able to find. So suppose $\exists \bar{x} \in \mathcal{X} \ : \ \nexists \ k \in \mathcal{K} \ : \ f(\bar{x}) = s_k(f(\hat{x}_k))$.
Combining point 1, 2, 3 and 4 we have that $\left\{\tilde{\mathcal{X}}_k\right\}$ is a partition of $\mathcal{X}$, implying

$$\bar{x} \in \mathcal{X} \ \Rightarrow \ \bar{x} \in \bigcup_{k \in \mathcal{K}} \tilde{\mathcal{X}}_k \ \Rightarrow \ \exists \ k \in \mathcal{K} \ : \bar{x} \in \tilde{\mathcal{X}}_k \ \Rightarrow \ \ \exists \ k \in \mathcal{K} \ : f(\bar{x}) \geq s_k(f(\hat{x}_k))$$

in contradiction with what just said. ∎

Another important result is given by the fact that we already know how many subproblems will be solved during the computation.

Theorem 4.8. *Suppose to have a Pareto Frontier with exactly m points, then the FPA algorithm will solve exactly $2m+1$ subproblems and $m+1$ of them have empty feasible set.*

Proof. From the previous theorem we know that the algorithm ends finding all the Pareto frontier and returning the partition $\left\{\tilde{\mathcal{X}}^k\right\}$ of the feasible set $\mathcal{X}$. A set $\tilde{\mathcal{X}}^k$ is generated with other sets forming a partition of $\mathcal{X}^k$, the three sets are $\left\{\tilde{\mathcal{X}}^k, \ \mathcal{X}^{k+1}, \ \mathcal{X}^{k+2}\right\}$. So by the end we have $3m$ sets generated by the algorithm. The algorithm solves integer programs only on sets of the form $\mathcal{X}^{k+1}$ and $\mathcal{X}^{k+2}$, so in the end we have exactly $2m$ problems plus the root one. So the total number of problems solved is $2m+1$. ∎

5 Equivalence

$$\begin{cases} \min & f(x) \\ s.t. & x \in \mathcal{X} \end{cases} \tag{3}$$

Where $f : \mathcal{X} \rightarrow \mathbb{R}^m$ and $\mathcal{X}$ are such that our optimization process can always find the global optimum (e.g. Integer Programming and Convex Programming).
Now formulate the problem

$$\begin{cases} \min & t \\ s.t. & t \geq f(x) \\ & x \in \mathcal{X} \end{cases} \tag{4}$$

Where $f : \mathcal{X} \rightarrow \mathbb{R}^m$ and $\mathcal{X}$ are the same of (3).

Proposition 5.1. $\hat{x} \in E_3 \ \Rightarrow \ \exists \ \hat{t} \in \mathbb{R}^m : (\hat{t}, \hat{x}) \in E_4$ *and vice versa* $(\hat{t}, \hat{x}) \in E_4 \ \Rightarrow \ \hat{x} \in E_3$

Proof. We split the proposition into two parts.

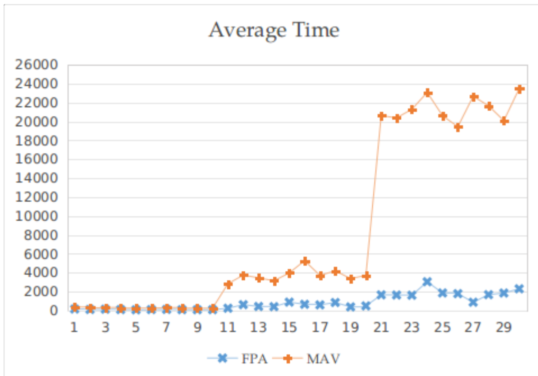
Part I. $\hat{x} \in E_3 \ \Rightarrow \ \exists \ \hat{t} \in \mathbb{R}^m : (\hat{t}, \hat{x}) \in E_4$.
Call $\hat{t} = f(\hat{x})$. Chosen $\hat{x} \in \mathcal{X}$, since $t \geq f(\hat{x}) = \hat{t}$ then $\nexists \ t \in \mathbb{R}^m : t \neq \hat{t}, \ t \preccurlyeq \hat{t}$. [3]
Since $\hat{x} \in E_3$ then $\nexists \ x \in \mathcal{X} : f(x) \preccurlyeq f(\hat{x})$ thus $\nexists \ x \in \mathcal{X}, \ t \in \mathbb{R}^m : t \neq \hat{t}, \ \hat{t} = f(\hat{x}) \geq t \geq f(x), \ f(x) \neq f(\hat{x}) \ \Rightarrow (\hat{t}, \hat{x}) \in E_4$.

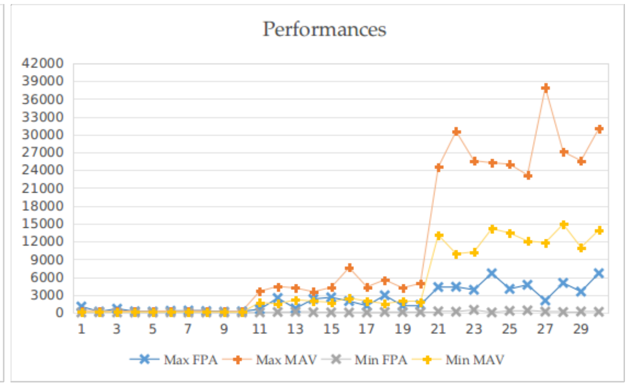
Part II. $(\hat{t}, \hat{x}) \in E_4 \ \Rightarrow \ \hat{x} \in E_3$.
We start proving

$$(\hat{t}, \hat{x}) \in E_4 \ \Rightarrow \ \hat{t} = f(\hat{x}) \tag{5}$$

In fact by contradiction suppose $(\hat{t}, \hat{x}) \in E_4$ and $\hat{t} \neq f(\hat{x})$, implying $\hat{t} > f(\hat{x})$. We can build the feasible point $\tilde{t} = f(\hat{x})$, obtaining $f(\hat{x}) = \tilde{t} < \hat{t} \ \Rightarrow \ \tilde{t} \preccurlyeq \hat{t}$ denying the fact that $(\hat{t}, \hat{x}) \in E_4$.

[3] $\nexists \ t \in \mathbb{R}^m : t \leq \hat{t}, \ t \neq \hat{t}$.





Now since we know that $(\hat{t}, \hat{x}) \in E_4$ we have that $\nexists\ (t, x) \in \mathcal{G}(x) \times \mathcal{X} : t \preccurlyeq \hat{t}$, where $\mathcal{G}(x) = \{t \in \mathbb{R}^m : t \geq f(x)\}$. Taking into account (5) we have that $\nexists\ (t, x) \in \mathcal{G}(x) \times \mathcal{X} : f(x) \leq t \preccurlyeq \hat{t} = f(\hat{x})$.
We want to show that

$$\nexists\ (t, x) \in \mathcal{G}(x) \times \mathcal{X} : f(x) \preccurlyeq f(\hat{x})$$
$$\Downarrow$$
$$\nexists\ x \in \mathcal{X} : f(x) \preccurlyeq f(\hat{x})$$

By contradiction suppose that $\exists\ x \in \mathcal{X} : f(x) \preccurlyeq f(\hat{x})$ but then there is a point $(t = f(x), x) \in \mathcal{G}(x) \times \mathcal{X} : f(x) \preccurlyeq f(\hat{x})$, and this is not possible. $\square$

6 Heuristic for Convex Problems

Even in the quadratic case once we have determined the constraint to add to the formulation, then the resulting problem is Quadratic Objective Quadratically Constrained Integer program. In general could be useful to use an heuristic which does not increment the difficulty of the feasible set.
Since we are supposing Convex objectives, once we know a value for ϵ_h (or an estimation) and comparison point $\hat{x}$, we can write

$$f_h(x) \leq f_h(\hat{x}) - \epsilon_h$$
$$\Downarrow$$
$$\nabla f_h(\hat{x})^{\intercal} (\hat{x} - y) \geq f_h(\hat{x}) - f_h(x) \geq \epsilon_h$$

We use this linearized constraints to reduce the hardness of the formulation.

7 Computational Experience

The focus of this section is on linear problems since we can mae a comparison between different methods. In particular we will provide results for FPA algorithm and the one proposed by Mavrotas in [**?**]. There is no

References

[Alves and Clímaco, 2007] Alves, M. J. and Clímaco, J. (2007). A review of interactive methods for multiobjective integer and mixed-integer programming. *European Journal of Operational Research*, 180(1):99–115.

Table 1: My caption

	Frontier Partitioner Algorithm Time (ms)				Mavrotas Algorithm Time (ms)			
Experiment	Average	Std Dev	Max	Min	Average	Std Dev	Max	Min
10 x 10 x 0	166.5	307.88	1038.0	25.0	219.2	66.35	339.0	138.0
10 x 10 x 1	117.9	39.83	188.0	68.0	181.2	23.69	230.0	148.0
10 x 10 x 2	148.3	212.27	668.0	35.0	187.2	44.97	280.0	131.0
10 x 10 x 3	104.1	65.62	228.0	24.0	161.9	34.74	214.0	103.0
10 x 10 x 4	82.6	50.84	186.0	16.0	183.1	47.64	284.0	114.0
10 x 10 x 5	108.1	88.82	321.0	50.0	177.3	37.81	239.0	119.0
10 x 10 x 6	124.3	117.78	360.0	24.0	202.8	51.14	314.0	122.0
10 x 10 x 7	96.9	75.49	271.0	30.0	162.4	27.45	203.0	125.0
10 x 10 x 8	90.6	53.35	170.0	28.0	163.4	42.34	222.0	92.0
10 x 10 x 9	94.7	65.33	230.0	31.0	174.3	48.60	257.0	90.0
20 x 10 x 0	250.7	186.44	577.0	44.0	2590.4	624.74	3618.0	1729.0
20 x 10 x 1	627.6	694.11	2476.0	117.0	3124.9	888.01	4427.0	1489.0
20 x 10 x 2	437.9	189.45	712.0	157.0	3028.2	644.90	4165.0	2184.0
20 x 10 x 3	419.2	673.93	2304.0	65.0	2731.8	624.99	3528.0	1939.0
20 x 10 x 4	896.3	830.70	2588.0	45.0	3082.4	790.46	4255.0	1749.0
20 x 10 x 5	665.4	641.08	1949.0	29.0	4614.4	1713.32	7589.0	2472.0
20 x 10 x 6	606.5	384.94	1170.0	27.0	3109.8	841.21	4344.0	1947.0
20 x 10 x 7	857.3	828.58	2932.0	119.0	3350.1	1151.39	5542.0	1533.0
20 x 10 x 8	394.6	321.91	1168.0	107.0	2988.5	645.95	4186.0	1940.0
20 x 10 x 9	496.7	390.92	1323.0	91.0	3227.5	867.74	4936.0	1776.0
30 x 10 x 0	1665.2	1474.33	4313.0	226.0	18946.5	3936.56	24517.0	13080.0
30 x 10 x 1	1646.1	1435.61	4366.0	166.0	18728.1	5238.12	30539.0	9928.0
30 x 10 x 2	1623.9	1304.13	3853.0	477.0	19658.1	5548.89	25602.0	10230.0
30 x 10 x 3	3051.2	2162.46	6631.0	38.0	20013.9	3703.17	25259.0	14154.0
30 x 10 x 4	1865.5	1433.32	4004.0	307.0	18737.6	3993.58	25010.0	13423.0
30 x 10 x 5	1791.0	1394.81	4693.0	386.0	17647.8	4006.52	23196.0	12072.0
30 x 10 x 6	885.7	605.58	2046.0	181.0	21776.9	8875.90	37929.0	11805.0
30 x 10 x 7	1686.1	1472.15	5016.0	105.0	19972.9	4039.23	27119.0	14916.0
30 x 10 x 8	1858.1	1251.97	3545.0	200.0	18225.1	5623.28	25606.0	11001.0
30 x 10 x 9	2304.7	2038.46	6645.0	149.0	21160.8	6340.89	31024.0	13923.0

Table 2: My caption

Pareto Points				FPA Number of nodes				MAV Number of nodes			
Average	Std Dev	Max	Min	Average	Std Dev	Max	Min	Average	Std Dev	Max	Min
4.1	1.85	7.0	1.0	8.2	3.71	14.0	2.0	422.4	101.20	668.0	314.0
4.9	1.10	7.0	3.0	9.8	2.20	14.0	6.0	406.8	56.25	518.0	316.0
4.4	1.58	7.0	2.0	8.8	3.16	14.0	4.0	425.6	108.77	672.0	300.0
4.3	1.89	7.0	2.0	8.6	3.78	14.0	4.0	365.4	78.93	486.0	230.0
4.3	1.64	7.0	2.0	8.6	3.27	14.0	4.0	417.0	114.29	662.0	252.0
4.2	1.48	7.0	2.0	8.4	2.95	14.0	4.0	391.6	86.18	540.0	256.0
5.4	2.07	9.0	3.0	10.8	4.13	18.0	6.0	455.2	124.29	750.0	276.0
4.9	2.33	10.0	2.0	9.8	4.66	20.0	4.0	367.2	68.11	472.0	280.0
5.1	2.73	12.0	2.0	10.2	5.45	24.0	4.0	370.2	96.98	508.0	210.0
3.6	1.78	6.0	1.0	7.2	3.55	12.0	2.0	395.6	116.09	594.0	206.0
5.1	1.97	7.0	2.0	10.2	3.94	14.0	4.0	3496.0	867.18	4952.0	2368.0
6.7	2.36	11.0	3.0	13.4	4.72	22.0	6.0	4274.8	1248.80	6186.0	2022.0
7.2	2.10	10.0	3.0	14.4	4.20	20.0	6.0	4170.8	872.65	5808.0	2924.0
6.7	2.16	10.0	3.0	13.4	4.33	20.0	6.0	3741.8	823.48	4882.0	2620.0
7.1	2.08	10.0	4.0	14.2	4.16	20.0	8.0	4232.8	1100.08	5916.0	2428.0
6.6	2.95	13.0	2.0	13.2	5.90	26.0	4.0	5282.0	1937.97	8896.0	2998.0
6.7	2.31	10.0	2.0	13.4	4.62	20.0	4.0	4037.8	985.95	5282.0	2528.0
6.9	2.96	13.0	4.0	13.8	5.92	26.0	8.0	4470.0	1516.27	7304.0	1998.0
6.9	2.33	12.0	4.0	13.8	4.66	24.0	8.0	4122.6	930.55	5852.0	2620.0
6.8	2.49	10.0	4.0	13.6	4.97	20.0	8.0	4407.6	1178.10	6734.0	2512.0
9.2	3.61	15.0	4.0	18.4	7.23	30.0	8.0	15035.0	3289.55	20074.0	10738.0
8.7	3.13	15.0	5.0	17.4	6.26	30.0	10.0	14869.4	4249.19	24760.0	8108.0
8.2	3.55	14.0	4.0	16.4	7.11	28.0	8.0	15317.2	4120.41	20376.0	8062.0
12.0	5.50	19.0	1.0	24.0	10.99	38.0	2.0	16277.4	2943.17	20772.0	11732.0
10.4	2.50	14.0	7.0	20.8	5.01	28.0	14.0	15174.0	3363.16	20428.0	10804.0
7.6	2.41	11.0	4.0	15.2	4.83	22.0	8.0	14350.4	3133.44	18680.0	9984.0
9.2	3.12	14.0	5.0	18.4	6.24	28.0	10.0	17732.2	7198.87	30698.0	9610.0
8.7	3.83	14.0	2.0	17.4	7.66	28.0	4.0	16305.6	3308.71	22354.0	12208.0
9.5	2.95	13.0	5.0	19.0	5.91	26.0	10.0	14874.0	4570.54	20740.0	9142.0
11.2	4.47	19.0	5.0	22.4	8.93	38.0	10.0	17367.8	5290.21	25890.0	11486.0

Table 3: My caption

	Time (ms)				Number of Pareto points			
Experiment	Average	Std Dev	Max	Min	Average	Std Dev	Max	Min
10 x 10 x 0	343.2	271.58	970	74	3.8	1.55	6	2
10 x 10 x 1	226.3	120.28	439	60	3.8	1.55	7	2
10 x 10 x 2	171.3	89.79	344	47	3.2	1.32	5	2
10 x 10 x 3	486.8	207.89	985	226	5.9	1.6	9	3
10 x 10 x 4	355.2	260.12	971	83	5.3	2.06	9	3
10 x 10 x 5	327.9	266.41	845	48	4.5	2.07	8	2
10 x 10 x 6	529	283.26	1209	292	6	2.31	12	4
10 x 10 x 7	144	98.9	376	51	3.4	1.17	5	2
10 x 10 x 8	464.2	259.54	875	170	4.6	1.9	7	2
10 x 10 x 9	272.1	233.35	723	24	4.3	1.89	8	1
20 x 10 x 0	1373.7	1013.7	3419	287	6.4	3.17	13	3
20 x 10 x 1	1391.7	616	2892	774	6.4	1.84	11	4
20 x 10 x 2	1564.9	1137.5	4212	329	6.6	3.1	12	3
20 x 10 x 3	1467.5	806.11	2632	313	6.5	2.64	10	3
20 x 10 x 4	1452	791.69	2827	385	6.1	2.13	10	4
20 x 10 x 5	1800.3	963.91	4084	666	7.7	2.71	13	4
20 x 10 x 6	1225.9	472.69	2192	552	5.9	1.73	10	4
20 x 10 x 7	1316	621.03	2294	333	6.2	1.81	9	4
20 x 10 x 8	1225.3	513.82	2349	334	5.6	1.35	8	3
20 x 10 x 9	1532.8	929.24	3406	222	6.6	2.67	10	2
30 x 10 x 0	3116.6	1414.71	6579	1590	8.8	3.71	18	5
30 x 10 x 1	3334.3	1713.68	6754	842	9.2	2.82	13	3
30 x 10 x 2	2753.5	1421.49	6029	1136	7.7	3.23	12	3
30 x 10 x 3	2572.7	1663.46	6294	557	7.2	3.36	14	2
30 x 10 x 4	2903.7	2422.01	7911	193	8.4	6.67	22	1
30 x 10 x 5	3385.4	1734.99	5930	475	8.4	3.92	15	2
30 x 10 x 6	3248.3	1725.48	6799	458	8.1	3.73	15	1
30 x 10 x 7	3655.4	3525.51	13027	816	9	5.58	22	3
30 x 10 x 8	3058.1	881.01	4648	1929	8	1.89	11	5
30 x 10 x 9	2297.5	966.16	4020	832	6	2.11	10	2

[Belotti et al., 2013] Belotti, P., Soylu, B., and Wiecek, M. M. (2013). A branch-and-bound algorithm for biobjective mixed-integer programs. *Optimization Online*.

[Belotti et al., 2016] Belotti, P., Soylu, B., and Wiecek, M. M. (2016). Fathoming rules for biobjective mixed integer linear programs: Review and extensions. *Discrete Optimization*, 22:341–363.

[Boland et al., 2016] Boland, N., Charkhgard, H., and Savelsbergh, M. (2016). The l-shape search method for triobjective integer programming. *Mathematical Programming Computation*, 8(2):217–251.

[Boland et al., 2017a] Boland, N., Charkhgard, H., and Savelsbergh, M. (2017a). A new method for optimizing a linear function over the efficient set of a multiobjective integer program. *European Journal of Operational Research*, 260(3):904–919.

[Boland et al., 2017b] Boland, N., Charkhgard, H., and Savelsbergh, M. (2017b). The quadrant shrinking method: A simple and efficient algorithm for solving tri-objective integer programs. *European Journal of Operational Research*, 260(3):873–885.

[Büsing et al., 2017] Büsing, C., Goetzmann, K.-S., Matuschke, J., and Stiller, S. (2017). Reference points and approximation algorithms in multicriteria discrete optimization. *European Journal of Operational Research*, 260(3):829–840.

[Cacchiani and DAmbrosio, 2017] Cacchiani, V. and DAmbrosio, C. (2017). A branch-and-bound based heuristic algorithm for convex multi-objective minlps. *European Journal of Operational Research*, 260(3):920–933.

[Che et al., 2017] Che, A., Kats, V., and Levner, E. (2017). An efficient bicriteria algorithm for stable robotic flow shop scheduling. *European Journal of Operational Research*, 260(3):964–971.

[Gabbani and Magazine, 1986] Gabbani, D. and Magazine, M. (1986). An interactive heuristic approach for multi-objective integer-programming problems. *Journal of the Operational Research Society*, pages 285–291.

[Gutjahr and Pichler, 2016] Gutjahr, W. J. and Pichler, A. (2016). Stochastic multi-objective optimization: a survey on non-scalarizing methods. *Annals of Operations Research*, 236(2):475–499.

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

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

[Martin et al., 2017] Martin, B., Goldsztejn, A., Granvilliers, L., and Jermann, C. (2017). Constraint propagation using dominance in interval branch & bound for nonlinear biobjective optimization. *European Journal of Operational Research*, 260(3):934–948.

[Mavrotas, 2009] Mavrotas, G. (2009). Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation*, 213(2):455–465.

[Mavrotas and Diakoulaki, 1998] Mavrotas, G. and Diakoulaki, D. (1998). A branch and bound algorithm for mixed zero-one multiple objective linear programming. *European Journal of Operational Research*, 107(3):530–541.

[Mavrotas and Diakoulaki, 2005] Mavrotas, G. and Diakoulaki, D. (2005). Multi-criteria branch and bound: A vector maximization algorithm for mixed 0-1 multiple objective linear programming. *Applied mathematics and computation*, 171(1):53–71.

[Moradi et al., 2015] Moradi, S., Raith, A., and Ehrgott, M. (2015). A bi-objective column generation algorithm for the multi-commodity minimum cost flow problem. *European Journal of Operational Research*, 244(2):369–378.

[Przybylski and Gandibleux, 2017] Przybylski, A. and Gandibleux, X. (2017). Multi-objective branch and bound. *European Journal of Operational Research*, 260(3):856–872.

[Przybylski et al., 2010] Przybylski, A., Gandibleux, X., and Ehrgott, M. (2010). A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives. *Discrete Optimization*, 7(3):149–165.

[Raith and Ehrgott, 2009] Raith, A. and Ehrgott, M. (2009). A two-phase algorithm for the biobjective integer minimum cost flow problem. *Computers & Operations Research*, 36(6):1945–1954.

[Ralphs et al., 2004] Ralphs, T. K., Saltzman, M. J., and Wiecek, M. M. (2004). An improved algorithm for biobjective integer programming and its application to network routing problems. *Annals of Operations Research*, 73:253–280.

[Ralphs et al., 2006] Ralphs, T. K., Saltzman, M. J., and Wiecek, M. M. (2006). An improved algorithm for solving biobjective integer programs. *Annals of Operations Research*, 147(1):43–70.

[Ramesh et al., 1986] Ramesh, R., Zionts, S., and Karwan, M. H. (1986). A class of practical interactive branch and bound algorithms for multicriteria integer programming. *European Journal of Operational Research*, 26(1):161–172.

[Rezaee et al., 2017] Rezaee, A., Dehghanian, F., Fahimnia, B., and Beamon, B. (2017). Green supply chain network design with stochastic demand and carbon price. *Annals of Operations Research*, 250(2):463–485.

[Sedeño-Noda and González-Martın, 2001] Sedeño-Noda, A. and González-Martın, C. (2001). An algorithm for the biobjective integer minimum cost flow problem. *Computers & Operations Research*, 28(2):139–156.

[Stidsen et al., 2014] Stidsen, T., Andersen, K. A., and Dammann, B. (2014). A branch and bound algorithm for a class of biobjective mixed integer programs. *Management Science*, 60(4):1009–1032.

[Sylva and Crema, 2004] Sylva, J. and Crema, A. (2004). 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.

[Villarreal and Karwan, 1981] Villarreal, B. and Karwan, M. H. (1981). Multicriteria integer programming: A (hybrid) dynamic programming recursive approach. *Mathematical programming*, 21(1):204–223.