Annals of Operations Research (2021) 296:35–69
https://doi.org/10.1007/s10479-019-03317-9

S.I.: MOPGP 2017

An exact scalarization method with multiple reference points for bi-objective integer linear optimization problems

Angelo Aliano Filho[1] · **Antonio Carlos Moretti**[2] · **Margarida Vaz Pato**[3] · **Washington Alves de Oliveira**[4]

Published online: 24 July 2019

Abstract
This paper presents an exact scalarization method to solve bi-objective integer linear optimization problems. This method uses diverse reference points in the iterations, and it is free from any kind of a priori chosen weighting factors. In addition, two new adapted scalarization methods from literature and the modified Tchebycheff method are studied. Each one of them results in different ways to obtain the Pareto frontier. Computational experiments were performed with random real size instances of two special problems related to the manufacturing industry, which involve lot sizing and cutting stock problems. Extensive tests confirmed the very good performance of the new scalarization method with respect to the computational effort, the number of achieved solutions, the ability to achieve different solutions, and the spreading and spacing of solutions at the Pareto frontier.

Keywords Bi-objective optimization problems · Integer linear optimization · Exact scalarization methods

1 Introduction

A number of practical optimization problems in various branches of science, such as engineering, physics, biology, and chemistry, involve concomitantly optimizing several conflicting objectives, as minimizing the cost of production, setup, time, environmental impact, and maximizing profit, machine efficiency, safety, etc. These situations cannot be modeled with mono-objective methodologies, and hence specialized methodologies are required. Optimization problems of this nature are reported as multi-objective optimization problems (MOPs)

✉ Angelo Aliano Filho
angeloaliano@utfpr.edu.br

1 Academic Department of Mathematics, Federal Technological University of Paraná, Apucarana, Brazil

2 Institute of Mathematics, Statistics and Scientific Computation, University of Campinas, Campinas, Brazil

3 ISEG and CMAFcIO, Universidade de Lisboa, Lisbon, Portugal

4 School of Applied Sciences, University of Campinas, Limeira, Brazil

(Ehrgott and Gandibleux 2002). Usually, no single point will optimize all given objective functions simultaneously and the concept of optimality can be replaced by the concept of Pareto optimality. A solution is called Pareto optimal, for the minimization case, if there does not exist a different solution with the same or smaller objective function values, such that there is a decrease in at least one objective function value. Consequently, a MOP has a non-unitary set of optimal solutions which establish a compromise between the objectives. This variety of solutions can support more reliable and secure decisions.

Applications for MOPs can be found in engineering (Solanki 1991; Clímaco et al. 1997a; Czyzżak and Jaszkiewicz 1998; Sylva and Crema 2004; Ehrgott and Gandibleux 2000, 2002; Laumanns et al. 2006; Ralphs et al. 2006; Dächert et al. 2012; Dai and Charkhgard 2018), in industry, namely, cutting stock problems (Golfeto et al. 2009; Aliano Filho et al. 2018), lot-sizing problems (Ustun and Demirtas 2008), scheduling problems (Xia and Wu 2005), in truss and structural optimization problems (Das and Dennis 1998; Messac et al. 2003; Kim and de Weck 2005; Deb 2014), in logistics, namely, routing problems, (Ehrgott and Gandibleux 2000; Bérubé et al. 2009), transportation (Madsen et al. 1995), assignment problems (Özlen and Azizoğlu 2009; Anthony et al. 2009), design (Deb and Sundar 2006; Shim et al. 2017), aircraft planning (Schandl et al. 2001), and in space exploration (Deb 2014). Applications arise also in statistics (Miettinen et al. 2009), management science (Miettinen 1999; Cohon 2003), social analysis (Luque 2015), cancer radiotherapy treatment planning (Shao and Ehrgott 2016), etc.

Approximation algorithms for combinatorial and continuous mono-objective problems have been adapted for MOP. Deb and Kalyanmoy (2001) have reported several evolutionary algorithms. Czyzżak and Jaszkiewicz (1998) have proposed a simulated annealing algorithm for approaching the Pareto solutions set while maintaining a uniform distribution of the generated solutions. Madsen et al. (1995) have developed a solution algorithm based on an insertion heuristic for a dial-a-ride problem, where each solution reflects the user's preferences through a flexible way for weighting of the various goals. The interactive polyhedral outer approximation algorithm of Lazimy (2013) was used to solve MOPs with non-linear and non-differentiable objective and constraint functions, and with continuous or discrete decision variables.

Moreover, current commercial solvers of excellent quality facilitate to refine and adapt exact methods for MOP. Some difficult problems can be solved in minutes or even seconds with the powerful tools available. Also, as described in the following sections, they are essential to prove that the solutions obtained by multi-objective methods are Pareto optimal or not.

One of the main exact solution strategies for MOPs is based on the scalarization approach where the original problem is transformed into a classical mono-objective optimization subproblem. (Bowman 1976; Benson 1978; Clímaco et al. 1997b; Das and Dennis 1998; Messac et al. 2003; Kim and de Weck 2005; Sayin and Kouvelis 2005; Özlen and Azizoğlu 2009; Bérubé et al. 2009; Dächert et al. 2012; Dai and Charkhgard 2018; Aliano Filho et al. 2018). Under certain conditions, one or several parameterized subproblems are solved, resulting in a corresponding number of Pareto optimal solutions. Thus, the scalarization methods differ from each other by the number of subproblems generated and the way of modifying each of them. It is advisable that the ratio between the number of subproblems solved and the number of distinct solutions found is close to one. However, for certain types of MOP, there are scalarization methods that do not provide all the Pareto optimal solutions.

Exact optimization methods for MOPs that do not scalarize have been developed, for instance, the lexicographic optimality, max-ordering optimality, lexicographic max-ordering optimization, lexicographic goal programming, and interactive methods (see, e.g., Miettinen 1999; Ehrgott 2005; Branke et al. 2008 for an overview on the subject).

This paper deals with exact methods based on scalarizations to solve the bi-objective integer linear optimization problem (BOILOP), which has practical applications. For instance, problems involving production dynamics have two conflicting aspects: the cost of production and the machine setup. The combinatorial nature of BOILOP turns it harder to solve by scalarization methods, due to the NP-hardness of each scalar optimization subproblem and also because some scalarization techniques cannot determine all the distinct solutions of BOILOP. These factors motivate the employment and development of new effective methods capable of generating a large and diversified set of solutions, while solving a smaller number of integer subproblems.

The central contribution of this study is to provide improvements and extensions on solution methods for BOILOP, by proposing and adapting existing optimization techniques to take advantage of the current powerful computers and solvers. We compare four scalarization methods on the basis of extensive computational tests. The main contributions include: (i) the proposal of a new scalarization method that considers various reference points during its iterations; (ii) the presentation of a slight modification of the Benson method; (iii) the adaptation of the normal constraints method for BOILOP; (iv) the proposal of a new bi-objective lot sizing model; and (v) the determination of new optimal solutions for a set of benchmark problems and the extensive computational tests for two production planning problems: the one-dimensional cutting stock and the lot sizing problems.

The paper is organized as follows. Section 2 presents a brief literature review of the studies on exact scalarization methods for MOP. Section 3 provides the basic concepts and notation used for the development of this article. Section 4 presents the new scalarization method here proposed, and its behavior is evidenced with a numerical example in Appendix A. Section 5 describes the other scalarization methods that were implemented and compared. Section 6 presents the bi-objective models for the one-dimensional cutting stock and the lot-sizing problems. Section 7 presents the computational experiments to validate these algorithms. The last section provides a summary of our findings and some clues for future research.

2 Literature review for exact scalarization methods

Exact methods for MOPs based on scalarizations were developed from the 1970s to the present day. This branch of operations research has received diversified attention with the discovery and improvement of several methods. Clímaco et al. (1997b), Ehrgott and Gandibleux (2000), Ehrgott and Gandibleux (2002), Ruzika and Wiecek (2003), and Ruzika and Wiecek (2005) have provided a complete review with up to 50 different exact methods, as well as approximation methods for MPO with discrete and continuous variables. The Benson method (Benson 1978) has been slightly changed and tested in this paper for problems with integer variables. Some features of other scalarization methods initially proposed for continuous variables were used to develop our new method for discrete variables, such as the methods from Bowman (1976) based in the Tchebycheff metric, and from Cohon (2003) for linear MOP.

Due to easy handling, specific literature for MOP reports weighted sum and ε-constrained methods as the most used approaches. However, the weighted sum method can generate a tiny number of Pareto optimal solutions for MOP, since it is not able to determine Pareto optimal solutions that do not belong to the convex hull of the Pareto frontier (Miettinen 1999). The ε-constrained method overcomes this difficulty, but it has the disadvantage of being influenced by the values taken by the objective functions in the feasible region (Ehrgott 2005).

The classification of the scalarization methods is done according to the way of modifying the constraints set or the objective function of the original MOP in each subproblem to calculate each Pareto optimal solution. The methods from Clímaco et al. (1997a) and Klein and Hannan (1982) specialized on finding an ideal branching within a branch-and-bound routine. Other papers have proposed methods based on weighted norms, such as Clímaco et al. (1997a), Dächert et al. (2012), Kettani et al. (2004), Neumayer and Schweigert (1994) and Ralphs et al. (2006); and methods based on suitable modifications of the constraints set, such as Bérubé et al. (2009), Sayin and Kouvelis (2005), Schandl et al. (2001), Solanki (1991) and Özlen and Azizoğlu (2009). Steuer and Choo (1983) have overcome specific difficulties from the original method of Bowman (1976), and developed an interactive method based on Tchebycheff metric, in which the decision maker can influence the search for Pareto optimal solutions on certain regions of the criterion space.

The normal constraint method (Messac et al. 2003) is a procedure with the structure similar to the classic ε-constraint method. It was developed to determine equally spaced non-dominated vectors for MOPs with continuous variables. In this paper, we present extensive computational tests relative to applications of this method with discrete variables, and we compare the normal constraint method with other procedures also presented in this study.

Sylva and Crema (2004) have used the ideas of Klein and Hannan (1982) to introduce a solution method for an integer linear MOP, in which a new solution is obtained from a set of the prior Pareto optimal solutions by eliminating regions of the criterion space. These regions contain the dominated vectors, and the outcome of the method are all Pareto optimal solutions. Benson (1998) has also developed a finite algorithm that generates all Pareto optimal solutions for a linear MOP. Anthony et al. (2009) have highlighted and compared four solution methods for the tri-objective assignment problem, and draw a parallel with the same method by Sylva and Crema (2004). The technique developed by Miettinen et al. (2009) corresponds to a multi-objective algorithm that uses reference points, aspiration levels and adjustment of weights in the scalar objective functions to determine the Pareto optimal solution set. Similarly, Luque (2015) has employed similar concepts of the ones in Miettinen et al. (2009), however for non-linear continuous and non-convex problems.

Das and Dennis (1998) have introduced an alternative method for finding Pareto optimal solutions for the general MOP when if an evenly distributed set of parameters for a specific subproblem is given. Their method produces an evenly distributed set of Pareto optimal solutions. By using additional inequality constraints, Kim and de Weck (2005) have adapted the weighted sum method to solve bi-objective optimization problems, so that the non-convex region of the Pareto frontier can be determined. Luque et al. (2010) have introduced an interactive technique for MOPs based on the Tchebycheff metric. In each iteration, the decision maker chooses a particular reference point to build an objective function that prioritizes a certain region of their preference in the criterion space.

Shukla (2007) has developed an algorithm to obtain a representative set for MOP having a uniformly spaced non-dominated vectors set. Luque et al. (2012) have proposed a scalarization procedure that introduces an objective function with particular weights, in which feasible and infeasible reference points are weighted. Thus the decision maker can better reflect on preferences. The paper of Luque et al. (2015) has also involved a procedure based on modifications in the reference point. Laumanns et al. (2006) have proposed a scalarization method that solves a sequence of constrained scalar subproblems to obtain an approximation of the Pareto optimal set. The right-side of the constraints set are not known, and an adaptive scheme generates appropriated values for it during each iteration of the procedure. Shao and

Ehrgott (2016) have developed a way to represent a set of well-distributed non-dominated vectors along the Pareto frontier for non-linear and non-convex MOPs. For solving a particular engineering problem, Shim et al. (2017) have compared several multi-objective optimization techniques and improved the method from Das and Dennis (1998). Ouaïl and Chergui (2018) have recently proposed an exact algorithm for integer quadratic convex problem, where a branch-and-bound based technique allows to reduce the search area by truncating domains containing non-efficient solutions without having to enumerate them.

Aliano Filho et al. (2018) have recently implemented a modified version of the Thebycheff method and tested the ε-constraint and weighted sum methods for the one-dimensional cutting stock problem. The modified method uses weighted deviations from the objective functions and updated constraints set for calculating each Pareto optimal solution. The statistical summary has shown a better computational efficiency than the ε-constraint method. However, the modified method is unable to find all Pareto optimal solutions. The new approach in the present study overcomes this disadvantage. The recent technique of Dai and Charkhgard (2018) that solves a smaller number (compared to the previous study) of mono-objective subproblems is based on the ε-constraint method, uses constraints for bounding the objective functions and employs two stages to calculate all Pareto optimal solutions of BOILOP.

Wierzbicki (1980) has reported techniques involving scalarization and modification of reference points. In this paper we find the initial results and developments on the application and use of reference points. Solanki (1991) has presented a method based on Tchebycheff metric for integer bi-objective problems, which was reasoned from the non-inferior set estimation method of Cohon (2003). The latter method determines many non-dominated vectors in the criterion space and evaluates the properties of the line segments between each pair. A reference (aspiration) point indicates the direction to find the next feasible solution by solving a weighted subproblem. However, some subproblems can generate the same Pareto optimal solution obtained in the previous iteration. Although Ralphs et al. (2006) have used the same former principle, they have improved the procedure of Cohon (2003) by employing a strategy to update the precision of guiding the consecutive reference points in the criterion space. Deb and Sundar (2006) have associated an evolutionary multi-objective algorithm with a preference based strategy and have obtained a preferred set of solutions near the multiple reference points. Luque et al. (2009) have incorporated the preference information in its interactive algorithm, where the main idea was to take the opinion of the decision maker into account more closely when projecting the reference point onto the set of non-dominated solutions. Molina et al. (2009) have changed the concept of Pareto dominance to **g**-dominance, which is based on the information included in a reference point and designed to be used with any evolutionary multi-objective algorithm. This new concept approximates the efficient set around the area of the most preferred point. The principal differences between the method proposed in our paper and the last two ones are: (i) the definition of each scalar subproblem, (ii) the absence of any kind of a prior chosen weighting factors, (iii) the updating of the reference point, and (iv) the stopping criterion. Although they are based on the similar ideas of Solanki (1991) and Ralphs et al. (2006), from the mathematical point of view, these four aspects are fundamental to characterize the novelty of our method.

BOILOP appears in several real situations. In particular, it can be applied to production engineering, such as cutting stock, lot sizing, scheduling and sequencing production, and integrated problems of the mentioned types. Aliano Filho et al. (2018) and Golfeto et al. (2009) have presented some applications for the one-dimensional cutting stock problem. Sawik (2007) and Xia and Wu (2005) studied some exact methods for the BOILOP applied to production scheduling, while Ustun and Demirtas (2008) have applied multi-objective methods for lot-sizing problems with supplier selections. Deb (2014) has presented a large number of applications for this particular field of operations research.

3 Bi-objective problem formulation

Let $\mathbb{R}_+$ be the set of non-negative real numbers and x^T the transpose of $x \in \mathbb{R}^n$. The inequality between two vectors is to be understood in a componentwise sense. Considering that $x, y \in \mathbb{R}^m$ and $i \in I = \{1, \ldots, m\}$, then $x < y$ if and only if $x_i < y_i, \forall i \in I$; $x \leqq y$ if and only if $x_i \leqq y_i, \forall i \in I$; and $x \leq y$ if and only if $x \leqq y$ and $\exists i \in I$ such that $x_i \neq y_i$.

The BOILOP involves determination an $n-$dimensional vector of variables $x = (x_1, \ldots, x_n)^T \in \mathbb{Z}^n_+$ that satisfies the following mathematical formulation:

$$\begin{aligned} &\text{Minimize } z(x) = (z_1(x), z_2(x))^T \\ &\text{subject to } Ax = b, \\ &\qquad\qquad x \in \mathbb{Z}^n_+, \end{aligned} \tag{1}$$

where the optimization is considered in the Pareto perspective, $z(x)$ is the objective vector, $z_1(x) = c^T x$, $z_2(x) = d^T x$, and $c, d \in \mathbb{Z}^n$ are parameters. $A \in \mathbb{R}^{m \times n}$ is the constraints matrix, and $b \in \mathbb{R}^m$ is the constraints right-side vector. We denote by X the *feasible* or *decision space* of elements $x \in \mathbb{Z}^n_+$ satisfying $Ax = b$. If $x \in X$, then x is a *feasible solution*. We denote by Z the *criterion* or *objective space* of elements $z \in \mathbb{Z}^2$ satisfying $z = z(x)$ and $x \in X$.

We assume that the objective functions $z_1 = z_1(x)$ and $z_2 = z_2(x)$ are in conflict with each other over X, i.e., we look for objective vectors such that none of the components of each of those vectors can be improved without deterioration of at least one of the other components of the vector.

3.1 Definitions

The following definitions are according to Ehrgott (2005). To simplify, wherever it is possible, we refer to $x \in X$ as a point or a solution, and $z \in Z$ as a vector.

Definition 1 (*Dominance*) If $z(x) \leq z(y)$, we say that $x \in X$ *dominates* $y \in X$ for BOILOP.

Definition 2 (*Pareto optimal solution*) A feasible solution x^* is said to be a *Pareto optimal solution* of BOILOP if there does not exist another solution $x \in X$ such that $z(x) \leq z(x^*)$.

Definition 3 (*Pareto optimal solution set*) The *Pareto optimal solution set* is denoted by X^*. Note that $X^* \subseteq X$.

Definition 4 (*Non-dominated vector*) A vector $z^* \in Z$ is said to be *non-dominated vector* of BOILOP if $z^* = z(x^*)$ such that $x^* \in X^*$.

Definition 5 (*Non-dominated vector set*) The *non-dominated vector set* or the *Pareto frontier*, the set of non-dominated vectors, is denoted by Z^*. Note that $Z^* \subseteq Z$.

Definition 6 (*Supported solution and vector*) A Pareto optimal solution x^* is said to be *supported* if there exists $w = (w_1, w_2)^T > 0$ such that x^* is the unique optimal solution of the weighted scalar problem $\min\{w^T z(x) \mid x \in X\}$. The resulting $z^* = z(x^*)$ is a *supported non-dominated vector*. And x^* and z^* are called *unsupported Pareto optimal solution* and

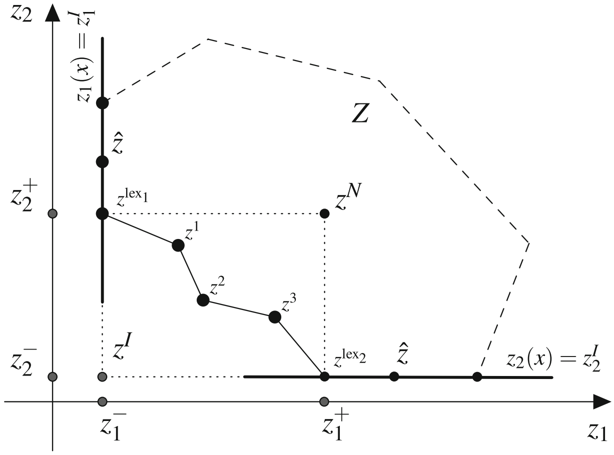

Fig. 1 Vectors in the objective space and their respective geometric interpretations

unsupported non-dominated vector, respectively, if x^* is a Pareto optimal solution, but it is not a unique optimal solution of the weighted scalar problem.

Definition 7 (*Ideal vector*) The vector $z^I = (z_1^I, z_2^I)^T \in Z$ is the *ideal vector* of BOILOP if its kth component is $z_k^I = \min\{z_k(x) \mid x \in X\}$ for $k = 1, 2$.

Definition 8 (*Nadir vector*) The vector $z^N = (z_1^N, z_2^N)^T \in Z$ is the *nadir vector* of BOILOP if its kth component is $z_k^N = \max\{z_k(x) \mid x \in X\}$ for $k = 1, 2$.

Definition 9 (*Lexicographic vectors*) The vectors z^{lex_1} and z^{lex_2} are the *lexicographic vectors* of BOILOP if each one of them dominates all vectors in lines $z_1(x) = z_1^I$ and $z_2(x) = z_2^I$, respectively.

Definition 10 (*Weakly Pareto optimal solution*) A feasible solution x^* is said to be *weakly Pareto optimal solution* of BOILOP if there does not exist another solution $x \in X$ such that $z(x) < z(x^*)$.

Definition 11 (*Weakly non-dominated vector*) A vector $\hat{z} \in Z$ is *weakly non-dominated vector* of BOILOP if $\hat{z} = z(\hat{x})$ such that $\hat{x} \in X$ is a weakly Pareto optimal solution.

If x^{*1} is the optimal solution of problem $\min\{z_2(x) \mid z_1(x) = z_1^I$ and $x \in X\}$, then we denote $z^{\text{lex}_1} = (z_1(x^{*1}), z_2(x^{*1}))^T = (z_1^-, z_2^+)^T$. Similarly, if x^{*2} is the optimal solution of problem $\min\{z_1(x) \mid z_2(x) = z_2^I$ and $x \in X\}$, then $z^{\text{lex}_2} = (z_1(x^{*2}), z_2(x^{*2}))^T = (z_1^+, z_2^-)^T$. Therefore, $z^I = (z_1^-, z_2^-)^T$ and $z^N = (z_1^+, z_2^+)^T$, and in this paper we define the solutions x^{*1} and x^{*2} as the *lexicographic solutions* for BOILOP.

Figure 1 illustrates Definitions 7–11. The Pareto frontier is represented by five vectors whose endpoints are the lexicographic vectors z^{lex_1} and z^{lex_2}. One supported vector is labeled by vector z^2 and unsupported vectors are illustrated by vectors z^1 and z^3. Ideal e nadir vectors, naturally, are not included in the Pareto frontier, but the nadir vector can belong to the objective space Z. The symbol $\hat{z}$ represents two examples of weakly Pareto optimal solutions.

3.2 Tchebycheff scalarization method (TCH method)

Bowman (1976) showed a relation between the Tchebycheff norm and the Pareto frontier. For $w \geqq 0$, the TCH method is based on the minimization of the subproblem:

$$\begin{array}{l} \text{Minimize } \max \left\{ w\left(z_1(x) - z_1^-\right),\ (1-w)\left(z_2(x) - z_2^-\right) \right\} \\ \text{subject to } x \in X. \end{array} \tag{2}$$

The following theoretical results for Problem (2) are well known (Ehrgott 2005).

Theorem 1 *If $w > 0$, then an optimal solution x^* of Problem (2) is weakly Pareto optimal solution of BOILOP.*

Theorem 2 *If Problem (2) has a single optimal solution x^*, then $x^* \in X^*$ for BOILOP.*

Theorem 3 *If $x^* \in X^*$ for BOILOP, then there is a weight $w > 0$ for which x^* is optimal solution for Problem (2).*

Note that the reference point for this method is z^I, and by selecting suitable weights w, Theorems 1–3 establish that all Pareto optimal solutions of BOILOP can be achieved.

Next, we present the four methods implemented in this paper. In each case, after performing an appropriate number of iterations, the Pareto filter procedure proposed by Deb and Kalyanmoy (2001) is applied to select from the list of solutions obtained only those that are Pareto optimal solutions. Each method takes advantage of the criterion space geometry to add new constraints involving the objective functions of each subproblem. Different Pareto frontiers can be obtained depending on the rule used. Moreover, the four methods achieve supported and unsupported solutions.

4 Exact scalarization method with multiple reference points

The multiple reference vectors scalarization method (MRV method) is based on the noninferior set estimation (NISE) method from Cohon (2003), and the methods from Ralphs et al. (2006), and Solanki (1991). While the TCH method uses weighted deviations from the ideal vector, the MRV method obtains different Pareto optimal solutions by modifying the vector of reference in each iteration, and it does not use weighted deviations from reference vectors. The MRV method can achieve all Pareto optimal solutions of BOILOP. It differs from TCH method in three aspects: (i) vector of reference; (ii) free from any kind of a priori chosen weighting factors; and (iii) stopping criterion.

In this approach, the subproblems are solved by applying the TCH method after a suitable adaptation. At each iteration k, at most 2^k subproblems are solved. For Subproblem t, $t = 1, \ldots, 2^k$, let x_t^{*1} and x_t^{*2} be two Pareto optimal solutions obtained in the iteration $k-1$, and by simple sorting we suppose that $z_1(x_t^{*1}) < z_1(x_t^{*2})$ and $z_2(x_t^{*1}) > z_2(x_t^{*2})$. In particular, $z(x_t^{*1})$ and $z(x_t^{*2})$ are endvectors of a portion of the Pareto frontier (see Fig. 2). In this portion, we choose the endpoints as the current lexicographic vectors $z_t^{\text{lex}_1} = z(x_t^{*1})$ and $z_t^{\text{lex}_2} = z(x_t^{*2})$, the ideal vector $z_t^I = (z_1(x^{*1}), z_2(x^{*2}))^T$, and the nadir vector $z_t^N = (z_1(x^{*2}), z_2(x^{*1}))^T$.

The coordinates of the vectors $A = z_t^{\text{lex}_1}$, $B = z_t^{\text{lex}_2}$, $R = z_t^I$, and $N = z_t^N$ shape the corners of the rectangle $ARBN$. If TCH method with the weight $w = 1/2$ and the reference vector $R = z_t^I$ is applied in this subproblem, the outcome of procedure may

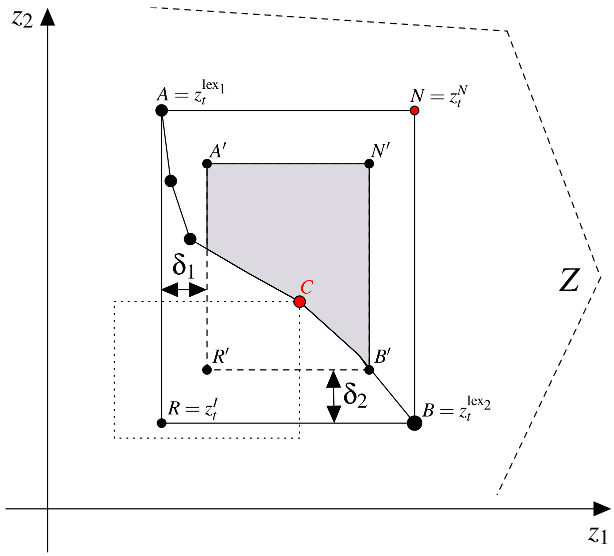

Fig. 2 Geometric illustration of Subproblem t in the criterion space for MRV method at iteration k

be a solution whose image coincides with the unwanted A or B. To avoid this, the shifted rectangle $A'R'B'N'$ inside the original rectangle $ARBN$ is created, where $A' = (A'_1, A'_2)^T = (A_1 + \delta_1, A_2 - \delta_2)^T$, $R' = (R'_1, R'_2)^T = R + \delta$, $B' = (B'_1, B'_2)^T = (B_1 - \delta_1, B_2 + \delta_2)^T$, and $\delta = (\delta_1, \delta_2)^T > 0$ is calculated from $\delta_1 = \frac{B_1 - A_1}{\ell}$ and $\delta_2 = \frac{A_2 - B_2}{\ell}$, and ℓ is a suitably chosen and sufficiently large positive real number. Thus, by solving the subproblem with the current reference vector R' (different from original ideal vector), a new non-dominated vector (for example, C in red color in Fig. 2) can be found. If such a vector does not exist, then the following subproblem (defined by Problem (3)) is infeasible:

$$\begin{aligned}
&\text{Minimize } u\\
&\text{subject to } z_1(x) - R'_1 \leqq u,\\
&\qquad\qquad z_2(x) - R'_2 \leqq u,\\
&\qquad\qquad A'_1 \leqq z_1(x) \leqq B'_1,\\
&\qquad\qquad A'_2 \leqq z_2(x) \leqq B'_2,\\
&\qquad\qquad x \in X,\ u \geqq 0.
\end{aligned} \tag{3}$$

The MRV procedure starts in iteration $k = 0$ with the original lexicographic vectors z^{lex_1} and z^{lex_2} (current successive pair of non-dominated vectors). Then, it searches for a new Pareto optimal solution whose image is within of the rectangle $A'R'B'N'$ (Subproblem $t = 1$) in the criterion space created from the coordinates of vectors $[z_1^{\mathrm{lex}_1}, z_1^{\mathrm{lex}_2}]$ by solving Problem (3). If a new Pareto optimal solution x^* is found in this rectangle, two new rectangles are created (Subproblem $t = 1, 2$) in the iteration $k = 1$ by using 2^1 new current successive pairs of non-dominated vector $[z_1^{\mathrm{lex}_1}, z_1^{\mathrm{lex}_2}] = [z^{\mathrm{lex}_1}, z(x^*)]$ and $[z_2^{\mathrm{lex}_1}, z_2^{\mathrm{lex}_2}] = [z(x^*), z^{\mathrm{lex}_2}]$. For up to 2^{k-1} new Pareto optimal solutions reached from Problem (3) in iteration $k-1$, the procedure creates at most 2^k new rectangles (Subproblem $t = 1, \ldots, 2^k$) by using each pair of non-dominated vectors $[z_t^{\mathrm{lex}_1}, z_t^{\mathrm{lex}_2}]$ in the criterion space to search for other solutions. This search is repeated for each successive pair of non-dominated vectors, and the algorithm ends when all new subproblems are infeasible (stopping criterion). Specific details of iterations $k = 0, 1$ for the MRV method are illustrated in Fig. 3.

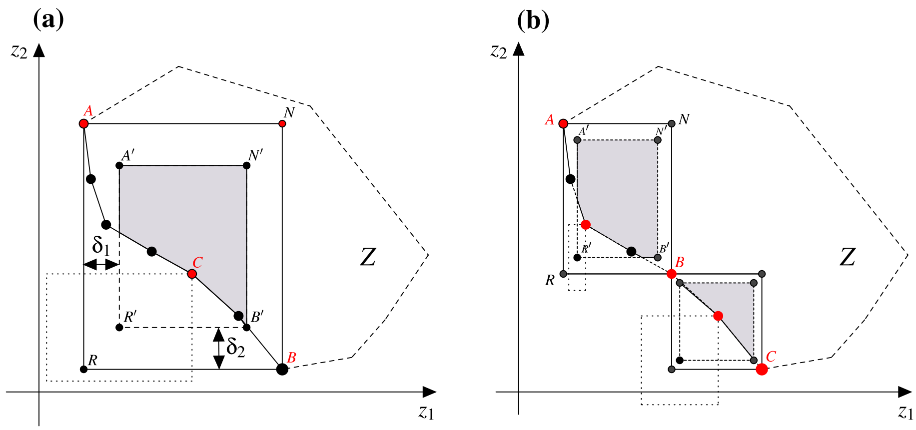

Fig. 3 Geometric illustration of two subproblems of MRV method: **a** iteration $k = 0$ and **b** iteration $k = 1$

Note that δ_1 and δ_2 are recalculated for each subproblem, thus the Euclidean distance between the non-dominated vectors found at the end of the method is at least $\sqrt{\delta_1^2 + \delta_2^2}$. If one subproblem does not have solution, then in the next iteration two subproblems will not be solved.

Since each subproblem is a particular case of Problem (2), the MRV method is well-funded in Theorems 1–3 and, depending on suitable choice of ℓ, it can be used to achieve all Pareto optimal solutions. Its main steps are described in Algorithm 4.1, where the pair of non-dominated vectors is denoted by PNDV.

Appendix A.1 provides an iterative numerical example for the MRV method and Sect. 6 presents extensive computational experiments to validate our approach.

5 Adaptation of other solution methods for comparison

This section includes three solution methods to solve BOILOP. Two of them are slightly modified methods from literature. A modification of the objective function in order to generate equally spaced non-dominated vectors is proposed for the Benson method. A way of varying the constraints set to determine well-distributed non-dominated vectors is developed for the normal constraints method. This modification was introduced in a previous work by Aliano Filho et al. (2018). The details are presented in the following subsections.

5.1 Modified Benson method

The scalarization Benson method was developed by Benson (1978) aiming to solve a more general case of MOP. However, with respect to BOILOP, if x^0 is a feasible solution and $z^0 = (z_1^0, z_2^0)^T$ is its image, Benson method is defined by the following subproblem:

$$
\begin{aligned}
&\text{Maximize } l_1 + l_2 \\
&\text{subject to } \; z_1^0 - z_1(x) = l_1 \\
&\qquad\qquad\quad z_2^0 - z_2(x) = l_2 \\
&\qquad\qquad\quad x \in X,\; l_1 \geqq 0,\; l_2 \geqq 0.
\end{aligned} \tag{4}
$$

Algorithm 4.1 Multiple reference vectors routine for BOILOP

Input: A, b, c, d, ℓ
Calculate the lexicographic solutions x^{*1} and x^{*2}, and compute $z^{\text{lex}_1} = z(x^{*1})$ and $z^{\text{lex}_2} = z(x^{*2})$
$k := 0$, $PNDV_{out} := \{z^{\text{lex}_1}, z^{\text{lex}_2}\}$, $X^* := \{x^{*1}, x^{*2}\}$, and $Z^* := \{z^{\text{lex}_1}, z^{\text{lex}_2}\}$
while $PNDV_{out} \neq \emptyset$ **do**
  $PNDV_{in} := PNDV_{out}$ and $PNDV_{out} := \emptyset$
  for $t = 1, \ldots, 2^k$ **do**
    if $PNDV_{in} = \emptyset$ **then**
      return
    else
      Remove $\{z^{\text{lex}_1}, z^{\text{lex}_2}\}$ from $PNDV_{in}$, and compute $z_t^{\text{lex}_1} := z^{\text{lex}_1}$, $z_t^{\text{lex}_2} := z^{\text{lex}_2}$, and $z_t^I := z^I$
      $A := z_t^{\text{lex}_1}$, $B := z_t^{\text{lex}_2}$, $R := z_t^I$, and $\delta := (\delta_1, \delta_2)^T = (\frac{B_1 - A_1}{\ell}, \frac{A_2 - B_2}{\ell})^T$
      Create Subproblem t with $(A_1', A_2')^T := (A_1 + \delta_1, A_2 - \delta_2)^T$, $(R_1', R_2')^T := R + \delta$, and $(B_1', B_2')^T := (B_1 - \delta_1, B_2 + \delta_2)^T$, and solve Subproblem t by using formulation of Problem(3)
      if Subproblem t has a solution x^* **then**
        $PNDV_{out} := PNDV_{out} \cup \{[A, z(x^*)], [z(x^*), B]\}$
        $X^* := X^* \cup \{x^*\}$ and $Z^* := Z^* \cup \{z(x^*)\}$
      end if
    end if
  end for
  $k := k + 1$
end while
Apply Pareto filter procedure on Z^* and delete the dominated vectors/solutions from Z^* and X^*
Output X^* and Z^*

Under some conditions, Ehrgott (2005) has showed that solutions of Problem (4) are Pareto optimal solutions for Problem (1). We present a modified Benson method (MB method) which improves its performance for solving BOILOP. In this study, when the objective function of Problem (4) is replaced by (maximize) l_1 (parallel to axis z_1), the results presented in Sect. 7 have shown that the next subproblem fits better to BOILOP:

$$\begin{aligned} &\text{Maximize } l_1 \\ &\text{subject to } \; z_1^0 - z_1(x) = l_1 \\ &\qquad\qquad\quad z_2^0 - z_2(x) = l_2 \\ &\qquad\qquad\quad x \in X,\; l_1 \geqq 0,\; l_2 \geqq 0. \end{aligned} \tag{5}$$

Notice that only one deviation from z^0 is maximized, and under simple conditions on the initial vector z^0 this modification can be used to generate all the Pareto optimal solutions of BOILOP.

In each iteration k of the MB method, the initial vector z^{0k} satisfies $z_2^- \leqq z_2^{0k} \leqq z_2^+$, i.e., it is between the lines r_1 and r_2 in the objective space Z, as shown in Fig. 4. After solving Subproblem k, the non-dominated vector z^k has the greatest deviation from z^0 on the direction z_1. The geometric interpretation in Fig. 4 illustrates iteration $k+1$ of MB method. At this stage, we want to determine the new non-dominated vector z^{k+1}. To achieve this, we choose the initial vector $z^{0(k+1)} = (z_1^k - \delta_1, z_2^k + \delta_2)^T$ for formulating a new subproblem, where $\delta_1 = \frac{z_1^+ - z_1^-}{\ell}$ and $\delta_2 = \frac{z_2^+ - z_2^-}{\ell}$, and ℓ has been suitably chosen and sufficiently large to be the maximum number of iterations.

The main steps of the MB method are given by Algorithm 5.1, and Appendix A.2 provides an iterative numerical example.

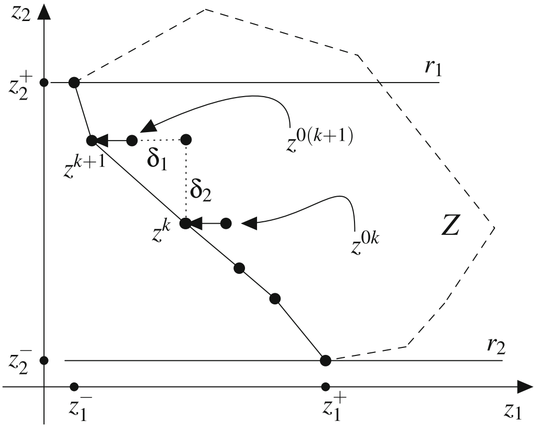

Fig. 4 Geometric illustration of a subproblem in iteration $k+1$ of the MB method

---

Algorithm 5.1 Modified Benson routine for BOILOP

1: **input:** A, b, c, d and ℓ
2: Calculate the lexicographic solutions x^{*1} and x^{*2}, and compute $z^{\text{lex}_1} = z(x^{*1})$ and $z^{\text{lex}_2} = z(x^{*2})$
3: $X^* := \{x^{*1}, x^{*2}\}$, $Z^* := \{z^{\text{lex}_1}, z^{\text{lex}_2}\}$, $\delta_1 := \frac{z_1^+ - z_1^-}{\ell}$, $\delta_2 := \frac{z_2^+ - z_2^-}{\ell}$, and $z^0 := z^{\text{lex}_2}$
4: **for** $k = 0, 1, \ldots, \ell - 1$ **do**
5: Create a subproblem with $z^{0(k+1)} := (z_1^k - \delta_1, z_2^k + \delta_2)^T$. Obtain the solution x^{k+1}, and compute $z^{k+1} = z(x^{k+1})$ by solving this subproblem from formulation of Problem (5)
6: $X^* := X^* \cup \{x^{k+1}\}$ and $Z^* := Z^* \cup \{z^{k+1}\}$
7: **end for**
8: Apply Pareto filter procedure on Z^* and delete the dominated vectors/solutions from Z^* and X^*
9: **output**: X^* and Z^*

5.2 Normal constraints method

The normal constraints method (NC method) is an interesting approach proposed by Messac et al. (2003) with a design similar to the classical ε-constrained method. It was developed to address continuous variables problems. To the best of our knowledge, there is no study published in the literature regarding discrete variables. In Fig. 5, the line r whose endpoints are the lexicographic vectors $z^{\text{lex}_1} = (z_1^-, z_2^+)^T$ and $z^{\text{lex}_2} = (z_1^+, z_2^-)^T$ is drawn. Line r is divided into $\ell - 1$ segments, resulting in (suitably chosen and sufficiently large) ℓ points at the same distance $\delta = \frac{\sqrt{(z_1^+ - z_1^-)^2 + (z_2^+ - z_2^-)^2}}{\ell}$ from each other. One of these points defines the intersection of the line r_k orthogonal to r. In Fig. 5 this orthogonal line r_k ($k = 1$) is used to reduce the feasible objective space Z. For each $k = 1, \ldots, \ell - 1$, we consider the new and reduced objective space on the orthogonal line r_k to determine a non-dominated vector $z^k \in Z$. By minimizing z_2, a weakly non-dominated vector z^k results.

Considering $\eta_k = k\delta \left(\frac{z_2^+ - z_2^-}{z_1^- - z_1^+} \right) \left(z_1^+ - z_1^- + z_2^- - z_2^+ \right)$ and the subset $S_k := \{(z_1, z_2) \mid (z_1^- - z_1^+) z_1 + (z_2^+ - z_2^-) z_2 \geqq \eta_k\}$, the objective space $Z_k = S_k \cap Z$ defines a new feasible objective space for each $k = 1, \ldots, \ell - 1$. By changing r_k, a corresponding set of solutions will be generated. In other words, a new weakly Pareto optimal solution can be calculated by including an additional constraint in Problem (1) and solving the following subproblem:

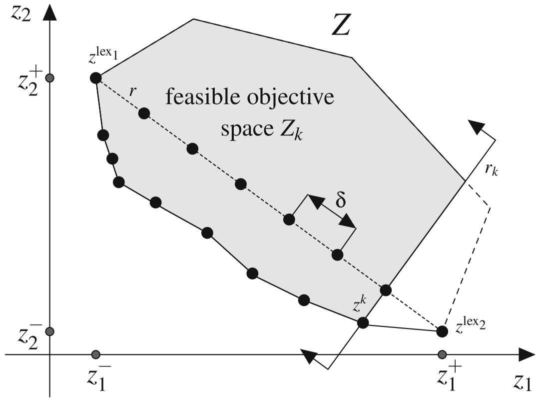

Fig. 5 Geometric illustration of a subproblem in iteration k of the NC method

$$\begin{aligned} &\text{Minimize } z_2(x) \\ &\text{subject to } (z_1^- - z_1^+)z_1(x) + (z_2^+ - z_2^-)z_2(x) \geqq \eta_k, \\ &\qquad\qquad x \in X. \end{aligned} \tag{6}$$

The main steps of the NC method are given by Algorithm 5.2.

Algorithm 5.2 Normal constraint routine for BOILOP

1: **input:** A, b, c, d, ℓ
2: Calculate the lexicographic solutions x^{*1} and x^{*2}, and compute $z^{\text{lex}_1} = z(x^{*1})$ and $z^{\text{lex}_2} = z(x^{*2})$
3: $X^* := \{x^{*1}, x^{*2}\}$, $Z^* := \{z^{\text{lex}_1}, z^{\text{lex}_2}\}$, $\delta := \left(\sqrt{\left(z_1^+ - z_1^-\right)^2 + \left(z_2^+ - z_2^-\right)^2}\right)/\ell$, and $\beta := \left(\frac{z_2^+ - z_2^-}{z_1^- - z_1^+}\right)\left(z_1^+ - z_1^- + z_2^- - z_2^+\right)$
4: **for** $k = 1, \ldots, \ell - 1$ **do**
5: Create a subproblem with $z_1^+, z_1^-, z_2^+, z_2^-, \eta_k := k\delta\beta$. Obtain the solution x^k, and compute $z^k = z(x^k)$ by solving this subproblem from formulation of Problem (6)
6: $X^* := X^* \cup \{x^k\}$ and $Z^* := Z^* \cup \{z^k\}$
7: **end for**
8: Apply Pareto filter procedure on Z^* and delete the dominated vectors/solutions from Z^* and X^*
9: **output:** X^* and Z^*

5.3 Modified Tchebycheff method

Aliano Filho et al. (2018) have introduced a modified Tchebycheff method (MTCH method) to solve a bi-objective cutting stock problem. Since the choice of appropriate weights in the classic TCH method turns it difficult to apply (a single Pareto optimal solution can be obtained from solving Problem (2) by using two distinct weights), two constraints were introduced in the scalarization to prevent the resolution of redundant subproblems. These helped to achieve the spreading and spacing of solutions at the Pareto frontier.

Consider the normalization of Problem (1) by replacing the parameters c and d with $\frac{c^T}{z_1^+ - z_1^-}$ and $\frac{d^T}{z_2^+ - z_2^-}$, respectively. Figure 6 represents the normalized Pareto frontier in the normalized objective space. In this space, we have $\bar{z}^{\text{lex}_1} = (1, 0)^T$, $\bar{z}^{\text{lex}_2} = (0, 1)^T$, $\bar{z}^I = (0, 0)^T$ and

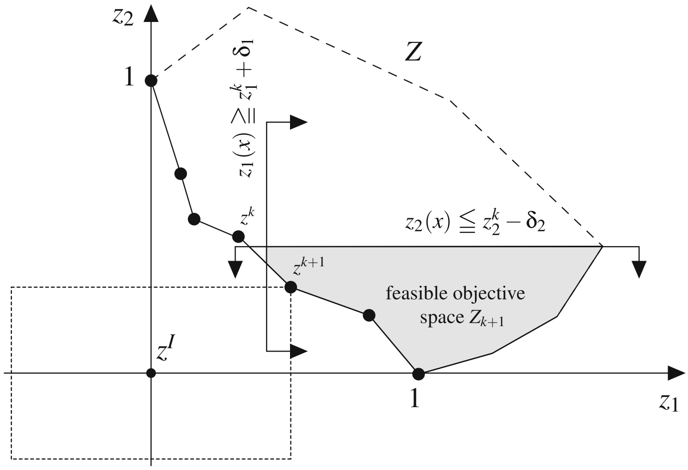

Fig. 6 Geometric illustration of a subproblem in iteration $k+1$ of the MTCH method (Aliano Filho et al. 2018)

$\bar{z}^N = (1, 1)^T$. Non-dominated vectors are illustrated in Fig. 6 when w varies in [0, 1] for Problem (2).

Let $k > 1$ be the index of an iteration of the MTCH method in which the weight w_k is known and the non-dominated vector $z^k = (z_1^k, z_2^k)^T$ was calculated by solving an appropriate subproblem. Definition 1 suggests that if we choose a new weight $w_{k+1} < w_k$, then a new non-dominated vector $z^{k+1} = (z_1^{k+1}, z_2^{k+1})^T$ would be achieved if $z_1^{k+1} > z_1^k$ and $z_2^{k+1} < z_2^k$. Also, if w varies from 0 to 1, we can choose $w_{k+1} > w_k$ for searching $z_1^{k+1} < z_1^k$ and $z_2^{k+1} > z_2^k$. To avoid calculating the same non-dominated vector obtained in the current iteration, we add some constraints to the subproblem in iteration $k+1$. These constraints generate a new feasible objective space Z_{k+1} by eliminating the parts from the normalized objective space where the previous non-dominated vector is located (see Fig. 6). Thereby choosing the weights from 1 to 0 such that $1 > w_1 > \cdots > w_k > w_{k+1} > \cdots > 0$, and determining the vectors $z^1, \ldots, z^k$, the following subproblem can be used to determine the non-dominated vector z^{k+1} in iteration $k+1$:

$$\begin{aligned} &\text{Minimize } u \\ &\text{subject to } w_{k+1} z_1(x) \leqq u, \\ &\qquad (1 - w_{k+1}) z_2(x) \leqq u, \\ &\qquad z_1(x) \geqq z_1^k + \delta_1, \\ &\qquad z_2(x) \leqq z_2^k - \delta_2, \\ &\qquad x \in X,\ u \geqq 0, \end{aligned} \tag{7}$$

where $\delta_1 = \frac{z_1^+ - z_1^-}{\ell}$ and $\delta_2 = \frac{z_2^+ - z_2^-}{\ell}$, and ℓ is suitably chosen and sufficiently large to be the maximum number of iteration k.

Figure 6 provides a geometric interpretation of Problem (7), where the additional constraints limit a new feasible objective space Z_{k+1}, and the non-dominated vectors of iterations from 1 to k will not be searched. New different non-dominated vectors (and consequently, new different Pareto optimal solutions) will be found in each iteration k, independently of being supported or unsupported. If ℓ is sufficiently large, the procedure can determine all non-dominated vectors in the Pareto frontier. However, we expect to solve a number of ℓ

Problems (7) that positively impacts the computational cost of the method. The main steps of the MTCH method are given by Algorithm 5.3.

Algorithm 5.3 Modified Tchebycheff routine for BOILOP

1: **input:** A, b, c, d, ℓ
2: Calculate the lexicographic solutions x^{*1} and x^{*2}, and compute $z^{\text{lex}_1} = z(x^{*1})$ and $z^{\text{lex}_2} = z(x^{*2})$
3: $X^* := \{x^{*1}, x^{*2}\}$, $Z^* := \{z^{\text{lex}_1}, z^{\text{lex}_2}\}$, $c := \frac{c^T}{z_1^+ - z_1^-}$, and $d := \frac{d^T}{z_2^+ - z_2^-}$
4: $w_0 := 1$, $(z_1^0, z_2^0)^T := z^{\text{lex}_1}$, $\delta_1 := \frac{z_1^+ - z_1^-}{\ell}$, and $\delta_2 := \frac{z_2^+ - z_2^-}{\ell}$
5: **for** $k = 0, 1, \ldots, \ell - 1$ **do**
6: Create a subproblem with $w_{k+1} := w_k - \frac{1}{\ell}$, z_1^k, z_2^k, δ_1, and δ_2. Obtain the solution x^{k+1}, and compute $z^{k+1} = z(x^{k+1})$ by solving the formulation of Problem (7)
7: $X^* := X^* \cup \{x^{k+1}\}$ and $Z^* := Z^* \cup \{z^{k+1}\}$
8: **end for**
9: Apply Pareto filter procedure on Z^* and delete the dominated vectors/solutions from Z^* and X^*
10: **output:** X^* and Z^*

6 Applications

We study two types of BOILOPs to evaluate and compare the performances of the methods: the bi-objective one-dimensional cutting stock problem (BCSP) and the bi-objective lot sizing problem (BLSP). The second objective function z_2 of both problems involves machine setup costs, whereas the first objective function z_1 represents production or inventory costs in accordance with each problem. These objective functions are conflicting, and the second one has a very strong impact on the production planning because it significantly increases the cost and time of the adjustment of machines. The Pareto optimal solution set establishes a compromise between the goals and can support a more comprehensive and secure decision-making.

6.1 BCSP formulation

To model this problem, consider a master roll in stock with width L (default widths) and let m be the number of items demanded. Each item i has width $\ell_i < L$ and at least d_i units need to be produced in order to meet demand, $i = 1, \ldots, m$.

Definition 12 (*Cutting pattern*) An instruction that lists the demanded items to be cut from the master roll is called a cutting pattern. It can be represented by an $m-$dimensional vector $a^j = (a_1^j, a_2^j, \ldots, a_m^j)^T$, where each component $a_i^j \in \mathbb{Z}_+$ denotes the quantity of item i present in the cutting pattern j, with $j = 1, \ldots, n$, and n is the total number of cutting patterns that can be made using the master roll.

Definition 13 (*Feasible cutting pattern*) A cutting pattern $a^j \in \mathbb{Z}_+^m$, with $j = 1, \ldots, n$, is feasible if it satisfies the following conditions:

$$
\begin{aligned}
\sum_{i=1}^{m} \ell_i \cdot a_i^j &\leqq L, \\
\sum_{i=1}^{m} \ell_i \cdot a_i^j &\geqq L - \iota, \\
\sum_{i=1}^{m} a_i^j \qquad &\leqq q,
\end{aligned} \tag{8}
$$

where $\iota = \min_{1 \leqq i \leqq m} \{\ell_i\}$ and q is the maximum number of knives allowed in the machine.

The decision variables are $x_j \in \mathbb{Z}_+$ and $y_j = \{0, 1\}$, denoting, respectively, the number of times the cutting pattern j must be cut off and the indication if it is used ($y_j = 1$) or not ($y_j = 0$).

For the application of the exact bi-objective programming methods presented in the previous section to generate the Pareto optimal solution set for this problem, we used $m < n$ a priori cutting patterns a^{j*} obtained from the Gilmore-Gomory column generation method (Gilmore and Gomory 1961, 1963), called *basic cutting patterns*. This strategy is supported by the fact that the linear relaxation of this problem is strong, considering the objective of minimizing z_1 without the setup. Therefore, it seems reasonable to use the basic cutting patterns as approximation to the Pareto optimal solutions. In this way, we can solve moderate and large scale bi-objective one-dimensional cutting stock instances in a reasonable computational time, thus providing a varied set of Pareto optimal solutions for the decision maker.

The objectives of this problem are to minimize (i) the total cost of production in order to meet the demand of each item and (ii) the cost of the machine setup. The mathematical formulation for this model is given by:

$$
\begin{aligned}
\text{Minimize } (z_1, z_2) = &\left(\sum_{j=1}^{m} c_j^1 \cdot x_j, \sum_{j=1}^{m} c_j^2 \cdot y_j \right) \\
\text{subject to } &\sum_{j=1}^{p} a_i^{j*} \cdot x_j \geqq d_i, \qquad i \in I, \\
&x_j \leqq N_j \cdot y_j, \qquad j = 1, \ldots, m, \\
&x_j \geqq y_j, \qquad j = 1, \ldots, m, \\
&y_j \in \{0, 1\},\ x_j \in \mathbb{Z}_+, \quad j = 1, \ldots, m,
\end{aligned} \tag{9}
$$

where c_j^1 and c_j^2 are the costs of each cutting pattern and setup, respectively, and N_j can be calculated by taking into account the cutting patterns, i.e.:

$$
N_j = \max_{1 \leqq i \leqq m} \left\{ \left\lceil \frac{d_i}{a_i^{j*}} \right\rceil \right\}, \quad j = 1, \ldots, m. \tag{10}
$$

Few papers have addressed the cutting stock problem with a bi-objective version. Aliano Filho et al. (2018) have recently developed and applied bi-objective programming methods to this problem, including the modification of the classical Tchebycheff method that may not determine all Pareto optimal solutions (depending of the ℓ value).

6.2 BLSP formulation

The lot-sizing problem is also classic in the area of operations research and has been extensively explored in the literature in its mono-objective version. For example Karimi et al. (2003) and Ben-Daya et al. (2008) have addressed and proposed solution methods for this problem. However, the authors did not find literature work that explored the bi-objective aspect of lot-sizing. With this study, we intend to take a bi-objective approach for this problem by generating the Pareto frontier, thus assisting the manager in making more secure decision.

Let us consider a planning horizon of T periods (hours, days, weeks, months) and let d_j be the quantity of items to be produced in the period $j = 1, \dots, T$. Let $x_j \in \mathbb{R}_+$ and $e_j \in \mathbb{R}_+$ be the quantity produced and stocked in period j, respectively. Consider y_j a binary variable that indicates whether there is production in the period j or not ($y_j = 0$). Considering the costs α_j and β_j, respectively, of the stock and setup period j, the objectives of this problem are (i) minimize the stock cost in the planning horizon and (ii) minimize the setup cost, with demand constraints in each period. The respective mathematical formulation is given by:

$$
\begin{array}{lll}
\text{Minimize } (z_1, z_2) = \left(\sum_{j=1}^{T} \alpha_j \cdot e_j, \sum_{j=1}^{T} \beta_j \cdot y_j \right) & & \\
\text{subject to } x_j + e_j - e_{j-1} = d_j, & j = 1, \dots, T, & \\
\qquad x_j \leqq N_j \cdot y_j, & j = 1, \dots, T, & \\
\qquad x_j \in \mathbb{R}_+, \; e_j \in \mathbb{R}_+, & j = 1, \dots, T, & \\
\qquad y_j \in \{0, 1\}, & j = 1, \dots, T, &
\end{array}
\tag{11}
$$

where N_j is an upper bound for x_j (for instance, $N_j = d_j$) and e_0 is the initial stock.

Problem (11) has two conflicting objectives. On the one hand, a solution with a low machine setup (i.e., a production mode where machines are used in few periods) tends to use high levels of stock for many periods. On the other hand, the minimization of the stock entails the use of the machines for more periods, thus causing a high setup cost.

Next section presents and compares, regarding the robustness of each procedure, the computational results of the four scalarization methods applied to the BCSP and BLSP formulations.

7 Computational results

The algorithms were implemented in MATLAB, Version 7.10.0 R2017a. Each scalar subproblem derived from the scalarization methods was solved through API of CPLEX 12.5. The tests were run on a PC with a Core i7, with 8 GB of RAM. To allow these tests to be reproduced, the random data for this study were generated with the same seed "2017" in MATLAB generator (see MATLAB manual).

Each bi-objective optimization problem is solved by the four scalarization methods presented in Sects. 4 and 5. The comparison and validation of these methods are based on the approximation of the Pareto frontier per instance. To simplify the notation, consider that the current set of non-dominated vectors obtained by a specific scalarization procedure is represented by the finite set Z^* such that its elements $z = (z_1, z_2)^T$ are ordered according to z_1 in ascending order. Note that, no alternative optimal solutions were found, hence the number of Pareto optimal solutions is the same as the number of non-dominated vectors.

Knowles and Corne (2002) and Collette and Siarry (2005) have proposed quality metrics concerning evaluation of performance of bi-objective optimization methods with respect to generation of the Pareto frontier. Based on those proposals, we have used the next three metrics to measure the quality of the set Z^*.

- σ^1 (**cardinality metric**): denotes the cardinality of the approximate Pareto frontier determined by each method. $\sigma^1 = |Z^*|$ measures the number of distinct non-dominated vectors produced. It is no scaling dependent, it is easy to compute, and it induces a complete ordering on the set of candidates (Knowles and Corne 2002).
- σ^2 (**proximity to the ideal vector**): denotes a specific accumulated inverse distance of the non-dominated and normalized vectors from the origin. We have proposed this metric. Let $\tilde{Z}$ be the normalized set obtained from Z^*. If $z^j \in Z^*$, then each $\tilde{z}^j \in \tilde{Z}$, $j = 1, \ldots, \sigma^1$, is calculated by $\tilde{z}_i^j = \frac{z_i^j}{\max\{z_i^j\} - \min\{z_i^j\}}$, $i = 1, 2$. Thus, we define $\sigma^2 = \sum_{j=1}^{\sigma^1} \frac{1}{\|\tilde{z}^j\|}$, where $\|\cdot\|$ denotes the Euclidean norm. σ^2 means how much the non-dominated vectors are close to the ideal vector. It is desirable that σ^2 takes a high value.
- σ^3 (**spacing metric**): denotes how evenly the non-dominated vectors are distributed. $\sigma^3 = \sqrt{\frac{1}{\sigma^1 - 1} \sum_{j=1}^{\sigma^1 - 1} \left(\bar{d} - d_j\right)^2}$, where $d_j = \|\tilde{z}^j - \tilde{z}^{j+1}\|$, $\tilde{z}^j \in \tilde{Z}$, $j = 1, \ldots, \sigma^1 - 1$, and $\bar{d}$ is the average of all d_j. This metric allows measuring the spacing of vectors along the Pareto frontier, and it provides information about the distribution of vectors obtained (standard deviation). When the vectors are evenly dispersed along the Pareto frontier, σ^3 takes low values. Unlike Knowles and Corne (2002) and Collette and Siarry (2005), we have used the Euclidean norm to calculate d_j.

Five performance indicators are used to quantify the quality of the studied methods: (1) CPU$time_{sub}$ is the average computational time to solve each scalar subproblem; (2) CPU$time$ is the total time to generate the approximate Pareto frontier; (3) $\lambda = \frac{\sigma^1}{\text{CPU}_{time}}$ is the average number of non-dominated vectors per second of runtime; (4) σ^4 is the number of subproblems solved per instance of the bi-objective problem; and (5) $\mu = \frac{\sigma^1}{\sigma^4}$ is the average number of non-dominated vectors obtained per subproblem solved.

The following two subsections present the computational results involving the four methods implemented, the two case studies, and four levels of approximation of the Pareto frontiers. We evaluated the computational effort of each procedure, which time is reported in seconds.

7.1 Experiments for BCSP

Random instances for the BCSP formulation were generated through an adaptation of the CUTGEN generator of Gau and Wascher (1995) that is widely used in the literature. Poldi and Arenales (2009) have proposed this adaptation. The instances were divided into three classes, each with 100 test instances.

These classes differ mainly by the number of items. The demand, the width of the items and the costs of the objective function were generated as follows:

- The number of items demanded was set equal to 50, 75 and 100, respectively, for Classes 1, 2 and 3.
- The width of item l_i was generated in the interval $[v_1 \cdot L, v_2 \cdot L]$, where L was set at 10,000. We use $v_1 = 0.01$ and $v_2 = 0.2$ to obtain small items and a great number of non-dominated solutions.

Table 1 Average computational results for σ^1 and σ^2 (BCSP)

	ℓ	σ^1				σ^2			
		NC	MTCH	MRV	MB	NC	MTCH	MRV	MB
$m = 50$	m	22.09	18.78	35.14	13.29	43.51	42.88	81.40	30.00
	$2m$	35.67	30.78	47.30	21.31	73.22	70.05	107.04	49.10
	$3m$	43.74	39.45	54.48	26.88	91.61	89.83	121.70	62.30
	$4m$	50.73	45.93	59.71	31.83	107.35	104.10	132.67	73.42
	Average	38.06	33.74	49.16	23.33	78.92	76.72	110.70	53.71
$m = 75$	m	30.46	29.52	55.33	18.09	59.02	67.27	128.89	39.02
	$2m$	51.58	49.61	82.91	30.37	104.23	113.39	188.88	67.11
	$3m$	66.39	64.74	98.82	40.24	137.14	147.34	222.63	89.61
	$4m$	78.13	77.05	108.10	48.33	162.61	174.32	242.34	107.65
	Average	56.64	55.23	86.29	34.26	115.75	125.58	195.69	75.85
$m = 100$	m	38.38	39.16	56.66	23.22	74.49	91.16	136.63	50.34
	$2m$	67.28	68.58	96.89	40.38	135.74	159.44	226.58	89.58
	$3m$	89.30	92.63	122.37	54.81	184.24	214.92	283.35	122.42
	$4m$	107.10	110.36	141.80	67.18	222.81	253.93	323.01	151.19
	Average	75.52	77.68	104.43	46.40	154.32	179.86	242.39	103.38

- The number of knives was fixed to $q = \left\lceil \sum_{i=1}^{m} \frac{L}{m \cdot l_i} \right\rceil$.
- The demand d_i for each item was randomly generated in the interval [10, 200].
- The costs c_j^1 and c_j^2 were random numbers generated in the intervals [0, 10] and [0, 100], respectively.
- As stated earlier, the number of a priori cutting patterns to be considered was set equal to $p = m$ and was determined by the Gilmore-Gomory procedure.

100 distinct instances, built with randomly generated data from a uniform distribution, within each of the Classes 1, 2 and 3, were solved by the four bi-objective programming methods with the following values for the parameter ℓ: m, $2m$, $3m$ and 4. At the end, we calculate the averages of all these metrics, and the results are presented from Tables 1, 2, 3 4, 5, 6 and 7.

Table 1 presents the average results for σ^1 and σ^2. Note that the different approximations of the Pareto frontier can be determined by increasing the value of ℓ. Figure 7 shows the evolution of the number of non-dominated vectors determined by each method for each approach level. Considering all the instances and the four methods, with $\ell = m, 2m, 3m$ and $4m$, we have on average 31.66, 51.88, 66.15 and 77.18 Pareto optimal solutions, respectively.

Each method is sensitive to modification of ℓ. For example, the methods NC and MTCH determine approximately the same number of solutions in all the simulations carried out, with a slight improvement in NC that determined 2.1% more vectors on average. However, the MRV method was able to determine more than a double of the number of vectors in relation to the MB method, thus showing a greater ability in relation to the other three methods with respect to σ^1. This can be understood because each new Problem (3) is feasible, consequently, there is always generation of a new Pareto optimal solution, which may not happen when Problems (6), (7) and (5) are optimized. With MRV we do not have the optimization of idle problems during the construction of the Pareto frontier, also the determination of the non-dominated vectors in the frontier are guided by the vectors of previous iterations and

Table 2 Average computational results for σ^3 and CPU$time$ (BCSP)

	ℓ	σ^3				CPU$time$			
		NC	MTCH	MRV	MB	NC	MTCH	MRV	MB
$m = 50$	m	0.062	0.105	0.072	0.121	10.25	5.58	9.54	6.42
	$2m$	0.048	0.072	0.052	0.087	20.99	9.67	17.39	14.87
	$3m$	0.043	0.059	0.047	0.073	29.65	13.78	26.24	24.25
	$4m$	0.041	0.053	0.044	0.064	40.34	18.65	39.55	34.59
	Average	0.049	0.072	0.054	0.086	25.31	11.92	23.18	20.03
$m = 75$	m	0.040	0.065	0.053	0.082	28.69	14.72	20.48	13.79
	$2m$	0.029	0.043	0.033	0.057	57.94	27.09	40.99	32.49
	$3m$	0.026	0.036	0.027	0.046	85.98	40.73	57.05	51.87
	$4m$	0.024	0.031	0.026	0.041	116.47	53.77	84.64	73.52
	Average	0.030	0.044	0.035	0.057	72.27	34.08	50.79	42.92
$m = 100$	m	0.029	0.049	0.063	0.067	56.43	27.91	30.45	22.09
	$2m$	0.019	0.031	0.032	0.044	116.64	58.07	66.72	53.89
	$3m$	0.017	0.024	0.024	0.034	177.40	85.96	111.03	89.07
	$4m$	0.015	0.021	0.018	0.028	229.55	110.87	146.12	125.68
	Average	0.020	0.031	0.034	0.043	145.01	70.70	88.58	72.68

Table 3 Average computational time per each scalar subproblem - CPU$time_{sub}$ (BCSP)

m	NC	MTCH	MRV	MB
50	0.204	0.095	0.201	0.154
75	0.388	0.180	0.292	0.225
100	0.539	0.252	0.352	0.276

Table 4 Average computational results for λ and μ (BCSP)

	ℓ	λ				μ			
		NC	MTCH	MRV	MB	NC	MTCH	MRV	MB
$m = 50$	m	2.54	4.49	4.40	2.31	0.44	0.38	0.49	0.27
	$2m$	1.99	4.25	3.52	1.58	0.36	0.31	0.49	0.21
	$3m$	1.68	3.80	2.60	1.17	0.29	0.26	0.49	0.18
	$4m$	1.43	3.27	2.36	0.97	0.25	0.23	0.49	0.16
	Average	1.91	3.95	3.22	1.51	0.34	0.29	0.49	0.20
$m = 75$	m	1.24	2.74	3.10	1.43	0.41	0.41	0.49	0.24
	$2m$	1.06	2.49	2.41	1.04	0.35	0.35	0.49	0.21
	$3m$	0.94	2.16	2.25	0.84	0.30	0.31	0.49	0.18
	$4m$	0.81	2.00	1.62	0.72	0.27	0.27	0.49	0.17
	Average	1.01	2.35	2.35	1.01	0.33	0.34	0.49	0.20
$m = 100$	m	0.85	1.99	1.90	1.15	0.38	0.39	0.49	0.23
	$2m$	0.73	1.59	1.54	0.79	0.34	0.34	0.50	0.20
	$3m$	0.65	1.39	1.32	0.64	0.30	0.31	0.50	0.18
	$4m$	0.57	1.22	1.08	0.57	0.27	0.28	0.50	0.17
	Average	0.70	1.55	1.46	0.79	0.32	0.33	0.50	0.20

Table 5 Average computational results for σ^1 and σ^2 (BLSP)

	ℓ	σ^1				σ^2			
		NC	MTCH	MRV	MB	NC	MTCH	MRV	MB
$m = 50$	m	58.69	33.51	85.04	47.44	143.64	89.52	222.09	122.46
	$2m$	77.55	46.65	110.10	64.96	192.92	125.16	285.57	167.22
	$3m$	92.37	57.80	127.95	79.45	230.42	154.02	329.42	203.88
	$4m$	104.78	68.72	142.22	91.17	262.40	182.76	365.38	232.73
	Average	83.35	51.67	116.33	70.76	207.35	137.87	300.62	181.57
$m = 75$	m	89.07	48.78	127.50	71.02	227.81	139.41	357.87	194.44
	$2m$	119.05	68.75	174.27	99.39	310.56	196.63	481.88	271.67
	$3m$	144.48	87.72	204.63	125.06	378.91	250.69	559.87	340.06
	$4m$	166.96	104.37	233.53	144.60	439.83	297.65	635.92	390.48
	Average	129.89	77.41	184.98	110.02	339.28	221.10	508.89	299.16
$m = 100$	m	109.54	61.27	165.08	98.99	305.29	188.88	497.23	288.12
	$2m$	148.45	87.71	223.95	141.93	422.33	270.62	668.95	411.49
	$3m$	182.32	112.94	268.79	180.83	523.89	349.18	796.62	522.87
	$4m$	213.19	135.71	315.10	214.35	615.23	419.54	928.79	618.23
	Average	163.38	99.41	243.23	159.03	466.69	307.06	722.90	460.18

Table 6 Average computational results for σ^3 and CPU$time$ (BLSP)

	ℓ	σ^3				CPU$time$			
		NC	MTCH	MRV	MB	NC	MTCH	MRV	MB
$m = 50$	m	0.031	0.064	0.029	0.043	13.58	14.87	12.01	13.21
	$2m$	0.027	0.047	0.024	0.032	20.65	25.55	18.94	20.66
	$3m$	0.025	0.040	0.021	0.028	28.93	34.31	23.87	27.63
	$4m$	0.023	0.034	0.021	0.024	33.39	46.14	28.02	34.60
	Average	0.027	0.046	0.024	0.032	24.14	30.22	20.71	24.03
$m = 75$	m	0.024	0.047	0.033	0.033	45.00	62.85	64.16	40.02
	$2m$	0.021	0.035	0.036	0.024	66.16	103.18	100.42	62.43
	$3m$	0.019	0.029	0.035	0.020	88.79	148.58	125.82	85.47
	$4m$	0.018	0.025	0.033	0.017	110.70	189.18	152.18	108.39
	Average	0.021	0.034	0.037	0.024	77.66	125.95	110.65	74.08
$m = 100$	m	0.027	0.043	0.028	0.029	85.40	116.19	154.04	84.34
	$2m$	0.023	0.031	0.024	0.021	128.50	183.26	241.46	129.80
	$3m$	0.020	0.025	0.023	0.016	171.30	268.40	304.67	177.27
	$4m$	0.019	0.022	0.021	0.014	214.25	328.17	382.70	224.40
	Average	0.022	0.030	0.024	0.020	149.86	224.01	270.72	153.95

that indicates that a new non-dominated vector can be found. This ability does not exist for the other three methods included in this study, where the determination of new solutions per iteration is not mandatory.

The values of σ^2 are influenced fundamentally by σ^1. However, σ^2 is also affected by the location of each non-dominated vector on the Pareto frontier.

Table 7 Average computational time per each scalar subproblem - CPU$time_{sub}$ (BLSP)

T	NC	MTCH	MRV	MB
50	0.139	0.170	0.176	0.138
75	0.298	0.472	0.588	0.281
100	0.429	0.632	1.099	0.438

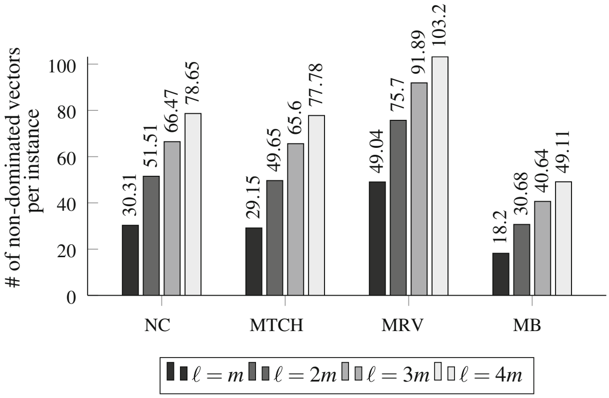

Fig. 7 Average number of non-dominated vectors for each value of ℓ and NC, MTCH, MRV and MB (BCSP)

The more vectors obtained in the central region of the frontier, the smaller the distance to the origin and, consequently, the higher the value of σ^2. This does not happen with non-dominated vectors situated near the lexicographic vectors. The results illustrated reflect that MRV determines the highest value for σ^2 because it produces more non-dominated vectors in relation to the others. A further analysis of these results allows us to draw other conclusions. For example, for $m = 100$, the average of σ^2 is equal to 154.32, 179.86, 242.39 and 103.38, respectively with the methods NC, MTCH, MRV and MB. Making the ratio of this figures by the number of Pareto optimal solutions determined, we have 2.04, 2.31, 2.32 and 2.22, showing that the non-dominated vectors found by these methods are located in different regions of the frontier. These figures lead us to conclude, for example, that NC determines its non-dominated vectors closer to the lexicographic vectors than MRV, which in turn, concentrates its solutions in the intermediate region of the frontier.

Table 2 shows the average results for the σ^3 metric, which unlike σ^2, is not influenced by σ^1. A poor distribution of the vectors along the frontier leads to a greater standard deviation of the distances between these vectors. The NC method more uniformly distributes its non-dominated vectors - both on the intermediate part and on the frontier's terminal region - as opposite the MB that produces a more irregular distribution along the boundary. This only confirms the discussion in the previous paragraph.

Table 2 also shows the CPU$time$ to obtain the Pareto frontier. The dimension of the instance entails a greater computational effort, either because time to optimize each problem is necessary, or because more problems are to be solved. Table 3 presents the average of CPU$time_{sub}$ necessary to solve each scalar subproblem. There is an approximate linearity between the computational time and the problem dimension m, as well as between computational time and the ℓ precision for the frontier. However, NC consumed more computational time due, mainly, to the higher computational cost to solve the cutting problems when using Problem (6).

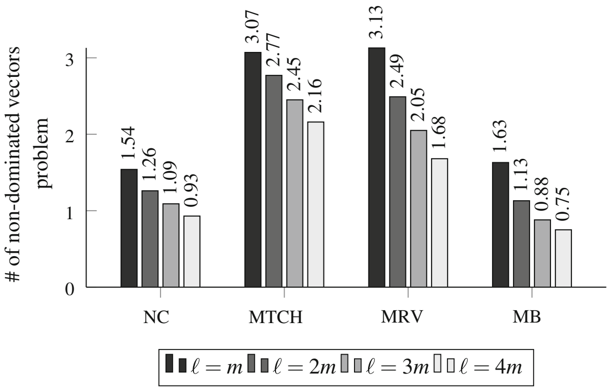

Fig. 8 Average of λ values per ℓ value (BCSP)

Fig. 9 General average number of non-dominated vectors per second with $m = 50$, 75 and 100 and $\ell = m$, $2m$, $3m$ and $4m$ (BCSP)

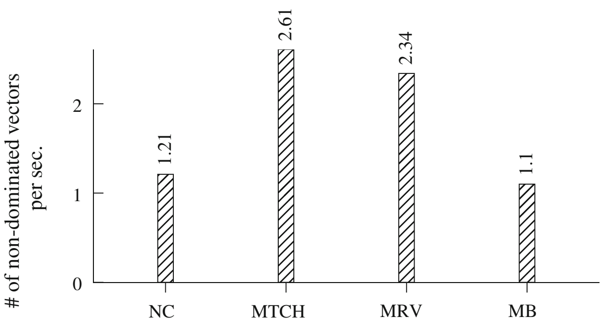

Table 4 illustrates the relationship between the efficiency measures λ and μ with each approximation level. We note that efficiency decreases as ℓ increases, i.e., the cost vs. benefit ratio is worsened as the desired precision for the Pareto frontier is increased. This happens because the number of scalar problems to be solved will generate a much smaller number of new non-dominated vectors.

Figure 8 presents the averages of the efficiencies for each approximation level, calculated for the four dimension of instances considered. The decrease in this efficiency occurs in a linear manner, but with a higher rate of decrease for MRV and MB. Especially, the MTCH method had the best efficiency λ because Problems (7) required less computational effort to be optimized.

Figure 9 provides an overall average for all instances and at all levels of approximation of the four methods tested. μ indicates the capacity of methods to define scalar problems with the ability to produce new Pareto optimal solutions. The relation between the number of scalar problems solved and the number of non-dominated vectors determined suffers significative variation mainly when ℓ is modified and a subtle modification with the dimension of the problem.

The values of μ for the MRV method remained constant during all tests. If we consider the number of scalar problems effectively solved, this ratio is exactly 1, because the infeasible Problems (3) are accounted when calculating μ. This result, undoubtedly, is the best mathematical property of this method, since the generation of a new non-dominated vector is independent of the distance between two consecutive non-dominated vectors. However, the

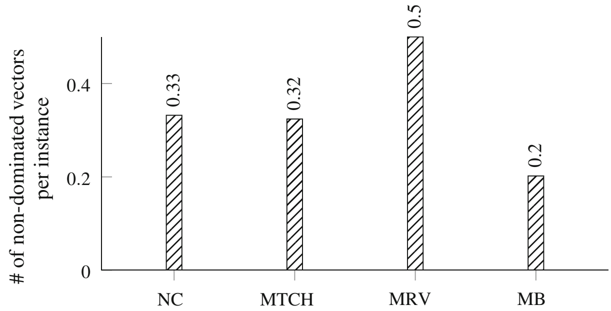

Fig. 10 General average number of non-dominated vectors per instance solved with $m = 50$, 75 and 100, considering $\ell = m$, $2m$, $3m$ and 4ℓ (BCSP)

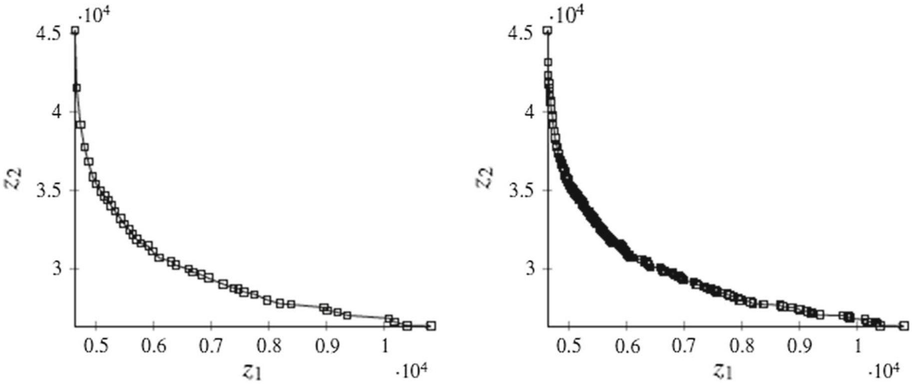

Fig. 11 Two Pareto frontiers obtained by MRV with an instance with $m = 100$ items by using $\ell = 100$ (at left) and $\ell = 400$ (at right) (BCSP)

same does not happen for the other three methods, where μ is decreasing with ℓ. The NC and MB which deal with constraints, although using the same principle, showed considerable differences with respect to this metric. This leads us to conclude that the oblique constraint forces NC to have a better success (similar to MTCH) in determining new non-dominated vectors as compared to the MB method, whose constraint is horizontally modified, parallel to the z_1 objective.

Figure 10 shows the average of μ, calculated for all the instances considered and for all levels of approximation ℓ, highlighting the best performance of MRV. In quantitative terms, the histogram shows that the NC, MTCH, MRV and MB methods need to optimize, on average, 3, 3, 2 and 5 scalar problems, respectively, to determine a Pareto optimal solution for the one-dimensional cutting stock problem.

To conclude this section of computational results, we present in Fig. 11 two approximations to the Pareto frontier of the problem with $m = 100$, using $\ell = 50$ and $\ell = 200$ for the MRV method. We obtained 45 and 167 non-dominated vectors, respectively. Note that an upper bound for the number of non-dominated vectors of this problem can be given by $z_2^+ - z_2^-$, which in the present example equals 18,846.

7.2 Experiments for BLSP

Instances for BLSP formulation were generated and divided into three classes, each with 100 test instances. The data parameters of each class were taken from Araujo and Arenales

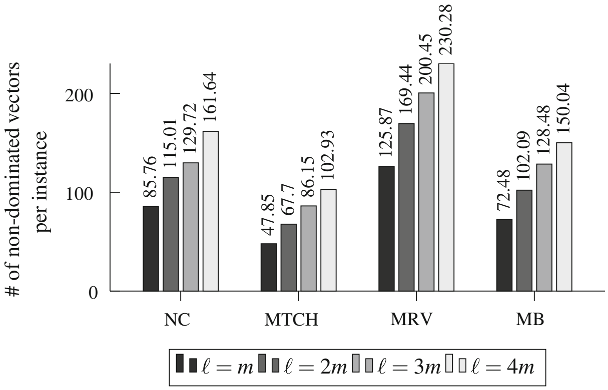

Fig. 12 Averages of the number of non-dominated vectors for each value of ℓ and NC, MTCH, MRV and MB methods (BLSP)

(2000). These classes differ with respect to the number of periods T and other parameters as follows:

- The number of periods in the planning horizon is equal to 50, 75 and 100, respectively, for Classes 1, 2 and 3, respectively.
- The demand d_j in each period was randomly generated in the interval [10, 200].
- The costs α_j and β_j are random numbers generated in the intervals [0, 10] and [0, 100], respectively.

100 distinct problems, within each of the Classes 1, 2 and 3, were solved by the four bi-objective programming methods and using four levels of approximation to the Pareto frontier, with the following values of the parameter ℓ: $2T$, $3T$, $4T$ and $5T$. At the end, we calculated the averages per instance and approximation level for the comparative metrics. The results are presented in the tables below.

Table 5 presents the results for metrics σ^1 and σ^2. As in the cutting stock problem, the results show the increase in the number of non-dominated vectors with the increase of ℓ. This interpretation is shown in Fig. 12, where we count σ^1 for each level of approximation, and considering each method.

Unlike the cutting problem, in this problem there was a change in the performance of the NC, MTCH and MB methods. The MTCH method determined the smallest number of non-dominated vectors (76.16 vectors), while the NC and MB methods determined, on average, 123.03 and 113.27 non-dominated vectors, respectively. However, the method we introduced in this article surpassed the other procedures in this metric, obtaining 181.51 vectors, hence it almost doubled the quantity in relation to MTCH. To get a better sense of this special difference, Fig. 12 shows that with $\ell = 2T$, MRV gets 22% more non-dominated vectors than MTCH with $\ell = 5T$.

As discussed earlier, the values for this metric are larger as more non-dominated vectors are determined and closer they are to the origin. The aspect to be analyzed is how the non-dominated vectors distribute along the Pareto frontier, depending on the scalarization method employed for this problem. By calculating the ratio between σ^2 and σ^1, we find the values of 2.69, 2.91, 2.81, and 2.76, respectively for NC, MTCH, MRV and MB, confirming a distribution of the most non-dominated vectors along the frontier when NC is applied as opposed to MRV.

Table 8 Average computational results for λ and μ (BLSP)

	ℓ	λ				μ			
		NC	MTCH	MRV	MB	NC	MTCH	MRV	MB
$m = 50$	$2T$	4.32	2.25	7.08	3.59	0.59	0.34	0.49	0.48
	$3T$	3.76	1.83	5.81	3.14	0.52	0.31	0.49	0.44
	$4T$	3.19	1.68	5.36	2.88	0.46	0.29	0.49	0.40
	$5T$	3.14	1.49	5.08	2.63	0.42	0.28	1.01	0.37
	Average	3.60	1.81	5.83	3.06	0.50	0.30	0.49	0.42
$m = 75$	$2T$	1.98	0.78	1.99	1.77	0.60	0.33	0.49	0.48
	$3T$	1.80	0.67	1.74	1.59	0.53	0.31	0.50	0.44
	$4T$	1.63	0.59	1.63	1.46	0.48	0.29	0.50	0.42
	$5T$	1.51	0.55	1.53	1.33	0.45	0.28	1.01	0.39
	Average	1.73	0.65	1.72	1.54	0.51	0.30	0.49	0.43
$m = 100$	$2T$	1.28	0.53	1.07	1.17	0.55	0.31	0.50	0.50
	$3T$	1.16	0.48	0.93	1.09	0.50	0.29	0.50	0.48
	$4T$	1.06	0.42	0.88	1.02	0.46	0.28	0.50	0.45
	$5T$	1.00	0.41	0.82	0.96	0.43	0.27	0.50	0.43
	Average	1.12	0.46	0.93	1.06	0.48	0.29	0.50	0.46

The averages of σ^3 are shown in Table 6 and the same behavior as compared with the cutting problem has been observed. The NC method, while generating less non-dominated vectors than MRV, distributes the non-dominated vectors more evenly along the frontier. Also, we present, in the same table, the results for CPU*time* to generate the approximate Pareto frontier.

There is a considerable increase in computational effort when the precision ℓ is increased and the instance of the problem grows. In quantitative terms, for $T = 50$ and $T = 100$, we have an increase in computational time of almost eight times. Besides this, NC and MB had a very similar computational effort for all three classes. While the MRV method was faster for the instances with $T = 50$, its computation time had a considerable increase for the larger ones.

Table 7 presents the average of CPU$time_{sub}$ necessary to solve the scalar subproblem that directly affects the total computational time of each method. Problems (3) are slower to be solved by CPLEX as compared to the subproblems of the other methods, whereas the subproblems of the scalarizations of the MB and NC methods require similar computational effort from CPLEX.

Table 8 presents the average values of λ and μ, giving a sense of the behavior of computational cost vs. benefit of each method, which is decreasing with ℓ. Clearly, the method with the worst performance for this problem was MTCH, being overtaken by the NC and MB methods, which had slight decrease of λ. The MRV method with the instances with $T = 50$ had the highest value for λ, but had a marked decrease in its performance when $T = 75$ and $T = 100$. Figure 13 shows the overall average of these efficiencies per method.

Considering the metric μ, there was a certain balance of the NC, MRV and MB methods and markedly a lower value when considering MTCH. Approximately, every 2.0, 3.4, 2.0 and 2.0 scalar Problems (6), (7), (3) and (5), respectively, generate a non-dominated vector.

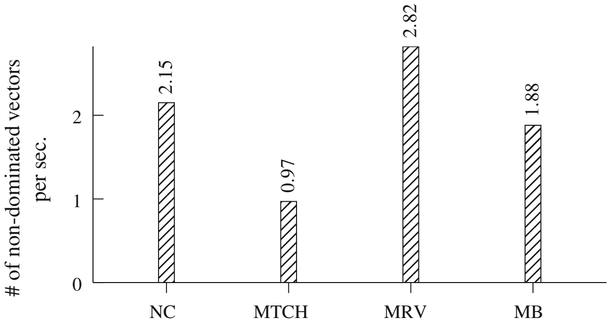

Fig. 13 General average of the number of non-dominated vectors per second with $T = 50$, 75 and 100, and $\ell = 2T, 3T, 4T$ and $5T$ (BLSP)

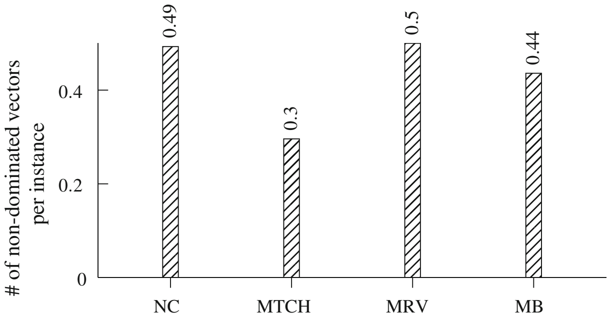

Fig. 14 General average of number of non-dominated vectors per number of instances solved with $T = 50$, 75 and 100, and $\ell = 2T, 3T, 4T$ and $5T$ (BLSP)

Figure 14 illustrates the general averages of μ for each method. However, when applying MRV, half of the scalar problems must effectively be optimized, and the remaining half are only checked for feasibility. The same fact is not valid for NC and MB, where half of the problems solved by these methods originated the same Pareto optimal solution.

Figure 15 shows two levels of approximation of the Pareto frontier, a typical illustration with an instance of the lot-sizing problem with $T = 100$ periods. On the left graphic we used $\ell = 200$ and determined 145 non-dominated vectors, while on the right graphic was obtained with $\ell = 500$ and has 271 non-dominated vectors. These levels can be further improved by increasing the values of ℓ, up to the point when all non-dominated vectors are determined, when ℓ is large enough.

However, a significant difference among values of ℓ for an approximation of the Pareto frontier of equal quality can be observed when using the different methods. Let us consider the special example on the right of Fig. 15 (MRV method with $\ell = 500$). The 271 vectors were obtained with the NC, MTCH and MB methods by programming them with ℓ equal to 1200, 1600 and 1400, respectively. This experiment shows, not only in this example but also throughout the entire phase of experiments, that the MRV method is the most useful to determine new non-dominated vectors, regardless the amplitude of the lexicographic vectors and the distance between any two consecutive non-dominated vectors. We consider this as one of the central contributions of this study.

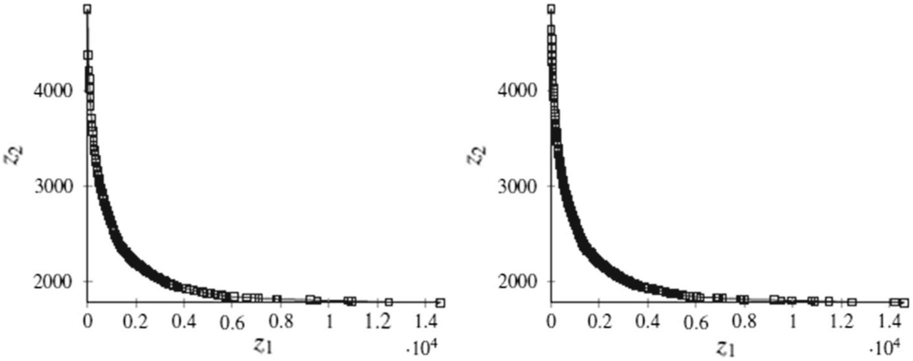

Fig. 15 Two approximation Pareto frontiers obtained by MRV with an instance with $m = 100$ items by using $\ell = 200$ (at left) and $\ell = 500$ (at right) (BLSP)

7.3 Summary of results for BCSP and BLSP

We emphasize the average of λ and μ to summarize the results for the two problems, BCSP and BLSP.

Figures 9 and 13 show that the MTCH method had showed a reverse performance concerning λ between the two problems, having the best output for BCSP. Although the MRV method has obtained figures for λ less than the ones obtained by the MTCH method, it was the second best with a ratio of 90% as compared to the MTCH method for BCSP, and it achieved the best output for BLSP. For the two problems BSPC and BLSP, the MRV method obtained an average of about 2.6 non-dominated vectors per second of runtime. Both the NC and MB methods achieved nearly the double of non-dominated vectors per second of runtime for BLSP when compared to BCSP.

Figures 10 and 14 show that the MRV method outperformed in the computational experience the other three methods concerning μ, where it has obtained, due to its feature, 0.5 non-dominated vectors per scalar subproblem solved. The NC method attained the second best performance for both problems, BCSP and BLSP, with a slight difference from the MRV method in the case of BLSP. Amongst the two worst methods, MB is the better, with $\mu = 0.44$ for BLSP.

8 Conclusions

The present paper has promoted extensions of exact scalarization methods to solve bi-objective integer linear programming problems. Specifically, the MRV method was presented in this research, and we also adapted and implemented the MB, NC and MTCH methods. A comparative analysis of these four methods was performed with randomly generated instances of bi-objective one-dimensional cutting stock and lot-sizing problems, which have the minimization of the machine setup in common. The results from the experimental tests showed that these exact techniques have different performances regarding the computational times, as well as the distribution and the number of non-dominated vectors that are determined.

Each subproblem of the MRV method satisfies the mathematical properties of Theorems 1–3 (TCH method). Moreover, the performance of the MRV method is not influenced by the amplitude of the objective function values and the distribution of the non-dominated vectors along the Pareto frontier. The metric μ confirms, through the experiments performed, that this method does not generate scalar problems whose optimal solutions had already been determined in previous iterations. This feature enables this method to determine up to twice the number of non-dominated vectors concerning the other methods when generating the same amount of subproblems. Also, the cost vs. benefit of this method is favorable since Problems (3) define tighter constraints for the objective functions, thus helping CPLEX to more efficiently optimize them.

From the practical perspective, the potentially large number of solutions of the Pareto frontier obtained via the MRV method requires some form of goal-setting or interactive method to guide the decision maker in selecting his final solution(s). As future research, we intend to adapt the MRV method to other combinatorial optimization problems and also extend it to problems with more than two objectives.

Acknowledgements The authors are indebted to the anonymous reviewers for their helpful comments. In addition, we thank the Federal Technological University of Parana for the support of this research. The research of Margarida Vaz Pato was supported by National Funding from FCT - Fundação para Ciência e a Tecnologia, Portugal, under Project UID/MAT/04561/2019 and UID/Multi/00491/2013.

A Appendices

Appendixes A.1 and A.2 consider the following illustrative instance for an integer linear bi-objective problem.

$$\begin{aligned} &\text{Minimize } z(x) = (c^T x, d^T x) \\ &\text{subject to } Ax = b \\ &\qquad\qquad x \in \mathbb{Z}^6_+, \end{aligned} \tag{12}$$

where $c^T = [-5\ \ 2]$, $d^T = [1\ \ -4]$, $A = [\bar{A} \mid I_4]$, $b = [3, 6, 8, 4]^T$, I_4 is the identity matrix of dimension 4, and $\bar{A}^T = \begin{bmatrix} -1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}$. Let $X = \{x \in \mathbb{Z}^6_+ | Ax = b\}$ be the feasible set of this problem.

A.1 Numerical example for MRV method

Example 1 Consider $\ell = 100$. This means that we expect to obtain at maximum 100 non-dominated vectors.

- **Calculation of lexicographic vectors**: First, we obtain the lexicographic vectors $z^{\text{lex}_1} = A = (-30, 6)^T$ and $z^{\text{lex}_2} = B = (3, -15)^T$ individually optimizing the objectives $z_1(x)$ and $z_2(x)$, respectively, and cheeking they if are dominated vectors.
  We calculate δ_1 and δ_2 as follows:
  $$\delta_1 = \frac{B_1 - A_1}{100} = 0.33 \quad \text{and} \quad \delta_2 = \frac{B_2 - A_2}{100} = 0.21.$$
  Let $t = 2$ and go to the next step.
- **Iteration 1**: first, $R' = (-30, -15)^T + (0.33, 0.21)^T = (-29.67, -14.79)^T$ and $N' = (3, 6)^T - (0.33, 0.21)^T = (2.67, 5.79)$.

We solve Problem (3) given by:

$$\begin{array}{ll} \text{Minimize } u & \\ \text{subject to } & -29.67 \leqq z_1(x) \leqq 2.67 \\ & -14.79 \leqq z_2(x) \leqq 5.79 \\ & z_1(x) - (-29.79) \leqq u \\ & z_2(x) - (-14.67) \leqq u \\ & x \in X,\ u \geqq 0, \end{array}$$

whose optimal solution is a Pareto optimal solution corresponding to the objective vector $C = (-19, -7)^T$. Let $t = 3$.

- **Iteration 2**: We organize the non-dominated vectors already determined, that is, $A = (-30, 6)^T$, $B = (-19, -7)^T$, $C = (3, -15)^T$. We consider the consecutive vectors A and B, hence $R' = (-30, -7)^T + (0.33, 0.21)^T = (-29.67, -6.79)^T$ and $N' = (-19, 6)^T - (0.33, 0.21)^T = (-19.33, 5.79)^T$ and solve the subproblem:

$$\begin{array}{ll} \text{Minimize } u & \\ \text{subject to } & -29.67 \leqq z_1(x) \leqq -19.33 \\ & -6.79 \leqq z_2(x) \leqq 5.79 \\ & z_1(x) - (-29.67) \leqq u \\ & z_2(x) - (-6.79) \leqq u \\ & x \in X,\ u \geqq 0, \end{array}$$

whose solution corresponds to the objective vector $H = (-26, -2)^T$. Considering the consecutive non-dominated vectors B and C, let $A \leftarrow B$ and $B \leftarrow C$, we have $R = (-19, -15)^T$ and $N = (3, -7)^T$ and so $R' = (-18.67, -14.79)^T$ and $N' = (2.67, -7.21)^T$. A new subproblem must be solved:

$$\begin{array}{ll} \text{Minimize } u & \\ \text{subject to } & -18.67 \leqq z_1(x) \leqq 2.67 \\ & -14.79 \leqq z_2(x) \leqq -7.21 \\ & z_1(x) - (-18.67) \leqq u \\ & z_2(x) - (-14.79) \leqq u \\ & x \in X,\ u \geqq 0. \end{array}$$

Its solution provides the objective vector $C = (-7, -13)^T$, and there are $t = 5$ non-dominated vectors.

- **Iteration 3**: Let us order the non-dominated vectors determined, that is, $A = (-30, 6)^T$, $B = (-26, -2)^T$, $C = (-19, -7)^T$, $D = (-7, -13)^T$ and $E = (3, -15)^T$. Hence, considering the first pair of consecutive vectors, we have $R' = (-29.67, -1.79)^T$, $N' = (-26.33, 6.21)^T$, and the new subproblem to be solved is the following:

$$\begin{array}{ll} \text{Minimize } u & \\ \text{subject to } & -29.67 \leqq z_1(x) \leqq -26.33 \\ & -1.79 \leqq z_2(x) \leqq 6.21 \\ & z_1(x) - (-29.67) \leqq u \\ & z_2(x) - (-26.33) \leqq u \\ & x \in X,\ u \geqq 0. \end{array}$$

It provides a solution whose objective vector is $F = (-28, 2)^T$. Now, $A \leftarrow B$ and $B \leftarrow C$ and thereby $R' = (-25.67, -6.79)^T$ and $N' = (-19.33, 1.79)^T$. We solve the subproblem:

$$\begin{aligned}&\text{Minimize } u\\&\text{subject to } -25.67 \leqq z_1(x) \leqq -19.33\\&\qquad -6.79 \leqq z_2(x) \leqq 1.79\\&\qquad z_1(x) - (-25.67) \leqq u\\&\qquad z_2(x) - (-6.79) \leqq u\\&\qquad x \in X,\ u \geqq 0,\end{aligned}$$

whose optimal solution corresponds to the objective vector $G = (-21, -3)^T$. Consequently, $A \leftarrow C$ and $B \leftarrow D$ and the new reference vectors are $R' = (-18.67, -12.79)^T$ and $N' = (-7.33, -7.21)^T$. The new subproblem is given by:

$$\begin{aligned}&\text{Minimize } u\\&\text{subject to } -18.67 \leqq z_1(x) \leqq -7.33\\&\qquad -12.79 \leqq z_2(x) \leqq -7.21\\&\qquad z_1(x) - (-18.67) \leqq u\\&\qquad z_2(x) - (-12.79) \leqq u\\&\qquad x \in X,\ u \geqq 0.\end{aligned}$$

This subproblem has an optimal solution whose associated objective vector is $H = (-14, -8)^T$. Let $A \leftarrow D$ and $B \leftarrow E$ and consequently, $R' = (-6.67, -14.79)^T$ and $N' = (2.67, -13.21)^T$, thus originating the following subproblem:

$$\begin{aligned}&\text{Minimize } u\\&\text{subject to } -6.67 \leqq z_1(x) \leqq 2.67\\&\qquad -14.79 \leqq z_2(x) \leqq -13.21\\&\qquad z_1(x) - (-6.67) \leqq u\\&\qquad z_2(x) - (-14.79) \leqq u\\&\qquad x \in X,\ u \geqq 0,\end{aligned}$$

which provides the objective vector $I = (-2, -14)^T$. Now, we have $t = 9$ non-dominated vectors.

- **Iteration 4**: Let us sort all the vectors already determined, that is, $A = (-30, 6)^T$, $B = (-28, 2)^T$, $C = (-26, -2)^T$, $D = (-21, -3)^T$, $E = (-19, -7)^T$, $F = (-14, -8)^T$, $G = (-7, -13)^T$, $H = (-2, -14)^T$ and $I = (3, -15)^T$. We have as reference vectors $R' = (-29.67, 2.21)^T$ and $N' = (-28.33, 5.79)^T$, thus resulting the following subproblem:

$$\begin{aligned}&\text{Minimize } u\\&\text{subject to } -29.67 \leqq z_1(x) \leqq -28.33\\&\qquad 2.21 \leqq z_2(x) \leqq 5.79\\&\qquad z_1(x) - (-29.67) \leqq u\\&\qquad z_2(x) - (2.21) \leqq u\\&\qquad x \in X,\ u \geqq 0,\end{aligned}$$

which is infeasible. The next four subproblems are infeasible. For the fifth problem, we have $R' = (-13.67, -12.79)^T$ and $N' = (-7.33, -8.21)^T$. Now, the subproblem to be solved is:

$$\begin{aligned} &\text{Minimize } u \\ &\text{subject to } -13.67 \leqq z_1(x) \leqq -7.33 \\ &\qquad -12.79 \leqq z_2(x) \leqq -8.21 \\ &\qquad z_1(x) - (-13.67) \leqq u \\ &\qquad z_2(x) - (-12.79) \leqq u \\ &\qquad x \in X,\ u \geqq 0, \end{aligned}$$

It provides a Pareto optimal solution whose associated objective vector is $J = (-12, -12)^T$. The other subproblems obtained from the consecutive vectors already found, are infeasible. Then, we update $t = 10$.

- **Iteration 5**: We sort all the non-dominated vectors determined: $A = (-30, 6)^T$, $B = (-28, 2)^T$, $C = (-26, -2)^T$, $D = (-21, -3)^T$, $E = (-19, -7)^T$, $F = (-14, -8)^T$, $G = (-12, -12)^T$, $H = (-7, -13)^T$, $I = (-2, -14)^T$ and $J = (3, -15)^T$. The first five subproblems determined were already checked in the previous iteration and are infeasible. It remains to analyze the sixth subproblem, hence $A \leftarrow F$ and $B \leftarrow G$, whose modified reference vectors are $R' = (-13.67, -11.79)^T$ and $N' = (-12.33, -8.21)^T$. Now we must solve the subproblem:

$$\begin{aligned} &\text{Minimize } u \\ &\text{subject to } -13.67 \leqq z_1(x) \leqq -12.33 \\ &\qquad -11.79 \leqq z_2(x) \leqq -8.21 \\ &\qquad z_1(x) - (-13.67) \leqq u \\ &\qquad z_2(x) - (-11.79) \leqq u \\ &\qquad x \in X,\ u \geqq 0, \end{aligned}$$

which is infeasible. For the seventh subproblem, $A \leftarrow G$ and $B \leftarrow H$, and, consequently, $R' = (-11.67, -12.79)^T$ and $N' = (-13.33, -12.21)^T$. The respective subproblem is the following:

$$\begin{aligned} &\text{Minimize } u \\ &\text{subject to } -11.67 \leqq z_1(x) \leqq -13.33 \\ &\qquad -12.79 \leqq z_2(x) \leqq -12.21 \\ &\qquad z_1(x) - (-11.67) \leqq u \\ &\qquad z_2(x) - (-12.79) \leqq u \\ &\qquad x \in X,\ u \geqq 0, \end{aligned}$$

which is also infeasible. In the previous iteration, it was verified that the nine subproblems are infeasible, thus attaining the stopping criterion for this procedure and with all the non-dominated vectors determined. The set of all non-dominated vectors for the problem instance given in this example is

$$\begin{aligned} Z^* = \Bigg\{ &\begin{bmatrix} -30 \\ 6 \end{bmatrix}, \begin{bmatrix} -28 \\ 2 \end{bmatrix}, \begin{bmatrix} -26 \\ -2 \end{bmatrix}, \begin{bmatrix} -21 \\ -3 \end{bmatrix}, \\ &\begin{bmatrix} -19 \\ -7 \end{bmatrix}, \begin{bmatrix} -14 \\ -8 \end{bmatrix}, \begin{bmatrix} -12 \\ -12 \end{bmatrix}, \begin{bmatrix} -7 \\ -13 \end{bmatrix}, \\ &\begin{bmatrix} -2 \\ -14 \end{bmatrix}, \begin{bmatrix} 3 \\ -15 \end{bmatrix} \Bigg\}. \end{aligned} \tag{13}$$

It is important to note the following aspect: in spite of using $\ell = 100$, thus expecting a maximum 100 iterations, the procedure only solved 10 subproblems up to optimality and tested nine for feasibility.

A.2 Numerical example for MB method

Example 2 The lexicographic vectors of Problem (12) are given by $A = z^{\text{lex}_1} = (3, -15)^T$ and $B = z^{\text{lex}_2} = (-30, 6)^T$. Let $\ell = 21$, and, consequently, $\delta_1 = 1.57$ and $\delta_2 = 1.00$.

- **Iteration 1:** For $k = 1$, $z^0 = (1.43, -14)$ and the following subproblem is solved:

$$\begin{aligned} &\text{Maximize } l_1 \\ &\text{subject to } \ 1.43 - z_1(x) = l_1 \\ &\qquad\qquad\quad -14 - z_2(x) = l_2 \\ &\qquad\qquad\quad x \in X,\ l_1 \geqq 0,\ l_2 \geqq 0, \end{aligned}$$

  thus giving a new Pareto optimal solution, whose associated objective vector is $z^1 = (-2, -14)^T$.
- **Iteration 2:** $k = 2$, $z^{0k} = (-3.57, -13)^T$, we have $z^0 = (1.43, -14)$ and the following subproblem is solved:

$$\begin{aligned} &\text{Maximize } l_1 \\ &\text{subject to } \ -3.57 - z_1(x) = l_1 \\ &\qquad\qquad\quad -13 - z_2(x) = l_2 \\ &\qquad\qquad\quad x \in X,\ l_1 \geqq 0,\ l_2 \geqq 0, \end{aligned}$$

  thus determining a new Pareto optimal solution, whose associated objective vector is $z^2 = (-7, -13)^T$.

By following the same steps, until the value of the second coordinate of z^0 researches 6, all the non-dominated vectors for Problem (12) are determined, as presented in (13).

References

Aliano Filho, A., Moretti, A. C., & Pato, M. V. (2018). A comparative study of exact methods for the bi-objective integer one-dimensional cutting stock problem. *Journal of the Operational Research Society*, *69*(1), 91–107.

Anthony, P., Xavier, G., & Ehrgott, M. (2009). Computational results for four exact methods to solve the three-objective assignment problem. In *Multiobjective programming and goal programming* (pp. 79–88). Berlin: Springer

Araujo, S. A., & Arenales, M. N. (2000). Problema de dimensionamento de lotes monoestágio com restrição de capacidade: modelagem, método de resolução e resultados computacionais. *Pesquisa Operacional*, *20*, 287–306.

Ben-Daya, M., Darwish, M., & Ertogral, K. (2008). The joint economic lot sizing problem: Review and extensions. *European Journal of Operational Research*, *185*(2), 726–742.

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

Benson, H. (1998). An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem. *Journal of Global Optimization*, *13*(1), 1–24.

Bérubé, J. F., Gendreau, M., & Potvin, J. Y. (2009). An exact ε-constraint method for bi-objective combinatorial optimization problems: Application to the traveling salesman problem with profits. *European Journal of Operational Research*, *194*(1), 39–50.

Bowman, V. (1976). On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives. In H. Thieriez (Ed.), *Multiple criteria decision making* (Vol. 130)., Lecture notes in economics and mathematical systems Berlin: Springer.

Branke, J., Deb, K., Miettinen, K., & Slowiński, R. (2008). *Multiobjective optimization: Interactive and evolutionary approaches* (Vol. 5252)., Lecture notes in computer science Berlin: Springer.

Clímaco, J., Ferreira, C., & Captivo, M. E. (1997a). Multicriteria integer programming: An overview of the different algorithmic approaches. In *Multicriteria analysis: Proceedings of the XIth international conference on MCDM, 1–6 August 1994, Coimbra, Portugal* (pp. 248–258). Berlin: Springer.

Clímaco, J., Ferreira, C., & Captivo, M. E. (1997b). Multicriteria integer programming: An overview of the different algorithmic approaches. In J. Clímaco (Ed.), *Multicriteria analysis* (pp. 248–258). Berlin: Springer.
Cohon, J. L. (2003). *Multiobjective programming and planning*. Mineola, NY: Dover Publications. (Originally published: New York: Academic Press, 2003. in series: Mathematics in Science and Engineering. With new pref.).
Collette, Y., & Siarry, P. (2005). Three new metrics to measure the convergence of metaheuristics towards the pareto frontier and the aesthetic of a set of solutions in biobjective optimization. *Computers and Operations Research*, *32*(4), 773–792.
Czyzżak, P., & Jaszkiewicz, A. (1998). Pareto simulated annealing—A metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, *7*(1), 34–47.
Dächert, K., Gorski, J., & Klamroth, K. (2012). An augmented weighted Tchebycheff method with adaptively chosen parameters for discrete bicriteria optimization problems. *Computational and Operational Research*, *39*(12), 2929–2943.
Dai, R., & Charkhgard, H. (2018). A two-stage approach for bi-objective integer linear programming. *Operations Research Letters*, *46*(1), 81–87.
Das, I., & Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, *8*(3), 631–657.
Deb, K. (2014). Multi-objective optimization. In *Search methodologies* (pp. 403–449). Boston: Springer.
Deb, K., & Kalyanmoy, D. (2001). *Multi-objective optimization using evolutionary algorithms*. New York, NY: Wiley.
Deb, K. & Sundar, J. (2006). Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th annual conference on genetic and evolutionary computation, GECCO '06* (pp. 635–642). New York, NY: ACM.
Ehrgott, M. (2005). *Multicriteria optimization* (Vol. 491). Berlin: Springer.
Ehrgott, M., & Gandibleux, X. (2000). A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spektrum*, *22*(4), 425–460.
Ehrgott, M., & Gandibleux, X. (2002). *Multiobjective combinatorial optimization—Theory, methodology, and applications* (pp. 369–444). Boston, MA: Springer.
Gau, T., & Wascher, G. (1995). CUTGEN: A problem generator for the standard one-dimensional cutting stock problem. *European Journal of Operational Research*, *84*(3), 572–579.
Gilmore, P. C., & Gomory, R. E. (1961). A linear programming approach to the cutting-stock problem. *Operations Research*, *9*, 848–859.
Gilmore, P. C., & Gomory, R. E. (1963). A linear programming approach to the cutting-stock problem—part II. *Operations Research*, *11*, 863–888.
Golfeto, R. R., Moretti, A. C., & Salles, L. L. N. (2009). A genetic symbiotic algorithm applied to the one-dimensional cutting stock problem. *Pesquisa Operacional (impresso)*, *9*, 365–382.
Karimi, B., Ghomi, S. F., & Wilson, J. (2003). The capacitated lot sizing problem: A review of models and algorithms. *Omega*, *31*(5), 365–378.
Kettani, O., Aouniïd, B., & Martel, J. M. (2004). The double role of the weight factor in the goal programming model. *Computers and Operations Research*, *31*(11), 1833–1845.
Kim, I., & de Weck, O. (2005). Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and Multidisciplinary Optimization*, *29*(2), 149–158.
Klein, D., & Hannan, E. (1982). An algorithm for the multiple objective integer linear programming problem. *European Journal of Operational Research*, *9*(4), 378–385.
Knowles, J., & Corne, D. (2002). On metrics for comparing nondominated sets. In *Proceedings of the 2002 congress on evolutionary computation, 2002. CEC'02* (Vol. 1, pp. 711–716). IEEE.
Laumanns, M., Thiele, L., & Zitzler, E. (2006). An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, *169*(3), 932–942.
Lazimy, R. (2013). Interactive polyhedral outer approximation (IPOA) strategy for general multiobjective optimization problems. *Annals of Operations Research*, *210*(1), 73–99.
Luque, M. (2015). Modified interactive Tchebyshev algorithm (MICA) for non-convex multiobjective programming. *Optimization Letters*, *9*(1), 173–187.
Luque, M., Lopez-Agudo, L., & Marcenaro-Gutierrez, O. (2015). Equivalent reference points in multiobjective programming. *Expert Systems with Applications*, *42*(4), 2205–2212.
Luque, M., Miettinen, K., Eskelinen, P., & Ruiz, F. (2009). Incorporating preference information in interactive reference point methods for multiobjective optimization. *Omega*, *37*(2), 450–462.
Luque, M., Miettinen, K., Ruiz, A. B., & Ruiz, F. (2012). A two-slope achievement scalarizing function for interactive multiobjective optimization. *Computers and Operations Research*, *39*(7), 1673–1681.

Luque, M., Ruiz, F., & Steuer, R. E. (2010). Modified interactive Tchebyshev algorithm (MICA) for convex multiobjective programming. *European Journal of Operational Research*, *204*(3), 557–564.
Madsen, O. B., Ravn, H. F., & Rygaard, J. M. (1995). A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research*, *60*(1), 193–208.
Messac, A., Ismail-Yahaya, A., & Mattson, C. (2003). The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization*, *25*(2), 86–98.
Miettinen, K. (1999). *Nonlinear multiobjective optimization.*, volume 12 of International series in operations research and management science Dordrecht: Kluwer.
Miettinen, K., Eskelinen, P., Luque, M., & Ruiz, F. (2009). On the use of preferential weights in interactive reference point based methods. In *Multiobjective programming and goal programming* (pp. 211–220). Berlin: Springer.
Molina, J., Santana, L. V., Hernández-Díaz, A. G., Coello, C. A. C., & Caballero, R. (2009). g-dominance: Reference point based dominance for multiobjective metaheuristics. *European Journal of Operational Research*, *197*(2), 685–692.
Neumayer, P., & Schweigert, D. (1994). Three algorithms for bicriteria integer linear programs. *Operations Research Spektrum*, *16*(4), 267–276.
Ouaïl, F. Z., & Chergui, M. E. A. (2018). A branch-and-cut technique to solve multiobjective integer quadratic programming problems. *Annals of Operations Research*, *267*(1–2), 431–446.
Özlen, M., & Azizoğlu, M. (2009). Multi-objective integer programming: A general approach for generating all non-dominated solutions. *European Journal of Operational Research*, *199*(1), 25–35.
Poldi, K. C., & Arenales, M. N. (2009). Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths. *Computers and Operations Research*, *36*, 2074–2081.
Ralphs, T. K., Saltzman, M. J., & Wiecek, M. M. (2006). An improved algorithm for solving biobjective integer programs. *Annals of Operations Research*, *147*(1), 43–70.
Ruzika, S., & Wiecek, M. M. (2003). A survey of approximation methods in multiobjective programming. *Journal of Optimization Theory and Applications*, *126*(3), 473–501.
Ruzika, S., & Wiecek, M. M. (2005). Approximation methods in multiobjective programming. *Journal of Optimization Theory and Applications*, *126*(3), 473–501.
Sawik, T. (2007). A lexicographic approach to bi-objective scheduling of single-period orders in make-to-order manufacturing. *European Journal of Operational Research*, *180*(3), 1060–1075.
Sayin, S., & Kouvelis, P. (2005). The multiobjective discrete optimization problem: A weighted min-max two-stage optimization approach and a bicriteria algorithm. *Management Science*, *51*(10), 1572–1581.
Schandl, B., Klamroth, K., & Wiecek, M. (2001). Norm-based approximation in bicriteria programming. *Computational Optimization and Applications*, *20*(1), 23–42.
Shao, L., & Ehrgott, M. (2016). Discrete representation of non-dominated sets in multi-objective linear programming. *European Journal of Operational Research*, *255*(3), 687–698.
Shim, H. S., Lee, S. H., & Kang, B. S. (2017). Pareto front generation for bridge deck management system using bi-objective optimization. *KSCE Journal of Civil Engineering*, *21*(5), 1563–1572.
Shukla, P. K. (2007). *On the normal boundary intersection method for generation of efficient front* (pp. 310–317). Berlin: Springer.
Solanki, R. (1991). Generating the noninferior set in mixed integer biobjective linear programs: An application to a location problem. *Computers and Operations Research*, *18*(1), 1–15.
Steuer, R., & Choo, E.-U. (1983). An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, *26*(3), 326–344.
Sylva, J., & 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.
Ustun, O., & Demirtas, E. A. (2008). An integrated multi-objective decision-making process for multi-period lot-sizing with supplier selection. *Omega*, *36*(4), 509–521. (Special Issue on Logistics: New Perspectives and Challenges.).
Wierzbicki, A. P. (1980). The use of reference objectives in multiobjective optimization. In *Multiple criteria decision making theory and application* (pp. 468–486). Berlin: Springer.
Xia, W., & Wu, Z. (2005). An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. *Computers and Industrial Engineering*, *48*(2), 409–425.