A method to identify a representation of the set of non-dominated points for discrete tri-objective optimization problems

Computers & Operations Research 176 (2025) 106928

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

A method to identify a representation of the set of non-dominated points for discrete tri-objective optimization problems

Sunney Fotedar *, Ann-Brith Strömberg

Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, Gothenburg, SE-412 96, Sweden

ARTICLE INFO

Keywords:
Decision support system
Multi-objective optimization
Discrete tri-objective optimization
Representation of the set of non-dominated points

ABSTRACT

Solving a discrete tri-objective optimization problem involves generating a set of non-dominated points. Most generation methods aim to identify all the non-dominated points to understand the trade-off between conflicting objectives. Finding all the non-dominated points is computationally demanding, which may discourage decision-makers from using generation methods that identify all the non-dominated points. Therefore, it is beneficial to identify a good representation of the Pareto front. In this work, we present an algorithm for computing a representation of the Pareto front for discrete tri-objective optimization problems for a user-defined coverage gap. Further, we present a parallelization approach to decompose the criterion space while avoiding redundancies. We present *constrained coverage gap* to measure performance of algorithms when the problems have incommensurable objective functions. Our algorithm is computationally compared with the state-of-the-art algorithms *Grid point based algorithm* (GPBA-A; Mesquita-Cunha et al., (2023) and *Territory-defining algorithm* (TDA; Ceyhan et al., (2019)). While our primary motivation comes from industrial applications of the generalized tri-objective tactical resource allocation problem (GTRAP; Fotedar et al., (2023)), we have also performed tests on standard benchmark instances of the multi-dimensional tri-objective knapsack problem (3KP) to further validate our approach. Out of 300 instances of 3KP, our proposed algorithm performs best (computationally) in 264 instances. For GTRAP, our algorithm is computationally superior in all the instances.

1. Introduction

When facing real-world decision problems, decision-makers (DMs) want to be mindful of the trade-off between different stakeholders' objectives/goals, which are often in conflict. Multi-objective optimization methods are classified into *a priori*, *a posteriori*, and *interactive* methods. In a priori methods the DM is aware of their preferences among the different objectives and thus, able to combine the objectives into a single objective function. A posteriori methods—also referred to as generation methods—yield a set of so-called *non-dominated points* (NDPs)[1] which contains feasible objective vectors such that none of the objective functions' values can be improved (reduced in a minimization problem) without degrading (increasing in a minimization problem) the value of at least one other objective function. Interactive methods combine elements of the two previous approaches and aim to find the most desired solution for a decision maker (DM) by progressively incorporating their revealed preferences. While bi-objective optimization problems are common, tri-objective optimization problems are gaining traction, though still less prevalent. Incorporating additional objectives into large-scale industrial applications, especially with integrality constraints on some decision variables, remains a challenging task. The development of specialized algorithms for bi- and tri-objective problems has proven valuable, often outperforming general multi-objective methods (referencing bi-objective: Boland et al., 2015a; Stidsen et al., 2014; Przybylski et al., 2008; Visée et al., 1998; Dächert et al., 2012 and tri-objective: Boland et al., 2015b, 2017; Dächert and Klamroth, 2014). The reluctance to expand beyond three objectives stems from the exponential increase in non-dominated points with increase in objectives (Brunsch et al., 2014) and interpretation/visualization of the results is also not easy. We develop a tri-objective optimization method, driven by the need to solve industrial instances of the Generalized Tactical Resource Allocation Problem (GTRAP) (Fotedar et al., 2023b) with applications in aerospace engine manufacturing.

A common category of generation methods comprises criterion/objective space search approaches, which involve a search of the objective space to find new non-dominated points.[2] Although obtaining

* Corresponding author.
E-mail address: sunneyfotedar@gmail.com (S. Fotedar).

[1] Corresponding solutions are called *efficient solutions*

[2] Our focus is on criterion space search methods as they exploit the power of MILP solvers; hence, any future solver enhancement is replicated for the end-user of our approach as well.

https://doi.org/10.1016/j.cor.2024.106928
Received 1 December 2023; Received in revised form 21 September 2024; Accepted 23 November 2024
Available online 9 December 2024

many or all non-dominated points (NDPs) is useful, their generation is computationally demanding. Besides, it is cumbersome for the DM to scan through all the NDPs to identify the most suitable one. To the best of our knowledge, the three types of methods that address such issues are *interactive* (Alves and Clímaco, 2007; Teghem et al., 2000), *inexact* (metaheuristic) (Ehrgott and Gandibleux, 2004) and *representation* (Mesquita-Cunha et al., 2023; Ceyhan et al., 2019; Sayın, 2000) methods. Interactive methods require more active participation of the DM; inexact methods are fast but are not guaranteed to find NDPs. In an ideal case (with no time limit), representation methods (of interest in this work) compute a subset of NDPs that provide a guarantee for a given performance criterion. The three common choices of performance criteria that are considered in representation methods are *coverage* (how accurately the set of NDPs is represented by the set of NDPs obtained), *uniformity* (how well-spread the obtained NDPs are), and *cardinality* (the number of NDPs identified).

1.1. Representation of the Pareto front

Let us consider the discrete tri-objective optimization problem

$$\min_{\mathbf{x}\in X} \{f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})\}, \tag{1}$$

where the set $X \subseteq \mathbb{Z}^n_+$ is defined by linear inequalities and integrality requirements. Let $f_i : X \to \mathbb{Z}_+$, $i = 1,2,3$, denote the objective functions, $F(X) := \{\mathbf{f}(\mathbf{x}) \in \mathbb{Z}^3_+ \mid \mathbf{x} \in X\}$ the objective/criterion space, and $F_{\text{ndp}} \subseteq F(X)$, where F_{ndp} is the set of all the NDPs. The findings discussed in the subsequent sections remain applicable to even some mixed-integer linear programming models, provided the variables have a finite number of distinct values (see an example in (Fotedar et al., 2023a, Prop. 3)). Such problems are also called *discrete multi-objective optimization problems* (DMOOP) as the set F_{ndp} is finite. Assume that we have computed a set of objective points $R \subseteq F(X)$. To assess how well a set R represents the set F_{ndp} calls for a performance measure. We describe one such measure, the so-called *coverage gap*, introduced in (Ceyhan et al., 2019, Def. 6) for a maximization DMOOP. An adaptation to a minimization DMOOP is as follows.

Definition 1 (*Coverage Gap and Representative Points*). Consider a representation R of the set F_{ndp} for the DMOOP (1). The coverage gap of R with respect to $\mathbf{z} \in F_{\text{ndp}}$ is defined as $\alpha_R(\mathbf{z}) := \min_{\mathbf{y}\in R} \{\max_{i=1,2,3} \{y_i - z_i\}\}$, where $\alpha_R(\mathbf{z}) \geq 0$ holds for $\mathbf{z} \in F_{\text{ndp}}$, by the definition of F_{ndp}. The coverage gap of R with respect to F_{ndp} is then defined as $\alpha^*_R := \alpha_R(\mathbf{z}^*)$, where $\mathbf{z}^* \in \operatorname{argmax}_{\mathbf{z}\in F_{\text{ndp}}} \{\alpha_R(\mathbf{z})\}$. A point $\mathbf{r} \in R$ is said to be representative for the point $\mathbf{z} \in F_{\text{ndp}}$ if it holds that $\max_{i=1,2,3} \{r_i - z_i\} = \alpha_R(\mathbf{z})$. □

Solving a representation problem for a *desired coverage gap* value $\overline{\alpha}$ provided by the DM means identifying a representation R of F_{ndp} such that $\alpha^*_R := \alpha_R(\mathbf{z}^*) \leq \overline{\alpha}$. Note that a reduction of the desired coverage gap $\overline{\alpha}$ may increase the number of points required in R and hence, increase the computing time.

Definition 2 (*Strictly $\overline{\alpha}$-dominated Point, Doğan et al., 2022, Def. 6*). Let $\mathbf{y}^k, \mathbf{y}^j \in F(X)$ and assume a desired coverage gap of $\overline{\alpha} > 0$. If it holds that $y^k_i > y^j_i - \overline{\alpha}$, $i = 1,2,3$, then $\mathbf{y}^k$ is said to be strictly $\overline{\alpha}$-dominated by $\mathbf{y}^j$. Furthermore, if only $y^k_i \geq y^j_i - \overline{\alpha}$, $i = 1,2,3$, is satisfied then $\mathbf{y}^k$ is weakly $\overline{\alpha}$-dominated □

Observation 3. *Consider $\mathbf{y}^k, \mathbf{y}^j \in F(X)$, a desired coverage gap $\overline{\alpha} > 0$, and a representation R such that $\mathbf{y}^j \in R$ and $\mathbf{y}^k \in F_{\text{ndp}} \setminus R$. Then, if $\mathbf{y}^k$ is strictly $\overline{\alpha}$-dominated by $\mathbf{y}^j$ it holds that $\alpha_R(\mathbf{y}^k) < \overline{\alpha}$. Hence, including $\mathbf{y}^k$ in R is not necessary for the given $\overline{\alpha}$, due to the relations $\alpha_R(\mathbf{y}^k) := \min_{\mathbf{y}\in R}\{\max_{i=1,2,3}\{y_i - y^k_i\}\} \leq \max_{i=1,2,3}\{y^j_i - y^k_i\} < \overline{\alpha}$, where the first inequality holds since $\mathbf{y}^j \in R$, while the last inequality is due to $\mathbf{y}^k$ being strictly $\overline{\alpha}$-dominated by $\mathbf{y}^j \in R$.*

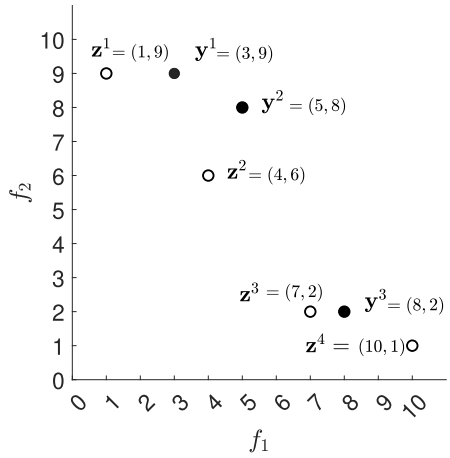


Fig. 1. The set of (all the) NDPs, $F_{\text{ndp}} = \{\mathbf{z}^1, \mathbf{z}^2, \mathbf{z}^3, \mathbf{z}^4\}$, and a representative set $R = \{\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3\}$.

The utility of Definition 2 and Observation 3 is to exclude certain regions from the criterion space when a new objective point is added to a representation R. Another type of performance measure that is proposed by Sayın (2000) is *uniformity level* denoted as Δ_R, which equals the minimum distance between any two points in the representation R, and which may be measured using, e.g., Chebyshev distance. Solving a representation problem for a desired uniformity level $\overline{\delta}$ means obtaining $\Delta_R \geq \overline{\delta}$. The third performance measure is the cardinality of R. The *discrete representation problem* is defined in (Shao and Ehrgott, 2016) as $\min_{|R|\geq 2} \{\alpha^*_R, -\Delta_R, |R|\}$. By improving (reducing) the desired coverage gap, the cardinality of the representative set R increases (Sayın, 2000). In this work—to facilitate the DM's interpretation—we focus on the coverage gap; its calculation is illustrated next.

Consider a discrete bi-objective optimization problem. The set of NDPs is $F_{\text{ndp}} = \{\mathbf{z}^1, \ldots, \mathbf{z}^4\}$ (unfilled circles in Fig. 1) and $R = \{\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3\}$ (solid circles in Fig. 1). The coverage gaps $\alpha_R(\mathbf{z})$, $\mathbf{z} \in F_{\text{ndp}}$, are then given by $\alpha_R(\mathbf{z}^1) = 2$, $\alpha_R(\mathbf{z}^2) = 2$, $\alpha_R(\mathbf{z}^3) = 1$, and $\alpha_R(\mathbf{z}^4) = 1$. Note that $\mathbf{z}^* \in \{\mathbf{z}^1, \mathbf{z}^2\}$ and $\alpha_R(\mathbf{z}^*) = 2$. This implies that the NDPs $\mathbf{z}^1, \mathbf{z}^2 \in F_{\text{ndp}}$ are those being least represented by R. In practice, the set F_{ndp} is not known beforehand; hence, an effective representation method must ensure that the identified set R does not violate a user-defined desired coverage gap i.e. $\overline{\alpha}$. An alternate (but quite similar) metric to coverage gap often employed in comparative algorithm analyses is the *coverage error*, as detailed in (Sayin, 2003). This metric is defined as $\epsilon := \max_{\mathbf{z}\in F_{\text{ndp}}} \{\min_{\mathbf{y}\in R} d(\mathbf{z}, \mathbf{y})\}$, where $d(\cdot,\cdot)$ indicates the (l_∞) distance between two vectors. However, this metric is not always chosen due to its limitations in accurately reflecting representation quality. For example, the coverage errors for $R = \{\mathbf{z}^2\}$ and $R = \{\mathbf{z}^4\}$ both equal 2, while $\alpha_R(\mathbf{z}^4) = 1$ and $\alpha_R(\mathbf{z}^2) = 2$. Based on our understanding of the DMs' requirements, the coverage gap presents as a more appropriate metric.

1.2. Survey of existing methods

Some of the earliest methods developed to identify a good representation of all the NDPs are the *method for finding well-dispersed subsets* (Sylva and Crema, 2007) and the *diversity maximization approach* (DMA) (Masin and Bukchin, 2008). Let $R := \{\mathbf{y}^1, \ldots, \mathbf{y}^r\} \subset \mathbb{R}^3_+$ be the representation found up to a given iteration, and $w_i > 0$, $i = 1,2,3$. The NDP that is worst represented by the set R is given by Sylva and Crema (2007) as

$$(\mathbf{y}^{r+1}, \alpha^*_R) \in \operatorname*{argmin}_{\mathbf{z},\alpha_R\geq 0} F := \operatorname{lexmin} \left\{ -\alpha_R, \textstyle\sum_{i=1}^3 w_i z_i \right\} \tag{2a}$$

$$\text{s.t.} \quad \alpha_R = \min_{j=1,\ldots,r} \left\{ \max_{i=1,2,3} \left\{ y^j_i - z_i \right\} \right\}, \tag{2b}$$

$$\mathbf{z} \in F_{\text{ndp}}. \tag{2c}$$

The constraint (2c) cannot be explicitly stated and lexicographic minimization can be implemented using a small enough multiplier $\epsilon > 0$ such that the minimum decrease in $-\alpha_R$ is greater than the maximum reduction in $\epsilon \sum_{i=1}^{3} w_i z_i$ (Mavrotas, 2009). An equivalent MILP model is thus stated in (Sylva and Crema, 2007) as

$$(\mathbf{y}^{r+1}, \alpha_R^*, \boldsymbol{\phi}^*, \mathbf{x}^*) \in \underset{\mathbf{z},\alpha_R \geq 0, \boldsymbol{\phi}, \mathbf{x}}{\operatorname{argmin}} \quad \left[-\alpha_R + \epsilon \textstyle\sum_{i=1}^{3} w_i z_i \right], \tag{3a}$$

$$\text{s.t.} \quad \alpha_R + z_i \leq y_i^j \phi_i^j + \overline{Y}_i (1 - \phi_i^j), \qquad i = 1,2,3;\ j = 1,\dots,r, \tag{3b}$$

$$\textstyle\sum_{i=1}^{3} \phi_i^j = 1, \qquad j = 1,\dots,r, \tag{3c}$$

$$\phi_i^j \in \{0,1\}, \qquad i = 1,2,3;\ j = 1,\dots,r, \tag{3d}$$

$$z_i = f_i(\mathbf{x}), \qquad i = 1,2,3, \tag{3e}$$

$$\mathbf{x} \in X, \tag{3f}$$

where $\overline{Y}_i \geq \max_{(\mathbf{y},\mathbf{z}') \in R \times F(X)} \{ \|\mathbf{y} - \mathbf{z}'\|_\infty \} + \max_{\mathbf{x} \in X} f_i(\mathbf{x})$. The algorithm in (Sylva and Crema, 2007, Sect. 3) consists of following steps: (1) Choose the desired coverage gap $\overline{\alpha} > 0$, generate an NDP $\mathbf{y}^1$, initialize the representative set $R := \{\mathbf{y}^1\}$, and set $r := |R| = 1$; (2) Solve the MILP model (3); (3) Set $R := R \cup \{\mathbf{y}^{r+1}\}$ and $r := r + 1$; if $\alpha_R^* \leq \overline{\alpha}$ holds, then exit, otherwise go to Step (2). A crucial aspect is that every point in a representative set R must be an NDP in order to guarantee that the final representative set has the desired coverage gap. This is guaranteed by practically all of the upcoming methods, i.e., $R \subseteq F_{\text{ndp}}$. This also implies that each optimization problem must be solved to optimality to obtain guarantees.

The size of the MILP model (3) grows with $r = |R|$, adding three binary variables and three inequality constraints per iteration of the algorithm. The computing time will typically grow exponentially with r (an illustration for the algorithm in (Sylva and Crema, 2007) applied to some randomly generated instances is presented in (Ceyhan et al., 2019, Fig. 2)). Next, we present the two algorithms in (Ceyhan et al., 2019) and their approaches to tackle the issue—that the model size increases with $|R|$—of the algorithm in (Sylva and Crema, 2007). The *subset-based approach* (SBA) utilizes the bounding procedure from Lokman and Köksalan (2012) to decompose the criterion space, while the scalarization resembles the model (3). Assume $\{\mathbf{y}^1, \dots, \mathbf{y}^r\} =: R$ is identified until a given iteration. Using a subset of the points in $R \subseteq F_{\text{ndp}}$, the idea is to identify local upper bound vectors (using the procedure in Lokman and Köksalan (2012, p. 356–359)) that determine the unexplored region of the criterion space. The set $\mathcal{K}(R) := \{\mathbf{k}^1, \dots, \mathbf{k}^{|\mathcal{K}(R)|}\}$, where $\mathbf{k}^j = (k_1^j, k_2^j)^\top$, $k_i^j \in \{0,\dots,r\}$, $i = 1,2$, $j \in \{1,\dots,|\mathcal{K}(R)|\}$, contains the vectors of indices defining local upper bounds b_i^j on f_i, $i = 1,2$. For $j = 1,\dots,|\mathcal{K}(R)|$, $\mathbf{b}^j \in \mathbb{R}^3$ is defined as $b_i^j = y_i^{k_i^j}$, $i = 1,2$, and $b_3^j = \min\{\, y_3 \,|\, \mathbf{y}^j \in R; y_i^j \leq b_i^j, i = 1,2 \,\}$.[3] Hence, given a set R, the set $\mathcal{K}(R)$ is found using (Lokman and Köksalan, 2012, p. 356–359) and for each $\mathbf{k}^j \in \mathcal{K}(R)$ and corresponding $\mathbf{b}^j$, the model

$$(\mathbf{y}^{r+1}, \alpha_R^{j^*}, \mathbf{x}^*) \in \underset{\mathbf{z},\alpha_R^j \geq 0, \mathbf{x}}{\operatorname{argmin}} \left\{ -\alpha_R^j + \epsilon \sum_{i=1}^{3} w_i z_i \;\middle|\; \alpha_R^j \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \mathbf{z} \leq \mathbf{b}^j; \text{(3e)–(3f)} \right\} \tag{4}$$

is solved. In Ceyhan et al. (2019, Prop. 4) it is shown that (4) and (3) possess equal optimal coverage gaps, i.e. the optimal value α_R^* of α_R for a given R in (3) equals $\max_{j \in \{1,\dots,|\mathcal{K}(R)|\}} \{\alpha_R^{j^*}\}$ (cf. (4)). The algorithm stops if, for a representative set R, it holds that $\max_{j \in \{1,\dots,|\mathcal{K}(R)|\}} \{\alpha_R^{j^*}\} \leq \overline{\alpha}$. SBA has superior computational performance than the algorithm by Sylva and Crema (2007). To further improve SBA, the same authors present the *Territory-defining algorithm* (TDA) (Ceyhan et al., 2019).

Given a representation R and a desired coverage gap $\overline{\alpha}$ some territories are removed from the criterion space. Let $T_R := \cup_{\mathbf{y}' \in R} T_{\mathbf{y}'}$, where $T_{\mathbf{y}'} = \{\mathbf{z} \,|\, z_i > y_i' - \overline{\alpha},\ i = 1,2,3\}$. According to Definition 2 the points in the set $T_{\mathbf{y}'}$, where $\mathbf{y}' \in R$ are all $\overline{\alpha}$-strictly dominated by $\mathbf{y}'$. Hence, as stated in Observation 3 that any point $\mathbf{z} \in T_{\mathbf{y}'}$ is a point with coverage gap $\alpha_R(\mathbf{z}) < \overline{\alpha}$. Therefore, not required to be added to the set R if the desired coverage gap is $\overline{\alpha}$. The only difference between the implementation of SBA and TDA is that the set R in (4) is replaced by $\overline{R} := \{\overline{\mathbf{y}} : \overline{\mathbf{y}} := \mathbf{y}' - (\overline{\alpha}, \overline{\alpha}, \overline{\alpha})^\top, \mathbf{y}' \in R\}$ in the latter.[4] As indicated in (Ceyhan et al., 2019) (the results section) this modification improves the computational performance as compared to SBA. A recent effort to further improve TDA is presented in (Doğan et al., 2022), which develops the so-called *Territory-excluded supported generating algorithm* (TSGA). The major strengths of TSGA over existing approaches (such as DMA, SBA, and TDA) is that it takes into account problem-specific distributional properties of the already identified NDPs and does a non-myopic search to reduce the cardinality of the obtained representation for a desired coverage value. TSGA starts with initializing the representation with one NDP, followed by solving the scalarization similar to (3), however, with some differences. In the scalarization used in TSGA, both α_R and ϵ (used in the objective function (3a)) are removed. Furthermore, the term α_R in (3b) is replaced by the desired coverage gap $\overline{\alpha}$. The rest of the scalarization is equivalent to (3). Instead of looking for the worst-represented NDP (as in TDA), it searches for an NDP that has the highest *individual representation power* (IRP) (Doğan et al., 2022, Def. 9).

The recent work (Mesquita-Cunha et al., 2023) suggests a *Grid point based algorithm* (GPBA); the variant tackling mainly the coverage gap aspect is called GPBA-A. The scalarization used is based on the ϵ-constraint-scalarization suggested in (Mavrotas and Florios, 2013), however, the bounding strategy is a modification of Zhang and Reimann (2014).

1.3. Mathematical background and notations

We start by presenting some useful definitions and previous results.

Definition 4 (*Two-Stage Scalarization; Kirlik and Sayın (2014)*). For $\mathbf{u} \in \mathbb{R}_+^2$, $\mathbf{f} : \mathbb{Z}_+^n \to \mathbb{Z}_+^3$ (linear and non-negative functions), $X \subset \mathbb{Z}_+^n$, and $k \in \{1,2,3\}$ the so-called axis of decomposition, the set $\Omega(k,\mathbf{u}) \subseteq X$ of optimal solutions to the two-stage scalarization of (1), is defined as

$$\Omega(k,\mathbf{u}) := \underset{\mathbf{x} \in X}{\operatorname{argmin}} \left\{ \sum_{j=1}^{3} f_j(\mathbf{x}) \;\middle|\; f_j(\mathbf{x}) \leq u_j, j = \{1,2,3\} \setminus \{k\}; f_k(\mathbf{x}) = \overline{f}_k(\mathbf{u}) \right\}, \tag{5a}$$

where

$$\overline{f}_k(\mathbf{u}) := \min_{\mathbf{x} \in X} \{\, f_k(\mathbf{x}) \,|\, f_j(\mathbf{x}) \leq u_j,\ j \in \{1,2,3\} \setminus \{k\} \,\}. \tag{5b}$$

Theorem 5 (*Kirlik and Sayın (2014, Theorem 1)*). *Let $k \in \{1,2,3\}$. For any $\mathbf{u} \in \mathbb{R}_+^2$ any optimal solution $\mathbf{x}^k(\mathbf{u}) \in \Omega(k,\mathbf{u})$ to the two-stage scalarization (5) is efficient in the problem (1).*

The objective $k \in \{1,2,3\}$ that is first optimized in the two-stage scalarization (see (5b)) is referred to as (in this text) the *axis of decomposition*. W.l.o.g. we consider the third objective as the axis of decomposition. In the QSM (Boland et al., 2017), the authors used the two-stage scalarization and combined it with their novel bounding strategy to identify appropriate upper bounds $\mathbf{u} \in \mathbb{R}_+^2$ to be used in (5b) and then (5a). It identifies regions in the (*projected two-dimensional*) criterion space where (projections of) unidentified NDPs may exist. Let $j \geq 2$; then any two vectors $\mathbf{u}, \mathbf{w} \in \mathbb{R}^j$ satisfies[5]

[3] By defining $y_i^0 = +\infty$, no upper bound is imposed on the ith objective in iteration j whenever $k_i^j = 0$ (which does not refer to any point in R).

[4] W.l.o.g., the vector $(\overline{\alpha}, \overline{\alpha}, \overline{\alpha})$ can be replaced by $(\overline{\alpha}_1, \overline{\alpha}_2, \overline{\alpha}_3)$ if we have multi-dimensional desired coverage gap.

[5] These notations are used only for comparing vectors; for scalars regular notations are used.

$$\mathbf{u} \lneqq \mathbf{w} \iff w_i \in [u_i, \infty) \quad \forall i \in \{1, \dots, j\}; \tag{6a}$$

$$\mathbf{u} \leq \mathbf{w} \iff w_i \in [u_i, \infty) \quad \forall i \in \{1, \dots, j\} \text{ and } \mathbf{u} \neq \mathbf{w}; \tag{6b}$$

$$\mathbf{u} < \mathbf{w} \iff w_i \in (u_i, \infty) \quad \forall i \in \{1, \dots, j\}. \tag{6c}$$

Definition 6 (*Orthogonal Projection*)**.** The orthogonal projection of a set $S \subset \mathbb{R}^{n+m}$ onto the linear space $\mathbb{R}^n \times \{0\}^m$ is $\text{proj}_y\{S\} := \{\mathbf{y} \in \mathbb{R}^n : \exists\, \mathbf{z} \in \mathbb{R}^m \text{ s.t. } (\mathbf{y}, \mathbf{z}) \in S\}$.

Definition 7 (ε-*constrained Subset of* X *w.r.t.* $\mathbf{f}$ *and* $\boldsymbol{\epsilon}$)**.** Let $k > 0$, $X \subset \mathbb{R}^n$, the function $\mathbf{f} : X \mapsto \mathbb{R}^k_+$, and $\boldsymbol{\epsilon} \in \mathbb{R}^k_+$. An ε-constrained subset of X w.r.t. $\mathbf{f}$ and $\boldsymbol{\epsilon}$ is defined as $\Theta^{\mathbf{f}}(X, \boldsymbol{\epsilon}) := \{\mathbf{x} \in X \,|\, \mathbf{f}(\mathbf{x}) \leq \boldsymbol{\epsilon}\} \subseteq X$, where $\mathbf{x} \in \mathbb{R}^n_+$.

1.4. Contributions

Our work presents a two-pronged contribution. Our goal is to develop an algorithm for discrete tri-objective optimization problems utilizing the desired coverage gap to reduce computational burden while guaranteeing (if time allows) the desired representation quality. Firstly, we devise a method to partition the criterion space using localized upper bounds, facilitating parallel subproblem resolution on a computing cluster. Secondly, our approach has procedures to mitigate redundancies typically arising when solving multiple subproblems concurrently, through a novel assignment process that allocates sub-problems with their respective upper bounds. Our algorithm undergoes rigorous testing on both relevant industrial instances and various instances of the multi-dimensional tri-objective binary knapsack problem (3KP).

2. A new approach to identify a representation of the set of all NDPs

An exact criterion space search method encompasses two fundamental elements: (a) scalarization, and (b) criterion space bounding/decomposition for computing unidentified non-dominated points (NDPs). For scalarization (a), we employ the *two-stage scalarization* defined in Definition 4 (Kirlik and Sayın, 2014). The reason to choose two-stage scalarization is that it enables our proposed decomposition of the criterion space. Regarding bounding/decomposition (b), we introduce a novel *hierarchical bounding* strategy to compute a representation of the Pareto front exploiting a pre-defined desired coverage gap, $\overline{\alpha}$.

2.1. Splitting the criterion space

The first level of our hierarchical approach is to compute the minimum value of any one of the objectives[6] followed by solving a bi-objective optimization problem which results in pruning the criterion space and computing a subset of NDPs. The identified NDPs are used to create a set of local upper bounds that decompose the so-called *projected two-dimensional criterion space* (orthogonal projections of all feasible objective vectors onto a two-dimensional criterion space). Let $F(X) := \{\mathbf{f}(\mathbf{x}) \in \mathbb{R}^3_+ \,|\, \mathbf{x} \in X\}$ and denote its orthogonal projection onto the (f_1, f_2)-criterion space by $\widehat{F}(X) := \{(f_1(\mathbf{x}), f_2(\mathbf{x})) \in \mathbb{R}^2_+ \,|\, \exists \mathbf{f}(\mathbf{x}) \in F(X) : f_3(\mathbf{x}) \in \mathbb{R}_+\} = \text{proj}_{(f_1,f_2)}\{F(X)\}$. For notational convenience, we denote by $\hat{\mathbf{f}} = (f_1, f_2)^\top$, the orthogonal projection of $\mathbf{f} = (f_1, f_2, f_3)^\top$ onto the (f_1, f_2)-criterion space. Proposition 8 elaborates on the characteristics of the proposed projection (the proof is in Appendix A.1).

[6] W.l.o.g., we consider the third objective to be f_3 in Proposition 8.

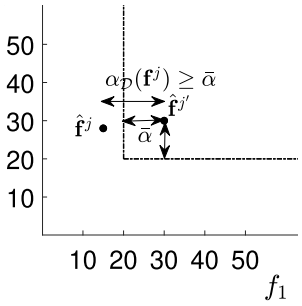


Fig. 2. Illustration of Corollary 8.2. As indicated in the proof, no NDP $\mathbf{f}^j$ exists such that $\alpha_D(\mathbf{f}^j) \geq \overline{\alpha} = 10$.

Proposition 8. *Consider a discrete tri-objective optimization problem as stated in (1). Let* $\mathcal{D}$ *and* $\widehat{\mathcal{D}}$ *denote the non-empty and finite sets of NDPs of the DMOOPs* $\min\{\{f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})\} \in \mathbb{Z}^3_+ \,|\, \mathbf{x} \in \Theta^{f_3}(X, f_3^{\min})\}$ *and* $\min\{\{f_1(\mathbf{x}), f_2(\mathbf{x})\} \in \mathbb{Z}^2_+ \,|\, \mathbf{x} \in \Theta^{f_3}(X, f_3^{\min})\}$*, respectively (cf. Definition 7). Then, it holds that* $\mathcal{D} = \{(\mathbf{u}, f_3^{\min}) \,|\, \mathbf{u} \in \widehat{\mathcal{D}}\}$*, where* $f_3^{\min} := \min\{f_3(\mathbf{x}) \,|\, \mathbf{x} \in X\}$.

Corollary 8.1. *It holds that* $\mathcal{D} \subseteq F_{\text{ndp}}$*, where* F_{ndp} *is the set of all NDPs.*

Proof. Let $\mathcal{D} \nsubseteq F_{\text{ndp}}$, which implies $\exists (\mathbf{u}^*, f_3^{\min})^\top \in \mathcal{D}$ such that $(\mathbf{u}^*, f_3^{\min})^\top \notin F_{\text{ndp}}$. Since $f_3^{\min}$ is the minimum value for f_3 corresponding to a feasible set X, it holds that $(\mathbf{u}^*, f_3^{\min})$ is dominated by an NDP, say $(\mathbf{u}, f_3^{\min})^\top \in F_{\text{ndp}}$. Consequently, $\mathbf{u} \leq \mathbf{u}^*$ must hold. This is not possible since by Proposition 8, $\mathbf{u}^* \in \widehat{\mathcal{D}}$, hence non-dominated w.r.t. the solution space $\Theta^{f_3}(X, f_3^{\min})$ (note the corresponding solution $\mathbf{x}$ such that $\mathbf{f}(\mathbf{x}) = (\mathbf{u}, f_3^{\min})^\top$ also satisfies $\mathbf{x} \in \Theta^{f_3}(X, f_3^{\min})$. □

We next exploit the points in $\widehat{\mathcal{D}}$ to define the region to be excluded in the projected two-dimensional criterion space as a consequence of Proposition 8.

Corollary 8.2. *Consider the non-empty sets* $\mathcal{D}$ *and* $\widehat{\mathcal{D}}$*, same as Proposition 8 and a desired coverage gap* $\overline{\alpha}$*. The region excluded while searching for an NDP in the projected criterion space* (f_1, f_2) *is*[7] $\widehat{\mathcal{D}}^{\overline{\alpha}} := \bigcup_{\hat{\mathbf{f}} \in \widehat{\mathcal{D}}} \left\{ \left[\hat{\mathbf{f}} - (\overline{\alpha}, \overline{\alpha})^\top\right]_+ + \mathbb{R}^2_{>0} \right\}$*. We claim that any point* $\mathbf{f}^j \in F_{\text{ndp}}$ *s.t.* $(f_1^j, f_2^j) \in \widehat{\mathcal{D}}^{\overline{\alpha}}$ *must satisfy* $\alpha_R(\mathbf{f}^j) < \overline{\alpha}$*, for any representation* R *satisfying* $R \supseteq \mathcal{D}$.

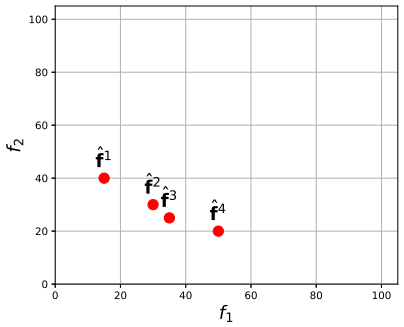
Proof. Let $\mathbf{f}^j \in F_{\text{ndp}} \setminus R$ be an NDP s.t. $(f_1^j, f_2^j)^\top \in \widehat{\mathcal{D}}^{\overline{\alpha}}$; we have a representation $R \supseteq \mathcal{D}$. Consider on the contrary that $\alpha_D(\mathbf{f}^j) \geq \alpha_R(\mathbf{f}^j) \geq \overline{\alpha}$ (the first inequality following from Definition 1 and the fact that $R \supseteq \mathcal{D}$). Assume that the desired coverage gap $\overline{\alpha} > 0$, and let $\mathbf{f}^{j'} \in \text{argmin}_{\mathbf{f} \in \mathcal{D}}\{\max_{1 \leq i \leq 3}\{f_i - f_i^j\}\}$ for a given $\mathbf{f}^j \in F_{\text{ndp}}$.

Case I: Assume that $f_3^{j'} - f_3^j \geq \overline{\alpha} > 0$. Since $\mathbf{f}^{j'} \in \mathcal{D}$, it holds that $f_3^{j'} = f_3^{\min}$, which implies that $f_3^{\min} - f_3^j \leq 0$ for any NDP $\mathbf{f}^j$. Hence, this case is invalid.

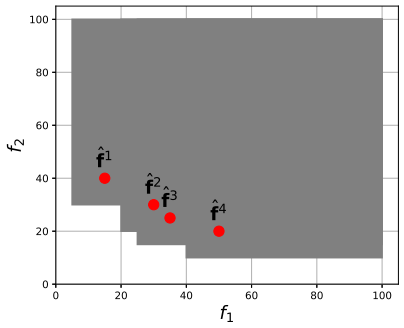
Case II: We conclude that at least one of the constraints $f_i^{j'} - f_i^j \geq \overline{\alpha} > 0$, $i = 1, 2$, must hold. Assume w.l.o.g. that $f_2^{j'} - f_2^j < \overline{\alpha}$, which then implies that $f_1^{j'} - f_1^j \geq \overline{\alpha} > 0$. Since $\widehat{\mathcal{D}}^{\overline{\alpha}}$ is an open set the two inequalities imply that $\mathbf{f}^j \notin \widehat{\mathcal{D}}^{\overline{\alpha}}$. Note that even if $\alpha_D(\mathbf{f}^j) = \overline{\alpha}$, the point $\mathbf{f}^j$ does not have a projection onto the open set $\widehat{\mathcal{D}}^{\overline{\alpha}}$; see Fig. 2. Hence, the contradiction yields that $\alpha_D(\mathbf{f}^j) < \overline{\alpha}$, and since $\alpha_R(\mathbf{f}^j) \leq \alpha_D(\mathbf{f}^j)$ the proposition follows. □

Next, we employ Proposition 8 and Corollaries 8.1–8.2 to enable a parallel computing approach. We first present an example to motivate it. Let the minimum value of the third objective be $f_3^{\min} :=$

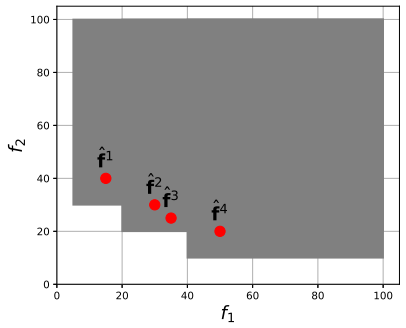
[7] $[\mathbf{a}]_+$ denotes the Euclidean projection of $\mathbf{a} \in \mathbb{R}^m$ onto the non-negative orthant $\mathbb{R}^m_+$, while $R^m_{>0} := \{\mathbf{a} \in \mathbb{R}^m \,|\, a_i > 0, i = 1, \dots, m\}$.



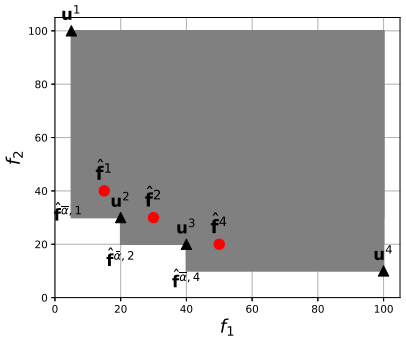
(a) Projections of points in $\mathcal{D}_1$ onto the (f_1, f_2)-criterion space



(b) Excluded region due to $\mathcal{D}_1$ for $\overline{\alpha}^{\text{dis}} = 1$ in Alg. 1



(c) Excluded region due to $\mathcal{D}_{10}$ for $\overline{\alpha}^{\text{dis}} = 10$ in Alg. 1



(d) Quadrants to explore for NDPs (white region)

Fig. 3. The region in the (f_1, f_2)-criterion space to be excluded for a given $\overline{\alpha}^{\text{dis}}$ using Alg. 1.

$\min\{\, f_3(\mathbf{x}) \,|\, \mathbf{x} \in X \,\}$. We compute the set $\mathcal{D}$ as stated in Proposition 8 by identifying the NDPs of the corresponding bi-objective optimization problem $\min\{\, [f_1(\mathbf{x}), f_2(\mathbf{x})] \,|\, \mathbf{x} \in \Theta^{f_3}(X, f_3^{\min}) \,\}$. To compute $\widehat{D}$ (corresponding to the bi-objective optimization model) we use a standard implementation of the ε-constraint method.[8]

Consider $f_3^{\min} = 30$ and compute the set[9] $\mathcal{D} = \mathcal{D}_1 = \left\{ \begin{pmatrix}15\\40\\30\end{pmatrix}, \begin{pmatrix}30\\30\\30\end{pmatrix}, \begin{pmatrix}35\\25\\30\end{pmatrix}, \begin{pmatrix}50\\20\\30\end{pmatrix} \right\}$. Fig. 3(a) illustrates the orthogonal projection of each of these four NDPs onto the criterion space (f_1, f_2). The excluded region (gray color in Fig. 3(b)) due to Corollary 8.2 for $\mathcal{D}_1$ and a (non-trivial[10]) desired coverage gap $\overline{\alpha} > 1$ is $\widehat{D}_1^{\overline{\alpha}}$. Note that the projection of NDP $\mathbf{f}^3$ lies in the excluded region of $\mathbf{f}^2$ in the projected two-dimensional criterion space. One can avoid identifying $\mathbf{f}^3$ in the ε-constraint method by applying the ε-constraint $f_2(\mathbf{x}) \leq f_2(\mathbf{x}^*) - \overline{\alpha}$, where $f_2(\mathbf{x}^*) = f_2^2$ is the second component of the last identified NDP (see Alg. 1, lines 5 and 9, for an updated version; for completeness see Proposition 9). Denote this parameter used to discretize the (f_1, f_2)-criterion space along the second axis (which, w.l.o.g., could be the first axis) by $\overline{\alpha}^{\text{dis}}$ and the corresponding set $\mathcal{D}_{\overline{\alpha}^{\text{dis}}} = \{\mathbf{f}^1, \mathbf{f}^2, \mathbf{f}^4\}$ (for $\overline{\alpha}^{\text{dis}} = \overline{\alpha} = 10$).

Now, we introduce local upper bounds (having similar properties as the local upper bounds defined for stable set in (Klamroth et al., 2015)). Let $K = |\mathcal{D}_{\overline{\alpha}^{\text{dis}}}|$ ($\mathbf{f}^i \in \mathcal{D}_{\overline{\alpha}^{\text{dis}}}$, $i = 1, \ldots, K$ such that $f_2^1 > \cdots > f_2^K$). Then, define $\mathbf{u}^1 := \left[(f_1^1 - \overline{\alpha}, f_2^{\max})^\top\right]_+$, $\mathbf{u}^{K+1} := \left[(f_1^{\max}, f_2^K - \overline{\alpha})^\top\right]_+$, and $\mathbf{u}^i := \left[(f_1^i - \overline{\alpha}, f_2^{i-1} - \overline{\alpha})^\top\right]_+$, $i \in \{2, \ldots, K\}$ In this example the local upper bounds are $\mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3, \mathbf{u}^4$ as indicated by solid triangles in Fig. 3(d). We define rectangle/quadrants $R(\mathbf{0}, \mathbf{u}^i) := \{\, \mathbf{y} \in \mathbb{R}^2_+ \,|\, \mathbf{0} \leqq \mathbf{y} \leqq \mathbf{u}^i \,\}$, $i = 1, 2, 3, 4$, where $\mathbf{u}^i \geq \mathbf{0}^2$. The projections of the remaining NDPs exist in the regions $\cup_{i=1}^4 R(\mathbf{0}, \mathbf{u}^i)$. Note that using $\overline{\alpha}^{\text{dis}} = \overline{\alpha}$ might reduce the computational time to identify the initial set of NDPs $\mathcal{D}_{\overline{\alpha}}$ as compared to $\mathcal{D}_1$, but on the other hand it reduces the number of sub-problems to be solved simultaneously and it is yet to be discussed how it affects the overall computational time of our algorithm. This is empirically tested in Section 3 (cf. Fig. 10). Note that the choice of $\overline{\alpha}^{\text{dis}}$ is consistent only if $\mathcal{D}_1$ is not a singleton; otherwise $\mathcal{D}_1 = \mathcal{D}_{\overline{\alpha}^{\text{dis}}}$ holds true for $\overline{\alpha}^{\text{dis}} > 1$. The proof of the next proposition is found in Appendix A.1.

Proposition 9. *Let $\mathcal{D}_1$ and $\mathcal{D}_{\overline{\alpha}}$ be the output from Alg. 1 with $\overline{\alpha}^{\text{dis}} = 1$ and $\overline{\alpha}^{\text{dis}} = \overline{\alpha}$ (desired coverage gap), respectively. Further, let R be a representation of F_{ndp} such that $\mathcal{D}_1 \subseteq R \subseteq F_{\text{ndp}}$ and $\alpha_R^* := \max_{\mathbf{f} \in F_{\text{ndp}}} \{\alpha_R(\mathbf{f})\} \leq \overline{\alpha}$. Let $R^- := \{R \setminus \mathcal{D}_1\} \cup \mathcal{D}_{\overline{\alpha}}$ and $\alpha_{R^-}^* := \max_{\mathbf{f} \in F_{\text{ndp}}} \{\alpha_{R^-}(\mathbf{f})\}$. Then it holds that $\alpha_R^* = \alpha_{R^-}^* \leq \overline{\alpha}$.*

Next, we present the second level of our hierarchical approach combining a procedure to compute required NDPs in the defined quadrants/rectangles as described above and a novel assignment procedure to mitigate redundancies caused by concurrently solving sub-problems with overlapping search space.

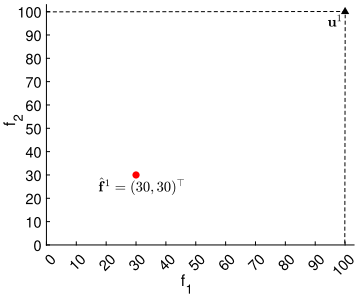
2.2. Incorporating the desired coverage gap

We need a new procedure to update the local upper bounds $\mathbf{u}$ used in the scalarization (5) of the two-stage scalarization to incorporate the desired coverage gap $\overline{\alpha}$. Firstly, we provide an example, present the algorithm and then state Proposition 10 to illustrate how the desired coverage gap is incorporated. W.l.o.g., consider the axis of decomposition as f_3, such that the bounds are applied only for the objectives f_1 and f_2 in (5). Consider the problem (5b) with $k = 3$ and

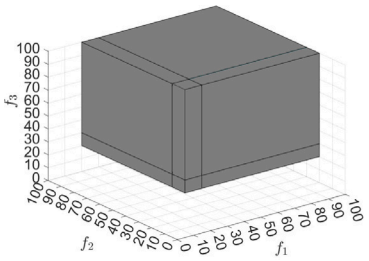
[8] This implies that for an integer-valued objective, the r.h.s. of the ε-constraint is updated by one; for a real-valued function, a user-defined discretization parameter is used.

[9] The index 1 of $\mathcal{D}_1$ represents this set being computed by updating the r.h.s. by one in the ε-constraint method.

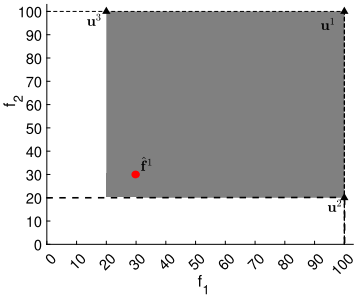
[10] a trivial $\overline{\alpha}$ for an integer programming problem is $\overline{\alpha} \leq 1$.



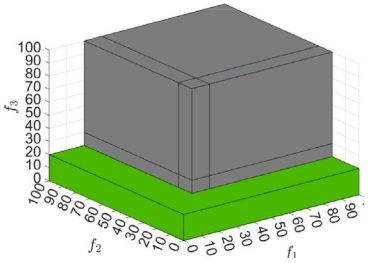
(a) $R = \{\mathbf{f}^1\}$, $U = \{\mathbf{u}^1\}$



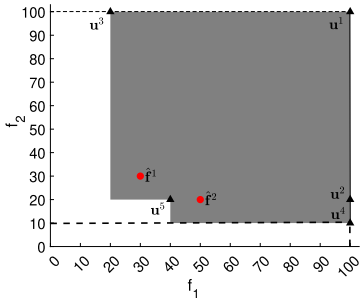
(b) Excluded region of $R = \{\mathbf{f}^1\}$



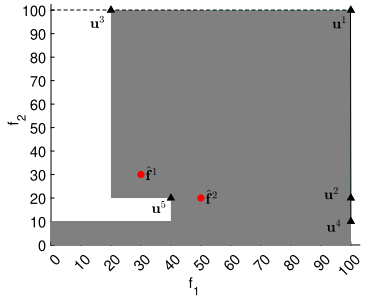
(c) $R = \{\mathbf{f}^1\}$, $U = \{\mathbf{u}^2, \mathbf{u}^3\}$



(d) Excluded region using the properties of two-stage scalarization (green).



(e) $R = \{\mathbf{f}^1, \mathbf{f}^2\}$, $U = \{\mathbf{u}^4, \mathbf{u}^5, \mathbf{u}^3\}$



(f) $R = \{\mathbf{f}^1, \mathbf{f}^2\}$, $U = \{\mathbf{u}^5, \mathbf{u}^3\}$

Fig. 4. Incorporating the desired coverage gap $\overline{\alpha} = 10$.

Algorithm 1 ε-constraint method for $\overline{\alpha}^{\text{dis}}$; axis of decomposition f_3

Input: $X, \mathbf{f}, f_3^{\min}, \mathbf{f}^{\max}, w$ ▷ $\mathbf{f}^{\max}$ is the global upper bound objective vector
$\hat{D}_{\overline{\alpha}^{\text{dis}}} := \{\}$; $\varepsilon := f_2^{\max}$
Initial model P: $\min\{f_1(\mathbf{x}) + w f_2(\mathbf{x}) \,|\, f_2(\mathbf{x}) \le \varepsilon; f_3(\mathbf{x}) \le f_3^{\min}; \mathbf{x} \in X\}$
Output: $\mathbf{f}^* :=$ Solve(P), $\hat{D}_{\overline{\alpha}^{\text{dis}}} := \hat{D}_{\overline{\alpha}^{\text{dis}}} \cup \{(f_1^*, f_2^*)^\top\}$
Update $\varepsilon := f_2^* - \overline{\alpha}^{\text{dis}}$
while True **do**
    $\mathbf{f}^* :=$Solve(P)
    if P is feasible **then**
        $\varepsilon := f_2^* - \overline{\alpha}^{\text{dis}}$; Update $\hat{D}_{\overline{\alpha}^{\text{dis}}}$
    else
        `break`
    end if
end while
`return` $D_{\overline{\alpha}^{\text{dis}}} := \{(f_1, f_2, f_3^{\min})^\top \,|\, (f_1, f_2)^\top \in \hat{D}_{\overline{\alpha}^{\text{dis}}}\}$

set $\mathbf{u} := (100, 100)^\top$ as the upper bounds for the objectives f_1, f_2. Let the corresponding optimal objective value to (5) be $\mathbf{f}^1 = (30, 30, 30)^\top$ (by Theorem 5 is an NDP). We consider a desired coverage gap of $\overline{\alpha} = 10$. Since the bounds are applied only for the first two objectives in (5b), we look for regions to exclude in the projected two-dimensional criterion space. A projection of $\mathbf{f}^1$ is $\hat{\mathbf{f}}^1 = (30, 30)^\top$, illustrated in Fig. 4(a). For $\mathbf{f}^1$, the regions marked in Fig. 4(b) (i.e. the region containing all the strictly $\overline{\alpha}$-dominated points) should not be searched. Now, if we project (orthogonally) the region marked (color gray) in Fig. 4(b) onto the criterion space (f_1, f_2), we get Fig. 4(c). The region marked in gray (Fig. 4(c)) should not be excluded unless we specify the type of scalarization used. For instance, if we had used the scalarization in (3) or (4) to compute $\mathbf{f}^1$ then there is no guarantee that the criterion space $L = \{y_i \in [20, 100], i = 1, 2 \,|\, y_3 \in [0, 20); y_i = f_i(\mathbf{x}), i = 1, 2, 3; \mathbf{x} \in X\}$ is devoid of a feasible objective point. In fact, any solution for which the objective value's projection is in this region (L) is neither strictly nor weakly $\overline{\alpha}$-dominated by $\mathbf{f}^1$, hence should not be excluded. However, if $\mathbf{f}^1$ is identified by the two-stage scalarization (Definition 4) then the region $\widetilde{L} = \{y_i \in [0, 100], i = 1, 2 \,|\, y_3 \in [0, 20)\} \supset L$ in Fig. 4(d) is devoid of any feasible objective point (see shaded region in green). This is because we minimize (in (5)) the function f_3 over all possible values such that $f_i(\mathbf{x}) \le u_i^1 = 100, i = 1, 2$, hence, it is not possible to have an NDP $\mathbf{f}$ with $f_3 < f_3^1 - \overline{\alpha} = 20$ and $f_i \le u_i^1 = 100, i = 1, 2$. Consequently, if we use the two-stage scalarization, it is possible to exclude the region in the projected two-dimensional criterion space marked in gray in Fig. 4(c). Note that some parts of the green-region illustrated in Fig. 4(d) are not illustrated in Fig. 4(c), because there may exist projections with $f_3 \ge 20$, that is an NDP in the white region in Fig. 4(c) (see Proposition 10 for a formal proposition). Hence, using the two-stage scalarization and the desired coverage gap value

$\overline{\alpha}$, the unidentified NDPs (of interest) have their projections in the rectangles/quadrants $R(\mathbf{0},\mathbf{u}^2) := \{\hat{\mathbf{f}} \in \mathbb{R}^2_+ \,|\, f_1 \leq u^2_1, f_2 \leq u^2_2\}$, where $\mathbf{u}^2 = (100, f^1_2 - \overline{\alpha})^\top = (100, 20)^\top$. Similarly, the other region of interest is $R(\mathbf{0},\mathbf{u}^3) = (20, 100)^\top$. If we apply the two-stage scalarization with $\mathbf{u}^2 = (100, 20)^\top$ we compute $\mathbf{f}^2 = (50, 20, 45)^\top$. Now, as in the previous step we exclude some regions and identify two new upper bounds $\mathbf{u}^4$ and $\mathbf{u}^5$ as shown in Fig. 4(e). Applying two-stage scalarization with $\mathbf{u}^4$ we do not find a feasible solution to (5a), hence, we discard the region $R(\mathbf{0},\mathbf{u}^4)$. To formally prove that the resulting representation indeed guarantees the desired coverage gap, we state Proposition 10 later on after presenting our algorithm.

The main idea is to first compute the local upper bounds as described for the first level and then utilize the second level to solve each of the corresponding subproblems simultaneously for a pre-defined $\overline{\alpha}$. This approach is summarized in Alg. 2 and we call it $\overline{\alpha}$-PQSM.

2.3. Improving performance

One drawback of solving sub-problems simultaneously is that the same NDP may be computed multiple times due to overlapping search quadrants. Consider again the example described in Fig. 3. After the (initial) local upper bounds $\mathbf{u}^i$, $i = 1, \ldots, 4$, are computed, the $\overline{\alpha}$-PQSM is applied to each quadrant $R(\mathbf{0},\mathbf{u}^i)$ simultaneously. Our proposed modification to $\overline{\alpha}$-PQSM (already added on Lines 16 and 35 as CheckClosest procedure) is to avoid exploring some of the resulting quadrants while exploring any given (initial) quadrant $R(\mathbf{0},\mathbf{u}^i), i = 1, \ldots, 4$. For instance, let us assume there are four yet-unidentified NDPs (their projections are marked with unfilled circles in Fig. 5(a)). Consider sub-problems 2 and 3 (i.e. with search space $R(\mathbf{0},\mathbf{u}^2)$ and $R(\mathbf{0},\mathbf{u}^3)$) as examples. Firstly, we describe the solution procedure for the second sub-problem. In Fig. 5(b), $R(\mathbf{0},\mathbf{u}^2)$ is explored and an NDP $\mathbf{f}^6$ is computed using Definition 4. Subsequently, the two resulting quadrants are $R\left(\mathbf{0}, \begin{pmatrix} u^2_1 \\ f^6_2 - \overline{\alpha} \end{pmatrix}\right)$ and $R\left(\mathbf{0}, \begin{pmatrix} f^6_1 - \overline{\alpha} \\ u^2_2 \end{pmatrix}\right)$. The modification is that each resulting local upper bound $\mathbf{u}$ computed while exploring any of the (initial) quadrant $R(\mathbf{0},\mathbf{u}^i)$ must be assigned to one of the (initial) local upper bounds $\mathbf{u}^{i^*}$, $i^* = 1, \ldots, 4$, such that $i^* \in \text{argmin}_{i\in\{1,\ldots,4\}}\{(u^i_2 - u_2) \,|\, u^i_2 \geq u_2\}$. If the computed $\mathbf{u}^{i^*} \neq \mathbf{u}^i$ then $R(\mathbf{0},\mathbf{u})$ should not be explored and thus $\mathbf{u}$ should be removed from the list of local upper bounds (to be explored) for the current sub-problem $R(\mathbf{0},\mathbf{u}^i)$.

For instance, in Fig. 5(b), the unfilled triangle i.e. the point $\begin{pmatrix} u^2_1 \\ f^6_2 - \overline{\alpha} \end{pmatrix}$ is actually assigned to $\mathbf{u}^3$; hence, it is removed from the list of yet-unexplored local upper bounds in the current sub-problem. The point $\mathbf{u}^5$ remains; it is explored in Fig. 5(c) and the point $\mathbf{f}^5$ is identified. Two new local upper bounds, $(-5, 30)$ and $(5, 10)$, are generated similarly, and of which the former is infeasible due to the non-negativity of the objective functions. The latter is again removed (unfilled triangle) from the list of unexplored quadrants for sub-problem 2 as it is assigned to $\mathbf{u}^4$. Similar steps are performed for sub-problem 3: two more quadrant searches are avoided (unfilled triangles in Fig. 5(d)) while two new NDPs, $\mathbf{f}^8$ and $\mathbf{f}^7$, are identified. It should be clear from this example that using this assignment rule while updating the list of unexplored local upper bounds in a given sub-problem, the number of quadrant searches may be reduced. In this simple example, we avoided four quadrant searches (counting unfilled triangles and not the ones that are not considered due to the non-negativity of the objective functions). Our suggested modification combined with a solution of the sub-problems in parallel should lead to a reduction in computing time.[11]

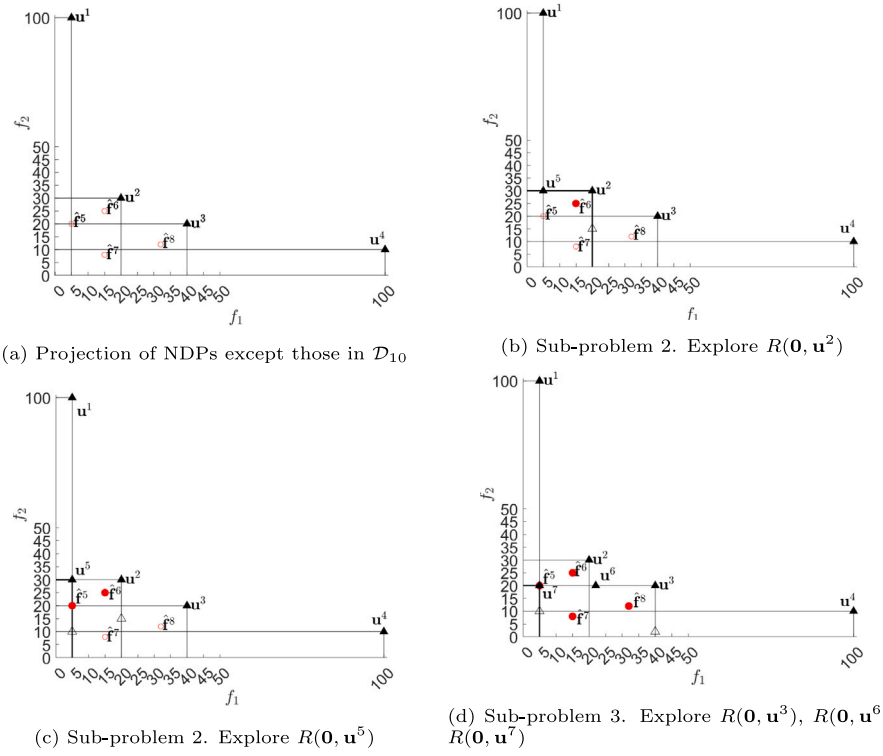
[11] The computing time is the cumulative time used to obtain $f^{\min}_k$ (k being the axis of decomposition), compute $\hat{D}_{\overline{\alpha}^{\text{dis}}}$ using Alg. 1, and the maximum sub-problem solution time.

Algorithm 2 Coverage gap aware PQSM ($\overline{\alpha}$-PQSM)

1: Input: $X, \mathbf{f}, \mathbf{f}^{\max}, \overline{\alpha}, k = 3$ (axis of decomposition), $\overline{\alpha}^{\text{dis}} \geq 1, w > 0$
2: $f^{\min}_3 := \min\{f_3(\mathbf{x}) | \mathbf{x} \in X\}$ $D_{\overline{\alpha}^{\text{dis}}}$ = From Alg. 1 ▷ Level I
3: Initialize a representation $R := D_{\overline{\alpha}^{\text{dis}}}$ and $K := |D_{\overline{\alpha}^{\text{dis}}}|$
4: Identify (initial) local upper bounds $\{\mathbf{u}^k\}^{k\in\{1,\ldots,K+1\}}$ using Section 2.1 ▷ Level II
5: **for** $i \in \{1, \ldots, K+1\}$ **do** ▷ Solve i^{th} sub-problem on i^{th} computer
6: $U := \{\mathbf{u}^i\}$
7: **while** U is not empty and time limit not reached **do**
8: Right_boundary := True
9: **while** Right_boundary = True **do**
10: $\mathbf{u}$:= Pop (extract and remove) the front element of U
11: $\mathbf{f}^*$:= Solve Two-stage scalarization (Definition 4) using $k = 3$ and $\mathbf{u}$
12: **if** $\mathbf{f}^*$ is Null **then**
13: Right_boundary := False
14: **else**
15: Add $\mathbf{f}^*$ to R ▷ if $\mathbf{f}^* \notin R$
16: a_r := CheckClosest$\left(\begin{pmatrix} u^1_1 \\ f^*_2 - \overline{\alpha} \end{pmatrix}\right)$; a_t := CheckClosest$\left(\begin{pmatrix} f^*_1 - \overline{\alpha} \\ u^2_2 \end{pmatrix}\right)$; ▷ Section 2.3
17: **if** a_t = $\mathbf{u}^i$ **then**
18: **if** $u^f_1 < f^*_1 - \overline{\alpha}$ or U is empty **then** ▷ $\mathbf{u}^f$ is the first element of U
19: Add $(f^*_1 - \overline{\alpha}, u_2)$ to the front of U ▷ If $f^*_1 - \overline{\alpha} \geq 0$
20: **end if**
21: **end if**
22: **if** a_r = $\mathbf{u}^i$ **then**
23: Add $(u_1, f^*_2 - \overline{\alpha})$ to the front of U ▷ If $f^*_2 - \overline{\alpha} \geq 0$
24: **end if**
25: **end if**
26: **end while**
27: Top_boundary := True
28: **while** Top_boundary = True **do**
29: Pop the back element of U and denote by $\mathbf{u}$
30: $\mathbf{f}^*$:= Solve Two-stage scalarization (Definition 4) using $k = 3$ and $\mathbf{u}$
31: **if** $\mathbf{f}^*$ is Null **then**
32: Top_boundary := False
33: **else**
34: Add $\mathbf{f}^*$ to R ▷ if $\mathbf{f}^* \notin R$
35: a_r := CheckClosest$\left(\begin{pmatrix} u^1_1 \\ f^*_2 - \overline{\alpha} \end{pmatrix}\right)$; a_t := CheckClosest$\left(\begin{pmatrix} f^*_1 - \overline{\alpha} \\ u^2_2 \end{pmatrix}\right)$;
36: **if** a_r = $\mathbf{u}^i$ **then**
37: **if** $u^b_2 < f^*_2 - \overline{\alpha}$ or U is empty **then**
38: Add $(u_1, f^*_2 - \overline{\alpha})$ to the back of U ▷ If $f^*_2 - \overline{\alpha} \geq 0$
39: **end if**
40: **end if**
41: **if** a_t = $\mathbf{u}^i$ **then**
42: Add $(f^*_1 - \overline{\alpha}, u_2)$ to the back of U
43: **end if**
44: **end if**
45: **end while**
46: **end while**
47: **end for**
48: return R

2.4. Overview of $\overline{\alpha}$-PQSM

Some of the main distinctions of our proposed algorithm $\overline{\alpha}$-PQSM to the PQSM (Parallel QSM) (Fotedar et al., 2023b) and the QSM (Boland et al., 2017) are marked by rectangles in Alg. 2. Broadly speaking the QSM also utilized the two-stage scalarization technique first proposed by Kirlik and Sayın (2014) but the bounding procedure neither results in parallelization nor incorporates the desired coverage gap $\overline{\alpha}$. Our recent work (Fotedar et al., 2023b) proposes PQSM which parallelized the QSM for the industrial problem GTRAP. In this work, we generalize and improve PQSM in two aspects (1) include desired coverage gap to compute a representation of the Pareto front for any discrete tri-objective optimization problem; (2) new subproblem allocation procedure i.e. CheckClosest is introduced to mitigate redundancies caused in PQSM due to parallelization. These advancements not only boost computational efficiency but also extend the applicability to various problem types with discrete three-dimensional Pareto front. To ensure



(a) Projection of NDPs except those in $\mathcal{D}_{10}$

(b) Sub-problem 2. Explore $R(\mathbf{0}, \mathbf{u}^2)$

(c) Sub-problem 2. Explore $R(\mathbf{0}, \mathbf{u}^5)$

(d) Sub-problem 3. Explore $R(\mathbf{0}, \mathbf{u}^3)$, $R(\mathbf{0}, \mathbf{u}^6)$, $R(\mathbf{0}, \mathbf{u}^7)$

Fig. 5. Adjustments to remove redundancies caused by overlapping regions of the search quadrants while solving sub-problems simultaneously.

the quality of our representation we present Proposition 10 (proof in Appendix).

Proposition 10. *Consider a discrete tri-objective optimization problem (1), and let $\mathbf{f}^i \in \mathbb{R}^3_+$ be identified in the ith iteration of the $\overline{\alpha}$-PQSM (with $k = 3$ and an upper bound $\mathbf{u}^i$ in the two-stage scalarization, as in Definition 4). Denote the set of NDPs (including $\mathbf{f}^i$) identified by the $\overline{\alpha}$-PQSM until iteration i by R. Then, the region $\widehat{F}^i := \text{proj}_{(f_1,f_2)}\{\mathbf{f} \in \mathbb{R}^3_+ \mid u^i_\ell \geq f_\ell > f^i_\ell - \overline{\alpha}, \ell = 1,2\}$ in the projected two-dimensional criterion space does not contain the projection of any NDP $\mathbf{f}^j \in F_{\text{ndp}}$ s.t. $\alpha_R(\mathbf{f}^j) > \overline{\alpha}$. Hence, it is not of interest for identifying a representation with a desired coverage gap $\overline{\alpha} > 0$ w.r.t. the set F_{ndp} of all the NDPs.*

2.5. Constrained coverage gap

In this section, we first highlight the importance of introducing the concept of a *vector coverage gap* and further extend this notion by defining a *constrained coverage gap*. In numerous practical applications, the objectives involved are inherently incommensurable, making their reduction to a single scalar coverage gap inadequate. For instance, in one of our test cases derived from a Generalized Tactical Resource Allocation Problem (GTRAP) appearing in the aerospace industry (Fotedar et al., 2023b), the minimization objectives include excess resource imbalance (real-valued), the number of one-time setups (integer-valued), and inventory levels (integer-valued). Consequently, a three-dimensional coverage gap, denoted by $\overline{\overline{\boldsymbol{\alpha}}} \in \mathbb{R}^3_+$, is adequate in such a case, while a scalar coverage gap $\overline{\alpha} \in \mathbb{R}$ is not. This issue is not limited to our particular test case but extends to many real-world applications having likewise incommensurable objectives.

We present an example from the GTRAP instances (Fotedar et al., 2023b); for model details, see the Appendix A.3. Consider one of the NDPs, $\mathbf{z} = \begin{pmatrix} 0 \\ 10 \\ 100 \end{pmatrix} \in F_{\text{ndp}}$, and assume we have computed a representation $R = \left\{\begin{pmatrix} 0.1 \\ 11 \\ 90 \end{pmatrix}, \begin{pmatrix} 0.18 \\ 11 \\ 85 \end{pmatrix}\right\}$. Let $f_i^{\max} := \max_{\mathbf{z} \in F_{\text{ndp}}}\{z_i\}$ and $f_i^{\min} := \min_{\mathbf{z} \in F_{\text{ndp}}}\{z_i\}$ represent the global maximum and minimum values for the ith objective over F_{ndp}, respectively. Assume fixed bounds are known $\mathbf{f}^{\max} = \begin{pmatrix} 2.2 \\ 15 \\ 405 \end{pmatrix}$ and $\mathbf{f}^{\min} = \begin{pmatrix} 0 \\ 9 \\ 0 \end{pmatrix}$. A user-imposed desired vector coverage gap is $\overline{\overline{\boldsymbol{\alpha}}} = \begin{pmatrix} 0.1 \\ 1 \\ 100 \end{pmatrix}$. Then, the coverage gap as per Definition 1 yields $\alpha_R(\mathbf{z}) = \min\left\{\max\begin{pmatrix} 0.1 \\ 1 \\ -10 \end{pmatrix}, \max\begin{pmatrix} 0.18 \\ 1 \\ -15 \end{pmatrix}\right\} = 1$, which is not useful as the obtained value cannot be compared with our desired vector coverage gap. To consider the required vector coverage, there are at least two natural paths to follow. (a) Define a scalar coverage gap $\overline{\alpha}$ for a given $\overline{\overline{\boldsymbol{\alpha}}}$ such that if R is found with the desired vector coverage then there is no need to add $\mathbf{z}$ to R. For instance, we can derive $\overline{\alpha} = \min_{j \in \{1,2,3\}}\left\{\frac{\overline{\overline{\alpha}}_j}{f_j^{\max} - f_j^{\min}}\right\}$. This clearly comes with a risk of finding R such that $|R|$ is large and hence, long computation time. (b) Define a new scalar coverage gap that takes the vector coverage into account in its definition. To illustrate why the first option is not viable, consider a normalization of $\mathbf{z}$ and R, which yields $\mathbf{z}^{\text{nm}} = \begin{pmatrix} 0 \\ 0.16 \\ 0.24 \end{pmatrix}$, $R^{\text{nm}} = \left\{\begin{pmatrix} 0.04 \\ 0.33 \\ 0.22 \end{pmatrix}, \begin{pmatrix} 0.08 \\ 0.33 \\ 0.20 \end{pmatrix}\right\}$, and $\alpha_{R^{\text{nm}}}(\mathbf{z}^{\text{nm}}) = \min\{\max\{0.04, 0.17, -0.02\}, \max\{0.08, 0.17, -0.04\}\} = 0.17$. So, for a desired vector coverage gap $\overline{\overline{\boldsymbol{\alpha}}} = (0.1, 1, 10)^\top$ we compute a scalar coverage gap $\overline{\alpha} = \min_{j=1,2,3}\left\{\frac{\overline{\overline{\alpha}}_j}{f_j^{\max} - f_j^{\min}}\right\} = 0.045$. This indicates that $\mathbf{z}$ needs to be added to the set R as $0.17 > 0.045$, which is not true. Hence, there is a risk that it would result in finding more NDPs than necessary. Hence, we need a definition that is tailored to a desired vector coverage gap.

Definition 11 (*Constrained Coverage Gap w.r.t. $\overline{\overline{\boldsymbol{\alpha}}} \in \mathbb{R}^p_+$*). Consider a representation (normalized) R (R^{nm}) of the set of all the NDPs F_{ndp} ($F^{\text{nm}}_{\text{ndp}}$) for the DMOOP (1). The constrained coverage gap of $\mathbf{z} \in F_{\text{ndp}}$

($\mathbf{z}^{\text{nm}} \in F^{\text{nm}}_{\text{ndp}}$) w.r.t. the desired vector coverage gap $\overline{\overline{\boldsymbol{\alpha}}} \in \mathbb{R}^p_+$ is defined as the scalar quantity

$$\alpha^{\text{c}}_{R^{\text{nm}}}(\mathbf{z}^{\text{nm}}, \overline{\overline{\boldsymbol{\alpha}}}) := \min_{\mathbf{y}^{\text{nm}} \in R^{\text{nm}}, \gamma \geq 0} \left\{ M \mathbb{1}_{\{\gamma > 0\}} + \max_{i=1,\dots,p} \{(y^{\text{nm}}_i - z^{\text{nm}}_i)\} \right\}, \tag{7a}$$

$$\text{where} \quad y^{\text{nm}}_i - z^{\text{nm}}_i - \gamma \leq \frac{\overline{\alpha}_i}{z^{\max}_i - z^{\min}_i}, \quad i = 1, \dots, p, \tag{7b}$$

$M \gg 1$, $\mathbb{1}_{\{\gamma>0\}}$ is an indicator function,[12] $z^{\text{nm}}_i := \frac{z_i - z^{\min}_i}{z^{\max}_i - z^{\min}_i}$ ($y^{\text{nm}}_i := \frac{y_i - z^{\min}_i}{z^{\max}_i - z^{\min}_i}$, $\mathbf{y} \in R$), $z^{\min}_i := \min\{ z_i : \mathbf{z} \in F_{\text{ndp}} \}$, and $z^{\max}_i := \max\{ z_i : \mathbf{z} \in F_{\text{ndp}} \}$. The constrained coverage gap corresponding to $\overline{\overline{\boldsymbol{\alpha}}}$ is defined as $\alpha^{\text{c}*}_R(\overline{\overline{\boldsymbol{\alpha}}}) = \alpha^{\text{c}*}_{R^{\text{nm}}}(\overline{\overline{\boldsymbol{\alpha}}}) := \max_{\mathbf{z}^{\text{nm}} \in F^{\text{nm}}_{\text{ndp}}} \left\{ \alpha^{\text{c}}_{R^{\text{nm}}}(\mathbf{z}^{\text{nm}}, \overline{\overline{\boldsymbol{\alpha}}}) \right\}$.[13] □

Transforming back y^{nm}_i and z^{nm}_i to y_i and z_i in (7b), yields the constraint $(y_i - z_i) - \gamma(z^{\max}_i - z^{\min}_i) \leq \overline{\alpha}_i, i = 1, \dots, p$. Note that if the optimal solution $(\mathbf{y}^*, \gamma^*)$ to (7) satisfies $\gamma^* > 0$, then this implies that no point in R satisfies the desired vector coverage gap $\overline{\overline{\boldsymbol{\alpha}}}$. Furthermore, if $\gamma^* = 0$ then at least one point $\mathbf{y}^* \in R$ satisfies the desired vector coverage gap for a given point $\mathbf{z} \in F_{\text{ndp}}$. Since $\alpha_{R^{\text{nm}}}(\mathbf{z}^{\text{nm}}) = \alpha^{c}_{R^{\text{nm}}}(\mathbf{z}^{\text{nm}}, \overline{\overline{\boldsymbol{\alpha}}})$ holds if $\gamma^* = 0$ (the l.h.s. refers to Definition 1), it follows that Definition 1 is a special case of Definition 11. Moreover, if for each $\mathbf{z}^{\text{nm}} \in F^{\text{nm}}_{\text{ndp}}$ there exist a point $\mathbf{y}^{\text{nm}} \in R^{\text{nm}}$ satisfying the condition that the corresponding $\gamma^* = 0$ (or $\alpha_{R^{\text{nm}}}(\mathbf{z}^{\text{nm}}) = \alpha^{c}_{R^{\text{nm}}}(\mathbf{z}^{\text{nm}}, \overline{\overline{\boldsymbol{\alpha}}})$), then $\alpha^*_{R^{\text{nm}}} = \alpha^{\text{c}*}_{R^{\text{nm}}}(\overline{\overline{\boldsymbol{\alpha}}})$ (cf. Definition 1).

3. Computational experiments and results

We provide an overview of the input data, the computational setup, and the experimental design. All the algorithms are run on 1.70 GHz processor, 16 GB RAM, and 4 cores. We use `Python 3.9` and `Gurobi 9` as the solver. The sub-problems (in Alg. 2) are solved on parallel identical computers in a cluster. Each computer solves a sub-problem and its full processor capacity and all CPU cores are available to `Gurobi` when solving a scalarization in a given sub-problem. For comparison, we have considered TDA from (Ceyhan et al., 2019) and GPBA-A from (Mesquita-Cunha et al., 2023). We implement the TDA (as mentioned in (Ceyhan et al., 2019, Alg. 2, p. 67)) and the code of GPBA-A is shared http://bit.ly/41vfEY7 by Mesquita-Cunha et al. (2023).[14]

3.1. Multi-dimensional tri-objective binary knapsack problem (3KP)

The first set of instances belong to the multi-dimensional tri-objective knapsack problem (3KP).[15] All three objectives are maximized, integer-valued and assumed to be commensurable i.e., we use a scalar desired coverage gap. There are 10 classes of the problem having different numbers of variables (n) and constraints (g). Each of these classes is studied for 10 instances. For each instance in a class, both profit and constraint coefficients are drawn from a uniform distribution using different seed values. These details are presented in (Mesquita-Cunha et al., 2023, Sec. 4.3.1); the ten chosen instances per class are uploaded at https://bit.ly/47zUb3M. The two stopping criteria used are the MIP duality gap of 10^{-4} (near-optimal) and a time limit of 7200 s (for class J we increase it to 10 800 s). In order to perform computational comparisons we must specify $\overline{\alpha} \in \mathbb{Z}_+$. The computational performance may vary across different values of $\overline{\alpha}$ as it influences the cardinality of the representation and consequently, the computational time. Hence, for a fair comparison, we need to try a few reasonable values of $\overline{\alpha}$ for each instance. For this purpose, we devise a procedure that is based on the fact that users may want to compute at least a certain fraction of all the NDPs. The number of computed NDPs as a fraction of all the NDPs is denoted as a parameter λ. For instance, the user may wish to compute at least $\lambda = 0.2$ (or 20%) of the total NDPs of a given instance. Given a value of λ, we use a simple procedure similar to a binary search (see Alg. 3) to pre-compute an $\overline{\alpha}$, such that the representation contains at least $\lceil \lambda \cdot |F_{\text{ndp}}| \rceil$ NDPs. We believe that this results in a fair comparison, as choosing $\overline{\alpha}$ randomly may yield results biased towards certain algorithms. For instance, some algorithms are good at computing a representative set with fewer NDPs satisfying desired coverage gap where some algorithms might have easy to solve individual scalarization. Hence, by pre-computing $\overline{\alpha}$ using the values of λ gives us a better understanding. The obtained desired scalar coverage gap $\overline{\alpha}$ is then used for comparing the three algorithms $\overline{\alpha}$-PQSM, TDA, and GPBA-A.

Summary statistics over all ten instances for each given class (A–I), value of $\lambda \in \{0.2, 0.5, 0.8\}$, and each of the three algorithms are presented in Table 1.[16] For instance, class A has $n = 10$ variables and $g = 5$ constraints.

Column 4 provides values of the parameter λ used in Alg. 3 to compute corresponding coverage gaps $\overline{\alpha}$. Column 5 provides averages of desired coverage gaps, $\overline{\alpha}$. Columns 6, 8, and 10 provide average solution times in seconds for the respective algorithms; the mark '*' indicates that the algorithm was stopped prematurely due to time limitation, with no guarantee of achieving the desired coverage gap. Column 12 shows the fraction of instances for which $\overline{\alpha}$-PQSM has the same or shorter computational time than the other two algorithms. Columns 7, 9, and 11 present the average number of NDPs identified in the representative set for the respective algorithms; notably, our proposed algorithm $\overline{\alpha}$-PQSM computes fewer NDPs but still guarantees $\overline{\alpha}$-coverage, which is a positive aspect in comparison with the other algorithms.

The key insight from our test results is that $\overline{\alpha}$-PQSM is the best performing algorithm in 264 out of 300 instances w.r.t. solution time. GPBA-A shows competitive performance on a few instances (36 out of 300) particularly in the classes I and J (i.e. simple knapsack problems, with $g = 1$). Hence, it is safe to say that GPBA-A is competitive for simple knapsack problems ($g = 1$) with $n \geq 75$ variables. However, for reverse cases, i.e. instances with more constraints and fewer variables, our algorithm performs significantly better. Notably, GPBA-A's efficacy is improved when $\lambda \in \{0.5, 0.8\}$ for classes I and J, indicating a strength when there are more points are needed in a representation.

We next illustrate a set of comparative measures of the results from the three methods. Fig. 6(a) shows that the classes I and J have substantial numbers of NDPs, while Fig. 6(b) reveals that the count of representative points identified by $\overline{\alpha}$-PQSM is markedly lower than those found by GPBA-A. This suggests the existence of a minor computational advantage in solving individual scalarization while using GPBA-A. This is accentuated in classes I and J, which comprise thousands of NDPs. Nevertheless, in such cases it could be wiser to employ smaller values of λ. Additionally, the presence of more NDPs does not inherently elevate the significance of any class, as the complexity of solving a single instance of the 3KP problem should theoretically depend on the product of the numbers of constraints and variables, $n \cdot g$. Therefore, classes I and J should not be considered more significant than, for instance, classes F and G; they hold equal importance in this context. For almost all the test instances and algorithms we managed to solve all the scalarizations to near-optimality.

[12] $\mathbb{1}_{\{\gamma>0\}} = 1$ if $\gamma > 0; 0$, otherwise.

[13] There is no need to solve problem (7) if the sets R and F_{ndp} are given. This definition is used in the computational section to compare algorithms when there is a desired vector coverage gap, since Definition 1 is not sufficient as a scalar performance measure.

[14] We modified the code to use Gurobi as the original code used CPLEX.

[15] We also tested on instances of simple knapsack problems, with one knapsack constraint.

[16] An `xlsx` file with detailed computational results is shared at https://bit.ly/3TyG5tH.

Table 1
Average results for the 3KP instances. For each of the classes A–I and values of $\lambda \in \{0.2, 0.5, 0.8\}$ ten instances were solved using the algorithms $\overline{\alpha}$-PQSM, TDA, and GPBA-A, respectively. time [s]: average solution times in seconds. #NDPs: average of $|R|$ needed to reach the desired coverage. best: fraction of instances for which the computing time for $\overline{\alpha}$-PQSM was the shortest (or equal).

Class	n	g	λ	$\overline{\alpha}$ (avg)	$\overline{\alpha}$-PQSM (avg)		TDA (avg)		GPBA-A (avg)		Best
					Time [s]	#NDPs	Time [s]	#NDPs	Time [s]	#NDPs	
A	10	5	0.2	12.8	0.16	3.5	0.49	5.0	0.86	5.8	1.0
			0.5	5.3	0.23	6.2	0.84	7.4	1.90	8.0	1.0
			0.8	2.0	0.30	9.0	0.95	9.1	3.60	9.6	1.0
B	15	10	0.2	10.8	0.71	8.4	5.72	17.9	2.07	10.8	1.0
			0.5	2.9	3.62	19.6	9.21	22.7	9.25	21.4	1.0
			0.8	2.0	3.88	20.6	8.77	22.2	10.73	22.6	1.0
C	20	15	0.2	9.5	1.58	12.1	15.04	29.5	3.85	17.5	1.0
			0.5	2.9	3.62	33.1	9.21	36.3	9.25	35.2	1.0
			0.8	2.0	3.88	33.1	8.77	36.3	10.73	35.2	1.0
D	25	2	0.2	44.6	1.15	19.2	46.99	65.9	5.14	32.0	1.0
			0.5	14.5	4.46	45.9	56.80	68.2	18.30	55.6	1.0
			0.8	3.7	14.54	74.3	65.50	76.9	72.81	77.2	1.0
E	25	3	0.2	44.2	1.74	19.5	47.75	61.5	5.29	51.7	1.0
			0.5	12.7	7.54	42.3	57.66	65.1	20.72	74.4	1.0
			0.8	2.4	25.01	72.5	57.91	74.6	90.52	25.8	1.0
F	25	4	0.2	50.6	1.23	17.4	28.80	54.1	3.99	45.4	1.0
			0.5	15.7	4.58	37.2	37.79	59.2	12.48	66.4	1.0
			0.8	3.3	18.83	64.1	42.84	66.9	66.56	27.2	1.0
G	25	5	0.2	46.0	1.66	17.5	37.10	60.5	4.30	27.2	1.0
			0.5	15.2	6.37	37.7	45.62	63.1	14.40	49.9	1.0
			0.8	3.6	25.53	67.1	51.35	69.9	70.64	71.3	1.0
H	50	1	0.2	35.1	7.87	70.9	1826.69	255.6	16.64	122.0	1.0
			0.5	9.8	53.24	205.2	1887.16	235.8	86.15	272.7	1.0
			0.8	2.0	188.73	346.6	1968.33	244.6	375.59	360.2	1.0
I	75	1	0.2	25.0	60.48	265.4	7200[a]	368.7	66.39	366.8	0.7
			0.5	7.30	365.48	686.7	7200[a]	864.3	352.33	844.0	0.5
			0.8	2.0	1570.63	1130.6	7200[a]	1187.5	1376.75	1175.3	0.5
J	100	1	0.2	20.8	224.47	625.5	10 800[a]	760.1	182.74	790.5	0.5
			0.5	6.0	1590.51	1130.6	10 800[a]	1790.1	1090.02	1873.5	0.2
			0.8	2.0	5202.00	2214.6	10 800[a]	2280.7	3203.33	2303.1	0.0

[a] The computing time limit is reached.

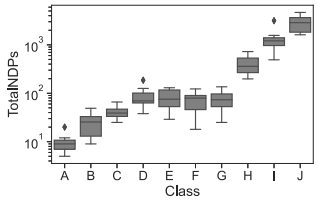
Hence, the desired coverage gap is guaranteed, except where '*' is marked in Table 1. Fig. 7(a) shows the ratio of solution time (obtained by applying a given algorithm for a given instance and λ) to the least solution time (over the three algorithms for a given instance and λ).[17] In Fig. 7(b) the effect of varying λ is analyzed; the variation in computational performance of TDA reduces as λ increases, while $\overline{\alpha}$-PQSM has a considerably smaller performance variation than the other two algorithms. Fig. 8(a) illustrates a performance profile (Dolan and Moré, 2002), which uses the performance ratio $r_{pa} := \frac{t_{pa}}{\min_{a\in\mathcal{A}}\{t_{pa}\}}$, where t_{pa} denotes the solution time for problem instance $p \in \mathcal{P}$ using algorithm $a \in \mathcal{A} := \{\text{GPBA-A}, \text{TDA}, \overline{\alpha}\text{-PQSM}\}$, and the cumulative distribution function $P_a(\omega) := |\mathcal{P}|^{-1}\left|\{p \in \mathcal{P} : r_{pa} \le \omega\}\right|$, $\omega \in [1, \infty)$, $a \in \mathcal{A}$. Fig. 8(b) illustrates a scatter plot of the cardinality of the representation, $|R|$, and the least number of NDPs desired. Note that the total number of NDPs $|F_{\text{ndp}}|$ (required to compute the value on x-axis in Fig. 8(b)) was computed separately to understand the reason behind bad performance of some algorithms and is not part of the procedure to find the representative set. There are 300 data points in total and on the x-axis we indicate corresponding $\lambda|F_{\text{ndp}}|$ which implies approximately how many NDPs were required and on y-axis we have illustrated $|R|$ for each instance and λ. It is clear that TDA identifies more NDPs for the same desired coverage gap as compared to the other algorithms. Of course, this comes at the cost of computation which is reflected in Table 1. As illustrated in Fig. 9, although the #NDPs is larger for TDA, the other two algorithms still manage to find representations that satisfy the desired coverage gap $\overline{\alpha}$ with fewer #NDPs and larger average coverage gaps (the average coverage gaps for each class and λ value is represented on the vertical axis). It is clear that TDA generally finds a representation possessing a coverage gap much lower than desired, but this has a negative impact on the computational time (see Table 1), and for class J most of the representations do not even satisfy the desired coverage gap. The superior performance of $\overline{\alpha}$-PQSM as compared to TDA and GPBA-A can be attributed to its novel two-stage decomposition technique (which is inherited from the original QSM by Boland et al., 2017), our proposed parallelization approach, and finally the allocation procedure from Section 2.3 to reduce redundancy. The drawback regarding weak *individual representative power* (IRP) (cf. Section 1.2) of the computed representative set of TDA is discussed in (Doğan et al., 2022).

3.2. Generalized tactical resource allocation problem

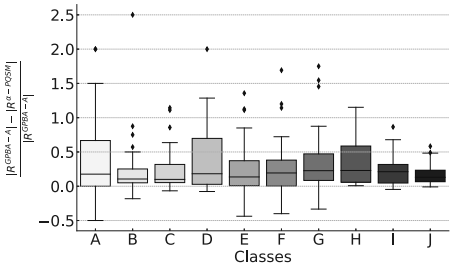
This test set is based on an industrial production planning problem of interest to us. The problem defined in (Fotedar et al., 2023b) is called generalized tactical resource allocation problem (GTRAP), a discrete tri-objective optimization problem having one real-valued and two integer-valued objective functions,[18] all three to be minimized. The objectives are defined as the sum (over multiple time periods) of the maximum resource loading above a given *threshold* over all the machines, the total *qualification cost*, and the total inventory, respectively. For details, see Appendix A.3; the public instances are shared at https://bit.ly/3keKzY4. Since the three objectives are incommensurable, applying a scalar desired coverage gap to all three objectives

[17] There are 300 data points: 10 classes × 10 instances × 3 values of λ.

[18] The discreteness of the Pareto front is proved in (Fotedar et al., 2023a, Prop. 3).

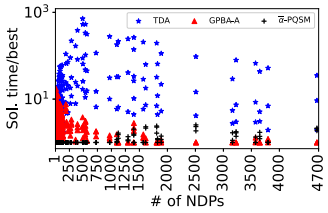


(a) Total #NDPs; 10 instances per class

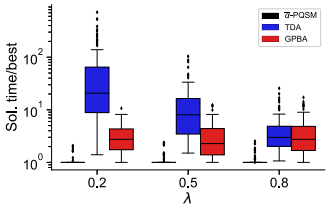


(b) Relative increase in numbers of representative points for GPBA-A as compared to $\overline{\alpha}$-PQSM

Fig. 6. Box plots for the classes A–I.

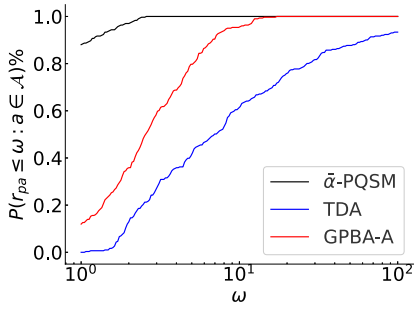


(a) Scatter plot of the ratio of solution time to the solution time of the best algorithm versus the total # of NDPs for each 3KP instance

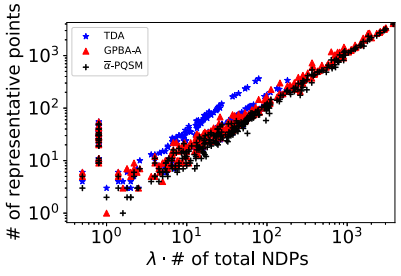


(b) Box plot of the ratio of solution time to the solution time of the best algorithm for each algorithm and λ

Fig. 7. Ratios of solution times.



(a) Performance profiles for the three algorithms with respect to solution times



(b) Scatter plot of $|R|$ versus corresponding number of NDPs desired as given by λ.

Fig. 8. Illustrations of computational results for the 3KP instances.

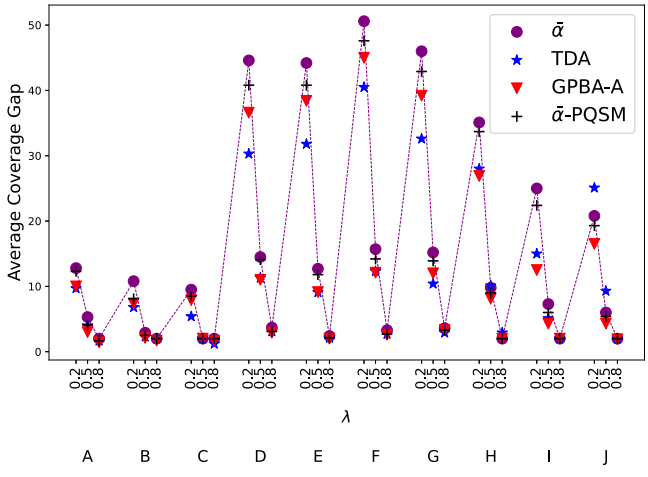


Fig. 9. Comparison of average values (cf. Table 1) of the desired coverage gap and the coverage gap obtained from each of the three algorithms, for the classes A–I and values of $\lambda \in \{0.2, 0.5, 0.8\}$.

may not be prudent (cf. Section 2.5). Hence, we applied a three-dimensional desired coverage gap, $\overline{\overline{\boldsymbol{\alpha}}} \in \mathbb{R}^3_+$. The guarantee of the desired vector coverage is derived from the scalar case, since all the ϵ-constraints in the two-stage scalarization separate (similar for the other two algorithms). Hence, the only change needed is that while excluding regions in lines 19, 23, 38, and 42 of Alg. 2, the correct components $\overline{\overline{\alpha}}_1$, $\overline{\overline{\alpha}}_2$, $\overline{\overline{\alpha}}_2$, and $\overline{\overline{\alpha}}_1$, respectively, are utilized. As mentioned above, this implies that all scalarizations must be solved to optimality in order to guarantee $\overline{\overline{\alpha}}$-coverage.

Computational set-up For each scalarized problem, we use three stopping criteria: (a) MIP duality gap of 10^{-4}; (b) a time limit of 1500 s; (c) MIP absolute gap limit is ≤ 0.01 and there have been no improvements in the duality gap for the previous 100 nodes of the branch-and-bound tree; hence, the solutions found are approximately efficient. We also employ a global computing time limit of 7200 s. The discretization of the real-valued function f_1 and time limitations introduce certain approximations. As a result, the identified representations may contain non-dominated points (NDPs) as well as their best-known approximations. In order to compare the performance of the algorithms, we must define the best-known approximation of the Pareto front. Then, the

Table 2
Results for the GTRAP instances for a vector coverage gap $\overline{\overline{\boldsymbol{\alpha}}}$. The decomposition is done using D_1; cf. Alg. 1.

Instance	$\overline{\overline{\boldsymbol{\alpha}}}$			$\overline{\alpha}$-PQSM			TDA			GPBA-A		
	$\overline{\overline{\alpha}}_1$	$\overline{\overline{\alpha}}_2$	$\overline{\overline{\alpha}}_3$	$\alpha^{c*}_{R^{nm}}(\overline{\overline{\boldsymbol{\alpha}}})$	Time	NDP	$\alpha^{c*}_{R^{nm}}(\overline{\overline{\boldsymbol{\alpha}}})$	Time	NDP	$\alpha^{c*}_{R^{nm}}(\overline{\overline{\boldsymbol{\alpha}}})$	Time	NDP
1	0.1	2	50	1.2	1308	10	1.0	2357	11	M	7200*	9
	0.1	1	100	0.5	4378	15	1.0	7200*	15	M	7200*	10
2	0.1	2	50	1.33	1226	12	0.66	4022	12	M	7200*	7
	0.1	1	100	0.50	6742	17	M	7200*	15	M	7200*	8
3	0.1	2	50	1.33	898	7	0.66	3446	10	M	7200*	9
	0.1	1	100	0.50	4294	17	M	6717	16	M	7200*	12
4	0.1	2	50	M	2782	13	M	7200*	14	M	7200*	12
	0.1	1	100	M	5045	21	M	7200*	20	M	7200*	15
5	0.1	2	50	0.50	3066	10	0.50	7200*	13	M	7200*	10
	0.1	1	100	0.92	7200*	14	M	7200*	15	M	7200*	13
6	0.1	2	50	0.68	6483	5	M	7200*	4	M	7200*	3
	0.1	1	100	M	7200*	6	M	7200*	5	M	7200*	3
7	0.1	2	50	0.50	520	10	0.50	3170	12	M	7200*	7
	0.1	1	100	0.50	3208	17	M	7200*	16	M	7200*	9
8	0.1	2	50	0.50	4064	12	0.66	7200*	14	M	7200*	11
	0.1	1	100	0.65	7200*	15	M	7200*	15	M	7200*	10
9	0.1	2	50	0.50	688	7	0.50	2896	7	M	7200*	5
	0.1	1	100	0.50	2257	12	0.50	7200*	12	M	7200*	7
10	0.1	2	50	0.73	5616	9	M	7200*	10	M	7200*	7
	0.1	1	100	M	3022	7	M	7200*	8	M	7200*	5

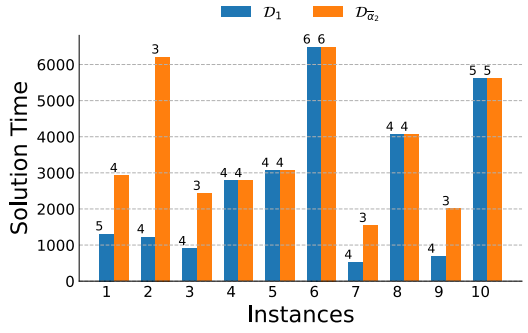


Fig. 10. Comparison of solution time when using D_1 and $D_{\overline{\alpha}_2}$ while solving the GTRAP for $\overline{\boldsymbol{\alpha}} = (0.1, 2, 50)^\top$. The bars are annotated with $|D_1|$ and $|D_{\overline{\alpha}_2}|$.

performance of each algorithm is evaluated based on solution time and the so-called constrained coverage gap of the respective representation (cf. Definition 11).

Due to the practical instances being large-scale, for some instances we are unable to solve each scalarization to optimality due to computing time limitation. Hence, we consider the best-known approximation of the set of NDPs by combining the output from all three algorithms ($\overline{\alpha}$-PQSM, TDA, and GPBA-A). Let $F^p_{\underset{\sim}{\text{ndp}}}(a)$ be the set of points identified by algorithm $a \in \mathcal{A}$ for instance p of the GTRAP. Furthermore, we define $F^p_{\underset{\sim}{\text{ndp}}}(\mathcal{A})$ as the set of points that are non-dominated in the union $\cup_{a\in\mathcal{A}} F^p_{\underset{\sim}{\text{ndp}}}(a)$. So, $F^p_{\underset{\sim}{\text{ndp}}}(\mathcal{A})$ is the best-known approximation[19] of the set of all the NDPs for an instance p. We compute $F^p_{\underset{\sim}{\text{ndp}}}(\mathcal{A})$ as the best-known approximation of the set of all the NDPs. Hence, the values of $\alpha^{c*}_{R^{nm}}(\overline{\overline{\boldsymbol{\alpha}}})$ reported in Table 2, columns 5, 8, and 11, are constrained coverage gaps for each instance and a given representation $F^p_{\underset{\sim}{\text{ndp}}}(a)$ obtained from algorithm $a \in \mathcal{A}$ and instance p. Consequently, in Definition 11, we set $F_{\text{ndp}} := F^p_{\underset{\sim}{\text{ndp}}}(\mathcal{A})$ and $R := F^p_{\underset{\sim}{\text{ndp}}}(a)$ for a given instance p and algorithm a. The mark "M" represents that for at least one of the points $\mathbf{z} \in F^p_{\underset{\sim}{\text{ndp}}}(\mathcal{A})$ there exists no point $\mathbf{y} \in F^p_{\underset{\sim}{\text{ndp}}}(a)$ for which the corresponding $\gamma^* = 0$ (cf. (7)). This means that the given representation does not satisfy the desired vector coverage gap $\overline{\overline{\boldsymbol{\alpha}}}$. The choice of the two variations of the desired coverage gap shown in Table 2 is based on our understanding of the range of values for these instances (and end-user preferences). Our experience with the instance of interest reveals that the second objective has limited distinct values due to its integer-valued function, which operates within a small interval. As a result, increasing $\overline{\overline{\alpha}}_2$ generally reduces solution time, as confirmed in Table 2. Our proposed algorithm $\overline{\alpha}$-PQSM has a lower or equal constrained coverage gap as compared to TDA in 17 (out of 20) instances. There are seven occurrences when $\overline{\alpha}$-PQSM satisfied the desired coverage gap $\overline{\overline{\boldsymbol{\alpha}}}$ but TDA did not (the opposite never happened). Furthermore, for $\overline{\alpha}$-PQSM we see "M" only four times as compared to eleven (out of 20) times for the TDA. Most likely, the reason for the poor performance of GPBA-A can be attributed to its bad performance when an optimization model has several knapsack constraints. In the GTRAP model, there are several knapsack constraints linked to the limitation on number of set-ups (9f) and qualifications (9i) (these constraints can be in thousands). Furthermore, as discussed for the 3KP instances, GPBA-A seems to perform well only when the ε-constraints on the objective is one. These ε-constraints are also a form of knapsack constraints.

3.3. Insights and conclusion

In our study, we introduce an algorithm—denoted as $\overline{\alpha}$-PQSM to identify a representation of the Pareto front and evaluated its performance on several instances of two distinct types of problems. For the 3KP instances, our algorithm generally outperforms the other two algorithms, particularly for 240 of the 300 instances. However, in some of the instances the Grid point based algorithm (GPBA-A) emerges as a formidable competitor. This dynamic shifts with the GTRAP instances, where GPBA-A falls significantly behind both TDA (Territory-Defining Algorithm) and $\overline{\alpha}$-PQSM. Our analysis suggests that the notable efficiency of $\overline{\alpha}$-PQSM for the industrial GTRAP instances can be attributed to the larger number of sub-problems i.e. $|D_1| > 1$ for the GTRAP instances (see Fig. 10). This contrasts with the 3KP instances, which

[19] This best-known approximation set is finite and stable w.r.t. the dominance relation $\preceq$. This implies for any two points $\mathbf{z}^1$ and $\mathbf{z}^2$ in $F^i_{\underset{\sim}{\text{ndp}}}(\mathcal{A})$, $\mathbf{z}^1 \npreceq \mathbf{z}^2$ holds.

predominantly feature only two sub-problems,[20] i.e. $|D_1| = 1$. Furthermore, how we select the parameter to discretize the projected two-dimensional criterion space also led to further improvement. In Proposition 9 we discussed the parameter to discretize the projected two-dimensional criterion space which sets up the number of parallel subproblems that required to be solved. It turns out that it is better to use $\overline{\alpha}^{\text{dis}} = 1$ rather than $\overline{\alpha}^{\text{dis}} = \overline{\alpha}_2 > 1$ w.r.t. the solution time. This is somewhat counterintuitive as it holds that $|D_1| \geq |D_{\overline{\alpha}_2}|$ when $\overline{\alpha}_2 > 1$, however, it was not clear how will this impact the solution time while computing rest of the NDPs in conjunction with the parallelization technique used. In Fig. 10 it is illustrated that opting for $\overline{\alpha}^{\text{dis}} = 1$ rather than $\overline{\alpha}^{\text{dis}} = \overline{\alpha}_2 > 1$ proves computationally advantageous, increasing the number of sub-problems solved simultaneously and consequently reducing computational burden. The blue bars are annotated with number of subproblems solved and it is consistently higher than $|D_{\overline{\alpha}_2}|$ bar in orange. However, the corresponding solution time is lower when the number of subproblems is larger.

Our proposed approach could be generalized to higher-dimensional objective vectors, making it an interesting area for further research.

CRediT authorship contribution statement

Sunney Fotedar: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ann-Brith Strömberg:** Writing – review & editing, Supervision, Funding acquisition.

Funding source

This work is financially supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA; project no 2017-04845), Chalmers University of Technology, and GKN Aerospace Sweden AB

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Funding was provided by VINNOVA, Sweden (Grant No. 2017-04845) and Chalmers University of Technology, Sweden.

Appendix

A.1. Proofs of propositions

Proof of Proposition 8. Since $\mathcal{D} \neq \emptyset$, it holds that $\Theta^{f_3}(X, f_3^{\min}) \neq \emptyset$. We first show that the inclusion $\{(\mathbf{u}, f_3^{\min}) \,|\, \mathbf{u} \in \widehat{\mathcal{D}}\} \subseteq \mathcal{D}$ holds; then that the inclusion $\mathcal{D} \subseteq \{(\mathbf{u}, f_3^{\min}) \,|\, \mathbf{u} \in \widehat{\mathcal{D}}\}$ holds. Let us consider an NDP $\mathbf{u}^{\text{b}} = (f_1(\mathbf{x}^{\text{b}}), f_2(\mathbf{x}^{\text{b}}))^\top \in \widehat{\mathcal{D}}$ (i.e. for the bi-objective MOOP) and assume the contradiction, that $(\mathbf{u}^{\text{b}}, f_3^{\min}) \notin \mathcal{D}$. This implies there exists an NDP $(\mathbf{u}^*, f_3^{\min}) \in \mathcal{D}$ (with corresponding solution $\mathbf{x}^* \in \Theta^{f_3}(X, f_3^{\min})$ and $(f_1(\mathbf{x}^*), f_2(\mathbf{x}^*))^\top = \mathbf{u}^*$) such that $(\mathbf{u}^*, f_3^{\min}) \leq (\mathbf{u}^{\text{b}}, f_3^{\min})$ (see (6)) which implies that $u_1^* \leq u_1^{\text{b}}$, $u_2^* \leq u_2^{\text{b}}$, and $\mathbf{u}^* \neq \mathbf{u}^{\text{b}}$. Hence, it holds that $f_i(\mathbf{x}^*) \leq f_i(\mathbf{x}^b)$, for $i = 1, 2$, and $(f_1(\mathbf{x}^*), f_2(\mathbf{x}^*)) \neq (f_1(\mathbf{x}^b), f_2(\mathbf{x}^b))$. We know that $\mathbf{x}^* \in \Theta^{f_3}(X, f_3^{\min})$. Hence, $\mathbf{x}^b$ is not an efficient solution in the bi-objective MOOP, which is a contradiction. The inclusion follows.

Now, consider $(\mathbf{u}^{\text{t}}, f_3^{\min}) \in \mathcal{D}$, where $u_i^{\text{t}} = f_i(\mathbf{x}^t)$, $i = 1, 2$, and $f_3(\mathbf{x}^t) = f_3^{\min}$. Assume that $\mathbf{u}^{\text{t}} \notin \widehat{\mathcal{D}}$; then there exists an NDP $\mathbf{u}^* \in \widehat{\mathcal{D}}$ such that $u_1^* \leq u_1^{\text{t}}$, $u_2^* \leq u_2^{\text{t}}$, and $\mathbf{u}^* \neq \mathbf{u}^{\text{t}}$. The corresponding solution fulfills $\mathbf{x}^* \in \Theta^{f_3}(X, f_3^{\min})$ such that $(f_1(\mathbf{x}^*), f_2(\mathbf{x}^*))^\top = \mathbf{u}^*$; which implies that $(\mathbf{u}^{\text{t}}, f_3^{\min}) \notin \mathcal{D}$, i.e. a contradiction. The proposition follows. □

[20] Maximum number of subproblems are $|D_1| + 1$.

Proof of Proposition 9. Since $\mathcal{D}_1 \supseteq \mathcal{D}_{\overline{\alpha}}$, it implies $R^- \subseteq R$, hence, $\alpha_R^* \leq \alpha_{R^-}^*$. We prove that $\alpha_R^* = \alpha_{R^-}^*$. Let us consider an NDP $\mathbf{f}^s \in R \setminus R^-$ which implies $\mathbf{f}^s \in \mathcal{D}_1 \setminus \mathcal{D}_{\overline{\alpha}}$. Now, we consider to the contrary that $\mathbf{f}^s$ satisfy $\alpha_{R^-}(\mathbf{f}^s) > \overline{\alpha}$. We consider that $f_2^k > f_2^s > f_2^{k+1}$, where $\mathbf{f}^k, \mathbf{f}^{k+1}$ are NDPs computed successively by Alg. 1 using $\overline{\alpha}^{\text{dis}} = \overline{\alpha}$ ($\{\mathbf{f}^k, \mathbf{f}^{k+1}\} \subseteq D_{\overline{\alpha}}$). Then $\alpha_{R^-}(\mathbf{f}^s) > \overline{\alpha}$ implies $f_2^k - f_2^s > \overline{\alpha}$ and $f_1^{k+1} - f_1^s > \overline{\alpha}$. The former inequality and the fact $\mathbf{f}^s$ is an NDP implies that $\mathbf{f}^s \in D_{\overline{\alpha}}$, which is a contradiction. □

Proof of Proposition 10. Consider the contrary, that there is an NDP $\mathbf{f}^j$ with projection $\hat{\mathbf{f}}^j \in \widehat{F}^i$ and $\alpha_R(\mathbf{f}^j) > \overline{\alpha}$. Note that $\hat{\mathbf{f}}^j$ is in at least one of the four regions of $\widehat{F}^i$ discussed below. For each region/case, we prove the existence of a distinct NDP $\mathbf{f}^j \neq \mathbf{f}^i$ such that $\alpha_R(\mathbf{f}^j) > \overline{\alpha}$ is not possible. (Region 1) Let the projection $\hat{\mathbf{f}}^j$ i.e. $(f_1^j, f_2^j)^\top \in \{\hat{\mathbf{f}} \in \widehat{F}^i \,|\, \mathbf{u}^i \geqq \hat{\mathbf{f}} \geqq \hat{\mathbf{f}}^i\}$. To have $\alpha_R(\mathbf{f}^j) > \overline{\alpha}$ we must have $f_3^j < f_3^i - \overline{\alpha}$ (for region 1). However, it is not possible (as indicated in Definition 4) $\mathbf{f}^i$ has the least value of the function f_3 for a given $\mathbf{u}^i$, and $\overline{\alpha} > 0$. Hence, it is not possible to find a feasible solution $\mathbf{x}^j$ such that $f_3^j = f_3(\mathbf{x}^j) < f_3^i - \overline{\alpha}$ and $\hat{\mathbf{f}}^i \leqq \hat{\mathbf{f}}^j \leqq \mathbf{u}^i$. (Region 2) Let the projection $\hat{\mathbf{f}}^j \in \{\hat{\mathbf{f}} \in \widehat{F}^i \,|\, \hat{\mathbf{f}}^i \geqq \hat{\mathbf{f}} > \hat{\mathbf{f}}^i - (\overline{\alpha}, \overline{\alpha})^\top\}$. We prove that $\mathbf{f}^j$ will have $\alpha_R(\mathbf{f}^j) < \overline{\alpha}$. We note that

$$\alpha_R(\mathbf{f}^j) = \min_{\mathbf{f} \in R}\Big\{\max_{\ell=1,2,3}\{f_\ell - f_\ell^j\}\Big\} \leq \min_{\mathbf{f} \in \{\mathbf{f}^i\} \subseteq R}\Big\{\max_{\ell=1,2,3}\{f_\ell - f_\ell^j\}\Big\}, \tag{8a}$$

$$\iff \alpha_R(\mathbf{f}^j) \leq \min\Big\{\max_{\ell=1,2}\{f_\ell^i - f_\ell^j\}\Big\} < \overline{\alpha}. \tag{8b}$$

For (8b) we use the fact that $\hat{\mathbf{f}}^j > \hat{\mathbf{f}}^i - (\overline{\alpha}, \overline{\alpha})^\top$ which implies $0 \leq f_\ell^i - f_\ell^j < \overline{\alpha}, \ell = 1, 2$ and we also know $f_3^j \geq f_3^i$ (because $\hat{\mathbf{f}}^j \leqq \mathbf{u}^i$ and then using Definition 4 we know $f_3^i \leq f_3^j$). Therefore, $\alpha_R(\mathbf{f}^j) < \overline{\alpha}$. (Region 3) Let the projection $\hat{\mathbf{f}}^j \in \{\hat{\mathbf{f}} \in \widehat{F}^i \,|\, f_1^i \leq f_1 \leq u_1^i; f_2^i - \overline{\alpha} < f_2 \leq f_2^i\}$. As shown in (8a) $\alpha_R(\mathbf{f}^j) \leq \max_{\ell=1,2,3}\{f_\ell^i - f_\ell^j\}$. But in region 3 we know that $f_1^i - f_1^j \leq 0$ and also by Definition 4 we know that $f_3^i - f_3^j \leq 0$, whereas $f_2^i - f_2^j \geq 0$. Therefore (8b) can be restated for this region as $\alpha_R(\mathbf{f}^j) \leq (f_2^i - f_2^j) < \overline{\alpha}$ (the last inequality should be clear by the definition of the region). (Region 4) Let the projection $\hat{\mathbf{f}}^j \in \{\hat{\mathbf{f}} \in \widehat{F}^i | f_2^i \leq f_2 \leq u_2^i; f_1^i - \overline{\alpha} < f_1 \leq f_1^i\}$. The proof is similar to Region 3 but with a desired coverage gap defined by the first function f_1. Therefore, the conclusion is $\alpha_R(\mathbf{f}^j) \leq (f_1^i - f_1^j) < \overline{\alpha}$. The proposition follows □

A.2. Algorithm

Algorithm 3 Binary search to find $\overline{\alpha}$ to obtain at least $100\lambda\%$ of all NDPs

```
Input: All data required for Alg. 2, λ, startAlpha,stepLength,alphaGapLimit
alphaCurr := startAlpha; repMax := Alg. 2(ᾱ = 1) ; maxNDP := |repMax|
while True do ▷ Find alphaCurr resulting in a singleton representation
    repCurr := Alg. 2(ᾱ = alphaCurr)
    alphaCurr:=alphaCurr + stepLength      ▷ if |repCurr| = 1 then break
end while
alphaCurrLeft := 1; alphaCurrRight := alphaCurr; alphaGap := alphaCurrRight − alphaCurrLeft
while (alphaGap ≤ alphaGapLimit and alphaCurrRight > alphaCurrLeft) do
    alphaCurr := alphaCurrLeft + (alphaCurrRight−alphaCurrLeft)/2; repCurr := Alg. 2(ᾱ = alphaCurr)
    if |repCurr| > max(⌈λ · maxNDP⌉, 1) then
        alphaCurrLeft := alphaCurr      ▷ Increase the value of alphaCurr
    else
        alphaCurrRight := alphaCurr     ▷ Reduce the value of alphaCurr
    end if
    alphaGap := (alphaCurrRight − alphaCurrRight)
end while
```

A.3. Extra material on the GTRAP test case

We consider three objective functions possessing equal priority, namely the excess resource loading, defined as $g_1 := \sum_{t \in \mathcal{T}} n_t$, the

Table 3
Notations for the GTRAP.

Sets	Description
$\mathcal{L} = \{1, \dots, L\}$	Set of part/product types
$\mathcal{J}_\ell = \{1, \dots, J_\ell\}$	Ordered set of indices of operations to produce part $\ell \in \mathcal{L}$; J_ℓ is the index of the final operation
$\mathcal{M}$	$:= \{(j,\ell) : j \in \mathcal{J}_\ell, \ell \in \mathcal{L}\}$ set of pairs (j,ℓ) of task indices
$\widetilde{\mathcal{J}}_\ell = \{0, \dots, J_\ell\}$	Extended ordered set of operations to produce a part type $\ell \in \mathcal{L}$ (the fictitious operation 0 represents raw material)
$\mathcal{K}$	Set of machines
$\mathcal{K}_j^\ell \subseteq \mathcal{K}$	Set of machines feasible for task $(j,\ell) \in \mathcal{M}$
$\mathcal{N}_j^\ell \subseteq \mathcal{K}_j^\ell$	Set of machines feasible but not qualified for task $(j,\ell) \in \mathcal{M}$
$\mathcal{T} = \{1, \dots, T\}$	Set of time periods (index 0 represents the beginning of time period 1)
Variables	**Description**
$x_{jkt}^\ell \in \mathbb{Z}_+$	Number of orders of task $(j,\ell) \in \mathcal{M}$ performed in machine $k \in \mathcal{K}_j^\ell$ in time period $t \in \mathcal{T}$
$s_{jkt}^\ell \in \{0,1\}$	Equals 1 if an order of task (j,ℓ) is allocated to machine $k \in \mathcal{K}_j^\ell$ in time period $t \in \mathcal{T}$; equals 0 otherwise
$z_{jkt}^\ell \in \{0,1\}$	Equals 1 if machine $k \in \mathcal{N}_j^\ell$ is qualified for task $(j,\ell) \in \mathcal{M}$, in time period $t \in \mathcal{T}$; equals 0 otherwise
$n_t \in \mathbb{R}_+$	Maximum resource loading above thresholds ζ, in time period $t \in \mathcal{T}$
$r_{jt}^\ell \in \mathbb{Z}_+$	Stock/inventory of raw material or (semi-)finished parts resulting from task $(j,\ell) \in \mathcal{M}$, at the end of time period $t \in \mathcal{T} \cup \{0\}$
$m_t^\ell \in \mathbb{Z}_+$	Total amount of raw materials ordered for part ℓ in time period t
Parameters	**Description**
$a_t^\ell \in \mathbb{Z}_+$	Demand/orders of part $\ell \in \mathcal{L}$ in time period $t \in \mathcal{T}$
$p_{jk}^\ell \in \mathbb{Q}_+$	Average processing time in machine $k \in \mathcal{K}_j^\ell$ for completing task $(j,\ell) \in \mathcal{M}$
$C_{kt} \in \mathbb{Z}_+$	Capacity (hours) available in machine $k \in \mathcal{K}$ in each time period $t \in \mathcal{T}$
$\beta_{jk}^\ell \in \mathbb{Z}_+$	Cost associated with qualifying machine $k \in \mathcal{N}_j^\ell$ for task $(j,\ell) \in \mathcal{M}$
$\gamma \in \mathbb{Z}_+$	Upper limit for the number of qualifications in any given time period
$\tau \in \mathbb{Z}_+$	Upper limit for the number of machines to which a given task can be assigned in each time period
$\zeta \in [0,1]$	Loading threshold for machines
M_{jkt}^ℓ	$:= \min\left\{\sum_{q\in\{t,\dots,T\}} a_q^\ell; \left\lfloor C_{kt}/p_{jk}^\ell \right\rfloor\right\}$ upper limit for machine $k \in \mathcal{K}_j^\ell$ on the production of task $(j,\ell) \in \mathcal{M}$, in time period $t \in \mathcal{T}$
$\bar{r}_{jt}^\ell \in \mathbb{Z}_+$	Upper limit for the inventory of raw material/semi-finished/finished part resulting from performing task $(j,\ell) \in \mathcal{M}$ in time period $t \in \mathcal{T}$
$\bar{r}_{j0}^\ell \in \mathbb{Z}_+$	Initial inventory ($t = 0$) of raw material/semi-finished/finished part for $j \in \widetilde{\mathcal{J}}_\ell$, $\ell \in \mathcal{L}$
$d \in \mathbb{Z}_+$	Limiting factor for the order of the raw material in any time period

sum of all qualification costs $g_2 := \sum_{t\in\mathcal{T}}\sum_{\ell\in\mathcal{L}}\sum_{j\in\mathcal{J}_\ell}\sum_{k\in\mathcal{N}_j^\ell}\beta_{jk}^\ell z_{jkt}^\ell$, and the total inventory, expressed as $g_3 := \sum_{t\in\mathcal{T}}\sum_{\ell\in\mathcal{L}}\sum_{j\in\widetilde{\mathcal{J}}_\ell} r_{jt}^\ell$. Please see supporting notation in Table 3. The constraints are expressed as

$$m_t^\ell + r_{0,t-1}^\ell - r_{0t}^\ell = \sum_{k\in\mathcal{K}_1^\ell} x_{1kt}^\ell, \qquad \ell \in \mathcal{L}, t \in \mathcal{T}, \tag{9a}$$

$$\sum_{k\in\mathcal{K}_j^\ell} x_{jkt}^\ell + r_{j,t-1}^\ell - r_{jt}^\ell = \sum_{k\in\mathcal{K}_{j+1}^\ell} x_{j+1,k,t}^\ell, \qquad j \in \mathcal{J}_\ell \setminus J_\ell, \ell \in \mathcal{L}, t \in \mathcal{T}, \tag{9b}$$

$$\sum_{k\in\mathcal{K}_{J_\ell}^\ell} x_{J_\ell kt}^\ell + r_{J_\ell,t-1}^\ell - r_{J_\ell t}^\ell = a_t^\ell, \qquad \ell \in \mathcal{L}, t \in \mathcal{T}, \tag{9c}$$

$$r_{j0}^\ell = \bar{r}_{j0}^\ell, \qquad j \in \widetilde{\mathcal{J}}_\ell, \ell \in \mathcal{L}, \tag{9d}$$

$$x_{jkt}^\ell \le M_{jkt}^\ell s_{jkt}^\ell, \qquad k \in \mathcal{K}_j^\ell, j \in \mathcal{J}_\ell, \ell \in \mathcal{L}, t \in \mathcal{T}, \tag{9e}$$

$$\sum_{k\in\mathcal{K}_j^\ell} s_{jkt}^\ell \le \tau, \qquad j \in \mathcal{J}_\ell, \ell \in \mathcal{L}, t \in \mathcal{T}, \tag{9f}$$

$$\frac{1}{C_{kt}} \sum_{\ell\in\mathcal{L}} \sum_{j\in\mathcal{J}_\ell} p_{jk}^\ell x_{jkt}^\ell \le n_t + \zeta \le 1 \qquad k \in \mathcal{K}, t \in \mathcal{T}, \tag{9g}$$

$$\sum_{t\in\mathcal{T}: t\le q} z_{jkt}^\ell \ge s_{jkq}^\ell, \qquad k \in \mathcal{N}_j^\ell, j \in \mathcal{J}_\ell, \ell \in \mathcal{L}, q \in \mathcal{T}, \tag{9h}$$

$$\sum_{\ell\in\mathcal{L}} \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{N}_j^\ell} z_{jkt}^\ell \le \gamma, \qquad t \in \mathcal{T}, \tag{9i}$$

$$r_{jt}^\ell \le \bar{r}_{jt}^\ell, \qquad j \in \widetilde{\mathcal{J}}_\ell, \ell \in \mathcal{L}, t \in \mathcal{T}, \tag{9j}$$

$$m_t^\ell \le d \cdot a_t^\ell, \qquad \ell \in \mathcal{L}, t \in \mathcal{T}, \tag{9k}$$

$$x_{jkt}^\ell \in \mathbb{Z}_+, \qquad k \in \mathcal{K}_j^\ell, j \in \mathcal{J}_\ell, \ell \in \mathcal{L}, t \in \mathcal{T}, \tag{9l}$$

$$s_{jkt}^\ell \in \mathbb{B}, \qquad k \in \mathcal{K}_j^\ell, j \in \mathcal{J}_\ell, \ell \in \mathcal{L}, t \in \mathcal{T}, \tag{9m}$$

$$z_{jkt}^\ell \in \mathbb{B}, \qquad k \in \mathcal{N}_j^\ell, j \in \mathcal{J}_\ell, \ell \in \mathcal{L}, t \in \mathcal{T}, \tag{9n}$$

$$m_t^\ell \in \mathbb{Z}_+, \qquad \ell \in \mathcal{L}, t \in \mathcal{T}, \tag{9o}$$

$$n_t \ge 0, \qquad t \in \mathcal{T}, \tag{9p}$$

$$r_{jt}^\ell \in \mathbb{Z}_+, \qquad j \in \widetilde{\mathcal{J}}_\ell, \ell \in \mathcal{L}, t \in \mathcal{T} \cup \{0\}. \tag{9q}$$

Data availability

Data will be made available on request.

References

Alves, M.J., Clímaco, J., 2007. A review of interactive methods for multiobjective integer and mixed-integer programming. European J. Oper. Res. 180 (1), 99–115.

Boland, N., Charkhgard, H., Savelsbergh, M., 2015a. A criterion space search algorithm for biobjective integer programming: The balanced box method. INFORMS J. Comput. 27 (4), 735–754.

Boland, N., Charkhgard, H., Savelsbergh, M., 2015b. The L-shape search method for triobjective integer programming. Math. Program. Comput. 8 (2), 217–251.

Boland, N., Charkhgard, H., Savelsbergh, M., 2017. The Quadrant Shrinking Method: A simple and efficient algorithm for solving tri-objective integer programs. European J. Oper. Res. 260 (3), 873–885.

Brunsch, T., Goyal, N., Rademacher, L., Röglin, H., 2014. Lower bounds for the average and smoothed number of Pareto- Optima. Theory Comput. 10 (10), 237–256.

Ceyhan, G., Köksalan, M., Lokman, B., 2019. Finding a representative nondominated set for multi-objective mixed integer programs. European J. Oper. Res. 272 (1), 61–77.

Dächert, K., Gorski, J., Klamroth, K., 2012. An augmented weighted Tchebycheff method with adaptively chosen parameters for discrete bicriteria optimization problems. Comput. Oper. Res. 39 (12), 2929–2943.

Dächert, K., Klamroth, K., 2014. A linear bound on the number of scalarizations needed to solve discrete tricriteria optimization problems. J. Global Optim. 61 (4), 643–676.

Doğan, I., Lokman, B., Köksalan, M., 2022. Representing the nondominated set in multi-objective mixed-integer programs. European J. Oper. Res. 296 (3), 804–818.

Dolan, E.D., Moré, J.J., 2002. Benchmarking optimization software with performance profiles. Math. Program. 91 (2), 201–213.

Ehrgott, M., Gandibleux, X., 2004. Approximative solution methods for multiobjective combinatorial optimization. Top 12 (1), 1–63.

Fotedar, S., Strömberg, A.-B., Almgren, T., 2023a. Bi-objective optimization of the tactical allocation of job types to machines: mathematical modeling, theoretical analysis, and numerical tests. Int. Trans. Oper. Res. 30 (6), 3479–3507.

Fotedar, S., Strömberg, A.-B., Almgren, T., Cedergren, S., 2023b. A criterion space decomposition approach to generalized tri-objective tactical resource allocation. Comput. Manag. Sci. 20 (1).

Kirlik, G., Sayın, S., 2014. A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. European J. Oper. Res. 232 (3), 479–488.

Klamroth, K., Lacour, R., Vanderpooten, D., 2015. On the representation of the search region in multi-objective optimization. European J. Oper. Res. 245 (3), 767–778.

Lokman, B., Köksalan, M., 2012. Finding all nondominated points of multi-objective integer programs. J. Global Optim. 57 (2), 347–365.

Masin, M., Bukchin, Y., 2008. Diversity maximization approach for multiobjective optimization. Oper. Res. 56 (2), 411–424.

Mavrotas, G., 2009. Effective implementation of the ϵ-constraint method in Multi-Objective Mathematical Programming problems. Appl. Math. Comput. 213 (2), 455–465.

Mavrotas, G., Florios, K., 2013. An improved version of the augmented ϵ-constraint method (AUGMECON2) for finding the exact Pareto set in multi-objective integer programming problems. Appl. Math. Comput. 219 (18), 9652–9669.

Mesquita-Cunha, M., Figueira, J.R., Barbosa-Póvoa, A.P., 2023. New ϵ-constraint methods for multi-objective integer linear programming: A Pareto front representation approach. European J. Oper. Res. 306 (1), 286–307.

Przybylski, A., Gandibleux, X., Ehrgott, M., 2008. Two phase algorithms for the bi-objective assignment problem. European J. Oper. Res. 185 (2), 509–533.

Sayın, S., 2000. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. In: Hybrid Metaheuristics. Math. Program. 87 (3), 543–560.

Sayin, S., 2003. A procedure to find discrete representations of the efficient set with specified coverage errors. Oper. Res. 51 (3), 427–436.

Shao, L., Ehrgott, M., 2016. Discrete representation of non-dominated sets in multi-objective linear programming. European J. Oper. Res. 255 (3), 687–698.

Stidsen, T., Andersen, K.A., Dammann, B., 2014. A branch and bound algorithm for a class of biobjective mixed integer programs. Manage. Sci. 60 (4), 1009–1032.

Sylva, J., Crema, A., 2007. A method for finding well-dispersed subsets of non-dominated vectors for multiple objective mixed integer linear programs. European J. Oper. Res. 180 (3), 1011–1027.

Teghem, J., Tuyttens, D., Ulungu, E., 2000. An interactive heuristic method for multi-objective combinatorial optimization. Comput. Oper. Res. 27 (7–8), 621–634.

Visée, M., Teghem, J., Pirlot, M., Ulungu, E.L., 1998. Two-phases method and branch and bound procedures to solve the Bi–objective Knapsack Problem. J. Global Optim. 12 (2), 139–155.

Zhang, W., Reimann, M., 2014. A simple augmented ϵ-constraint method for multi-objective mathematical integer programming problems. European J. Oper. Res. 234 (1), 15–24.