An approximation algorithm for multi-objective mixed-integer convex optimization

Ina Lammel[*], Karl-Heinz Küfer[*], Philipp Süss[*]

February 28, 2023

In this article we introduce an algorithm that approximates Pareto fronts of multiobjective mixed-integer convex optimization problems. The algorithm constructs an inner and outer approximation of the front exploiting the convexity of the patches and is applicable to 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.

1 Introduction

Multicriteria optimisation 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 Pareto fronts of multiobjective mixed-integer convex (MOMIC) optimization problems

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

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

where $f(x)$ denotes the vector of d convex objective functions $f_i : \mathbb{R}^{k+m} \to \mathbb{R}$ and the decision vectors $x \in \mathcal{X} \subset \mathbb{R}^k \times \mathbb{Z}^m$ have continuous and integer components. The subset of the feasible set $\mathcal{X}_C$ containing the continuous variables is convex. We assume that the set of feasible integer assignments $\mathcal{X}_I$ is finite.

We can decompose the MOMIC problem 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*

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

The Pareto front of a patch problem is called a *patch*. This term is also used in Diessel (2022), Eichfelder and Warnow (2021) and Serna Hernandez (2011).

Many strategies for the approximation of Pareto fronts 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 nonlinear 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 Pareto fronts of problems with disconnected feasible sets can also be applied to the approximation of Pareto fronts of MOMI problems. They compute points of the Pareto front 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 Pareto set of a MOMIC 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.

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

There is a group of algorithms using a branch-and-bound approach to solve MOMIC 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 DAmbrosio (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 DAmbrosio (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 (2021) 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 Pareto front and that are used to compute new Pareto 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.

An algorithm for bicriteria 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 Pareto points. Then, an inner and outer approximation of the Pareto front of the MOMIC 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 current approximation error.

In this article, we will present an algorithm that is designed for multiobjective mixed-integer convex problems where the integer aspect is not the dominant difficulty. These can be problems with only few patches. 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 Pareto front of the MOMIC problem and is applicable to problems with any number of criteria. We will approximate the Pareto front of the MOMIC problem by iteratively computing Pareto 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 MOMIC Pareto front using boxes as in Eichfelder and Warnow (2021) is convenient, it is not the best-possible approximation of convex parts of the Pareto front. We define the inner and outer approximation of the patch Pareto fronts and the Pareto front of the MOMIC problem 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 Pareto front 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 Pareto front will then not be refined in following steps of the algorithm.

2 Constructing a Sandwich approximation of the Pareto front

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

2.1 Lower and upper bounds on the patch Pareto fronts

We construct inner and outer approximations of the Pareto fronts of the convex patch problems. Let $\{z_p^1, \ldots, z_p^n\}$ be the computed Pareto points of patch p. The inner approximation of the patch Pareto front is given by the convex hull of the Pareto points, extended by the standard domination cone

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

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

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

When a new Pareto 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 Pareto front 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 Pareto fronts (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 bicriteria case by Cabrera-Guerrero et al. (2021).

2.2 Lower and upper bounds on the Pareto front of the MOMIC problem

The Pareto front of the MOMIC problem is the nondominated set of the union of the patch Pareto fronts. Let the Pareto 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 $I_1, \ldots, I_q$ and $O_1, \ldots, O_q$ 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

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

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

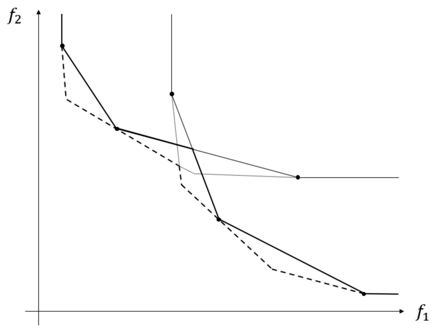


Figure 1: A MOMIC problem consisting of two patches. The patches are approximated by three patch Pareto points each. The boundaries of the inner and outer approximations of the patch Pareto fronts are marked with thin lines. The boundary of the inner approximation of the MOMIC problem is marked with 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.

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

The approximation quality of a MOMIC approximation can be measured by obtaining the patch approximation qualities.

Lemma 2.1. *Let a MOMIC problem have q patches and let $\delta^\epsilon(I_p, O_p) \leq \epsilon$ for every patch p, $p \in \{1,\ldots,q\}$ with inner and outer approximation I_p, O_p. Then, the MOMIC Pareto front is also approximated with quality ϵ: $\delta^\epsilon(\mathbf{I},\mathbf{O}) \leq \epsilon$.*

Proof. From lemma 1 of Diessel (2021) and using the definition of the inner approximation we obtain

$$\delta^\epsilon(\mathbf{I},\mathbf{O}) = \delta^H(\mathbf{I},\mathbf{O}) := \max\left\{\max\left\{\|x,\mathbf{I}\|_\infty, x \in \mathbf{O}\right\}, \max\left\{\|y,\mathbf{O}\|_\infty, y \in \mathbf{I}\right\}\right\}.$$

where δ^H is the Hausdorff distance defined using the maximum metric. Let $x \in O_l, l \in \{1,\ldots,q\}, x \in \mathbf{O}$ and let $\bar{y} \in I_l$ such that

$$\|x, I_l\|_\infty = \min\left\{\|y - x\|_\infty,\ y \in I_l\right\} = \|x, \bar{y}\|_\infty.$$

Then

$$\begin{aligned}
\|x, I^{\text{union}}\|_\infty =& \|x, I_1 \cup \ldots \cup I_q\|_\infty \\
=& \min\left\{\|x, I_1\|_\infty, \ldots, \|x, I_q\|_\infty\right\} \\
\leq& \|x, I_l\|_\infty = \|x, \bar{y}\|_\infty \leq \epsilon.
\end{aligned}$$

This implies that either $\bar{y} \in \mathbf{I}$ and therefore $\|x,\mathbf{I}\|_\infty = \|x, I^{\text{union}}\|_\infty \leq \epsilon$ or $\bar{y} \in I^l \subset I^{\text{union}}$ but $\bar{y} \notin \mathbf{I}$ since $\bar{y}$ is dominated by some $\bar{z} \in \mathbf{I}$. Then, since $\bar{z}_i \leq \bar{y}_i\ \forall i = 1,\ldots,d$, it holds $|\bar{z}_i - x_i| \leq |\bar{y}_i - x_i|\ \forall i = 1,\ldots,d$ and therefore $\|\bar{z} - x\|_\infty \leq \|\bar{y} - x\|_\infty$. Thus,

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

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

The epsilon indicator value of a Sandwiching approximation can be computed by solving a small linear program for every vertex of the outer approximation, see Serna Hernandez (2011), Bokrantz and Forsgren (2013). Due to Lemma 2.1, the same procedure can be used to obtain a bound on the error of the MOMIC Pareto front approximation.

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 MOMIC Pareto front. We detect parts of patches that are dominated and avoid generating more points in these areas.

Often, only a small part of the Pareto front 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 Pareto front will not be refined in following steps of the algorithm. When the approximation process is completed, the Pareto front approximation can 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 Pareto points to patches so that the approximation quality of the MOMIC Pareto front 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 Sss 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 section 2.1: the inner approximation is the convex hull of the Pareto points, extended by the domination cone. The outer approximation is given by the cut of the supporting half-spaces in the Pareto points. Then, the approximation quality, e.g. the epsilon-indicator value (cf. section 2.3) is computed. To improve the approximation, a new Pareto point is computed so that its tangential hyperplane is parallel to the inner approximation facet where the worst quality value was attained.

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 Pareto front

We want to identify dominated patch parts to avoid refining parts of patches that do not contribute to the MOMIC Pareto front. However, we relax the nondomination requirement a bit: in our algorithm it is enough if a point z is *ϵ_{dom}-nondominated* for some $\epsilon_{\mathrm{dom}} \geq 0$ to be kept for refinement, i.e. if $z - \epsilon_{\mathrm{dom}} := (z^1 - \epsilon_{\mathrm{dom}}, \ldots, z^d - \epsilon_{\mathrm{dom}})$ is nondominated. We do this to allow the decision maker to incorporate goals in the decision-making process that are not captured in the optimization problem. After computing an initial approximation for every patch, we apply a two-phase approach.

First, we try to identify as many fully dominated patches as possible by looking only at the Pareto points of the patches. 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 Pareto point of such a patch.

Then, we move on to another procedure that is guaranteed to identify any dominated patch and that can detect dominated parts of patches. We propose to check for every new patch Pareto 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 dominated until a dominating patch is found.

Although this approach is sufficient in our applications where the computational effort of computing a new Pareto 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 when a significant number of the patches is already discarded in the first phase. If the patch approximation problem can be formulated as a mixed-integer program, for large numbers of patches the algorithm presented in Eichfelder and Warnow (2021) may be the better choice since it avoids computing a small initial approximation for every integer assignment. In section 5 we will compare the performance 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 $Z \in \mathbb{R}^{d \times k}$ contain the Pareto 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 $\xi \in \mathbb{R}^k$ describes the convex combination of the patch Pareto points and $\mu \in \mathbb{R}^d_{\geq 0}$ adds the standard domination cone $\mathbb{R}^d_{\geq}$.

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

$$\begin{aligned} \min\ & - e^T \cdot d \\ \text{s.t. } & Z\xi + \mu + d = z - \epsilon_{\mathrm{dom}} \\ & e^T \xi = 1 \\ & d, \xi, \mu \geq 0 \end{aligned} \tag{3.1}$$

can be solved. The vector e is the vector of ones. If (3.1) has an optimal solution (d, ξ, μ), a dominating point is given by $z^{\text{dom}} = Z\xi + \mu$.

If the computational effort for generating a Pareto 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 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

$$\begin{aligned} &\min -d \\ &\text{s.t. } (Z\xi)^j + \mu^j + d = z^j - \epsilon_{\text{dom}} \\ &\quad (Z\xi)^i + \mu^i = z^i - \epsilon_{\text{dom}} \ \forall i \neq j \\ &\quad e^T \xi = 1 \\ &\quad d, \xi, \mu \geq 0 \end{aligned} \tag{3.2}$$

where Z, ξ, μ, e are defined as above. If (3.2) has an optimal solution (d, ξ, μ), 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 3.1. *If there exists a point on the inner approximation of a patch dominating z, then the programs 3.1 and 3.2 have a solution (d, ξ, μ) and $z^{dom} = Z\xi + \mu$ dominates z.*

Proof. Let z' be a point of patch p dominating z. Then $z' + \mathbb{R}^d_{\geq}$ is part of the inner approximation of patch p and there exists a $\mu \in \mathbb{R}^d$, $\mu \geq 0$ so that $z' + \mu$ is a feasible point of (3.1) and (3.2). Since for an optimal solution z^{dom} of (3.1) or (3.2) it holds $d \geq 0$ and therefore $z^{\text{dom}} = Z\xi + \mu \leq z$, so z^{dom} dominates z. □

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

Lemma 3.2. *If the solution (d, ξ, μ) of problems 3.1 or 3.2 fulfils $\mu_i > 0$ for some $i \in \{1, \ldots, d\}$, then $\tilde{z}^{dom} = Z\xi$ dominates z.*

Proof. From lemma 3.1 we know that $Z\xi + \mu$ dominates z. Since $\mu \geq 0$, it holds $\tilde{z}^{\text{dom}} = Z\xi \leq Z\xi + \mu \leq z$. □

The method is illustrated in figure 2. In a run of the patch approximation algorith, however, patch 2 would have already been marked as fully 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

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 dominated by some patch (e.g. 3.2 or 3.1)
a method that marks a dominated part of a patch Pareto front to avoid placing more points there

Remove patches whose ideal points are dominated by patches with nondominated ideal points.
for evey patch **do**
    Determine the ideal point of the patch
    Check whether the ideal point is ϵ_{dom}-dominated by a different patch
    if a dominating point is found **then**
        Mark the whole patch as dominated.
    else
        for every patch point that is not already known to be dominated **do**
            Check if the point is dominated by a different patch
            if a dominating point z^{dom} is found **then**
                Mark the region of the patch intersecting $z^{\text{dom}} + \mathbb{R}^d_{\geq}$ as dominated
            end if
        end for
    end if
end for

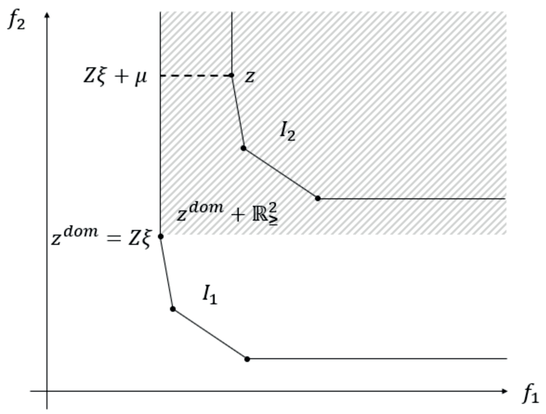


Figure 2: A MOMIC problem consisting of two patches. We check whether the second patch is dominated by solving (3.2) for the patch Pareto point z in the first coordinate direction. The solution $Z\xi + \mu$ dominates z, as well as the dominating point $z^{\text{dom}} := Z\xi$. The whole second patch is contained in the cone $z^{\text{dom}} + \mathbb{R}^2_{\geq}$, so that it can be marked as fully dominated.

and DAmbrosio (2017), Adelgren and Gupte (2022)) is to check whether the ideal point of a patch is dominated by some other patch.

Lemma 3.3. *If the ideal point z^I of a patch is ϵ_{dom}-dominated by a different patch, a part of the patch will be fully marked as 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 (3.1) (or (3.2)). Using lemma 3.1, we know that if the ideal point of the patch is dominated, then some dominating point z^{dom} will be found using (3.1) (or (3.2)). Then, the part of the patch intersecting $z^{\text{dom}} + \mathbb{R}^d_{\geq}$ will be marked as dominated. Since for every patch Pareto point z it holds $z^I \leq z$ and $z^{\text{dom}} \leq z^I$, we have $z \in z^{\text{dom}} + \mathbb{R}^d_{\geq}$. Since every Pareto point lies in the dominated region, every facet can be marked as dominated. $\square$

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

Once a region of a Pareto front is known to be dominated, we want to avoid computing new Pareto points there. We introduce two strategies that make sure that new Pareto points are only placed where they contribute to the MOMIC Pareto front.

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 $z^{\mathrm{dom}} + \mathbb{R}^2_{\geq}$. If the cone cuts the patch on the f_1-boundary, the convex constraint $f_1(x) \leq p_1^{dom}$ is added to the patch optimization problem. If the cone cuts the patch on the f_2, boundary, then the constraint $f_2(x) \leq p_2^{dom}$ is added to the patch optimization problem. Since the ranges that are known to be dominated are excluded within the patch problem formulation, the patch approximation algorithm can not place new points there.

A third case can occur that cannot be represented by box constraints: the cone can intersect the patch in the middle. Then, one can cut the patch in half and create two new patch problems from it. Alternatively, the approach of creating perfectly approximated facets by adding fake Pareto points can be used. This approach can also be applied to problems with more than two criteria and is presented in the following section.

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 Pareto points that satisfy $f_1 \leq p_1^{dom}$, they implicitly remove those parts of the Pareto front that have steeper trade-offs with respect to f_1 than the gradient of the Pareto front at $f_1 = z_1^{\mathrm{dom}}$. Specifying trade-off bounds can be incorporated into the approximation process by modifying the domination cone. Then, different domination cones would have to be used for each patch approximation problem.

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

We present a method to avoid computing any more Pareto points in regions that are known to be dominated that is tailored to the Sandwiching algorithm. If a whole facet of the convex hull of the Pareto points is dominated by a different patch, we will ignore it by adding a so-called fake Pareto point. By adding these artificial points to the Pareto front, 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 Pareto point there.

The convex hull of Pareto points of a multiobjective convex optimization problem may contain dominated facets. This phenomenon does not occur in bicriteria problems.

Definition 3.4. *A* facet *is a full-dimensional face of a polytope. A facet of the convex hull of Pareto points is called* dominated *if there exist points in* *its interior that are dominated by some Pareto point. This means that the relative interior of the facet lies in the interior of $Z + \mathbb{R}^d_{\geq}$.*

Lemma 3.5. *If the outer normal of a facet of the convex hull of Pareto points of a convex multiobjective optimization problem has at least one positive component, the facet is dominated.*

Proof. Let Z be the set of computed Pareto points. From lemma 2 of Rennen et al. (2011), we know that the outer normal of a supporting hyperplane of a facet of conv $(Z + \mathbb{R}^d_{\geq})$ always has nonpositive components. Thus, a facet of conv Z with a normal with positive components can not be nondominated since $(Z)_N = \left(Z + \mathbb{R}^d_{\geq}\right)_N$. □

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

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

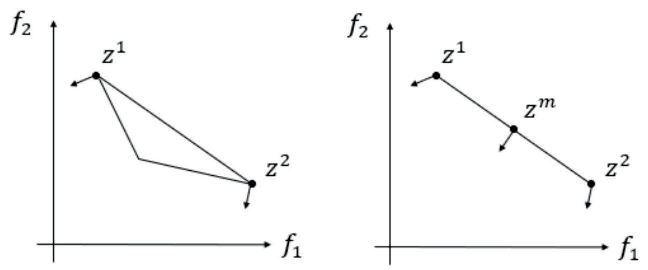


Figure 3: Let a facet be given by two Pareto points z^1 and z^2 and their normals which form the outer approximation. When adding the mean point z^m with the normal of the facet to the Pareto front, the outer and inner approximations coincide in this facet.

Lemma 3.6. *Let f be the vector of objectives of a convex multi-objective optimization problem. The solution $\tilde{z}$ of the weighted sum scalarization* $\min \lambda^T f$ *with weights $\lambda \geq 0$ has the tangent $\lambda^T z = \lambda^T \tilde{z},\ z \in \mathbb{R}^d$, which is orthogonal to λ.*

Proof. In objective space, $\lambda^T f(x) = c$, $c \in \mathbb{R}$ forms a hyperplane with normal λ. This hyperplane is shifted, i.e. the value of c is decreased, until c attains its minimal value $\tilde{c}$ in $\tilde{x}$. Then, the hyperplane is a tangent on $\tilde{x}$. The solution $\tilde{z} := f(\tilde{x})$ of the weighted sum problem lies on $\lambda^T f(\tilde{x}) = \lambda^T \tilde{z} = \tilde{c}$. Therefore, we obtain that $\lambda^T z = \lambda^T \tilde{z}$ is the tangent on $\tilde{z}$. □

Lemma 3.7. *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 Pareto front approximation, the outer and inner approximation of the facet will coincide, i.e. the facet has perfect approximation quality.*

Proof. The tangential hyperplane of a Pareto point calculated by the weighted sum scalarization with some weight will be orthogonal to the weight vector (lemma 3.6). Thus, the tangential hyperplane of z^m is orthogonal to the normal of the inner approximation facet, i.e. the tangential hyperplane H is parallel to the inner approximation facet. The outer approximation of the Pareto front will be updated by cutting it with $z^m + H$. Since the point z^m lies on the inner approximation facet, $z^m + H$ contains the inner approximation facet. □

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

Handling dominated facets We cannot add a fake Pareto point to a dominated facet as described above. By adding few additional points to the approximation, we can construct several nondominated facets from one dominated one. For each of these nondominated facets, fake Points can then be added to remove them from further refinement.

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 pareto points Z may contain dominated facets, the set $Z + \mathbb{R}^d_{\geq}$ only consists of nondominated facets. Therefore, the approach by Rennen et al. (2011) adds d additional points for every computed Pareto point that represent the domination cone. These additional points are defined as follows.

Definition 3.8 (Definition 7 of Rennen et al. (2011))**.** *Let an upper bound on the Pareto front $\tilde{y}^N$, e.g. a nadir point approximation (see Equation 2.15 of Ehrgott (2005)), be given. The standard domination cone $\mathbb{R}^d_{\geq}$ added to a Pareto point $z \in \mathcal{Y}$ can be represented by the points $d_1(z), \dots, d_d(z) \in \mathbb{R}^d$.*

The j-th point is defined as

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

But adding d of these points for every Pareto point produces a large amount of additional points. Many of these points are actually not necessary.

Lemma 3.9. *Let a dominated facet be defined by the points $Z := \{z^1, \ldots, z^d\}$. Let the facet have an outer normal with positive entries in directions $I \subset \{1, \ldots, d\}$. To construct nondominated facets, we only need to add points that represent the domination cone (see Definition 3.8) in directions I.*

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

We want to add points representing the domination cone to obtain facets with fully non-positive normals. When adding points $d_j(z)$ $\forall z \in Z$, 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 conv $(Z \cup \{d_i \forall i \in I\})$ are nondominated.

□

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 Pareto point z with normal w, the normal of $d_j(z)$ has entries $\min\{w_i, 0\}$, $i = 1, \ldots, d$.

Lemma 3.10. *The part of a patch Pareto front marked as ϵ_{dom}-dominated using fake Pareto points is actually ϵ_{dom}-dominated by some other patch.*

Proof. If for some patch Pareto point, an ϵ_{dom}-dominating point z^{dom} is found using (3.1) or (3.2), then fake Pareto points ensure that in those facets that fully lie inside of $z^{\text{dom}} + \mathbb{R}^d_{\geq}$, no new Pareto points are added. These facets form a subset of the cut of the patch with $z^{\text{dom}} + \mathbb{R}^d_{\geq}$. Every point $z \in z^{\text{dom}} + \mathbb{R}^d_{\geq}$ is ϵ_{dom}-dominated by z^{dom}. □

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

```
Determine all facets that lie inside of z^dom + R^d_≥.
for all of these facets do
    if the facet is dominated then
        Add points representing the domination (see Lemma 3.9)
    end if
    Add a fake Pareto point to every facet
end for
```

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 Pareto optimal 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)) Pareto fronts. 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 Pareto front, 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 MOMIC Pareto front to the decision maker in the patch navigation tool. The decision maker can the explore the trade-offs of the Pareto front and specify parts of the Pareto front 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 into the optimization problem using algorithm 3. When a patch Pareto point z has been removed by the restrictor, a new Pareto 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 Pareto points can be used when removing dominated parts of patches.

Algorithm 3 Process restrictors set in navigation step

Require: restrictor positions, all patch Pareto points
for every patch **do**
  for every restrictor **do**
    Add the constraint $f_i \leq \bar{z}$ to the patch optimization problem
    if there are patch points z with weight w fulfilling $z^i > \bar{z}$ **then**
      Remove the patch point from the approximation
      Compute a new Pareto point for this patch using weight w.
    end if
  end for
end for

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

Some algorithms that approximate Pareto fronts contain steps to remove dominated points that were computed during the course of the algorithm (e.g. Rennen et al. (2011), Cacchiani and DAmbrosio (2017)). Since the linear programs used in navigation make sure that we only navigate on the MOMIC Pareto front, we do not need to remove patch Pareto points that are dominated with respect to the MOMIC Pareto front.

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 figure 6 on a bicriteria case with five patches. Patches 1 and 2 partially contribute to the Pareto front, patch 3 is fully dominated and intersects patch 1 and 2, and patch 4 and 5 are fully dominated and far behind the Pareto front. We select method (3.2) to check for dominating points and remove dominated parts using fake Pareto points (algorithm 2).

Step 1 (figure 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 (figure 6b): The decision maker removed the shaded area from further refinement using restrictors. I_4 and I_5 are updated with new Pareto points (bold 'x').

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

Step 4 phase 2 (figure 6d): Patch 3 is identified as fully dominated by computing dominating points (bold 'x') using (3.2). Parts of patches 1 and

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 dominated by some patch (e.g. programs (3.2) or (3.1)),
a method for each patch that calculates points using weighted sum scalarization

Generate initial approximation for each patch: compute the extreme compromises and the weighted sum solution with equal weights

Present the initial approximation to the decision maker in a patch navigation tool (see section 3.4). The decision maker can exclude parts of the objective space that they are not interested in from further refinement using restrictors.

Process the restrictor positions using algorithm 3

Check the initial approximations for ϵ_{dom}-domination and remove dominated patch parts using algorithm 1.

while the stopping criterion is not met **do**

  Calculate a new patch Pareto point using Sandwiching in the patch that has the worst approximation quality

  Check whether the new point is ϵ-dominated by some other patch. If this is the case, mark this part of the patch Pareto front as dominated using algorithm 2.

  Determine the approximation quality of the updated patch and update the MOMIC approximation quality as the maximum of the patch qualities.

end while

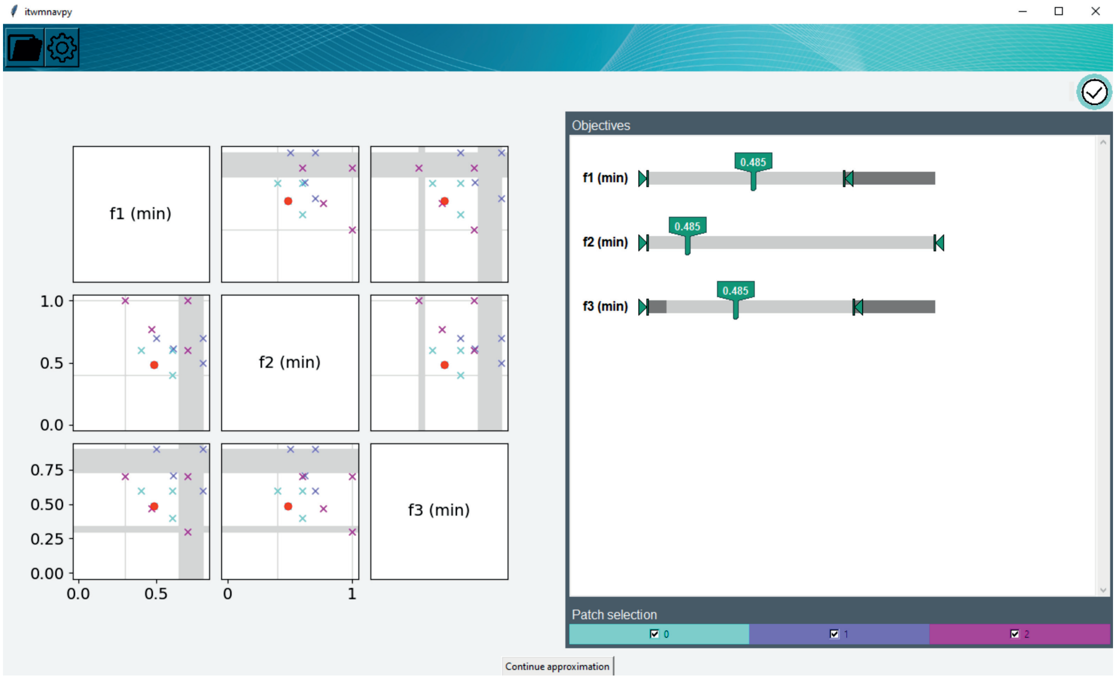


Figure 4: An initial approximation of a tri-criteria problem with three patches is presented to the decision maker in the patch navigation tool. The decision maker decided to restrict the objective space in the first and third objective (shaded).

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.

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

Step 6 (figure 6e): Add a new Pareto point $\bar{z}$ to patch 2.

Step 7 (figure 6f): We check $\bar{z}$ for domination. Two Pareto 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 Pareto point.

In the following chapter, 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 section 2.3), but also hold for some other metrics, e.g. the polyhedral gauge and the Hausdorff metric.

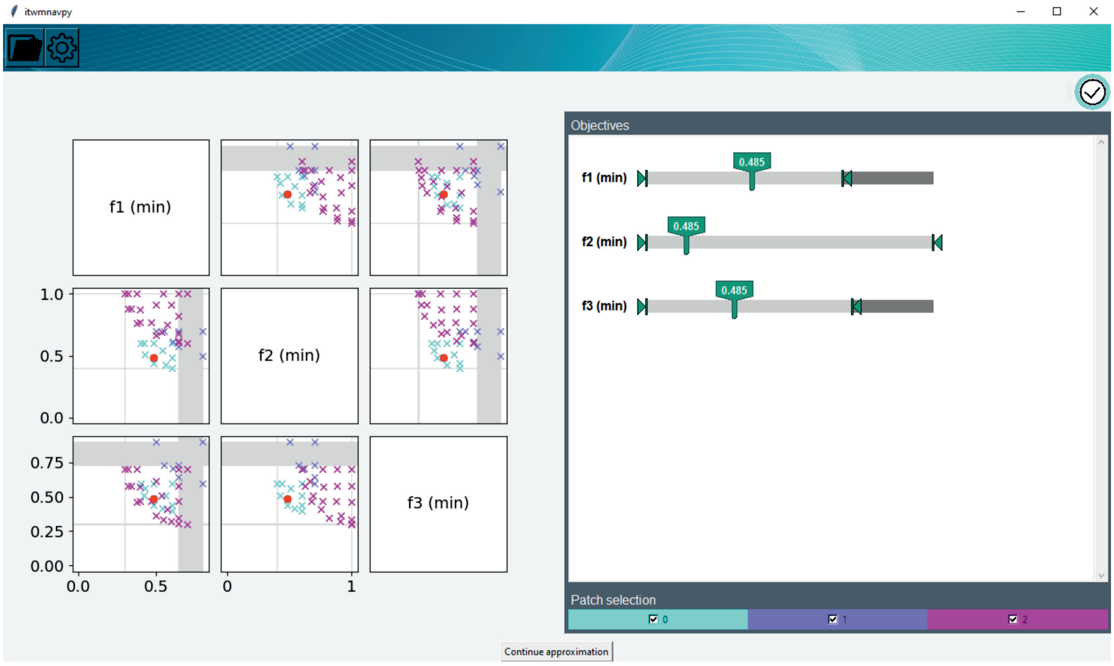


Figure 5: After incorporating the restrictor positions, the patch approximation is refined using the patch approximation algorithm. No new Pareto points have been computed in the restricted (shaded) areas. The decision maker can now explore the trade-offs of the finely approximated MOMIC Pareto front.

Theorem 4.1. *The algorithm converges, i.e.* $\lim_{n\to\infty} \delta^{\epsilon}(\mathbf{I}^n, \mathbf{O}^n) = 0$.

Proof. For every patch Pareto front, the Sandwiching algorithm converges (Theorem 4.2.22 of Lammel (2023)). Since the overall approximation quality of the patch Pareto front is the maximum of all patch approximation qualities and a new Pareto point is added to the patch that attains the worst quality, all patch Pareto fronts will eventually be approximated because we assume that there are only finitely many patches. Some parts of the Pareto front of the original patch problem will not be approximated finely since they are marked as dominated in the course of the algorithm. $\square$

In the worst case, no part of one patch dominates another patch. Then, the MOMIC Pareto front is just the union of the Pareto fronts 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 4.2 shows that the approximation error is reduced whithin one iteration in the order of $n^{1/(d-1)}$ with n the number of iterations. With additional regularity of the patch Pareto fronts, 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 4.2. *Let $\{(\mathbf{I}^n, \mathbf{O}^n)\}_{n=0,1,\ldots}$ be a sequence of inner and outer approximations generated by the patch approximation algorithm 4 for q patches. The set C^i denotes the Pareto front of patch i. If each patch Pareto front forms a convex compact set and is not a sphere, 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 4.3. *Let $\{(\mathbf{I}^n, \mathbf{O}^n)\}_{n=0,1,\ldots}$ be a sequence of inner and outer approximations generated by the patch approximation algorithm 4 for q patches. The set C^i is the Pareto front of patch i, I_0^i denotes the initial inner approximation of the patch. If each patch Pareto front forms a convex compact set with twice continuously differentiable boundary, 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

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

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 (2021). Additionally, we demonstrate that our algorithm performs well in higher-dimensional cases using a seven-criteria problem. Since we want to compute the entire Pareto front 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.

Comparing our algorithm to the HyPaD algorithm

From experimental data presented in Eichfelder and Warnow (2021), it appears that the HyPaD algorithm introduced in Eichfelder and Warnow (2021) is currently the best-performing patch algorithm that can be applied to problems of arbitrary objective dimension. The HyPaD algorithm solves MOMIC problems, focusing on methods to handle large numbers of possible integer assignments i.e. patches. Our algorithm, on the other hand, is designed for MOMIC problems where the integer aspect is not the dominant difficulty. These can be problems with only few patches. In the following, we apply our algorithm to the three examples discussed in Eichfelder and Warnow (2021) to demonstrate the abilities of our patch approximation algorithm and compare the performance to the HyPaD algorithm.

To compare the results we use the computation times given in Eichfelder and Warnow (2021). In Eichfelder and Warnow (2021), the HyPaD algorithm was implemented in MATLAB, the numerical tests were performed on a machine with 32 GB of RAM and a an Intel Core i9-10920X processor with a clock rate of 3.5 GHz and 12 cores.

The HyPaD algorithm approximates the Pareto front using boxes. As a quality criterion, the largest width of all boxes, is used. We will use the epsilon indicator as our quality criterion. In bicriteria problems, the enclosure computed by HyPaD consists of rectangles, the inner and outer approximation computed in the patch approximation algorithm consists of triangles where one side of the algorithm is the rectangle diagonal. The epsilon indicator based on the triangle cannot be directly compared to the smallest rectangle length used as a quality criterion in Eichfelder and Warnow (2021). To obtain more-or-less comparable results, when the HyPaD algorithm used a box width of ϵ as a stopping criterion, we will therefore target an epsilon indicator value of $\epsilon/2$.

We will determine whether a part of a patch is dominated using one search direction by solving (3.1). We will remove dominated parts using fake Pareto points.

Bicriteria problem with five patches

This bi-objective 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.

$$\begin{aligned} \min\ & (x_1 + x_3, x_2 + \exp(-x_3))^T \\ \text{s.t.}\ & x_1^2 + x_2^2 \leq 1 \\ & x_1, x_2 \in [-2, 2] \\ & x_3 \in \{-2, -1, 0, 1, 2\}. \end{aligned} \tag{5.1}$$

To apply our patch approximation algorithm, we create one optimization problem for every possible integer assignment. The HyPaD algorithm computes an enclosure of the Pareto front with maximal box width $\epsilon = 0.1$ within 1.34 seconds and for $\epsilon = 0.01$ within 6.09 seconds, 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 seconds and achieves quality 0.005 within 0.32 seconds which is 3.2 times the approximation time of the lower approximation quality.

Bicriteria problem with scalable number of continuous and integer variables

The next test case (example H1 of Eichfelder and Warnow (2021)) 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

$$\begin{aligned} \min\ & \begin{pmatrix} \sum_{i=1}^{n/2} x_i + \sum_{i=n+1}^{n+m/2} x_i^2 - \sum_{i=n+m/2+1}^{n+m} x_i \\ \sum_{i=n/2+1}^{n} x_i - \sum_{i=n+1}^{n+m/2} x_i + \sum_{i=n+m/2+1}^{n+m} x_i^2 \end{pmatrix} \\ \text{s.t.}\ & \sum_{i=1}^{n} x_i^2 \leq 1, \\ & x_1, \dots, x_n \in [-2, 2], \\ & x_{n+1}, \dots, x_m \in \{-2, -1, 0, 1, 2\}. \end{aligned} \tag{5.2}$$

The number of patches is given by 5^m. We perform tests using 2 and 4 integer variables, i.e. 25 and 625 patches.

For two continuous and two discrete variables, i.e. 25 patches, the computed patch Pareto points are shown in figure 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 Pareto 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 (2021)) for 25 and 625 patches are shown in figure 9. As expected, our patch approximation algorithm performs significantly better in cases with small amounts of 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 5.2 with two continuous and six discrete variables, i.e. $5^6 = 15625$ patches to an epsilon indicator quality of 0.05 within 75 seconds. The HyPaD algorithm can solve the even larger problem with two continuous and eight discrete variables in 25.70 seconds.

Tricriteria problem

The third test case that we use to compare our algorithm to the HyPaD algorithm from Eichfelder and Warnow (2021) is a tri-objective test instance, originally presented as T5 in De Santis et al. (2020). Its optimization problem is given by

$$\begin{aligned} \min\ & \begin{pmatrix} x_1 + x_4 \\ x_2 - x_4 \\ x_3 + x_4^2 \end{pmatrix} \\ \text{s.t. } & \sum_{i=1}^{3} x_i^2 \leq 1, \\ & x_1, x_2, x_3 \in [-2, 2], \\ & x_4 \in \{-2, -1, 0, 1, 2\}. \end{aligned} \tag{5.3}$$

Thus, there are five patches. The approximated Pareto front is shown in figure 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 dominated parts never form a full facet so that no parts of the patch Pareto front can be removed from further refinement by adding fake Pareto points.

The HyPaD algorithm computes an approximation with maximal box width of 0.1 within about 9 seconds (see Eichfelder and Warnow (2021)). The patch approximation algorithm presented in this article approximates the same Pareto front to an epsilon indicator quality of 0.05 within 0.62 seconds.

An example with seven criteria

To illustrate that our algorithm can approximate the Pareto fronts of problems with an arbitrary number of criteria by applying it to a problem with seven objectives.

The Pareto front 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 $c^1 = (1, \ldots, 1)$ 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

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

Using the patch approximation algorithm (algorithm 4) using problem (3.1) as the criterion to check for dominating patches, this Pareto front can be approximated to an epsilon indicator value of less than 0.1 using 106 Pareto points within 14.5 seconds. and to an epsilon indicator value of 0.05 using 325 Pareto points within 765 seconds.

6 Conclusion

We introduced a novel algorithm that approximates Pareto fronts of multiobjective mixed-integer convex problems. To the best of our knowledge, it is the first algorithm that constructs an inner and outer approximation of the front exploiting the convexity of the patches and that is applicable to 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 introduce methods that determine parts of patches that are dominated by other patches. Then, we remove these patch parts from further refinement by adding artificial Pareto points. An interactive step allows a decision maker to exclude parts of the Pareto front from further refinement after a coarse approximation of the Pareto front 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 Pareto fronts with small and medium (up to a few hundred) numbers of patches faster than the HyPaD algorithm introduced by Eichfelder and Warnow (2021), which then outperforms our algorithm for larger numbers of patches. To illustrate that our algorithm can also approximate the Pareto fronts 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 mxied-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. Typically, the number of technologies that are compared is small, e.g. less than 10, and the computational effort of computing a Pareto point is high.

The technologies can then be interpreted as patches and their Pareto front can be computed using our patch approximation algorithm.

Acknowledgments 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.

Declarations

Funding The research leading to these results received funding from DFG under Grant Agreement No 441208898.

Competing interests 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.

Availability of data and materials Not applicable.

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

References

Adelgren, N. and Gupte, A. (2022). Branch-and-bound for biobjective mixed-integer linear programming. *INFORMS Journal on Computing*, 34(2):909–933.

Allmendinger, R., Ehrgott, M., Gandibleux, X., Geiger, M. J., Klamroth, K., and Luque, M. (2017). Navigation in multiobjective optimization methods. *Journal of Multi-Criteria Decision Analysis*, 24(1-2):57–70.

Bokrantz, R. and Forsgren, A. (2013). An algorithm for approximating convex pareto surfaces based on dual techniques. *INFORMS Journal on Computing*, 25(2):377–393.

Bortz, M., Burger, J., Asprion, N., Blagov, S., Böttcher, R., Nowak, U., Scheithauer, A., Welke, R., Küfer, K.-H., and Hasse, H. (2014). Multi-criteria optimization in chemical process design and decision support by navigation on pareto sets. *Computers & chemical engineering*, 60:354–363.

Burachik, R. S., Kaya, C. Y., and Rizvi, M. (2017). A new scalarization technique and new algorithms to generate pareto fronts. *SIAM Journal on Optimization*, 27(2):1010–1034.

Burachik, R. S., Kaya, C. Y., and Rizvi, M. M. (2022). Algorithms for generating pareto fronts of multi-objective integer and mixed-integer programming problems. *Engineering Optimization*, 54(8):1413–1425.

Cabrera-Guerrero, G., Ehrgott, M., Mason, A. J., and Raith, A. (2021). Bi-objective optimisation over a set of convex sub-problems. *Annals of Operations Research*, pages 1–26.

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

Ceyhan, G., Köksalan, M., and Lokman, B. (2019). Finding a representative nondominated set for multi-objective mixed integer programs. *European Journal of Operational Research*, 272(1):61–77.

Collicott, C., Bonacker, E., Lammel, I., Teichert, K., Walzcak, M., and Süss, P. (2021). Interactive navigation of multiple convex patches. *Journal of Multi-Criteria Decision Analysis*, 28(5-6):311–321.

Craft, D. L., Halabi, T. F., Shih, H. A., and Bortfeld, T. R. (2006). Approximating convex pareto surfaces in multiobjective radiotherapy planning. *Medical physics*, 33(9):3399–3407.

De Santis, M., Eichfelder, G., Niebling, J., and Rocktäschel, S. (2020). Solving multiobjective mixed integer convex optimization problems. *SIAM Journal on Optimization*, 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., and Weißing, B. (2021). A benson-type algorithm for bounded convex vector optimization problems with vertex selection. *Optimization Methods and Software*, pages 1–21.

Ehrgott, M. (2005). *Multicriteria optimization*, volume 491. Springer Science & Business Media.

Ehrgott, M., Shao, L., and Schöbel, A. (2011). An approximation algorithm for convex multi-objective programming problems. *Journal of Global Optimization*, 50(3):397–416.

Eichfelder, G., Stein, O., and Warnow, L. (2022). A deterministic solver for multiobjective mixed-integer convex and nonconvex optimization. Preprint at `https://optimization-online.org/?p=18696`.

Eichfelder, G. and Warnow, L. (2021). A hybrid patch decomposition approach to compute an enclosure for multi-objective mixed-integer convex optimization problems. Preprint at `https://optimization-online.org/?p=17282`.

Eskelinen, P., Miettinen, K., Klamroth, K., and Hakanen, J. (2010). Pareto navigator for interactive nonlinear multiobjective optimization. *OR spectrum*, 32:211–227.

Halffmann, P., Schäfer, L. E., Dächert, K., Klamroth, K., and Ruzika, S. (2022). Exact algorithms for multiobjective linear optimization problems with integer variables: A state of the art survey. *Journal of Multi-Criteria Decision Analysis*, 29(5-6):341–363.

Hartikainen, M., Miettinen, K., and Klamroth, K. (2019). Interactive nonconvex pareto navigator for multiobjective optimization. *European Journal of Operational Research*, 275(1):238–251.

Klamroth, K., Tind, J., and Wiecek, M. M. (2003). Unbiased approximation in multicriteria optimization. *Mathematical Methods of Operations Research*, 56(3):413–437.

Lammel, I. (2023). *Approximation of Pareto Surfaces in Multi-Criteria Optimization*. PhD thesis, RPTU Kaiserslautern.

Löhne, A., Rudloff, B., and Ulus, F. (2014). Primal and dual approximation algorithms for convex vector optimization problems. *Journal of Global Optimization*, 60(4):713–736.

Monz, M., Küfer, K.-H., Bortfeld, T. R., and Thieke, C. (2008). Pareto navigationalgorithmic foundation of interactive multi-criteria IMRT planning. *Physics in Medicine & Biology*, 53(4):985.

Nowak, D., Johnson, T., Mark, A., Ireholm, C., Pezzotti, F., Erhardsson, L., Ståhlberg, D., Edelvik, F., and Küfer, K.-H. (2021). Multicriteria optimization of an oven with a novel ε-constraint-based sandwiching method. *Journal of Heat Transfer*, 143(1):012101.

Nowak, D. and 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., and Bortz, M. (2022). Rather the rule than the exception: Non-convex pareto sets and their navigation in distillation processes. *Chemie Ingenieur Technik*.

Rennen, G., van Dam, E. R., and Hertog, D. d. (2011). Enhancement of sandwich algorithms for approximating higher-dimensional convex pareto sets. *INFORMS Journal on Computing*, 23(4):493–517.

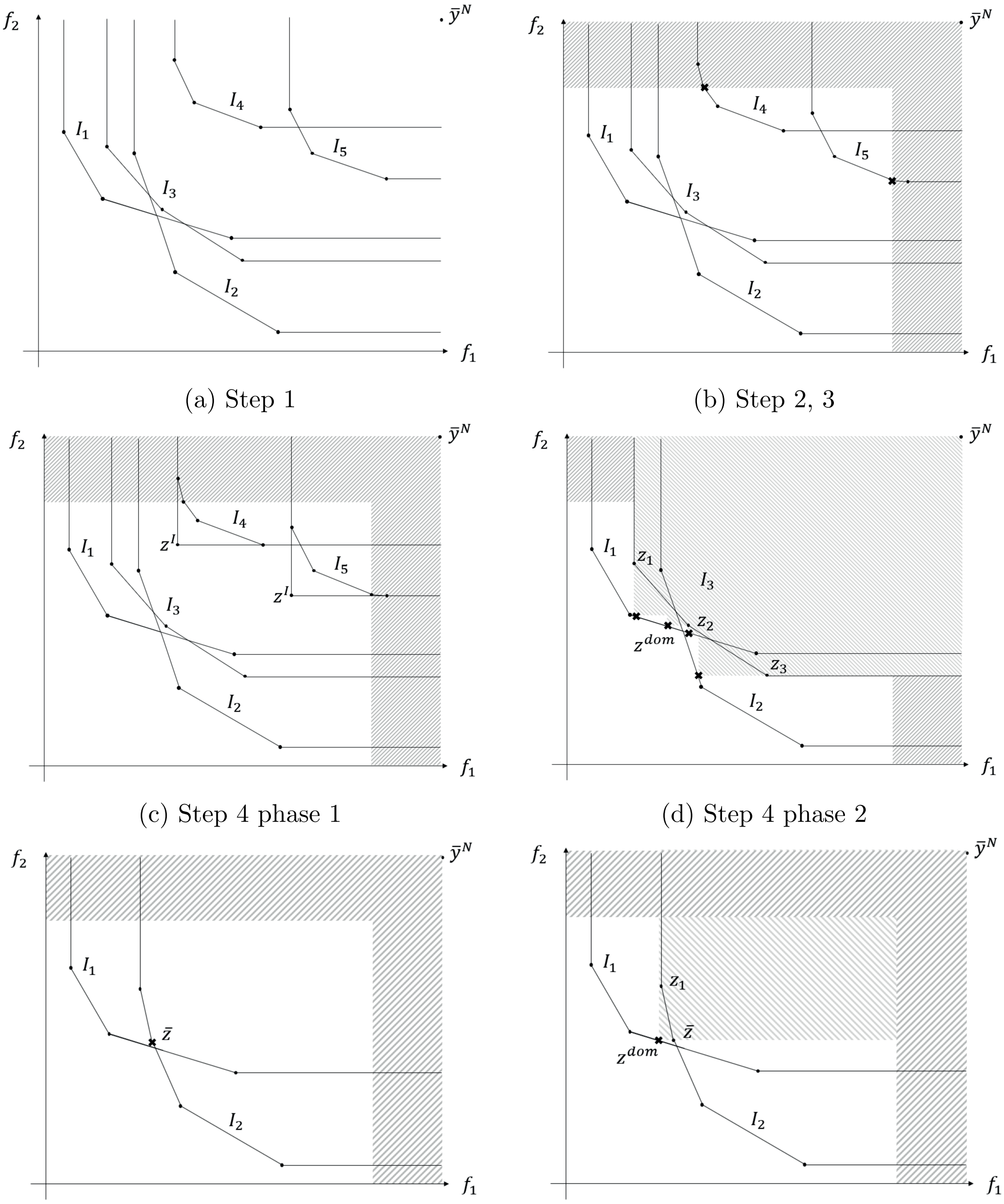
Ruzika, S. and Wiecek, M. M. (2005). Approximation methods in multiobjective programming. *Journal of optimization theory and applications*, 126(3):473–501.

Serna, J., Monz, M., Küfer, K.-H., and Thieke, C. (2009). Trade-off bounds for the pareto surface approximation in multi-criteria IMRT planning. *Physics in Medicine & Biology*, 54(20):6299.

Serna Hernandez, J. I. (2011). *Multi-objective optimization in Mixed Integer Problems with application to the Beam Selection Optimization Problem in IMRT*. PhD thesis, TU Kaiserslautern.

Sss, P., Foltin, G., Heidgen, M., Hajnal, D., Diaz, J., Schultze, H., Gattermayer, J., and Lehner, S. (2022). Chapter 12 - product development in a multicriteria context. In Bortz, M. and Asprion, N., editors, *Simulation and Optimization in Process Engineering*, pages 321–338. Elsevier.

Teichert, K., Süss, P., Serna, J., Monz, M., Küfer, K., and Thieke, C. (2011). Comparative analysis of pareto surfaces in multi-criteria IMRT planning. *Physics in Medicine & Biology*, 56(12):3669.



(a) Step 1 (b) Step 2, 3

(c) Step 4 phase 1 (d) Step 4 phase 2

(e) Step 6 (f) Step 7

Figure 6: Illustration of the patch approximation algorithm (algorithm 4) in a bicriteria example with five patches

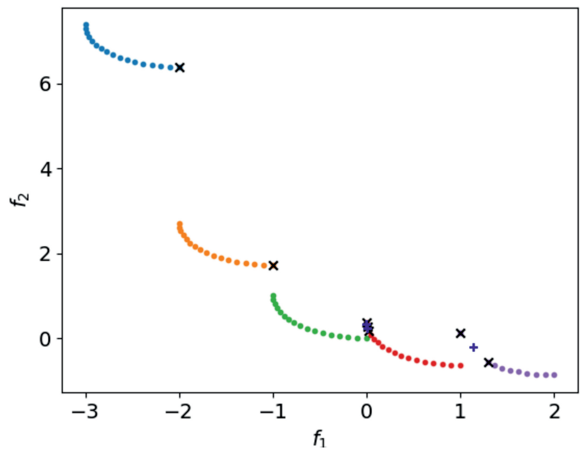


Figure 7: Pareto front of example 5.1 generated by our patch approximation algorithm. Dominated points are marked with an 'x', added fake Pareto points are marked with a '+'. Those parts of patches that contribute to the Pareto front are coloured.

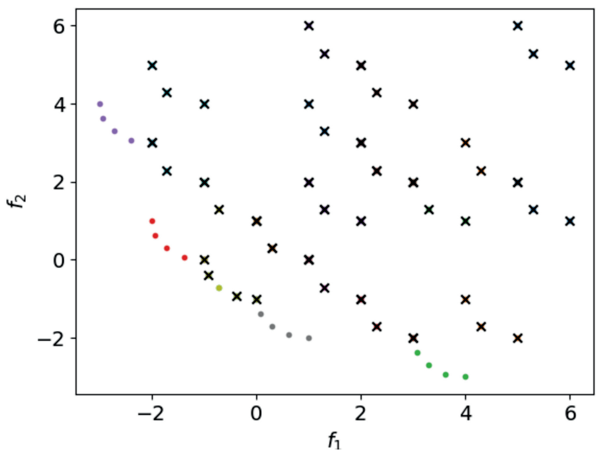
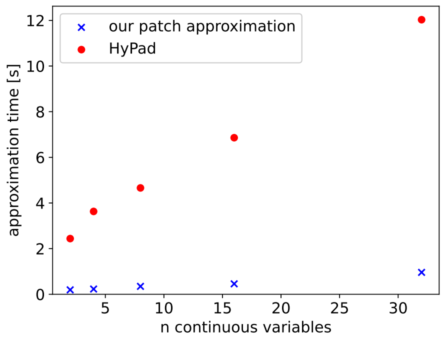
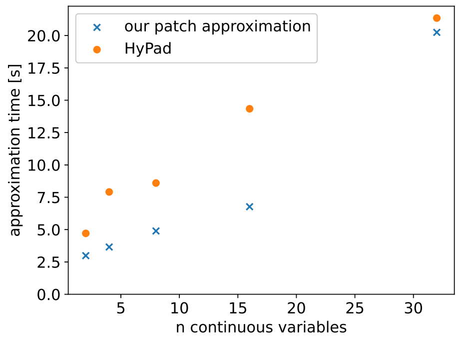


Figure 8: Pareto front of example 5.2 with two continuous and two discrete variables, e.g. 25 patches, generated by our patch approximation algorithm. Dominated points are marked with an 'x'. Those parts of patches that contribute to the Pareto front are coloured.



(a) Approximation times of our patch algorithm and the HyPaD algorithm for example 5.2 with 25 patches



(b) Approximation times of our patch algorithm and the HyPaD algorithm for example 5.2 with 625 patches

Figure 9: Comparison of the approximation time of the Pareto front of example 5.2, with 25 and 625 patches.

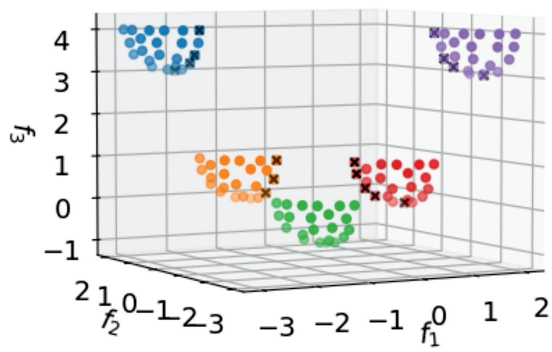


Figure 10: Approximated Pareto front of example 5.3 with three objectives and five patches, approximated to an epsilon indicator quality of 0.05.