

Lammel, Ina; Küfer, Karl-Heinz; Süss, Philipp

Article — Published Version

An approximation algorithm for multiobjective mixed-integer convex optimization

Mathematical Methods of Operations Research

Suggested Citation: Lammel, Ina; Küfer, Karl-Heinz; Süss, Philipp (2024) : An approximation algorithm for multiobjective mixed-integer convex optimization, Mathematical Methods of Operations Research, ISSN 1432-5217, Springer Berlin Heidelberg, Berlin/Heidelberg, Vol. 100, Iss. 1, pp. 321-350,
<https://doi.org/10.1007/s00186-024-00870-3>

This Version is available at:

<https://hdl.handle.net/10419/314975>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<http://creativecommons.org/licenses/by/4.0/>



An approximation algorithm for multiobjective mixed-integer convex optimization

Ina Lammel¹ · Karl-Heinz Küfer¹ · Philipp Süß¹

Received: 11 April 2023 / Revised: 19 January 2024 / Accepted: 7 July 2024 /

Published online: 29 July 2024

© The Author(s) 2024

Abstract

In this article we introduce an algorithm that approximates the nondominated sets of multiobjective mixed-integer convex optimization problems. The algorithm constructs an inner and outer approximation of the front exploiting the convexity of the patches for problems with an arbitrary number of criteria. In the algorithm, the problem is decomposed into patches, which are multiobjective convex problems, by fixing the integer assignments. The patch problems are solved using (simplicial) Sandwiching. We identify parts of patches that are dominated by other patches and ensure that these patch parts are not refined further. We prove that the algorithm converges and show a bound on the reduction of the approximation error in the course of the algorithm. We illustrate the behaviour of our algorithm using some numerical examples and compare its performance to an algorithm from literature.

Keywords Multiobjective optimization · Mixed-integer optimization · Approximation algorithm · Convex optimization

1 Introduction

Multiobjective optimization deals with problems in which several objective functions are optimised simultaneously. If such a problem depends on both continuous and integer variables, we call it a multiobjective mixed-integer optimization problem. In this paper, we present a novel algorithm for the approximation of the nondominated

Karl-Heinz Küfer and Philipp Süß have contributed equally to this work.

✉ Ina Lammel
ina.lammel@itwm.fraunhofer.de

Karl-Heinz Küfer
karl-heinz.kuefer@itwm.fraunhofer.de

Philipp Süß
philipp.suess@itwm.fraunhofer.de

¹ Department of Optimization, Fraunhofer Institute for Industrial Mathematics (ITWM), Fraunhofer Platz 1, 67663 Kaiserslautern, Germany

sets of multiobjective mixed-integer convex optimization problems of the form

$$\begin{aligned} \min \quad & f(x) = (f_1(x), \dots, f_d(x)) \\ \text{s.t.} \quad & x = (x_C, x_I) \\ & x_I \in \mathcal{X}_I, \quad x_C \in \mathcal{X}_C(x_I) \end{aligned} \quad (1)$$

where $f(x)$ denotes the vector of d convex objective functions $f_i : \mathbb{R}^{k+m} \rightarrow \mathbb{R}$ and the decision vectors x have continuous and integer components. We assume that the set of feasible integer assignments $\mathcal{X}_I \subset \mathbb{Z}^m$ is finite and known exactly. For every feasible integer assignment x_I , the set of corresponding feasible continuous variables $\mathcal{X}_C(x_I) \subset \mathbb{R}^k$ is convex. We note that the algorithm proposed in this article can also be applied to problems of the form (1) with linear objective functions and feasible sets defined by linear constraints.

We can decompose problem (1) into purely continuous multiobjective convex problems by fixing the integer assignments. Let q be the number of feasible integer assignments and let $\bar{x}_I^p \in \mathcal{X}_I$ be the p 'th feasible integer assignment. We define the p 'th *patch problem*

$$\begin{aligned} \min \quad & f(x) := f(x, \bar{x}_I^p) \\ \text{s.t.} \quad & x \in \mathcal{X}_C(\bar{x}_I^p). \end{aligned}$$

The nondominated set of a patch problem is called a *patch*. This term is also used in Diessel (2022), Eichfelder and Warnow (2023) and Serna Hernandez (2011), although the definitions differ slightly.

We additionally assume that the extreme compromise solutions, i.e. the solutions of the lexicographic optimization problems, are finite (see Section 5.1 of Ulus (2018)). This requirement is necessary for the construction of the inner and outer approximations.

In our algorithm, we will approximate the nondominated set of (1) by computing nondominated points of patches. Since the integer feasible set \mathcal{X}_I is finite and known, we can enumerate the patch problems. While this approach is not efficient for large numbers of patches, we argue that for small numbers of patches, enumerating them is more efficient than to apply algorithms that search the integer feasible set. Additionally, as discussed in Cabrera-Guerrero et al. (2021), problem-specific knowledge may allow to exclude a large number of patches from the start.

Many strategies for the approximation of nondominated sets of continuous linear, convex and nonconvex multiobjective optimization problems have been developed in the last decades, see the survey Ruzika and Wiecek (2005). A survey of algorithms for multiobjective linear optimization problems with integer variables has recently been presented in Halffmann et al. (2022).

In recent years, the first algorithms for solving multiobjective mixed-integer non-linear problems have been introduced. Burachik et al. demonstrate in Burachik et al. (2022) that their algorithms developed in Burachik et al. (2017) for the approximation of nondominated sets of problems with disconnected feasible sets can also be applied to the approximation of nondominated sets of multiobjective mixed-integer

problems. They compute points of the nondominated set using a specialized scalarization method. The parameters of the scalarization problem are determined using a fixed grid. The algorithms presented by Ceyhan et al. (2019) compute a representation of the nondominated set of a multiobjective mixed-integer convex problem and aim at reducing the coverage gap of the representation set. This approach differs from our approximation approach where the convex hull of the computed points is used as a representative set.

There is a group of algorithms using a branch-and-bound approach to solve multiobjective mixed-integer convex problems. A branch-and-bound approach in the decision space where bounds are computed using relaxations was introduced by Serna Hernandez (2011). The algorithm presented by Cacchiani and D'Ambrosio (2017) constructs an initial approximation by solving single-objective mixed-integer convex problems in epsilon constraint problems. Leaf nodes, i.e. patches, are solved by varying the weights of weighted sum problems. The ideal point of nodes is used as a lower bound. Two other algorithms based on a branch-and-bound approach in the decision space are De Santis et al. (2020) and Eichfelder et al. (2022) which use stronger bounds than Cacchiani and D'Ambrosio (2017). While De Santis et al. (2020) operates only in the decision space and obtains bounds from piecewise linear approximations, Eichfelder et al. (2022) combines information from the decision space and the image space: branching steps are performed in the decision space while the node selection and specialized cuttings are performed in the image space. These algorithms have the disadvantage that they are not well suited for problems with many decision variables.

The HyPaD algorithm introduced in Eichfelder and Warnow (2023) constructs a lower bound of the front of the mixed-integer problem and lower bounds of the patches. Using an upper bound set, boxes are constructed that enclose the nondominated set and that are used to compute new nondominated points. The algorithm does not rely on a given set of feasible integer assignments but constructs them in the course of the algorithm. This approach is especially useful if the number of feasible integer assignments is large.

A method for the solution of biobjective mixed-integer convex problems introduced by Diessel (2022) iteratively constructs line segments, in order to approximate the nondominated set. The algorithm can not be easily extended to solve problems with more than two criteria.

Another algorithm for biobjective patch problems with convex patches was presented in Cabrera-Guerrero et al. (2021). Inner and outer approximations of the patches are constructed using tangents and the convex hull of the nondominated points. Then, an inner and outer approximation of the nondominated set of the multiobjective mixed-integer convex problem can be defined as the nondominated set of the union of the patch approximations. A fixed grid defines values for epsilon constraint scalarization problems for every patch. The selection of the next epsilon value does not use information on the distance between the inner and outer approximation. When an epsilon value has been fixed, the patch for the next nondominated point is determined by evaluating the outer approximation for every patch and choosing the smallest value.

In this article, we will present an algorithm that is designed for multiobjective mixed-integer convex optimization problems of the form (1).

To the best of our knowledge, it is the first algorithm that exploits the convexity of the patches to construct an inner and outer approximation of the nondominated set of multiobjective mixed-integer convex problems with any number of criteria. We will approximate the nondominated set of (1) by iteratively computing nondominated points of its patch problems. The algorithm does not use the structure of the original mixed-integer problem. Therefore, the patch problems can also arise from different optimization problems each.

While approximating the nondominated set of a multiobjective mixed-integer convex problem using boxes as in Eichfelder and Warnow (2023) is convenient, it is not the best-possible approximation of convex parts of the nondominated set. We define the inner and outer approximation of the patches and the nondominated set of (1) exploiting the convexity of the patches as in Cabrera-Guerrero et al. (2021). We use a simplicial Sandwiching approximation approach for each patch but only add new points to a patch when necessary. We detect parts of patches that are dominated and make sure to avoid generating more points in these areas. We also add an interactive step. After an initial coarse approximation of the nondominated set has been computed, the decision maker can explore the approximated front in a patch navigation tool (e.g. using the approach presented in Collicott et al. (2021)) and mark parts of the objective space that they are not interested in. These parts of the nondominated set will then not be refined in following steps of the algorithm.

2 Constructing a Sandwich approximation of the nondominated set

When approximating the nondominated set of a multiobjective mixed-integer convex optimization problem, it suffices to approximate the patches. Every nondominated point of (1) is a nondominated point of one of the patch problems. The nondominated set of (1) is just the nondominated set of all patches (Cabrera-Guerrero et al. (2021), Proposition 1).

2.1 Lower and upper bounds on the patches

We construct inner and outer approximations of the convex patches.

We compute a new point of patch p by solving the weighted sum scalarization problem with weights $\lambda = (\lambda_1, \dots, \lambda_d)^T \in \mathbb{R}_{\geq}^d$.

$$\begin{aligned} \min \quad & \sum_{j=1}^d (\lambda_j)^T f_j(x, \bar{x}_I^p) \\ \text{s.t. } & x \in \mathcal{X}_C(\bar{x}_I^p). \end{aligned} \quad (2)$$

Every solution of the weighted sum scalarization problem is a weakly efficient point. Since the patch problems are convex, every efficient point can be obtained by solving weighted sum scalarization problems Ehrgott (2005).

Let $\{z_p^1, \dots, z_p^n\}$ be the computed points of patch p . The inner approximation of the patch is given by the convex hull of the nondominated points, extended by the standard domination cone

$$I_p^n := \text{conv} \{z_p^1, \dots, z_p^n\} + \mathbb{R}_{\geq}^d.$$

When a nondominated point z_p^{n+1} is added to the patch, the inner approximation can be updated by $I_p^{n+1} = \text{conv} \{z_p^{n+1}, I_p^n\}$.

The outer approximation of the patch is defined as the intersection of the half-spaces containing the nondominated set that support the patch points. Such a supporting hyperplane can be given by the nondominated point and using the negative weight of the weighted sum scalarization problem that was used to compute it as its outer normal. Let $H(w^i, b^i) := \{z : (w^i)^T z = b^i\}$ be the supporting hyperplane of the patch in z_p^i . Then the half-space $HS(w^i, b^i) := \{z : (w^i)^T z \leq b^i\}$ contains the patch due to convexity. The outer approximation is defined as

$$O_p^n := \cap \{HS(w^i, b^i), i = 1, \dots, n\}.$$

When a new nondominated point z_p^{n+1} has been computed, the outer approximation can be updated by $O_p^{n+1} := HS(w^{n+1}, b^{n+1}) \cap O_p^n$.

Since the true nondominated set of the patch problem lies between the inner and the outer approximation, this kind of approximation is known as a Sandwiching approximation. It is used in the well-known Sandwiching algorithm for convex nondominated sets (e.g. Bokrantz and Forsgren (2013), Dörfler et al. (2021), Nowak et al. (2021)) and it is an extension of the patch approximation used in the biobjective case by Cabrera-Guerrero et al. (2021).

2.2 Lower and upper bounds on the nondominated set of the MOMIC problem

The nondominated set of the MOMIC problem (1) is the nondominated set of the union of the patches. Let the nondominated set of a set $M \subset \mathbb{R}^d$ be defined as $(M)_N := \{m \in M : \nexists n \in M, n \text{ dominates } m\}$. Using the inner and outer approximations of the q patches I_1, \dots, I_q and O_1, \dots, O_q we define the inner and outer approximation of the front as the nondominated set of the union of the inner and outer approximations of the patches.

$$\begin{aligned} \mathbf{I} &:= \left(I^{\text{union}}\right)_N = (I_1 \cup \dots \cup I_q)_N \\ \mathbf{O} &:= \left(O^{\text{union}}\right)_N = (O_1 \cup \dots \cup O_q)_N. \end{aligned}$$

We illustrate this definition in Fig. 1. These sets will never need to be computed in practice. They can be seen as a Sandwich approximation of the nondominated set of (1). The same definition is used in Cabrera-Guerrero et al. (2021).

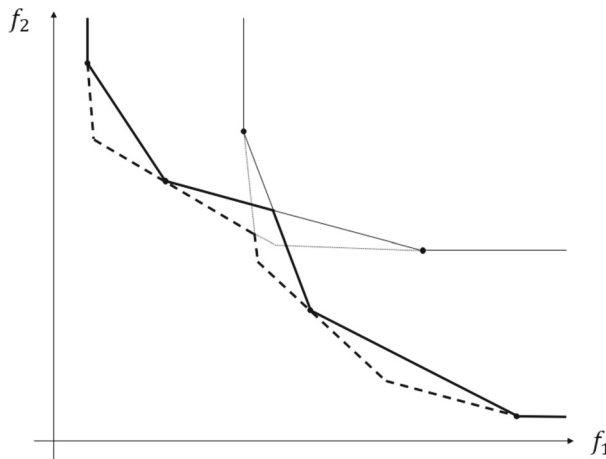


Fig. 1 A MOMIC problem consisting of two patches. The patches are approximated by three points each. The boundaries of the inner and outer approximations of the patches are marked with thin lines. The boundary of the inner approximation of the MOMIC problem is marked with a bold line, the boundary of the outer MOMIC approximation with a dashed bold line

2.3 Measuring the approximation quality

In Sandwiching algorithm variants, a variety of quality indicators are used to assess the approximation quality, e.g. the epsilon indicator in Bokrantz and Forsgren (2013) and Rennen et al. (2011), the Hausdorff metric in Löhne et al. (2014) and Dörfler et al. (2021) or the polyhedral gauge in Klamroth et al. (2003) and Serna Hernandez (2011). In this work we will present all results for the epsilon indicator (Definition 1 and 2 of Diessel (2021)). Similar results for the other quality metrics can be derived, but are not shown here.

Definition 1 The *epsilon-indicator* $\delta^\epsilon(I, O)$ of a Sandwiching approximation I, O is the smallest number $\epsilon \geq 0$ such that for every $o \in O$ there exists a point in the inner approximation $\iota \in I$ such that $\iota \leq o + \epsilon \cdot e$ where $e = (1, \dots, 1) \in \mathbb{R}^d$.

The approximation quality of the approximation of the nondominated set of (1) can be determined by obtaining the patch approximation qualities.

Lemma 1 Let problem (1) have q patches and let $\delta^\epsilon(I_p, O_p) \leq \epsilon$ for every patch p , $p \in \{1, \dots, q\}$ with inner and outer approximation I_p, O_p .

Then, the nondominated set of (1) is also approximated with quality ϵ : $\delta^\epsilon(\mathbf{I}, \mathbf{O}) \leq \epsilon$.

Proof We use the maximum metric $\|\iota, o\|_\infty := \max \{|o_j - \iota_j|, j \in \{1, \dots, d\}\}$ for $\iota, o \in \mathbb{R}^d$ and use the notation $\|o, \mathbf{I}\|_\infty := \min \{\|\iota - o\|_\infty, \iota \in \mathbf{I}\}$ for $\mathbf{I} \subset \mathbb{R}^d$. We first show that

$$\delta^\epsilon(\mathbf{I}, \mathbf{O}) = \max \{ \min \{ \|\iota, o\|_\infty, \iota \in \mathbf{I} \}, o \in \mathbf{O} \},$$

using a similar argument as in Lemma 1 of Diessel (2021).

If $\delta^\epsilon(\mathbf{I}, \mathbf{O}) = \epsilon$, then for every $o \in \mathbf{O}$ there exists an $\iota \in \mathbf{I}$ such that $\iota \leq o + \epsilon e$. Due to the definition of the inner approximation, it holds $o + \epsilon e \in \mathbf{I}$. Since $\|o + \epsilon e - o\|_\infty = \epsilon$, it holds $\max \{\min \{\|\iota, o\|_\infty, \iota \in \mathbf{I}\}, o \in \mathbf{O}\} \leq \epsilon$.

If, on the other hand, it holds $\max \{\min \{\|\iota, o\|_\infty, \iota \in \mathbf{I}\}, o \in \mathbf{O}\} \leq \epsilon$, then for every $o \in \mathbf{O}$ there exists a $\iota \in \mathbf{I}$ fulfilling $\|\iota, o\|_\infty \leq \epsilon$ since \mathbf{I} is closed as a finite union of closed sets. Therefore, we can choose an $\iota \in \mathbf{I}$ such that $\iota \leq o + \epsilon e$. Thus, $\delta^\epsilon(\mathbf{I}, \mathbf{O}) \leq \epsilon$.

Let $o \in O_l, l \in \{1, \dots, q\}, o \in \mathbf{O}$ and let $\bar{\iota} \in I_l$ such that

$$\|o, I_l\|_\infty := \min \{\|\iota - o\|_\infty, \iota \in I_l\} = \|o, \bar{\iota}\|_\infty.$$

Then

$$\begin{aligned} \|o, I^{\text{union}}\|_\infty &= \|o, I_1 \cup \dots \cup I_q\|_\infty \\ &\leq \|o, I_l\|_\infty = \|o, \bar{\iota}\|_\infty \leq \epsilon. \end{aligned}$$

This implies that either $\bar{\iota} \in \mathbf{I}$ and therefore $\|o, \mathbf{I}\|_\infty = \|o, I^{\text{union}}\|_\infty \leq \epsilon$ or $\bar{\iota} \in I^l \subset I^{\text{union}}$ but $\bar{\iota} \notin \mathbf{I}$ since $\bar{\iota}$ is dominated by some $\bar{z} \in \mathbf{I}$. Then, since $\bar{z}_j \leq \bar{\iota}_j \forall j = 1, \dots, d$, it holds $|\bar{z}_j - o_j| \leq |\bar{\iota}_j - o_j| \forall j = 1, \dots, d$ and therefore $\|\bar{z} - o\|_\infty \leq \|\bar{\iota} - o\|_\infty$. Thus,

$$\|o, \mathbf{I}\|_\infty \leq \|\bar{z} - o\|_\infty \leq \|\bar{\iota} - o\|_\infty \leq \epsilon.$$

The same reasoning can be applied to $\|\iota, \mathbf{O}\|_\infty, \iota \in \mathbf{I}$. □

In the algorithm, we will use the maximum of the patch qualities as an estimate of the approximation quality of the nondominated set of (1) since it is much easier to compute.

The epsilon indicator value of a Sandwiching approximation can be computed by solving a small linear program for every vertex of the outer approximation. The vertices of the outer approximation can be determined using a convex hull algorithm like qhull Barber et al. (1996) and the dual polyhedron, as described e.g. in Section 4.4 of Bokrantz and Forsgren (2013).

Although the number of vertices of the outer approximation of a patch can be very high, the computation of the epsilon indicator is still fast because solving most of the linear programs can be avoided while still computing the epsilon indicator approximation quality exactly as shown in Lammel et al. (2023). However, computing the epsilon indicator for every patch becomes inefficient for large numbers of patches.

Due to Lemma 1, the epsilon indicator values of the patches can be used to obtain a bound on the error of the approximation of the nondominated set of (1).

3 A new patch approximation algorithm

The general idea of the algorithm is to use a simplicial Sandwiching approximation approach for every patch to increase the approximation quality of the nondominated set of (1). We detect parts of patches that are dominated and avoid generating more points in these areas.

Often, only a small part of the nondominated set is of interest to the decision maker. Therefore, we propose to include an interactive step. After an initial approximation of the patches has been computed, the decision maker can explore the approximated front in a patch navigation tool (e.g. using the approach presented in Collicott et al. (2021)) and mark parts of the objective space that they are not interested in. We make sure that these parts of the nondominated set will not be refined in following steps of the algorithm. When the approximation process is completed, the approximation of the nondominated set can be presented to the decision maker again for the final decision-making process.

In the following sections, we will first develop the most important aspects of the algorithm. We summarize the simplicial Sandwiching method that we use to approximate the patch problems. We introduce methods that identify dominated parts of patches and methods that ensure that dominated areas are excluded from further refinements. Finally, we use these elements to present the patch approximation algorithm scheme.

3.1 Approximating the convex patches: Sandwiching

In the patch approximation algorithm, we use the well-known Sandwiching algorithm to add new nondominated points to patches so that the approximation quality of the nondominated set of (1) is improved. There exist several variants of the (simplicial) Sandwiching algorithm which are widely used, e.g. in Dörfler et al. (2021), Klamroth et al. (2003), Löhne et al. (2014), Rennen et al. (2011), Ehrgott et al. (2011), Serna Hernandez (2011) and have been applied to intensity-modulated radiotherapy Craft et al. (2006), Rennen et al. (2011), Bokrantz and Forsgren (2013), product development Süss et al. (2022), chemical process design Bortz et al. (2014) and industrial processes Nowak et al. (2021).

The general idea of the algorithm is as follows. After an initial approximation has been computed, the inner and outer Sandwiching approximation is constructed as presented in Sect. 2.1: the inner approximation is the convex hull of the nondominated points, extended by the domination cone. The outer approximation is given by the cut of the supporting half-spaces in the nondominated points. Then, the approximation quality, e.g. the epsilon-indicator value (cf. Sect. 2.3) is computed. To improve the approximation, a new nondominated point is computed so that its tangential hyperplane is parallel to the inner approximation facet where the worst quality value was attained.

While the algorithm introduced in Cabrera-Guerrero et al. (2021) for biobjective problems also computes an initial approximation for every patch and constructs the same inner and outer approximations as in our algorithm, it is not used to determine the location of the next nondominated point. Instead, the distance between unsolved scalarization parameters within a fixed grid of scalarization parameters is used.

We do not want to add points to a patch approximation using Sandwiching if this part of the patch is dominated by a different patch. In the next section we will discuss how dominated parts of patches can be identified.

3.2 Determining ϵ -dominated parts of a nondominated set

We want to identify dominated patch parts to avoid refining parts of patches that do not contribute to the nondominated set of (1). However, we relax the nondomination requirement a bit: in our algorithm it is enough if a point z is ϵ_{dom} -nondominated for some $\epsilon_{\text{dom}} \geq 0$ to be kept for refinement, i.e. if $z - \epsilon_{\text{dom}} := (z^1 - \epsilon_{\text{dom}}, \dots, z^d - \epsilon_{\text{dom}})$ is nondominated.

If $\epsilon_{\text{dom}} = 0$ is chosen, then ϵ_{dom} -nondominance is just classical nondominance. If a value $\epsilon_{\text{dom}} > 0$ is chosen, then parts of patches are added to the approximation which do not contribute to the overall nondominated set but which are slightly dominated by it. This is useful in practical applications to present more solutions than just the nondominated set to the user as long as these solutions are marked as such. This allows the decision maker to incorporate goals in the decision-making process that are not objective functions.

After computing an initial approximation for every patch, we apply a two-phase approach.

First, we try to identify as many fully ϵ_{dom} -dominated patches as possible by looking only at the patch points. We successively obtain patches whose ideal points are nondominated with respect to the ideal points of the other patches. Then we remove all patches whose ideal point is ϵ_{dom} -dominated by a point of such a patch.

Then, we move on to another procedure that is guaranteed to identify any ϵ_{dom} -dominated patch and that can detect ϵ_{dom} -dominated parts of patches. We propose to check for every new patch point z whether there is a point in the inner approximation of a different patch that ϵ_{dom} -dominates z . To do this, we solve a small linear program for those other patches that are not marked as fully ϵ_{dom} -dominated until a dominating patch is found.

Although this approach is sufficient in our applications where the computational effort of computing a new nondominated point clearly outweighs the effort of solving the linear programs, the proposed method does not scale well with the number of patches. This effect is mitigated (at the cost of having calculated at least d points for each of these dominated patches in the first phase) when a significant number of the patches is already discarded in the first phase. If problem (1) can be formulated as a multiobjective mixed-integer convex program where every patch has the same feasible set, for large numbers of patches the algorithm presented in Eichfelder and Warnow (2023) may be the better choice since it avoids computing a small initial approximation for every integer assignment. In Sect. 5 we will compare the behaviour of the two algorithms for some exemplary cases.

We propose two variants of determining whether a patch point is ϵ_{dom} -dominated by the inner approximation of a different patch. Let the matrix $Z \in \mathbb{R}^{d \times n}$ contain the nondominated points of a patch as column vectors. The term $Z\xi + \mu$, $\sum \xi = 1$, $\xi \geq 0$, $\mu \geq 0$ then represents the inner approximation of the patch, where $\xi \in \mathbb{R}^n$ describes the convex combination of the patch points and $\mu \in \mathbb{R}_{\geq 0}^d$ adds the standard domination cone \mathbb{R}_{\geq}^d .

To check whether the patch ϵ_{dom} -dominates the point z , the problem

$$\begin{aligned} & \min -e^T \cdot \delta \\ & \text{s.t. } Z\xi + \delta = z - \epsilon_{\text{dom}} \\ & \quad e^T \xi = 1 \\ & \quad \delta, \xi \geq 0 \end{aligned} \quad (3)$$

can be solved. The vector e is the vector of ones. If (3) has an optimal solution (δ, ξ) , with some $\delta^i > 0$, $i \in \{1, \dots, d\}$, a dominating point is given by $z^{\text{dom}} = Z\xi$.

If the computational effort for generating a nondominated point is very high, it may be worthwhile to solve a linear program for each coordinate direction in the objective space. This way, several dominating points are found and even larger ϵ_{dom} -dominated areas can be identified. To check whether a point z is ϵ_{dom} -dominated by some patch in coordinate direction j , we solve the linear program

$$\begin{aligned} & \min -\delta \\ & \text{s.t. } (Z\xi)^j + \mu^j + \delta = z^j - \epsilon_{\text{dom}} \\ & \quad (Z\xi)^i + \mu^i = z^i - \epsilon_{\text{dom}} \quad \forall i \neq j \\ & \quad e^T \xi = 1 \\ & \quad \delta, \xi, \mu \geq 0 \end{aligned} \quad (4)$$

where Z, ξ, μ, e are defined as above. If (4) has an optimal solution (δ, ξ, μ) , $\delta > 0$, a dominating point is then given by $z^{\text{dom}} = Z\xi + \mu$.

We first establish that if a patch dominates z , then a dominating point is found and that z^{dom} actually dominates z .

- Lemma 2** 1. Let $z' \in I_p^n$ for some patch p , z' ϵ_{dom} -dominating z . Then the program (3) has a solution (δ, ξ) where $z^{\text{dom}} = Z\xi$ ϵ_{dom} -dominates z .
2. Let $z' \in I_p^n$ for some patch p , z' ϵ_{dom} -dominating z with $(z')^j < z^j$. Then (4) has a solution (δ, ξ, μ) and $z^{\text{dom}} = Z\xi + \mu$ ϵ_{dom} -dominates z .

Proof 1. Since $z' \in I_p^n$, there exist $\xi \in \mathbb{R}^n$, $\sum \xi = 1$ and $\mu \in \mathbb{R}^d$, $\xi, \mu \geq 0$ with $z' = Z\xi + \mu$. Since z' ϵ_{dom} -dominates z , it holds $(z')^i = (Z\xi)^i + \mu^i \leq z^i - \epsilon_{\text{dom}}$ and therefore also $(Z\xi)^i \leq z^i - \epsilon_{\text{dom}}$ for all $i \in \{1, \dots, d\}$ and for some $j \in \{1, \dots, d\}$ it holds $(z')^j = (Z\xi)^j + \mu^j \leq z^j - \epsilon_{\text{dom}}$ and therefore also $(Z\xi)^j \leq z^j - \epsilon_{\text{dom}}$. For $\delta = z - \epsilon_{\text{dom}} - (Z\xi)$ it holds $\delta^i \geq 0 \quad \forall i \in \{1, \dots, d\}$ and $\delta^j < 0$. Therefore, (δ, ξ) is a feasible point of (3) and $Z\xi$ ϵ_{dom} -dominates z .

2. Since $z' \in I_p^n$, there exist $\xi \in \mathbb{R}^n$, $\sum \xi = 1$ and $\mu \in \mathbb{R}^d$, $\xi, \mu \geq 0$ with $z' = Z\xi + \mu$. Since z' ϵ_{dom} -dominates z in direction j , $z' = Z\xi + \mu \leq z - \epsilon_{\text{dom}}$ and $(z')^j = (Z\xi)^j + \mu^j < z^j - \epsilon_{\text{dom}}$. We choose $\delta = z^j - \epsilon_{\text{dom}} - (Z\xi)^j - \mu^j$ with $\delta > 0$. For $\bar{\mu}, v \in \mathbb{R}^d$ and $\bar{\mu} = \mu + v$ with $v^j = 0$, $v^i = z^i - \epsilon_{\text{dom}} - (Z\xi)^i - \mu^i$ for $i \in \{1, \dots, d\}$, $i \neq j$, it holds $v \geq 0$. Therefore, $(\delta, \xi, \bar{\mu})$ is feasible for (4). Since $\delta > 0$, $Z\xi + \bar{\mu}$ ϵ_{dom} -dominates z .

□

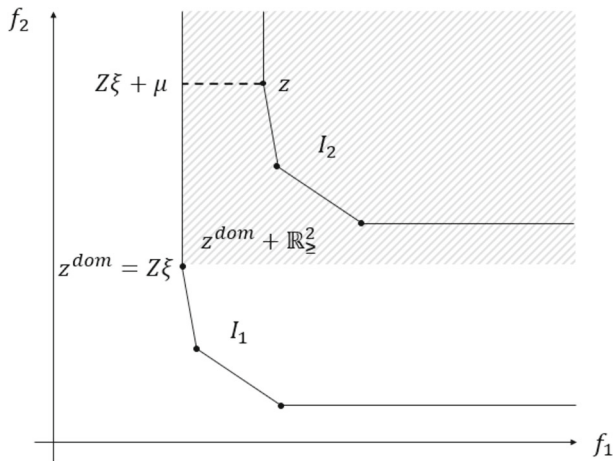


Fig. 2 A MOMIC problem consisting of two patches. We check whether the second patch is ϵ_{dom} -dominated by solving (4) with $\epsilon_{\text{dom}} = 0$ for the patch point z in the first coordinate direction. The solution $Z\xi + \mu$ dominates z , as well as the dominating point $z^{\text{dom}} := Z\xi$. The whole second patch is contained in the cone $z^{\text{dom}} + \mathbb{R}_{\geq}^2$, so that it can be marked as fully dominated

If there is a coordinate i with $\mu_i > 0$, then the point $Z\xi$ also dominates z and $Z\xi + \mathbb{R}_{\geq}^d \supset (Z\xi + \mu) + \mathbb{R}_{\geq}^d$.

Lemma 3 *If the solution (δ, ξ, μ) of (4) fulfils $\delta > 0$ and $\mu_i > 0$ for some $i \in \{1, \dots, d\}$, then $\tilde{z}^{\text{dom}} = Z\xi$ ϵ_{dom} -dominates z .*

Proof Since $\delta > 0$, from the first two equality constraints of (4) we obtain that $Z\xi + \mu$ ϵ_{dom} -dominates z . Since $\mu \geq 0$, it also holds $(Z\xi)^j < z^j - \epsilon_{\text{dom}}$ and $(Z\xi)^i \leq z^i - \epsilon_{\text{dom}}$.

The method is illustrated in Fig. 2. In a run of the patch approximation algorithm, however, patch 2 would have already been marked as fully ϵ_{dom} -dominated in the first phase without solving a linear program.

The task of checking whether the initial approximation of a patch is ϵ_{dom} -dominated is summarized in Algorithm 1.

A common fathoming rule in branch-and-bound based approximation algorithms for mixed-integer multiobjective problems (e.g. used in Cacchiani and D'Ambrosio (2017), Adelgren and Gupte (2022)) is to check whether the ideal point of a patch is ϵ_{dom} -dominated by some other patch.

Lemma 4 *If the ideal point z^I of a patch is ϵ_{dom} -dominated by a different patch, the patch will be fully marked as ϵ_{dom} -dominated the first time it is considered by the patch approximation algorithm.*

Proof After an initial approximation of the patches has been computed, the ideal points of the patches are checked for ϵ_{dom} -dominance by other patches by (3) (or (4)). Using Lemma 2, we know that if the ideal point of the patch is ϵ_{dom} -dominated, then some ϵ_{dom} -dominating point z^{dom} will be found using (3) (or (4)). Then, the part of the patch intersecting $z^{\text{dom}} + \mathbb{R}_{\geq}^d$ will be marked as ϵ_{dom} -dominated. Since for every patch point

Algorithm 1 Check the initial approximation of every patch for epsilon-domination**Require:** Initial approximation of every patch,dominance precision ϵ_{dom} ,a method that checks if a point is ϵ_{dom} -dominated by some patch (e.g. 3 or 4)a method that marks a ϵ_{dom} -dominated part of a patch to avoid placing more points there

```

1: for every patch do
2:   Determine the ideal point of the patch
3:   Check whether the ideal point is  $\epsilon_{\text{dom}}$ -dominated by a different patch
4:   if a dominating point is found then
5:     Mark the whole patch as  $\epsilon_{\text{dom}}$ -dominated.
6:   else
7:     for every patch point that is not already known to be  $\epsilon_{\text{dom}}$ -dominated do
8:       Check if the point is  $\epsilon_{\text{dom}}$ -dominated by a different patch
9:       if a dominating point  $z^{\text{dom}}$  is found then
10:        Mark the region of the patch intersecting  $z^{\text{dom}} + \mathbb{R}_{\geq}^d$  as  $\epsilon_{\text{dom}}$ -dominated
11:       end if
12:     end for
13:   end if
14: end for

```

z it holds $z^I \leq z$ and $z^{\text{dom}} \leq z^I$, we have $z \in z^{\text{dom}} + \mathbb{R}_{\geq}^d$. Since every nondominated point lies in the ϵ_{dom} -dominated region, every facet can be marked as ϵ_{dom} -dominated.

3.3 Ensuring that no more nondominated points are calculated in regions that are known to be dominated

Once a region of a nondominated set is known to be ϵ_{dom} -dominated, we want to avoid computing new nondominated points there. We introduce two strategies that make sure that new nondominated points are only placed where they contribute to the nondominated set of (1).

3.3.1 Biobjective problems: objective box constraints

In biobjective problems, parts on the boundary of a patch can be easily excluded from further consideration by adding objective box constraints to the patch optimization problem.

Let a point z^{dom} be given that dominates a patch. We want to avoid computing new points in those parts of the patch that lie in the cone $z^{\text{dom}} + \mathbb{R}_{\geq}^2$.

We look for intersections of $D_1 = \{z^{\text{dom}} + \delta e_1, \delta > 0\}$ and $D_2 = \{z^{\text{dom}} + \delta e_2\}$ where $e_1 = (1, 0)^T$ and $e_2 = (0, 1)^T$ with the patch. If an intersection of D_1 with the patch is found in \bar{z} , a new patch problem is created with additional constraint $f_1(x) \leq \bar{z}^1$. If an intersection of D_2 with the patch is found in \tilde{z} , a new patch problem is created with additional constraint $f_2(x) \leq \tilde{z}^2$. If a new patch problem has been created, i.e. there was at least one intersection point, the original patch problem is removed from (1) and the corresponding patch is removed from the approximation. If there are two intersection points, the number of patches increases by one.

Adding box constraints to a patch is equivalent to setting trade-off bounds as described by Serna et al. (2009): when the decision maker removes those patch points

that satisfy $f_1(x) \leq \bar{z}^1$, they implicitly remove those parts of the nondominated set that have steeper trade-offs with respect to f_1 than the gradient of the patch at \bar{z} . Specifying trade-off bounds can be incorporated into the approximation process by modifying the domination cone. Then, different domination cones need to be used for each patch.

A different strategy is used in Cabrera-Guerrero et al. (2021). Once a scalarization parameter has been selected for the next epsilon-constraint scalarization, the other component of the outer approximation for every patch is evaluated there. The scalarization problem is then solved for the patch with the smallest outer approximation value. The algorithm therefore avoids having to actively remove parts of patches from further approximation.

3.3.2 General number of objective dimensions: placing a perfectly approximated Sandwiching facet

We present a method to avoid computing any more nondominated points in regions that are known to be ϵ_{dom} -dominated that is tailored to the Sandwiching algorithm. If a whole facet of the inner approximation of a patch is ϵ_{dom} -dominated by a different patch, we will ignore it by adding a so-called fake nondominated point. By adding these artificial points to the nondominated set, we generate facets in which the outer and inner approximation coincide. Therefore, the Sandwiching algorithm will never select one of those facets to compute a new nondominated point there.

The inner approximation of a patch consists of the convex hull of its nondominated points with added domination cone. In this section, we want to work with the facets, i.e. the full-dimensional faces, of the inner approximation. Since these are not easy to compute directly, we work with those facets of the convex hull of the patch points which lie on the boundary of the inner approximation. We use a nadir point approximation or a user-defined upper bound on the patch $\tilde{y}^N \in \mathbb{R}^d$ and define

$$F := \{f : f \text{ is facet of } \text{conv}(z_p^1, \dots, z_p^n, \tilde{y}^N) \text{ with } \tilde{y}^N \notin f\}.$$

The set F can easily be computed using a convex hull algorithm, e.g. Barber et al. (1996). We justify the usage of F instead of I_p^n in the following lemma.

Definition 2 A facet is called *dominated* if there exist points in its interior that are dominated by a convex combination of nondominated points.

A facet is called *nondominated* if each of its points is nondominated.

There are no dominated facets in biobjective problems. Let \mathcal{Z} be the set of computed nondominated points.

Lemma 5 1. If the outer normal of a facet $f \in F$ has at least one positive component, the facet is dominated.

2. Every facet of I_p^n that is not a facet of the standard domination cone is also a facet of F .

3. Every facet $f \in F$ is either a facet of I_p^n or it is dominated.

Proof 1. From the definition of F and I_p^n we have $f \subset I_p^n$ for every $f \in F$.

From Lemma 2 of Rennen et al. (2011), we know that the outer normal of a supporting hyperplane of a facet of $\text{conv}(\mathcal{Z} + \mathbb{R}_{\leq}^d)$ always has nonpositive components. Therefore, if a facet $f \in F$ has an outer normal with a component $v_i > 0$ for some $i \in \{1, \dots, d\}$, it holds $f \not\subset \partial I_p^n$, where ∂I_p^n denotes the boundary of I_p^n .

Thus, there exists a point $z \in f$ with $z \notin I_p^n \setminus \partial I_p^n$. Then there is a $\delta \in \mathbb{R}^d$, $\delta \geq 0$ with at least one component $\delta_i > 0$ with $z - \delta \in I_p^n$. Thus, $z \in f \subset I_p^n$ is dominated by $z - \delta \in I_p^n$. As an element of I_p^n , $z - \delta$ can be decomposed into the sum of a convex combination of nondominated points and an element of \mathbb{R}_{\geq}^d which shows that f is a dominated facet.

2. Due to the added domination cone, the boundary of I_p^n consists of weakly nondominated points. Next, we show that every nondominated facet of $\text{conv}(\mathcal{Z})$ is an element of F . Let us assume that a nondominated facet of $\text{conv}(\mathcal{Z})$ is not a facet of $\text{conv}(\mathcal{Z}, \tilde{y}^N)$. Then f would need to be visible from \tilde{y}^N , i.e. the hyperplane extending f would need to separate $\text{conv}(\mathcal{Z})$ from \tilde{y}^N . But this is not possible since f consists of nondominated points of $\text{conv}(\mathcal{Z})$ and \tilde{y}^N is an approximation of the nadir point and therefore is dominated by $\text{conv}(\mathcal{Z})$.
3. Every facet in F is a facet of $\text{conv}(\mathcal{Z})$. If the facet's outer normal fulfills $v \leq 0$, each of its points is nondominated. And every nondominated facet of $\text{conv}(\mathcal{Z})$ is a facet of I_p^n . We already showed in Step 1 that every facet in F whose outer normal has a component $v_i > 0$ is dominated.

□

We can thus use the nondominated facets in F instead of the nondominated facets of I_p^n .

We first discuss the approach for a nondominated facet and then describe how we handle dominated facets.

When a nondominated facet of the nondominated set consisting of d linearly independent nondominated points z^0, \dots, z^{d-1} is known to be fully dominated by a different patch, we calculate the mean point z^m defined as $z_i^m := \frac{1}{d} \sum_{j=0}^{d-1} z_i^j$. As a convex combination of the defining nondominated points, z^m lies on the inner approximation facet. We then take the unit outer normal vector v^n of the facet and add the “fake nondominated point” z^m with v^n as its optimization weight to the approximation of the nondominated set. This idea is illustrated in Fig. 3.

Lemma 6 *Let f be the vector of objectives of a convex bounded multiobjective optimization problem. For the solution \tilde{x} of the weighted sum scalarization (2) with weights $\lambda \geq 0$, the point $f(\tilde{x})$ has the supporting hyperplane $H(\lambda^T, \lambda^T f(\tilde{x})) = \{z \in \mathbb{R}^d, \lambda^T z = \lambda^T f(\tilde{x})\}$.*

Proof In objective space, $\lambda^T f(x) = c$, $c \in \mathbb{R}$ forms a hyperplane with normal λ . When minimizing $\lambda^T f(x)$ in the weighted sum scalarization problem, this hyperplane is shifted by decreasing c until the boundary of the feasible set is reached at \tilde{c} .

Since for the solution \tilde{x} of the weighted sum scalarization the point $f(\tilde{x})$ lies on the hyperplane, it holds $\tilde{c} = \lambda^T f(\tilde{x})$. □

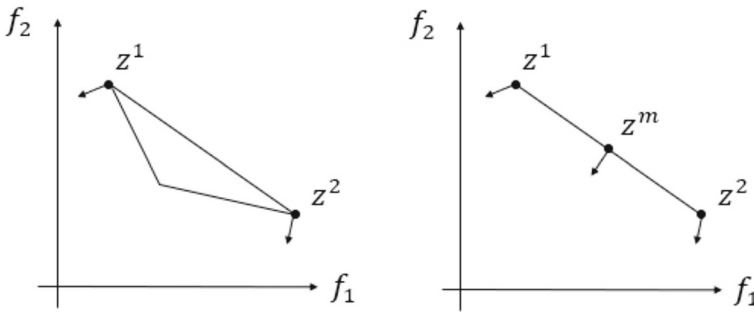


Fig. 3 Let a facet be given by two nondominated points z^1 and z^2 and their normals which form the outer approximation. When adding the mean point z^m with the normal of the facet to the nondominated set, the outer and inner approximations coincide in this facet

Lemma 7 *When adding the mean point z^m of an inner approximation facet, setting the unit outer normal vector v^n of the facet as its normal to the approximation of the nondominated set, the outer and inner approximation of the facet will coincide, i.e. the facet has perfect approximation quality.*

Proof When adding the point z^m with outer normal v^n to the approximation, the inner approximation is updated by $I_p^{n+1} = \text{conv}(I_p^n, z^m)$. Since z^m is a convex combination of patch points, it holds $I_p^{n+1} = I_p^n$.

The outer approximation is updated by the halfspace through z^m with outer normal v^n

$$\{z \in \mathbb{R}^d : (v^n)^T \cdot (z - z^m) \leq 0\} = \{z \in \mathbb{R}^d : (v^n)^T z \leq (v^n)^T z^m\} = HS(v^n, (v^n)^T z^m)$$

Since z^m lies on the facet and the facet normal is v^n , the hyperplane extending the facet is given by

$$\{z \in \mathbb{R}^d : (v^n)^T \cdot (z - z^m) = 0\} = \{z \in \mathbb{R}^d : (v^n)^T z = (v^n)^T z^m\} \subseteq HS(v^n, (v^n)^T z^m).$$

If for a patch point z an ϵ_{dom} -dominating point z^{dom} is determined, then only facets fully lying in $z^{\text{dom}} + \mathbb{R}_{\geq}^d$ can be marked as ϵ_{dom} -dominated using our proposed method. To fully mark the intersection of the patch with $z^{\text{dom}} + \mathbb{R}_{\geq}^d$ as ϵ_{dom} -dominated, additional optimization problems would have to be solved to compute new nondominated points. We do not follow this approach since we cannot know whether such additional points would improve the approximation quality of the nondominated set of (1) substantially.

Handling dominated facets If a region of the patch containing a dominated facet $f \in F$ is identified as ϵ_{dom} -dominated, we cannot add a fake nondominated point to f to mark it as dominated since its normal is not a suitable parameter for the weighted sum scalarization.

In literature, a variety of different approaches has been proposed to avoid dominated facets (Craft et al. (2006), Rennen et al. (2011), Bokrantz and Forsgren (2013)). The

basis of the approach introduced in Rennen et al. (2011) is that while the convex hull of the nondominated points \mathcal{Z} may contain dominated facets, the set $\mathcal{Z} + \mathbb{R}_{\geq}^d$ only consists of nondominated facets. We will now make use of the same approach.

We originally introduced F to avoid having to compute the facets of I_p^n which does not have dominated facets since the domination cone \mathbb{R}_{\geq}^d is added to all of its points.

Instead, we now construct facets of $f + \mathbb{R}_{\geq}^d$ by adding points to f such that the nondominated set of $f + \mathbb{R}_{\geq}^d$ only consists of nondominated facets. Then, for each of these nondominated facets, fake Points can be added to remove them from further refinement.

Definition 3 (Definition 7 of Rennen et al. (2011)) Let an upper bound on the nondominated set \tilde{y}^N , e.g. a nadir point approximation (see Equation 2.15 of Ehrgott (2005)), be given. For a nondominated point $z \in \mathcal{Y}$ we define the points $d_1(z), \dots, d_d(z) \in \mathbb{R}^d$. The j -th point is defined as

$$d_j^i(z) := \begin{cases} z_j & \text{if } j \neq i \\ \tilde{y}_j^N & \text{if } j = i. \end{cases}$$

Adding all d points to every nondominated point $z \in f$ creates all facets of $f + \mathbb{R}_{\geq}^d$, but produces a large amount of additional points. Many of these points are actually not on the boundary of the nondominated set of $f + \mathbb{R}_{\geq}^d$ and therefore do not need to be computed, as shown in the following lemma.

Lemma 8 Let a facet $f \in F$ be defined by the points $p = \{z^1, \dots, z^d\}$. Let the facet be dominated with an outer normal with positive entries in directions $J \subset \{1, \dots, d\}$. To construct all facets of the nondominated set of $f + \mathbb{R}_{\geq}^d$, it suffices to add the points $\{d_j(z) \mid z \in p \mid j \in J\}$ (see Definition 3) to the approximation.

Proof From Lemma 2 of Rennen et al. (2011) we obtain that if d points representing the domination cone are added for every computed nondominated point, then the convex hull of the nondominated points and the points representing the domination cone will not contain any dominated facets.

We want to add points representing the domination cone to obtain facets with fully non-positive normals. When adding points $d_j(z) \forall z \in \mathcal{Z}$, those points represent the j -th extreme ray of the domination cone. Since the ray's normal is zero in direction j , we eliminated the positive normal direction $j \in I$.

After repeating this process for all $i \in I$, all facets of $\text{conv}(\mathcal{Z} \cup \{d_i \mid i \in I\})$ are nondominated. \square

For a dominated facet in d -dimensional objective space with n_{dom} dominated directions, we add $n_{dom}d$ artificial points that represent the domination cone. Since adding a point to a d -dim simplex will split the simplex into d simplices, $(d - 1)$ facets are added to the convex hull for every added representative point.

To construct the outer approximation of the patch, we will need a normal on the points representing the domination cone as well. For a point $d_j(z)$ constructed using nondominated point z with normal w , the normal of $d_j(z)$ has entries $\min\{w_i, 0\}$, $i = 1, \dots, d$.

Algorithm 2 Mark the part of a patch front ϵ_{dom} -dominated by z^{dom} as ϵ_{dom} -dominated using fake nondominated points

```

1: Determine all facets that lie inside of  $z^{\text{dom}} + \mathbb{R}_{\geq}^d$ .
2: for all of these facets do
3:   if the facet is dominated then
4:     Add points representing the domination (see Lemma 8)
5:   end if
6:   Add a fake nondominated point to every facet
7: end for

```

Lemma 9 *The part of a patch marked as ϵ_{dom} -dominated using fake nondominated points is indeed ϵ_{dom} -dominated by some other patch.*

Proof If for some patch point, an ϵ_{dom} -dominating point z^{dom} is found using (3) or (4), then fake nondominated points ensure that in those facets that fully lie inside of $z^{\text{dom}} + \mathbb{R}_{\geq}^d$, no new nondominated points are added. These facets form a subset of the cut of the patch with $z^{\text{dom}} + \mathbb{R}_{\geq}^d$. Every point $z \in z^{\text{dom}} + \mathbb{R}_{\geq}^d$ is ϵ_{dom} -dominated by z^{dom} .

3.4 Patch Navigation

Navigation is a common tool in multi-criteria decision making. It is the interactive procedure of traversing through a set of points in the objective space guided by a decision maker. The ultimate goal of this procedure is to identify the single most preferred nondominated solution (Definition 1.1 of Allmendinger et al. (2017)). Numerous methods have been introduced for convex (e.g. Monz et al. (2008), Eskelinen et al. (2010)) and nonconvex (e.g. Nowak and Küfer (2020), Nowak et al. (2022), Hartikainen et al. (2019)) nondominated sets. An overview is given in Allmendinger et al. (2017). A first method for the comparison of two patches was introduced in Teichert et al. (2011). More recently, navigation approaches for patch problems have been developed, for example Hartikainen et al. (2019) and Collicott et al. (2021).

We will briefly outline the patch navigation approach published in Collicott et al. (2021). For the exploration of the nondominated set, two main navigation features can be used. The decision maker can change the solution in coordinate direction and observe the related trade-offs in real time (selection) and they can set bounds to individual objectives (restriction) and monitor their effect on the obtainable range in other objectives. This approach naturally extends to patches as shown by the authors in Collicott et al. (2021). Additionally, the distance of the solution that was chosen by the decision maker to the closest solutions on other patches is displayed. This helps the decision maker evaluate whether a solution on a different patch could be an alternative to the chosen solution.

In the patch approximation algorithm, we present the initial approximation of the nondominated set of (1) to the decision maker in the patch navigation tool. The decision maker can then explore the trade-offs of the nondominated set and specify parts of the nondominated set that are not of interest to them and therefore do not need a refinement using the restriction process of the navigation. The restriction choices are incorporated



Fig. 4 An initial approximation of a tri-criteria problem with three patches is presented to the decision maker in the patch navigation tool. The decision maker decided to restrict the objective space in the first and third objective (shaded)

into the optimization problem using Algorithm 3. When a patch nondominated point z_p has been removed by the restrictor, a new nondominated point is computed with the same weight to ensure that the initial approximation consists of $d + 1$ points. This is necessary so that the convex hull of the nondominated points can be used when removing ϵ_{dom} -dominated parts of patches.

Algorithm 3 Process restrictors set in navigation step

Require: restrictor positions, all patch points

```

1: for every patch do
2:   for every restrictor do
3:     Add the constraint  $f_i \leq \bar{z}$  to the patch optimization problem
4:     if there are patch points  $z$  with weight  $w$  fulfilling  $z^i > \bar{z}$  then
5:       Remove the patch point from the approximation
6:       Compute a new nondominated point for this patch using weight  $w$ .
7:     end if
8:   end for
9: end for

```

We have implemented the approach of Collicott et al. (2021). In Figs. 4 and 5, the two navigation stages of the patch approximation algorithm are illustrated.

Some algorithms that approximate nondominated sets contain steps to remove dominated points that were computed during the course of the algorithm (e.g. Rennen et al. (2011), Cacchiani and D'Ambrosio (2017)). Since the linear programs used in navigation make sure that we only navigate on the nondominated set of (1), we do not

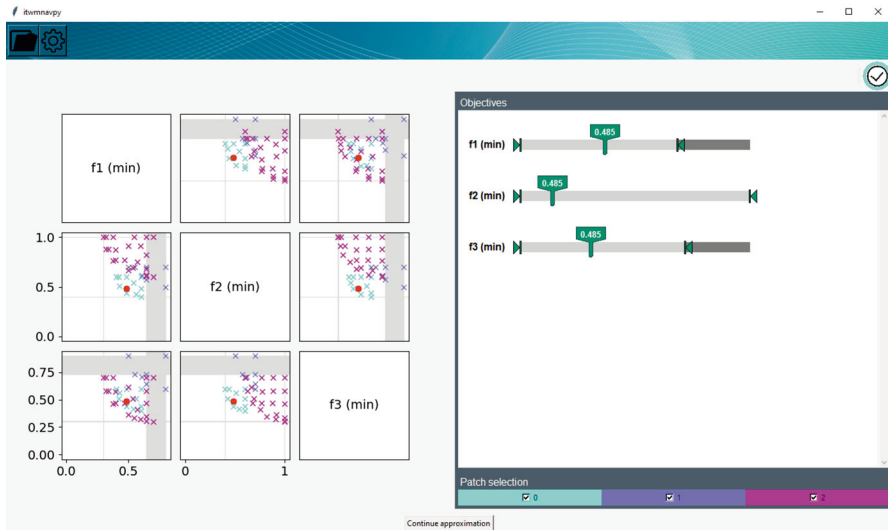


Fig. 5 After incorporating the restrictor positions, the approximation is refined using the patch approximation algorithm. No new nondominated points have been computed in the restricted (shaded) areas. The decision maker can now explore the trade-offs of the finely approximated nondominated set of (1)

need to remove nondominated patch points that are dominated with respect to the nondominated set of the overall problem (1).

3.5 Patch approximation algorithm scheme

After introducing the elements of the patch approximation algorithm, we can now state the algorithm scheme in Algorithm 4.

The algorithm is illustrated in Fig. 6 on a biobjective case with five patches. Patches 1 and 2 partially contribute to the nondominated set, patch 3 is fully dominated and intersects patch 1 and 2, and patch 4 and 5 are fully dominated and far behind the nondominated set. We select method (4) to check for dominating points and remove dominated parts using fake nondominated points (Algorithm 2).

Step 1 (Fig. 6a): An initial approximation is computed for every patch. For simplicity, we only show the inner approximation I_p , $p = 1, \dots, 5$.

Step 2 and 3 (Fig. 6b): The decision maker removed the shaded area from further refinement using restrictors. I_4 and I_5 are updated with new nondominated points (bold 'x').

Step 4 phase 1 (Fig. 6c): patches 4 and 5 are identified as fully dominated by their ideal points (both labelled z^I).

Step 4 phase 2 (Fig. 6d): Patch 3 is identified as fully dominated by computing dominating points (bold 'x') using (4). Parts of patches 1 and 2 are identified as dominated, but cannot be removed using fake Points (see Algorithm 2) since they contain only a single point each, no facet.

Algorithm 4 Multiobjective patch approximation algorithm

Require: target approximation quality ϵ_{approx} or other stopping criterion,

dominance precision ϵ_{dom} ,

a method that checks if a point is ϵ_{dom} -dominated by some patch (e.g. programs (4) or (3)),
 a method for each patch that calculates points using weighted sum scalarization

- 1: Generate initial approximation for each patch: compute the extreme compromises and the weighted sum solution with equal weights
 - 2: Present the initial approximation to the decision maker in a patch navigation tool (see Sect. 3.4). The decision maker can exclude parts of the objective space that they are not interested in from further refinement using restrictors.
 - 3: Process the restrictor positions using Algorithm 3
 - 4: Check the initial approximations for ϵ_{dom} -domination and remove ϵ_{dom} -dominated patch parts using Algorithm 1.
 - 5: **while** the stopping criterion is not met **do**
 - 6: Calculate a new patch point using Sandwiching in the patch that has the worst approximation quality
 - 7: Check whether the new point is ϵ_{dom} -dominated by some other patch. If this is the case, mark this part of the patch as ϵ_{dom} -dominated using Algorithm 2.
 - 8: Determine the approximation quality of the updated patch and update the estimate of the approximation quality of the nondominated set of (1) as the maximum of the patch qualities.
 - 9: **end while**
-

Since for simplicity we only show the inner approximation in Fig. 6, we cannot see that patch 2 attained the worst patch approximation quality.

Step 6 (Fig. 6e): Add a new nondominated point \bar{z} to patch 2.

Step 7 (Fig. 6f): We check \bar{z} for domination. Two nondominated points (\bar{z} and z_1) of patch 2 form a facet dominated by patch 1. The facet can then be removed by adding a fake nondominated point.

In the following section, we will investigate the convergence behaviour of the patch approximation algorithm.

4 Convergence properties of the patch approximation algorithm

We can obtain convergence results of the patch approximation algorithm (Algorithm 4) by using corresponding results for the Sandwiching algorithm introduced in Lammel (2023). The results are formulated using the epsilon indicator δ^ϵ (see Sect. 2.3), but also hold for some other metrics, e.g. the polyhedral gauge and the Hausdorff metric.

To apply the algorithms, we need the additional assumption that all patches need to be full-dimensional. Requirements for a full-dimensional nondominated set have been derived in Hillermeier (2001).

Theorem 10 *Algorithm 4 applied to a problem of the form (1) with full-dimensional patches converges, i.e. $\lim_{n \rightarrow \infty} \delta^\epsilon(\mathbf{I}^n, \mathbf{O}^n) = 0$.*

Proof For every patch, the Sandwiching algorithm converges (Theorem 3.3.18 of Lammel (2023)). Since the overall approximation quality of the nondominated set of (1) can be bounded by the maximum of all patch approximation qualities (Lemma 1) and a new nondominated point is added to the patch that attains the worst quality, all patches will eventually be approximated because we assume that there are only finitely

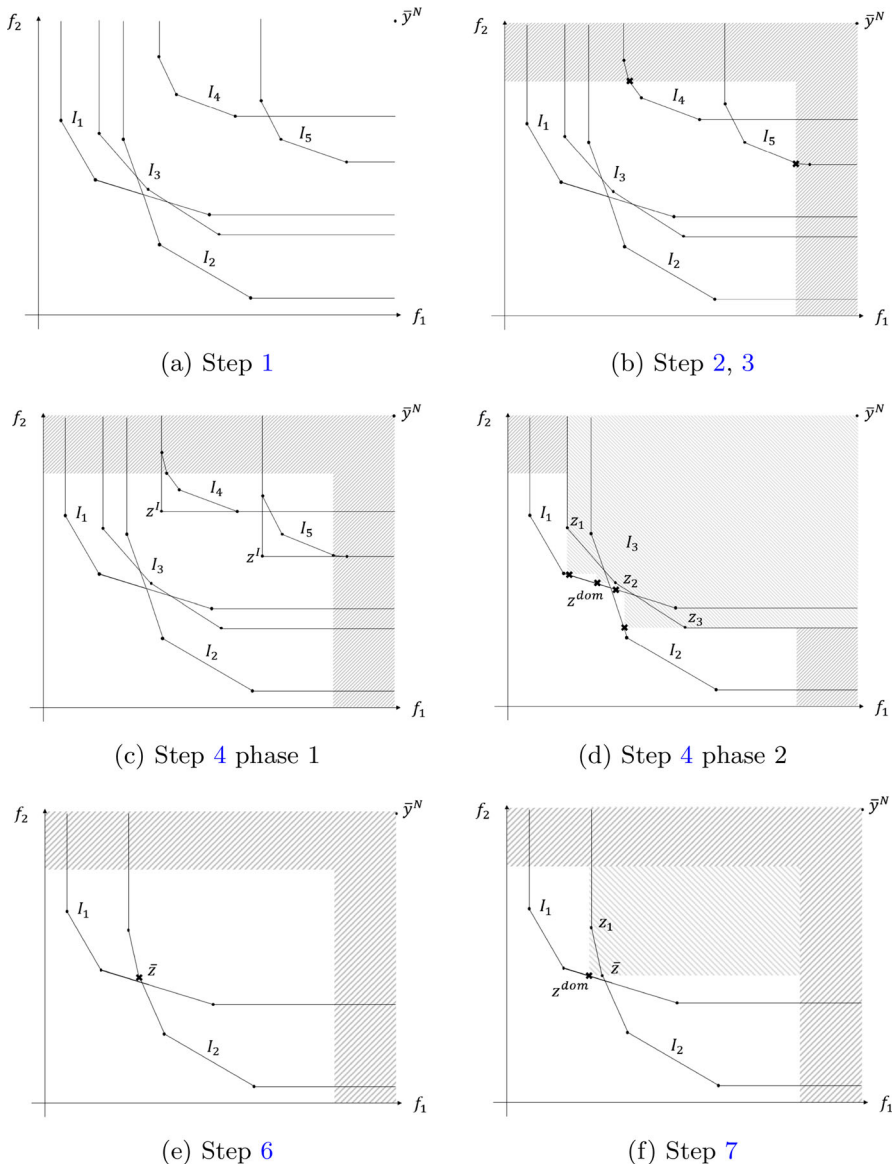


Fig. 6 Illustration of the patch approximation algorithm (Algorithm 4) in a biobjective example with five patches

many patches. Some parts of the nondominated set of the original problem (1) will not be approximated finely since they are marked as ϵ_{dom} -dominated in the course of the algorithm.

In the worst case, no part of one patch dominates another patch. Then, the nondominated set of (1) is just the union of the nondominated sets of all patches. In Lammel

(2023), a bound on the reduction of the approximation error of the Sandwiching approximation was introduced. We extend this result to the patch approximation algorithm. Theorem 11 shows that the approximation error is reduced within one iteration in the order of $n^{1/(d-1)}$ with n the number of iterations. With additional regularity of the patches, we can obtain an improved rate of $n^{2/(d-1)}$.

We introduce some notation that will be used in the following theorems. The asphericity $\alpha(C)$ is defined as the minimal ratio of the radii of concentric outer and inner spheres of C , $r_{\text{inner}}(C)$ is the radius of the largest ball included in C and $r_{\text{outer}}(C)$ is the radius of the smallest ball containing C . The constant π_d denotes the volume of a d -dimensional unit ball.

Theorem 11 *Let $\{(\mathbf{I}^n, \mathbf{O}^n)\}_{n=0,1,\dots}$ be a sequence of inner and outer approximations generated by the patch approximation algorithm (Algorithm 4) for q patches. The set C^i denotes patch i , all patches are full-dimensional. Then for any $\epsilon > 0$ there exists a number n_0 such that for $n \geq n_0$ it holds*

$$\delta^\epsilon(\mathbf{I}^n, \mathbf{O}^n) \leq (1 + \epsilon) \sum_{i=1}^q \left(c_1^i n^{1/(d-1)} \right)^{-1}$$

where

$$c_1^i = \frac{1}{2} \left(\frac{d-1}{d} \frac{\pi_{d-1}}{d} \frac{1}{\sigma(C^i)} \right)^{1/(d-1)} (\alpha(C^i)^2 - 1)^{-1/2} \alpha(C^i)^{2d/(1-d)} \left(\frac{1}{\sqrt{d}} \right)^{d/(d-1)}.$$

Theorem 12 *Let $\{(\mathbf{I}^n, \mathbf{O}^n)\}_{n=0,1,\dots}$ be a sequence of inner and outer approximations generated by the patch approximation algorithm (Algorithm 4) for q patches. The set C^i denotes patch i , I_0^i denotes the initial inner approximation of the patch. If each patch is full-dimensional, the objective functions and the functions forming the feasible set are three times continuously differentiable, the function mapping a weight λ of (2) to a weakly nondominated point is locally injective, and for the KKT points of the weighted sum problem the strict complementary slackness condition, LICQ and SOSC are fulfilled, then for any $\epsilon > 0$ there exists a number n_0 such that for $n \geq n_0$ it holds*

$$\delta^\epsilon(\mathbf{I}^n, \mathbf{O}^n) \leq (1 + \epsilon) \sum_{i=1}^q \left(c_2^i n^{2/(d-1)} \right)^{-1}$$

where

$$c_2^i = \left(\frac{d-1}{d+1} \frac{\pi_{d-1}}{d} \left(\frac{r_{\text{inner}}(I_0^i)}{\sqrt{d} r_{\text{outer}}(C^i)} \right)^d \frac{1}{\sigma(C^i)} \right)^{2/(d-1)} \frac{\rho_{\min}(C^i)}{8}.$$

5 Numerical results

We will demonstrate the abilities of our patch approximation algorithm (Algorithm 4). We compare our algorithm to the HyPaD algorithm presented in Eichfelder and Warnow (2023). Additionally, we demonstrate that our algorithm performs well in higher-dimensional cases using a seven-criteria problem. Since we want to compute the entire nondominated set in the following examples, we will omit the navigation step of the patch approximation algorithm.

We implemented the patch approximation algorithm in Python 3.9. All convex optimization problems are solved using `scipy.optimize.minimize`'s SLSQP algorithm. The tests were performed on a laptop with 32 GB of RAM and an Intel Core i7-11850H processor with a clock rate of 2.5 GHz and 8 cores.

5.1 Comparing our algorithm to the HyPaD algorithm

From experimental data presented in Eichfelder and Warnow (2023), it appears that the HyPaD algorithm introduced in Eichfelder and Warnow (2023) is currently the best-performing patch algorithm that can be applied to problems of arbitrary objective dimension. The HyPaD algorithm also approximates the nondominated set of multiobjective mixed-integer convex problems. It requires that every patch problem is defined using the same feasible set. However, their algorithm allows an infinite amount of feasible integer assignments. In the following, we apply our algorithm to the three examples discussed in Eichfelder and Warnow (2023) to demonstrate the abilities of our patch approximation algorithm and compare the behaviour to the HyPaD algorithm.

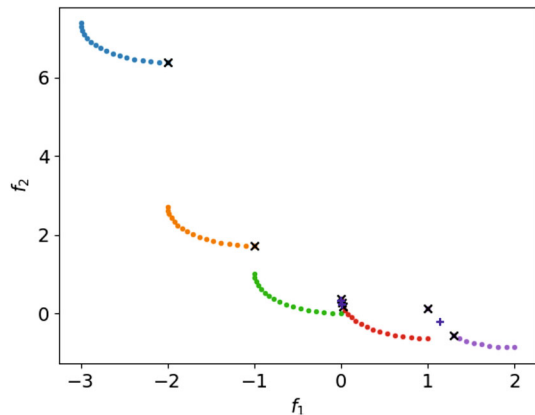
To compare the results we use the computation times given in Eichfelder and Warnow (2023). In Eichfelder and Warnow (2023), the HyPaD algorithm was implemented in MATLAB, the numerical tests were performed on a machine with 32 GB of RAM and an Intel Core i9-10920X processor with a clock rate of 3.5 GHz and 12 cores.

The HyPaD algorithm constructs an enclosure of the nondominated set by excluding dominated and dominating regions of the objective space. The space between the lower and upper bounds can be separated into boxes (rectangles in the biobjective case) and the quality criterion is the largest smallest length of all boxes. This criterion only uses the computed nondominated points and dominance relations.

In our proposed algorithm, the approximation of the nondominated set is not only given by the nondominated points, but also uses the convexity of the patches in the definition of the inner and outer approximation. The epsilon indicator measures the distance not between nondominated points but between the inner and outer approximation.

As an example, assume that a patch in biobjective space is given by a line between two points. Our algorithm would only need to compute the two points to approximate the nondominated set exactly as the inner and outer approximation are an exact linear model of this linear nondominated set. The epsilon indicator would then be zero and the algorithm would terminate. The quality criterion of the HyPaD algorithm evaluated

Fig. 7 Nondominated set of example (5) generated by our patch approximation algorithm. Dominated points are marked with an 'x', added fake nondominated points are marked with a '+'. Those parts of patches that contribute to the nondominated set are coloured



for the approximation of the line by two nondominated points is the minimal length of the rectangle defined by the two points.

As the behaviour of the quality criteria is therefore very different, none of the two algorithms can be evaluated by the quality criterion of the other algorithm.

We can therefore not compare the performance of the two algorithms quantitatively. For the following examples, we will still document values of the two quality criteria for their respective algorithms as they still represent a notion of the approximation quality.

We will determine whether a part of a patch is dominated using one search direction by solving (3) for $\epsilon_{\text{dom}} = 0$. We will remove dominated parts using fake nondominated points.

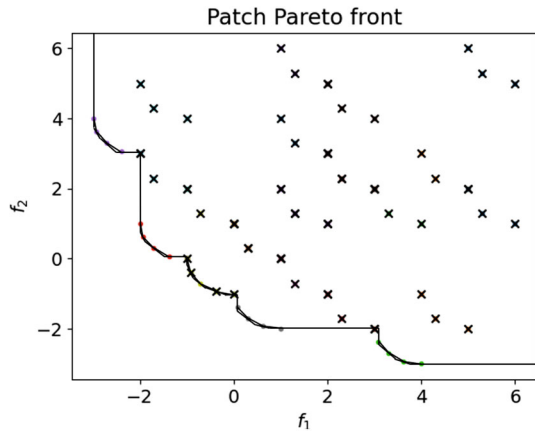
Biobjective problem with five patches

This biobjective test instance with quadratic constraint functions and a non-quadratic objective function was introduced in De Santis et al. (2020) under the name T6. There are two continuous variables and one integer variable which leads to five patches (Fig. 7).

$$\begin{aligned}
 &\min (x_1 + x_3, x_2 + \exp(-x_3))^T \\
 &\text{s.t. } x_1^2 + x_2^2 \leq 1 \\
 &\quad x_1, x_2 \in [-2, 2] \\
 &\quad x_3 \in \{-2, -1, 0, 1, 2\}.
 \end{aligned} \tag{5}$$

To apply our patch approximation algorithm, we create one optimization problem for every possible integer assignment. The HyPaD algorithm computes an enclosure of the nondominated set with maximal box width $\epsilon = 0.1$ within 1.34 s and for $\epsilon = 0.01$ within 6.09 s, that is, the improved quality is achieved in approximately 4.5 times the approximation time. Our patch approximation algorithm computes an approximation with epsilon indicator quality of 0.05 within 0.1 s and achieves quality 0.005 within 0.32 s which is 3.2 times the approximation time of the lower approximation quality.

Fig. 8 Nondominated set of example (6) with two continuous and two discrete variables, e.g. 25 patches, generated by our patch approximation algorithm. Dominated points are marked with an 'x'. Those parts of patches that contribute to the nondominated set are coloured



Bicriteria problem with scalable number of continuous and integer variables

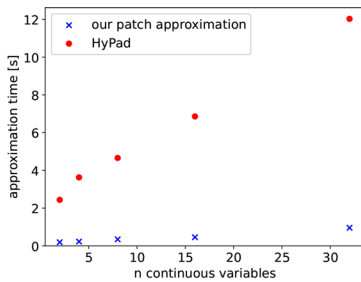
The next test case (example H1 of Eichfelder and Warnow (2023)) has two quadratic objective functions and a quadratic constraint function. The number of continuous variables n and the number of integer variables m are even natural numbers. The problem is given as

$$\begin{aligned} \min & \left(\sum_{i=1}^{n/2} x_i + \sum_{i=n+1}^{n+m/2} x_i^2 - \sum_{i=n+m/2+1}^{n+m} x_i \right) \\ & \left(\sum_{i=n/2+1}^n x_i - \sum_{i=n+1}^{n+m/2} x_i + \sum_{i=n+m/2+1}^{n+m} x_i^2 \right) \\ \text{s.t. } & \sum_{i=1}^n x_i^2 \leq 1, \\ & x_1, \dots, x_n \in [-2, 2], \\ & x_{n+1}, \dots, x_m \in \{-2, -1, 0, 1, 2\}. \end{aligned} \quad (6)$$

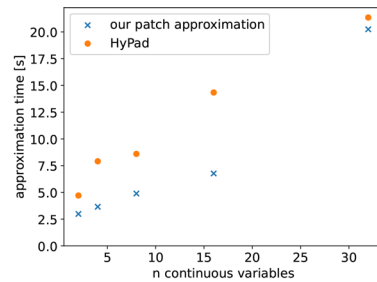
The number of patches is given by 5^m . We perform tests using $m = 2$ and $m = 4$, i.e. 25 and 625 patches. We use an epsilon indicator value of 0.05 as the stopping criterion for our patch approximation algorithm. The HyPaD approximation was improved to a width of the enclosure of 0.1.

For two continuous and two discrete variables, i.e. 25 patches, the computed patch points and the inner and outer approximation of the biobjective mixed-integer convex optimization problem are shown in Fig. 8. We can see that many patches are fully dominated. All of the fully dominated patches are discarded from the start since their ideal point is dominated by a different patch. Therefore, no fake nondominated points need to be added. In the figure, it looks like nondominated points are removed from the patch situated around the origin. In fact, there are two patches lying on top of each other. Only one patch is refined, the other is marked as dominated.

The results of the computation time of our patch approximation algorithm and the HyPaD algorithm (as documented in Eichfelder and Warnow (2023)) for 25 and 625 patches are shown in Fig. 9.



(a) Approximation times of our patch algorithm and the HyPaD algorithm for example (6) with 25 patches



(b) Approximation times of our patch algorithm and the HyPaD algorithm for example (6) with 625 patches

Fig. 9 Comparison of the approximation time of the nondominated set of example (6), with 25 and 625 patches

From the construction of our algorithm, which for example computes several non-dominated points for every patch, it is expected that it performs best for cases with small amounts of patches. This behaviour can be observed in this example. Our patch approximation algorithm performs significantly better than HyPaD in the case with 25 patches. But even for 625 patches our algorithm still outperforms the HyPaD algorithm.

For even larger numbers of patches, however, the HyPaD algorithm performs significantly better than our algorithm. Our patch approximation algorithm solves problem (6) with two continuous and six discrete variables, i.e. $5^6 = 15625$ patches to an epsilon indicator quality of 0.05 within 75 s. The HyPaD algorithm can solve the even larger problem with two continuous and eight discrete variables in 25.70 s.

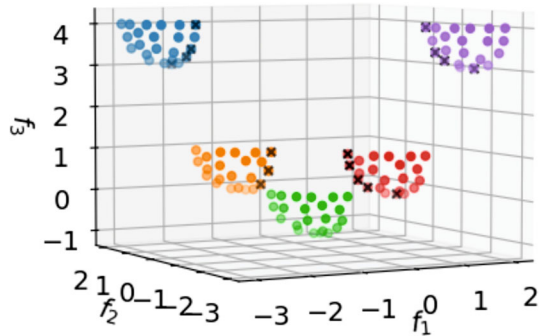
Tricriteria problem

The third test case that we use to compare our algorithm to the HyPaD algorithm from Eichfelder and Warnow (2023) is a tri-objective test instance, originally presented as T5 in De Santis et al. (2020). Its optimization problem is given by

$$\begin{aligned}
 & \min \begin{pmatrix} x_1 + x_4 \\ x_2 - x_4 \\ x_3 + x_4^2 \end{pmatrix} \\
 & \text{s.t. } \sum_{i=1}^3 x_i^2 \leq 1, \\
 & \quad x_1, x_2, x_3 \in [-2, 2], \\
 & \quad x_4 \in \{-2, -1, 0, 1, 2\}.
 \end{aligned} \tag{7}$$

Thus, there are five patches. The approximated nondominated set is shown in Fig. 10. We can see that the five patches mostly do not intersect. Some pieces of the boundary of some patches are determined as dominated (see the black 'x' in the figure). But the

Fig. 10 Approximated nondominated set of example (7) with three objectives and five patches, approximated to an epsilon indicator quality of 0.05



dominated parts never form a full facet so that no parts of the patch can be removed from further refinement by adding fake nondominated points. Since every patch contributes to the nondominated set, an integer search method instead of our approach to enumerate all patches would not improve the approximation process in this case.

The HyPaD algorithm computes an approximation with maximal width of 0.1 within about 9 s (see Eichfelder and Warnow (2023)). The patch approximation algorithm presented in this article approximates the same nondominated set to an epsilon indicator quality of 0.05 within 0.62 s.

5.2 An example with seven objectives

We illustrate that our algorithm can approximate the nondominated sets of problems with an arbitrary number of objectives by applying it to a problem with seven objectives.

The nondominated set consists of three spherical patches. One patch is fully dominated by the others. The patch problems are parametrized by the center point of the sphere and the radius. Patch 1 has center $c^1 = (1, \dots, 1)$ and radius $r^1 = 0.2$, patch 2 has $c^2 = (0.8, 0.8, 0.8, 0.6, 0.6, 0.6, 0.6)$, $r^2 = 0.4$ and patch 3 has $c^3 = (0.6, 0.6, 0.6, 0.7, 0.7, 0.7, 0.7)$, $r^3 = 0.3$. The j 'th patch problem is then given by

$$\begin{aligned} \min \quad & f_1(x) = x_1, \dots, f_7(x) = x_7 \\ \text{s.t.} \quad & \sum_{i=1}^7 \left(\frac{x_i - c_i^j}{r^j} \right)^2 \leq 1 \\ & 0 \leq x_1, \dots, x_7 \leq 1. \end{aligned}$$

Using the patch approximation algorithm (Algorithm 4) using problem (3) as the criterion to check for dominating patches, this nondominated set can be approximated to an epsilon indicator value of less than 0.1 using 106 nondominated points within 14.5 s. and to an epsilon indicator value of 0.05 using 325 nondominated points within 765 s.

6 Conclusion

We introduced a novel algorithm that approximates the nondominated sets of multiobjective mixed-integer convex problems of the form (1). To the best of our knowledge, it is the first algorithm that constructs an inner and outer approximation of the nondominated set exploiting the convexity of the patches for problems with an arbitrary number of criteria.

In the algorithm, the problem is decomposed into patches, which are bounded multiobjective convex problems, by fixing the integer assignments. The patch problems are solved using simplicial Sandwiching. We introduce methods that determine parts of patches that are ϵ_{dom} -dominated by other patches. Then, we remove these patch parts from further refinement by adding artificial nondominated points. An interactive step allows a decision maker to exclude parts of the nondominated set from further refinement after a coarse approximation of the nondominated set has been computed. We proved that the algorithm converges and showed a bound on the reduction of the approximation error.

We demonstrated that the algorithm can approximate nondominated sets with small and medium (up to a few hundred) numbers of patches faster than the HyPaD algorithm introduced by Eichfelder and Warnow (2023), which then outperforms our algorithm for larger numbers of patches. To illustrate that our algorithm can also approximate the nondominated sets of problems with an arbitrary number of criteria, we apply the algorithm to a problem with seven objectives.

Many problems arising from applications can be modeled as multiobjective mixed-integer convex problems. In other cases, for example in radiotherapy or chemical process engineering, a practical problem can be solved using different technologies that are modeled as multiobjective convex problems. Approximating the nondominated set of multiple multiobjective convex problems at once is also captured by our problem formulation (1). Typically, the number of technologies that are compared is small, e.g. less than 10, and the computational effort of computing a nondominated point is high. The technologies can then be interpreted as patches and their nondominated set can be computed using our patch approximation algorithm.

Acknowledgements We thank the employees of the optimization department of Fraunhofer ITWM who developed the experimental software for the Sandwiching algorithm and Patch Navigation that were used in the implementation of our patch approximation algorithm.

Funding Open Access funding enabled and organized by Projekt DEAL. The research leading to these results received funding from DFG under Grant Agreement No 441208898.

Data availability Not applicable.

Code availability The software used to calculate the results is closed-source and intellectual property of Fraunhofer Institute for Industrial Mathematics.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Adelgren N, Gupte A (2022) Branch-and-bound for biobjective mixed-integer linear programming. *INFORMS J Comput* 34(2):909–933
- Allmendinger R, Ehrgott M, Gandibleux X, Geiger MJ, Klamroth K, Luque M (2017) Navigation in multiobjective optimization methods. *J Multi-Criteria Decis Anal* 24(1–2):57–70
- Barber CB, Dobkin DP, Huhdanpaa H (1996) The quickhull algorithm for convex hulls. *ACM Transact Math Softw (TOMS)* 22(4):469–483
- Bokrantz R, Forsgren A (2013) An algorithm for approximating convex pareto surfaces based on dual techniques. *INFORMS J Comput* 25(2):377–393
- Bortz M, Burger J, Aspiron N, Blagov S, Böttcher R, Nowak U, Scheithauer A, Welke R, Küfer K-H, Hasse H (2014) Multi-criteria optimization in chemical process design and decision support by navigation on pareto sets. *Comput Chem Eng* 60:354–363
- Burachik RS, Kaya CY, Rizvi M (2017) A new scalarization technique and new algorithms to generate pareto fronts. *SIAM J Optim* 27(2):1010–1034
- Burachik RS, Kaya CY, Rizvi MM (2022) Algorithms for generating pareto fronts of multi-objective integer and mixed-integer programming problems. *Eng Optim* 54(8):1413–1425
- Cabrera-Guerrero G, Ehrgott M, Mason AJ, Raith A (2021) Bi-objective optimisation over a set of convex sub-problems. *Ann Oper Res*, 1–26
- Cacchiani V, D'Ambrosio C (2017) A branch-and-bound based heuristic algorithm for convex multi-objective minlps. *Eur J Oper Res* 260(3):920–933
- Ceyhan G, Köksalan M, Lokman B (2019) Finding a representative nondominated set for multi-objective mixed integer programs. *Eur J Oper Res* 272(1):61–77
- Collicott C, Bonacker E, Lammel I, Teichert K, Walczak M, Süß P (2021) Interactive navigation of multiple convex patches. *J Multi-Criteria Decis Anal* 28(5–6):311–321
- Craft DL, Halabi TF, Shih HA, Bortfeld TR (2006) Approximating convex pareto surfaces in multiobjective radiotherapy planning. *Med Phys* 33(9):3399–3407
- De Santis M, Eichfelder G, Niebling J, Rocktäschel S (2020) Solving multiobjective mixed integer convex optimization problems. *SIAM J Optim* 30(4):3122–3145
- Diessel E (2021) Precise control of approximation quality in multicriteria optimization. Preprint at <https://optimization-online.org/?p=16724>
- Diessel E (2022) An adaptive patch approximation algorithm for bicriteria convex mixed-integer problems. *Optimization* 71(15):4321–4366
- Dörfler D, Löhne A, Schneider C, Weißing B (2021) A benson-type algorithm for bounded convex vector optimization problems with vertex selection. *Optim Methods Softw*, 1–21
- Ehrgott M (2005) *Multicriteria optimization*. Springer Science & Business Media (491)
- Ehrgott M, Shao L, Schöbel A (2011) An approximation algorithm for convex multi-objective programming problems. *J Global Optim* 50(3):397–416
- Eichfelder G, Stein O, Warnow L (2022) A solver for multiobjective mixed-integer convex and nonconvex optimization. Preprint at <https://optimization-online.org/?p=18696>

- Eichfelder G, Warnow L (2023) A hybrid patch decomposition approach to compute an enclosure for multi-objective mixed-integer convex optimization problems. *Math Methods Oper Res*, 1–30
- Eskelinen P, Miettinen K, Klamroth K, Hakanen J (2010) Pareto navigator for interactive nonlinear multiobjective optimization. *OR Spectr* 32:211–227
- Halffmann P, Schäfer LE, Dächert K, Klamroth K, Ruzika S (2022) Exact algorithms for multiobjective linear optimization problems with integer variables: a state of the art survey. *J Multi-Criteria Decis Anal* 29(5–6):341–363
- Hartikainen M, Miettinen K, Klamroth K (2019) Interactive nonconvex pareto navigator for multiobjective optimization. *Eur J Oper Res* 275(1):238–251
- Hillmermeier C (2001) Generalized homotopy approach to multiobjective optimization. *J Optim Theory Appl* 110(3):557–583
- Klamroth K, Tind J, Wiecek MM (2003) Unbiased approximation in multicriteria optimization. *Math Methods Oper Res* 56(3):413–437
- Lammel I (2023) Approximation of pareto surfaces in multi-criteria optimization. PhD thesis, RPTU Kaiserslautern
- Lammel I, Küfer K-H, Süß P (2023) Efficient Computation of the Approximation Quality in Sandwiching Algorithms. Preprint at <https://optimization-online.org/?p=24383>
- Löhne A, Rudloff B, Ulus F (2014) Primal and dual approximation algorithms for convex vector optimization problems. *J Global Optim* 60(4):713–736
- Monz M, Küfer K-H, Bortfeld TR, Thieke C (2008) Pareto navigation-algorithmic foundation of interactive multi-criteria IMRT planning. *Phys Med Biol* 53(4):985
- Nowak D, Johnson T, Mark A, Ireholm C, Pezzotti F, Erhardsson L, Ståhlberg D, Edelvik F, Küfer K-H (2021) Multicriteria optimization of an oven with a novel ε -constraint-based sandwiching method. *J Heat Transfer* 143(1):012101
- Nowak D, Küfer K-H (2020) A Ray Tracing Technique for the Navigation on a Non-convex Pareto Front. Preprint at <https://arxiv.org/abs/2001.03634>
- Nowak D, Teichert K, Asprion N, Bortz M (2022) Rather the rule than the exception: non-convex pareto sets and their navigation in distillation processes. *Chemie Ingenieur Technik*
- Rennen G, Dam ER, Hertog DD (2011) Enhancement of sandwich algorithms for approximating higher-dimensional convex pareto sets. *INFORMS J Comput* 23(4):493–517
- Ruzika S, Wiecek MM (2005) Approximation methods in multiobjective programming. *J Optim Theory Appl* 126(3):473–501
- Serna Hernandez JI (2011) Multi-objective optimization in mixed integer problems with application to the beam selection optimization problem in IMRT. PhD thesis, TU Kaiserslautern
- Serna J, Monz M, Küfer K-H, Thieke C (2009) Trade-off bounds for the pareto surface approximation in multi-criteria IMRT planning. *Phys Med Biol* 54(20):6299
- Süß P, Foltin G, Heidgen M, Hajnal D, Diaz J, Schultze H, Gattermayer J, Lehner S (2022) Simulation and optimization in process engineering chapter 12 - Product development in a multicriteria context. Elsevier, 321–338
- Teichert K, Süß P, Serna J, Monz M, Küfer K, Thieke C (2011) Comparative analysis of pareto surfaces in multi-criteria IMRT planning. *Phys Med Biol* 56(12):3669
- Ulus F (2018) Tractability of convex vector optimization problems in the sense of polyhedral approximations. *J Global Optim* 72(4):731–742