Author(s): Hartikainen, Markus; Miettinen, Kaisa; Klamroth, Kathrin

Title: Interactive Nonconvex Pareto Navigator for Multiobjective Optimization

Year: 2019

Version: Accepted version (Final draft)

Interactive Nonconvex Pareto Navigator for Multiobjective Optimization

Markus Hartikainen, Kaisa Miettinen, Kathrin Klamroth

PII: S0377-2217(18)30979-2
DOI: https://doi.org/10.1016/j.ejor.2018.11.038
Reference: EOR 15487

To appear in: *European Journal of Operational Research*

Received date: 12 June 2015
Revised date: 30 October 2018
Accepted date: 15 November 2018

Please cite this article as: Markus Hartikainen, Kaisa Miettinen, Kathrin Klamroth, Interactive Nonconvex Pareto Navigator for Multiobjective Optimization, *European Journal of Operational Research* (2018), doi: https://doi.org/10.1016/j.ejor.2018.11.038

Highlights

- Characterizing navigation methods via their desirable properties
- New cones to facilitate navigation on nonconvex Pareto fronts
- Reflecting trade-off rates on navigation using the cones
- Introducing the new Nonconvex Pareto Navigator method
- The new method uses cones and satisfies the desirable properties

Interactive Nonconvex Pareto Navigator for Multiobjective Optimization

Markus Hartikainen*, Kaisa Miettinen

University of Jyvaskyla, Faculty of Information Technology, P.O. BOX 35 (Agora), FI-40014 University of Jyvaskyla, Finland

Kathrin Klamroth

School of Mathematics and Natural Sciences, University of Wuppertal, Gaußstr. 20, 42119 Wuppertal, Germany

Abstract

We introduce a new interactive multiobjective optimization method operating in the objective space called Nonconvex Pareto Navigator. It extends the Pareto Navigator method for nonconvex problems. An approximation of the Pareto optimal front in the objective space is first generated with the PAINT method using a relatively small set of Pareto optimal outcomes that is assumed to be given or computed prior to the interaction with the decision maker. The decision maker can then navigate on the approximation and direct the search for interesting regions in the objective space. In this way, the decision maker can conveniently learn about the interdependencies between the conflicting objectives and possibly adjust one's preferences. To facilitate the navigation, we introduce special cones that enable extrapolation beyond the given Pareto optimal outcomes. Besides handling nonconvexity, the new method contains new options for directing the navigation that have been inspired by the classification-based interactive NIMBUS method. The Nonconvex Pareto Navigator method is especially well-suited for computationally expensive problems, because the navigation on the approximation is computationally inexpensive. We demonstrate the method with an example. Besides proposing the new method, we characterize interactive navigation based

*The corresponding author, email *hartikainen.markus@gmail.com*

Email addresses: kaisa.miettinen@jyu.fi (Kaisa Miettinen), klamroth@math.uni-wuppertal.de (Kathrin Klamroth)

methods in general and discuss desirable properties of navigation methods overall and in particular with respect to Nonconvex Pareto Navigator.

Keywords: Multiple objective programming, interactive multiobjective optimization, navigation, nonconvex problems, Pareto optimality

1. Introduction

We study whether and how navigation methods developed for multiobjective optimization can be extended from linear and convex problems to nonconvex ones and what theoretical considerations this necessitates. Multiobjective optimization refers to optimizing multiple conflicting objectives at the same time, see, e.g., [24, 37, 39] for more introduction in this topic. Multiobjective optimization problems exist in many application areas, because optimization involves balancing between various needs like cost and quality. Typically, multiobjective optimization problems have multiple Pareto optimal solutions, where none of the objectives can be improved without impairing at least one of the others (see [32] for an original reference). However, typically one of the Pareto optimal solutions needs to be chosen for implementation. Thus, there is a need for preference information concerning the conflicting objectives and it is usually assumed that there exists a decision maker (DM) who can give this preference information. The ultimate aim of multiobjective optimization is to help the DM in finding a solution that is preferable to him/her with as little cognitive load as possible.

One can classify multiobjective optimization methods (see, e.g. [14, 24]) based on when the DM expresses his/her preferences. If no preferences are available, the method is called a no-preference method. In a priori methods, the DM first specifies preferences and then a Pareto optimal solution is found with respect to them, while, in a posteriori methods, many Pareto optimal solutions are generated and the DM is expected to choose one among them. Finally, interactive methods employ an iterative procedure and allow the DM to correct his/her preferences and, also, enable the DM to learn about the problem, see, e.g., [25] for a recent review. Interactive methods have in many instances been seen as a practically efficient approach because they allow the DM to gain more insight about the problem while solving it without introducing too much cognitive load at a time. In other words, the DM can consider only those Pareto optimal solutions that have been generated based on his/her preferences and learn about the feasibility of the preferences. For these reasons, we focus on interactive methods.

One type of interactive methods is navigation-based methods that were pioneered by [19] with the Pareto Race method. Pareto Race was originally developed for multiobjective linear problems and has inspired many further developments, see, e.g., [16, 17, 18, 21] for linear or quadratic problems, [3, 28, 29] for general convex problems, and [23] for a first application to nonconvex problems. See also [2] for a brief survey of navigation based and related methods.

Navigation-based methods dynamically generate and visualize Pareto optimal outcomes in the objective space and this is directed by a DM so that (s)he can smoothly move on the Pareto optimal front (i.e., the image of the Pareto optimal set in the objective space). Thanks to the navigation, the DM can learn about the interdependencies among the objective functions. The idea is to support the DM in the identification of a most preferred Pareto optimal outcome. In what follows, we refer to such methods as *navigation methods*. For example, the DM can express preference information to specify a direction along which Pareto optimal solutions are generated. One can also connect the decision space in the consideration.

Unfortunately, many real-life problems are computationally expensive (see, e.g., [8, 13]), i.e., it takes a long time to compute the values of the objective and/or constraint functions for given variable values. For these problems, it takes a long time to find new Pareto optimal outcomes. This implies a complication with navigation methods, because the DM has to wait while new Pareto optimal outcomes are generated according to his/her updated preferences in each iteration. This may lead to the unwillingness of the DM in exploring different outcomes and the DM may end up with less than preferable outcomes.

For computationally expensive problems, surrogate representations and approximations of the Pareto optimal front can be constructed to facilitate navigation methods. It is important to note that the DM is not involved in the construction of the approximation which may take time. An approximation that is computationally inexpensive to optimize allows the DM to avoid waiting times in navigation. Examples of such approaches for convex problems are given in [3, 28, 29]. For example, the Pareto Navigator method [3] allows the DM to continuously move on a convex hull approximation of the Pareto optimal front by providing reference points. Like in [21], the reference point is interpreted to set a reference direction and the DM can move in the reference direction as long as he/she wants. A Two-Step Framework for general nonconvex problems is introduced in a report [23]. In this method, a specific sampling method (aiming at an approximately equidistant representation of the Pareto optimal front) is first applied and then a Delaunay triangulation (as in [10]) of a lower dimension is created using linear combinations of subsets of samples. This is used to approximate the Pareto optimal front with simplices. Once this is done, an approach of finding preferable solutions on the relevant simplices of that approximation is introduced that uses preferences provided by the DM (changing the value of one objective function, fixing objective function values and/or setting a constraint or giving corresponding preference for variables). The implementation is based on solving multiple linear problems, each restricted to one of the simplices. The simplex with the best solution is presented to the DM who can then choose to continue the search, and/or to verify the result by computing a projection onto the real Pareto optimal front using the scalarization method of [33].

Recently, many neurobiological and behavioral experiments have shown that decision making tasks include a learning phase and a decision phase (e.g., [44]), which often also are inherent in computational models of decision making (e.g., [35]). This is in line with findings in multiobjective optimization e.g., in [20, 27]. In the learning phase in multiobjective optimization, the DM explores different solutions in order to find the most interesting region of the Pareto optimal set or the Pareto optimal front. The final solution is then to be identified in the decision phase. Navigation methods support the learning phase, in particular. Instead of jumping to the solution that best matches with the preferences of the DM, navigation methods continuously show to the DM how objective function values evolve when moving from the current solution along the reference direction. (S)he can stop the movement at any time and provide new preference information. This is in line with the brain's mechanisms of movement selection that incorporates all the elements of a deliberate decision, i.e., decision making designed to achieve goals in a dynamic environment [7]).

By following in real time how objective function values evolve, the DM can gain insight of the interdependencies involved as well as understand the feasibility of one's own preferences and modify the preferences (i.e., change one's mind) if needed. One can also return to any of the previously seen solutions and re-specify preferences there in order to navigate to a new direction. Thus, the speed of finding the final solution is not an end in itself but the confidence of the DM that (s)he has found a satisfactory solution and can justify it. (Once the most preferred area is found in the learning phase, the DM may continue with any other interactive multiobjective optimization method to fine-tune and find the final preferred outcome).

In this paper, we extend the Pareto Navigator method of [3] to nonconvex, computationally expensive multiobjective optimization problems and demonstrate that the positive properties of navigation methods can be extended for nonconvex problems. The paper includes a discussion of the technical difficulties that are induced by nonconvex and possibly disconnected Pareto optimal fronts and suggests a general approach to overcome these difficulties. The new method is called Nonconvex Pareto Navigator. The Nonconvex Pareto Navigator method has some similarities with the Two-Step Framework introduced in [23]. Both methods are aimed at nonconvex and computationally expensive problems. The main similarity is that both methods create a piecewise linear approximation of the Pareto optimal front using samples of Pareto optimal outcomes, computed prior to the involvement of the DM. However, the approximations differ significantly with respect to how difficulties occuring in nonconvex problems (like, for example, disconnected Pareto optimal fronts and partial dominance in the approximation) are handled. This also leads to significant differences in user experience, and what general properties the navigation can be guaranteed to satisfy. Moreover, in the Nonconvex Pareto Navigator method the navigation is realized using moving reference points in combination with the achievement scalarizing problem of [43], while [23] suggests to use bounds and fixing values or changing the value of an objective function in combination with the direction method of Pascoletti and Serafini [33]. We also provide a rigorous analysis of the properties of the Nonconvex Pareto Navigator method.

Advancing navigation methods for nonconvex problems is important as many real problems actually are nonconvex. For example, simulation-based multiobjective optimization problems, where function evaluations are derived from simulation and modelling tools which can be of black-box nature, cannot be assumed to be convex in general, see, e.g., [40].

The contribution of this paper is three-fold: First, we formally characterize navigation methods by proposing their desirable properties from both technical and user experience perspectives, and hence give a more formal definition of navigation methods compared to what has been presented in the literature so far, c.f. [2]. Because nonconvex problems pose challenges not present in convex problems (like, for example, potentially disconnected Pareto optimal fronts), second, we introduce a new type of cones to facilitate navigation on nonconvex and disconnected Pareto optimal fronts. We refer to these extrapolating cones as e-cones. The use of the e-cones is different from what we have found in the multiobjective optimization literature. The e-cones also reflect trade-off rates in the Pareto optimal front. Finally, we introduce the new Nonconvex Pareto Navigator method utilizing the approximation method PAINT [10, 11] and the e-cones information. Inspired by the interactive NIMBUS method [26], the new method employs both reference points or bounds not to be exceeded as preference information. As concluded in the literature survey in [40], there are not many interactive methods for computationally expensive problems and the Nonconvex Pareto Navigator method has been directed for such problems whenever the DM wishes to express preference information in the form or aspiration levels and/or bounds.

In what follows, we summarize the basic concepts of multiobjective optimization used in Section 2. Then, we characterize navigation methods for (computationally expensive) multiobjective optimization problems and their desirable properties in Section 3. In Section 4, we introduce the new Nonconvex Pareto Navigator and in Section 5, we evaluate how well the new method performs with respect to the properties of Section 3. Section 6 discusses the connections between the e-cones introduced in Section 4 and trade-off rates. Nonconvex Pareto Navigator is demonstrated with an example in wastewater treatment plant operation in Section 7. Finally, we conclude in Section 8.

2. Preliminaries

We deal with multiobjective optimization problems of the form

$$\begin{array}{ll} \min & \mathbf{f}(\mathbf{x}) := (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T \\ \text{s.t.} & \mathbf{x} \in S, \end{array} \tag{1}$$

where $k \geq 2$ is the number of objective functions $f_i : S \to \mathbb{R}$ $(i = 1, \ldots, k)$ and the set $S \subseteq \mathbb{R}^n$ is the feasible set formed by constraint functions. A decision vector $\mathbf{x} \in S$ is called a feasible solution and the vector $\mathbf{f}(\mathbf{x}) \in \mathbf{f}(S)$ is called an outcome or an objective vector. The Euclidean spaces $\mathbb{R}^n$ and $\mathbb{R}^k$ are called the decision and objective space, respectively.

A solution $\mathbf{x}^1 \in S$ and the corresponding outcome $\mathbf{f}(\mathbf{x}^1)$ are said to dominate another solution $\mathbf{x}^2 \in S$ and the corresponding $\mathbf{f}(\mathbf{x}^2)$, if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for all $i = 1, \ldots, k$ and $f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2)$ for some $j \in \{1, \ldots, k\}$. A solution $\mathbf{x}^1 \in S$ and the corresponding $\mathbf{f}(\mathbf{x}^1)$ are called Pareto optimal (PO), if there does not exist another solution $\mathbf{x}^2 \in S$ (and $\mathbf{f}(\mathbf{x}^2)$) that dominates it. A solution $\mathbf{x}^1 \in S$ is said to be weakly PO if there does not exist a solution

$\mathbf{x}^2 \in S$ such that $f_i(\mathbf{x}^2) < f_i(\mathbf{x}^1)$ for all $i = 1, \ldots, k$. The set of PO outcomes in the objective space is called the Pareto optimal front, or for short PO front. Analogously, the nondominated subset of a set $A \subseteq \mathbb{R}^k$ is given by

$$\mathrm{ND}(A) := \left\{\mathbf{a} \in A : \quad \text{there does not exist } \mathbf{b} \in A \text{ that dominates } \mathbf{a} \right\}$$

and is called the PO front of A.

Furthermore, we denote the nonnegative orthant as $\mathbb{R}^k_+ := \{\mathbf{x} \in \mathbb{R}^k : x_i \geq 0 \text{ for all } i = 1, \ldots, k\}$. The boundary of a set $A \subseteq \mathbb{R}^m$ (where $m \in \mathbb{N}$) is denoted by $\partial(A)$ and the sum of sets $A, B \subseteq \mathbb{R}^m$ (where $m \in \mathbb{N}$) is defined as

$$A + B = \{\mathbf{a} + \mathbf{b} : \mathbf{a} \in A, \mathbf{b} \in B\}.$$

In this paper, we assume that the multiobjective optimization problem is computationally expensive, i.e., it takes a relatively long time to compute the values of the objective or the constraint functions for a given solution $\mathbf{x} \in \mathbb{R}^n$. Computationally expensive problems occur, e.g., when the functions to be evaluated are only available through computer simulations (as, e.g., in [8, 13]) or real experiments.

There exist many interactive methods allowing the DM to learn about the problem and find preferred outcomes and these methods differ, e.g., in the type of preference information obtained from the DM (see, e.g., [25] for a recent survey). In this paper, we use mostly reference point based methods. A reference point $\mathbf{z}^{\mathrm{ref}} \in \mathbb{R}^k$ contains desirable objective function values for the DM. Its components are also often called aspiration levels. A reference point can be used to generate preferred PO outcomes in many ways. In this paper, we mostly use the augmented version of the achievement scalarizing problem of Wierzbicki [43] that can be formulated for problem (1) as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \max_{i=1,\ldots,k} \left[\frac{f_i(\mathbf{x}) - z_i^{\mathrm{ref}}}{z_i^{\mathrm{nad}} - z_i^{\mathrm{utopia}}} \right] + \rho \sum_{i=1}^{k} \frac{f_i(\mathbf{x})}{z_i^{\mathrm{nad}} - z_i^{\mathrm{utopia}}} \\ \text{s.t.} \quad & \mathbf{x} \in S, \end{aligned} \tag{2}$$

where the term $\rho \sum_{i=1}^{k} \frac{f_i(\mathbf{x})}{z_i^{\mathrm{nad}} - z_i^{\mathrm{utopia}}}$ is called an augmentation term, $\rho > 0$ is a small augmentation constant, and

$$\mathbf{z}^{\mathrm{nad}} := \left(\max_{\mathbf{x} \in S \text{ is PO}} f_1(\mathbf{x}), \ldots, \max_{\mathbf{x} \in S \text{ is PO}} f_k(\mathbf{x}) \right)^T \text{ and}$$

$$\mathbf{z}^{\mathrm{utopia}} := \left(\min_{\mathbf{x}\in S} f_1(\mathbf{x}) - \gamma, \ldots, \min_{\mathbf{x}\in S} f_k(\mathbf{x}) - \gamma \right)^T$$

are, respectively, the nadir objective vector and the utopian objective vector of problem (1), where $\gamma > 0$ is a small constant. They represent ranges of objective functions in the PO front. An optimal solution to (2) is PO [43]. If the augmentation term is removed from (2), then the optimal solution is weakly PO. For further information on the optimality results and estimating the nadir objective vector, see [24] and references therein. In the context of this paper, estimations of the nadir objective vector $\mathbf{z}^{\mathrm{nad}}$ can be computed, for example, based on an upper bound on acceptable objective values or on a precomputed finite set of PO outcomes.

3. Navigation Methods for Computationally Expensive Problems

In this section, we first characterize the core of navigation methods for computationally expensive problems in terms of modules. Then, we introduce desirable properties of these methods to formally characterize them.

As mentioned in the introduction, to be able to navigate efficiently in computationally expensive problems, approximations of the PO front are indispensable. For this reason, we consider two types of outcomes: approximate outcomes and outcomes of the original computationally expensive problems. In what follows, for brevity, we refer to the latter type simply as solutions and outcomes (in decision and objective spaces, respectively). In addition, we assume to have available an approximation of the PO front and it is created based on a set of given PO outcomes $P \subseteq \mathrm{ND}(\mathbf{f}(S))$.

3.1. General Modules

Navigation methods for computationally expensive problems can be characterized with a clear modular structure, which can be named as a

navigation set, which is a set in the objective space, where the DM may navigate and find preferable values for the objectives,

navigation control that allows the DM to set the direction and the speed of navigation (i.e., movement), and

projection that allows the DM to find the closest PO outcome to any point in the navigation set that he/she desires.

Note that the first two modules are also valid for navigation methods applied in computationally inexpensive problems, where the navigation set is the PO front (instead of an approximation). Thus, the projection is needed only in computationally expensive problems where navigation takes place in the approximation.

In this paper, we assume that the navigation set is an approximation of the PO front and it is created based on the set P. Furthermore, the navigation control keeps solving a single objective optimization problem involving a reference point that moves on the ray originating from the last PO outcome that was shown to the DM and aims at the reference point given by the DM.

In what follows, the reference point that is given by the DM is called merely a reference point and the reference point that is a technical element in creating new approximate PO outcomes and moves within the method (after the DM has given his/her preferences) is called a *moving reference point*. Furthermore, the point on the navigation set that is the optimal solution to the single objective optimization problem involving the moving reference point is called the *navigated point*. The vector in the navigation set where the DM has stopped the navigation is called the *current navigated point*. The continuous-like movement in the navigation set is realized in this way. To be more specific, the following steps are repeated: (1) the DM provides a reference point and (2) new navigated points are iteratively calculated for the DM as the moving reference point travels along the ray originating from the current navigated point aiming at the reference point.

As a concrete example of the modules and concepts introduced, let us consider the Pareto Navigator method [3] for convex problems. There, the navigation set is the boundary of the convex hull of the given PO outcomes P. In this case, the boundary of the convex hull can be assumed to approximate the complete PO front. In navigation control, the single objective optimization problem is the achievement scalarizing problem (2) (with the moving reference point as the reference point) and its feasible set is the above-mentioned convex hull. The DM can also control the speed how fast the moving reference point moves towards the reference point given by the DM. The speed is controlled indirectly through a step length for shifting the moving reference point.

An essential part of any navigation method is the graphical user interface to visualize the dynamic movement in the navigation set. The Pareto Navigator method has been implemented in the IND-NIMBUS® software framework [30]. Designing the user interface is discussed in [41] and the implementation of Nonconvex Pareto Navigator to be introduced follows similar ideas.

3.2. Desirable Properties

In this section, we introduce desirable properties of navigation methods for computationally expensive multiobjective optimization problems. We distinguish between technical and user experience related properties. For characterizing navigation methods in general, the desirable properties are based on findings and common characteristics between related methods in the literature (we mainly rely on [16, 17, 18, 19, 21] in this context). As far as computational cost is concerned, the desirable properties originate from both the literature and analysis of the authors based on practical experience.

3.2.1. Technical Properties

The properties in this subsection are technical by nature and a method either fulfills or does not fulfill them. They differ from the properties related to user experience given in Subsection 3.2.2 that may depend on the DM in question.

Navigation is complete. The method should not restrict the DM in what (approximate) PO outcomes he/she can find by navigating. The completeness of navigation is analogical to the completeness property of a scalarization method given in [37], which states that every PO outcome must be reachable with some parameters of the scalarization. To be more precise, this property can be posed as follows: given a current outcome $\mathbf{n}^1$ in the navigation set N and a PO outcome $\mathbf{f}(\mathbf{x}) \in \mathbf{f}(S)$, there exist preferences of the DM so that the PO outcome $\mathbf{f}(\mathbf{x})$ is the projection of an outcome $\mathbf{n}^2 \in N$ and the preferences of the DM lead to navigating on the outcome $\mathbf{n}^2$.

Navigation is computationally efficient. It is essential that the navigation is computationally efficient, because the use of an approximation based navigation set and not the original PO front is intended to alleviate the computational burden. This means that calculating outcomes in the navigation set must be fast. The speed of calculation is naturally relative, but the navigation on the navigation set should be substantially more efficient than that on the PO front. In particular, the interaction with the DM should be realisable in "real time", i.e., the DM should not have to wait more than, say, a couple of seconds for a new outcome.

Construction of the navigation set is computationally efficient. The construction of the navigation set should not take a very long time. Although the construction can be done offline before involving the DM, it should not take very long. Especially, the construction should take less time than computing more PO outcomes.

Accuracy of the navigation set can be measured. As the navigation set is an approximation of the PO front, the accuracy of the approximation should be measurable. The accuracy information can be used to determine when the navigation set is accurate enough to involve the DM and accuracy information can also be conveyed to the DM.

Accuracy of the navigation set can be improved. If we find that the navigation set is not accurate enough, then it has to be possible to increase the accuracy. This can be done, e.g., by adding more PO outcomes to the set P and then reconstructing the navigation set.

3.2.2. Properties Related to the User Experience

In this section, we introduce desirable properties related to the user experience of the DM. They are much more difficult to measure than the technical properties and may depend on the DM in question.

The DM can control the navigation. The DM should be able to control the navigation. To measure this, the following tests can be performed on DMs:

1. If a DM has in mind preferable values that are attainable, i.e., that correspond to a vector in the navigation set, is he/she able to reach it by navigating?
2. If a DM has in mind values of objectives that he/she does not like, can he/she restrict this area from being navigated to?

Low cognitive load is set on the DM. The navigation should cognitively burden the DM as little as possible. This can be separated into (at least) two concrete sub-properties: 1. controlling the navigation is intuitive and 2. navigation is visualized in an understandable way.

The DM is allowed to learn. The DM must be allowed to learn and change one's mind. The navigation should allow the DM to re-examine areas of the navigation set that have already been passed. In this way, the method should support psychological convergence of the DM rather than mathematical convergence as discussed, e.g., in [27].

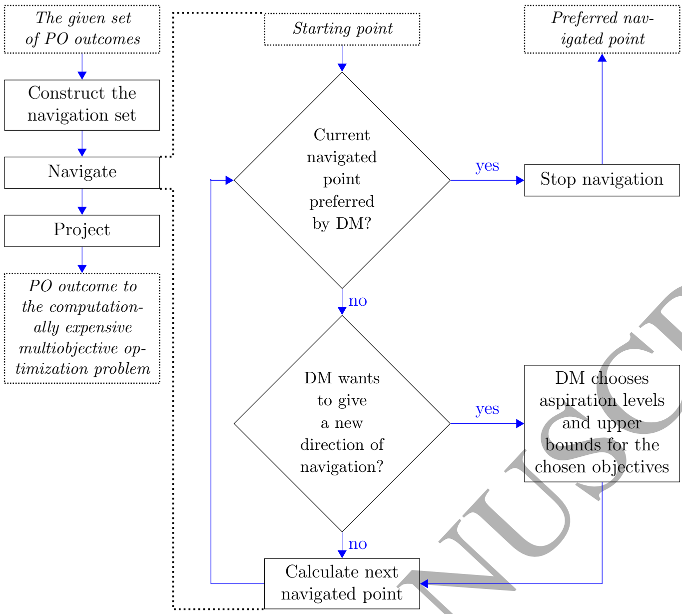


Figure 1: The flowchart of the Nonconvex Pareto Navigator method

The DM can get additional information of the navigation set. In addition to the navigated points, the DM should be given further information about the navigation set. For example, the DM can be shown the ranges of the objective function values in the navigation set. Another possibility is to show a set of feasible directions from a current navigated point (if it is in a corner).

4. The Nonconvex Pareto Navigator Method

In this section, we introduce the Nonconvex Pareto Navigator method. The new method includes the modules introduced in Section 3.1: a navigation set, a navigation control and a projection. Figure 1 shows the flowchart of the Nonconvex Pareto Navigator method. The main steps of the method are given on the left. Once a set of PO outcomes P is available, the navigation set is constructed with the help of the PAINT method and e-cones. The navigation step is described in more detail on the right of the figure. To get the navigation started, the DM can select one of the points in P as a starting point. From the current navigated point (the starting point in the first iteration), the DM can specify desirable aspiration levels for improving the components or specify upper bounds for some components that should not be exceeded. The DM navigates like this and once (s)he has found a preferred navigated point, the solution process proceeds to the projection step. There, problem (2) is used for projection and the result is a final PO solution to the original problem. If the final PO solution is not preferable, although the navigated point was, then we find that the navigation set did not reflect the set of PO outcomes well enough, and the accuracy of the navigation set must be improved and a new navigation must be performed.

The steps and the related modules of the method are described in more detail in the following subsections.

4.1. Constructing the Navigation Set

As the name suggests, the step called constructing the navigation set refers to the module navigation set. The essence of constructing the navigation set is an approximation of the PO front based on a given set of PO outcomes P (see Section 2), for which we suggest to use the PAINT method [10, 11]. However, the PAINT approximation method only interpolates and we want to allow the DM to also search for outcomes that may be outside the convex hull of P. This is crucial for the navigation to be complete, as defined in Section 3.2, and will be realised by an extension based on e-cones.

Approximation. Many approximation methods are available in the field of multiobjective optimization, see, e.g., [36] for a survey. Many of these may be used to construct the navigation set. For example, [23] suggest to use an (at most) $(k-1)$ dimensional representation of the PO front that is composed of simplices. While this approximation can usually be computed very efficiently, we note that it may produce undesirable solutions even for small examples. Consider, e.g., the set $P = \{(0,0,1)^T, (0,1,0)^T, (0.8,0.8,0)^T\}$ of three PO outcomes in $\mathbb{R}^3$. In this situation, the method of [23] yields a 2-dimensional polytope with corners at the three sampled PO outcomes. The approximation contains, e.g., the point $(0.5,0.5,0)^T$ which dominates the PO outcome $(0.8,0.8,0)^T$ and would, thus, mislead the DM in falsely indicating that the point $(0.8,0.8,0)^T$ was dominated. To overcome this difficulty, we suggest to use an approximation that, in addition to containing all points from the set P, is inherently nondominated, i.e., the approximation does not contain any points that dominate other points of the approximation. This implies in particulary that none of the points from the set P is dominated by a point in the approximation. The PAINT approximation has this property

[10, 11] and is thus used as the underlying approximation method in the Nonconvex Pareto Navigator method. Note that the PAINT approximation applied to the above set P yields two line segments joining the PO outcomes $(0,0,1)^T$ and $(0.8,0.8,0)^T$, and the PO outcomes $(0,1,0)^T$ and $(0.8,0.8,0)^T$, respectively, which is more meaningful.

PAINT takes as input a set of given PO outcomes $P \subseteq \mathbf{f}(S)$ and employs Delaunay triangulations. The resulting approximation is piecewise linear and inherently nondominated (as defined above), contains all points in P, and can be represented as a mixed integer linear multiobjective optimization problem. Thus, one can use most multiobjective optimization methods to find a preferred outcome on the approximation by replacing the original problem by the surrogate problem generated by PAINT.

e-Cones. The core element of the Nonconvex Pareto Navigator that makes it complete for nonconvex problems in the sense of Subsection 3.2.1 is that the PAINT approximation is extended by adding e-cones in the objective space. The goal is to enable navigation on disconnected navigation sets by joining the disconnected parts in the objective space, but without giving misleading (i.e., overly optimistic) information to the DM. The e-cones can be viewed as an outer approximation of the positive orthant $\mathbb{R}^k_+$. They are convex polyehdral cones that contain the positive orthant $\mathbb{R}^k_+$. For more information on polyhedral cones and polyhedral computation, we refer, e.g., to [4, 5]; see also [42, 45] for a different interpretation in the context of dominance cones.

As said, we introduce e-cones to join disconnected parts of the PAINT approximation to facilitate navigation. A drawback of this approach is that the DM may navigate to a point on the navigation set for which no corresponding feasible outcome exists. Since we do not want to raise unrealistic expectations by suggesting overly optimistic outcomes, the e-cones are defined as outer approximations of the positive orthant, i.e., the dominance cone. Using such a definition, it is unlikely that a point in the navigation set that is induced by e-cones is preferred by the DM since such points are "almost" dominated by at least one feasible alternative in the navigation set. If desired, outcomes on e-cones can be associated with a search direction that leads to an "almost dominating" outcome in the original approximation. In any case, the final projection step will at the latest make sure that the DM gets to see the closest outcome which does not involve approximation. Note that the property that none of the points in the navigation set dominates any of the given Pareto optimal outcomes remains valid even in this case.

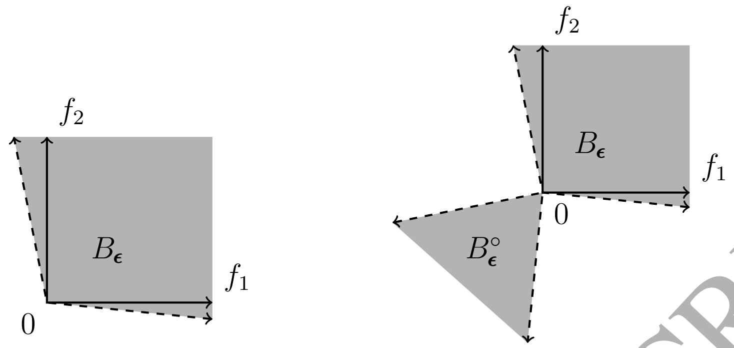


Figure 2: (a) On the left: An example of an e-cone B_ϵ (the grey area bounded by the dashed halflines) in $\mathbb{R}^2$ with $\epsilon = (0.1, 0.2)^T$. (b) On the right: The e-cone B_ϵ drawn together with its polar cone B_ϵ°, c.f. Section 5.1 and Example 1 below.

More formally, we define an e-cone using a k-dimensional parameter vector $\boldsymbol{\epsilon}$ which controls the tradeoffs defined by the e-cone. For each $\boldsymbol{\epsilon} \in \text{int}(\mathbb{R}^k_+)$ (chosen sufficiently small), the e-cone B_ϵ is defined by k extreme rays as

$$B_\epsilon = \left\{ \sum_{j=1,\dots,k} \lambda_j \mathbf{v}^j \, : \, \lambda_j \geq 0 \text{ for all } j = 1, \dots, k \right\}, \tag{3}$$

where the extreme rays $\mathbf{v}^j$, $j = 1, \dots, k$, are given by

$$v_i^j = \begin{cases} 1 & , \text{ if } j = i \text{ and} \\ -\epsilon_j & , \text{ otherwise} \end{cases} \tag{4}$$

and $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_k)^T \in \text{int}(\mathbb{R}^k_+)$ satisfies

$$0 < \epsilon_j < \frac{1}{k-1} =: \hat{\epsilon} \tag{5}$$

for all $j = 1, \dots, k$. In what follows, we call all $\boldsymbol{\epsilon} \in \mathbb{R}^k_+$ satisfying inequality (5) *admissible*. Note that B_ϵ is a convex cone and that $\mathbf{0} \in B_\epsilon$ by construction, and thus B_ϵ is pointed for all $\boldsymbol{\epsilon} \in \mathbb{R}^k_+$. See Figure 2(a) for an illustration.

The most important reason for inequality (5) is that the e-cones B_ϵ should include the positive orthant $\mathbb{R}^k_+$. This is not the case with too large values of ϵ_i, $i = 1, \dots, k$. To see this, we transform the representation of e-cones based on extreme rays given in (3) into a description based on linear inequalities. More generally, (3) is an extreme ray representation of the form

$$B_{\boldsymbol{\epsilon}} = \left\{ \mathbf{z} \in \mathbb{R}^k \,:\, \mathbf{z} = V^T \boldsymbol{\lambda} \text{ with } \boldsymbol{\lambda} \in \mathbb{R}^k_+ \right\},$$

where $V \in \mathbb{R}^{k \times k}$ is the matrix containing the extreme rays $\mathbf{v}^j$, $j = 1, \ldots, k$ of $B_{\boldsymbol{\epsilon}}$ as its rows. Then the polar cone $B^\circ_{\boldsymbol{\epsilon}}$ of $B_{\boldsymbol{\epsilon}}$ is given by

$$B^\circ_{\boldsymbol{\epsilon}} = \left\{ \mathbf{z} \in \mathbb{R}^k \,:\, -V\mathbf{z} \geq \mathbf{0} \right\},$$

i.e., by a description based on k linear inequalities. Clearly, both sets $B_{\boldsymbol{\epsilon}}$ and $B^\circ_{\boldsymbol{\epsilon}}$ are convex cones and, according to the Weyl-Minkowski theorem, both have a representation based on extreme rays *and* a description based on linear inequalities. In this particular case, $B_{\boldsymbol{\epsilon}} \subseteq \mathbb{R}^k$ is represented by k extreme rays, and $B^\circ_{\boldsymbol{\epsilon}} \in \mathbb{R}^k$ is described by k inequalities, i.e., both $B_{\boldsymbol{\epsilon}}$ and $B^\circ_{\boldsymbol{\epsilon}}$ are simplicial cones. Thus, the extreme ray representation of $B^\circ_{\boldsymbol{\epsilon}}$ and, by polarity, the inequality description of $B_{\boldsymbol{\epsilon}}$ can be obtained (as a special case of the double description method, see, e.g., [4, 5]) as

$$\begin{aligned} B^\circ_{\boldsymbol{\epsilon}} &= \left\{ \mathbf{z} \in \mathbb{R}^k \,:\, \mathbf{z} = (-V)^{-1} \boldsymbol{\lambda} \text{ with } \boldsymbol{\lambda} \in \mathbb{R}^k_+ \right\}, \text{ and} \\ B_{\boldsymbol{\epsilon}} &= \left\{ \mathbf{z} \in \mathbb{R}^k \,:\, (V^{-1})^T \mathbf{z} \geq \mathbf{0} \right\}. \end{aligned}$$

Here,

$$V = \begin{pmatrix} 1 & -\epsilon_1 & \cdots & -\epsilon_1 \\ -\epsilon_2 & 1 & \cdots & -\epsilon_2 \\ & & \ddots & \\ -\epsilon_k & -\epsilon_k & \cdots & 1 \end{pmatrix} = \underbrace{\begin{pmatrix} 1+\epsilon_1 & 0 & \cdots & 0 \\ 0 & 1+\epsilon_2 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & 1+\epsilon_k \end{pmatrix}}_{=:D} + \underbrace{\begin{pmatrix} -\epsilon_1 \\ -\epsilon_2 \\ \cdots \\ -\epsilon_k \end{pmatrix}}_{=:\mathbf{u}} \cdot \underbrace{(1, \ldots, 1)}_{=:\mathbf{v}^T}.$$

The inverse V^{-1} always exists and can be computed by the Sherman-Morrison formula as $V^{-1} = D^{-1} - (\frac{1}{1+\mathbf{v}^t D^{-1} \mathbf{u}})(D^{-1}\mathbf{u}\mathbf{v}^T D^{-1})$, see, e.g., [38]. Setting $\xi_j := \frac{1}{1+\epsilon_j}$ for $j = 1 \ldots, k$, we obtain

$$V^{-1} = \begin{pmatrix} \xi_1 & 0 & \cdots & 0 \\ 0 & \xi_2 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & \xi_k \end{pmatrix} + \frac{1}{1 - \sum\limits_{j=1}^{k} \epsilon_j \xi_j} \begin{pmatrix} \xi_1^2 \epsilon_1 & \xi_1 \xi_2 \epsilon_1 & \cdots & \xi_1 \xi_k \epsilon_1 \\ \xi_2 \xi_1 \epsilon_2 & \xi_2^2 \epsilon_2 & \cdots & \xi_2 \xi_k \epsilon_2 \\ & & \ddots & \\ \xi_k \xi_1 \epsilon_k & \xi_k \xi_2 \epsilon_k & \cdots & \xi_k^2 \epsilon_k \end{pmatrix}. \quad (6)$$

Note that all components of V^{-1} are nonnegative if $\boldsymbol{\epsilon}$ is admissible, i.e., if $\boldsymbol{\epsilon}$ satisfies (5). Indeed, let $\bar{\epsilon} := \max_{j=1,\dots,k} \epsilon_j$. The only value that may potentially be negative in (6) is the scalar $1 - \sum_{j=1}^{k} \epsilon_j \xi_j = 1 - \sum_{j=1}^{k} \frac{\epsilon_j}{1+\epsilon_j} \geq 1 - k \cdot \frac{\bar{\epsilon}}{1+\bar{\epsilon}}$, which is positive if $1 + \bar{\epsilon} - k\bar{\epsilon} > 0$, i.e., if $\bar{\epsilon} < \frac{1}{k-1} = \hat{\epsilon}$.

Example 1. *Consider a two-dimensional example with* $\boldsymbol{\epsilon} = (0.1, 0.2)^T$. *Then*

$$V = \begin{pmatrix} 1 & -0.1 \\ -0.2 & 1 \end{pmatrix} \text{ and } V^{-1} = \begin{pmatrix} 1.020408 & 0.102041 \\ 0.204082 & 1.020408 \end{pmatrix},$$

and hence

$$\begin{aligned} B_{\boldsymbol{\epsilon}} &= \left\{ \mathbf{z} \in \mathbb{R}^2 \,:\, \mathbf{z} = \begin{pmatrix} 1 & -0.2 \\ -0.1 & 1 \end{pmatrix} \boldsymbol{\lambda} \text{ with } \boldsymbol{\lambda} \in \mathbb{R}^2_+ \right\} \\ &= \left\{ \mathbf{z} \in \mathbb{R}^2 \,:\, \begin{pmatrix} 1.020408 & 0.204082 \\ 0.102041 & 1.020408 \end{pmatrix} \mathbf{z} \geq \mathbf{0} \right\}, \quad \text{and} \\ B^{\circ}_{\boldsymbol{\epsilon}} &= \left\{ \mathbf{z} \in \mathbb{R}^2 \,:\, \begin{pmatrix} -1 & 0.1 \\ 0.2 & -1 \end{pmatrix} \mathbf{z} \geq \mathbf{0} \right\} \\ &= \left\{ \mathbf{z} \in \mathbb{R}^2 \,:\, \mathbf{z} = \begin{pmatrix} -1.020408 & -0.102041 \\ -0.204082 & -1.020408 \end{pmatrix} \boldsymbol{\lambda} \text{ with } \boldsymbol{\lambda} \in \mathbb{R}^2_+ \right\}. \end{aligned}$$

See Figure 2(b) for an illustration.

The following result is an immediate consequence of the discussion above.

Theorem 1. *Let* $\boldsymbol{\epsilon}$ *be admissible. Then* $\mathbb{R}^k_+ \subseteq B_{\boldsymbol{\epsilon}}$ *and* $\mathbb{R}^k_+ \setminus \{\mathbf{0}\} \subset \operatorname{int}(B_{\boldsymbol{\epsilon}})$.

Proof. The claim follows from the fact that $B_{\boldsymbol{\epsilon}} = \{\mathbf{z} \in \mathbb{R}^k : (V^{-1})^T \mathbf{z} \geq \mathbf{0}\}$, where $(V^{-1})^T$ is a matrix whose entries are all strictly positive if $\boldsymbol{\epsilon}$ is admissible. In particular, $(V^{-1})^T \mathbf{z} \geq \mathbf{0}$ for all $\mathbf{z} \in \mathbb{R}^k_+$, and $(V^{-1})^T \mathbf{z} > \mathbf{0}$ for all $\mathbf{z} \in \mathbb{R}^k_+ \setminus \{\mathbf{0}\}$. □

Theorem 1 will be needed when discussing the completeness of navigation. The following property interrelates e-cones for different parameter vectors $\boldsymbol{\epsilon}$ and is included here for the sake of completeness.

Theorem 2. *Let* $\boldsymbol{\epsilon}^2$ *be admissible. If* $\boldsymbol{\epsilon}^1, \boldsymbol{\epsilon}^2 \in \mathbb{R}^k_+$ *such that* $0 < \epsilon^1_j \leq \epsilon^2_j$ *for all* $j = 1, \dots, k$, *then* $B_{\boldsymbol{\epsilon}^1} \subseteq B_{\boldsymbol{\epsilon}^2}$.

Proof. First note that if $\boldsymbol{\epsilon}^2$ is admissible, then so is $\boldsymbol{\epsilon}^1$. Let $V^\ell \in \mathbb{R}^{k\times k}$ be the matrix containing the extreme rays of $B_{\boldsymbol{\epsilon}^\ell}$ in its rows, $\ell = 1, 2$. Then the respective polar cones are given by $B^\circ_{\boldsymbol{\epsilon}^\ell} = \{\mathbf{z} \in \mathbb{R}^k : -V^\ell \mathbf{z} \geq \mathbf{0}\}$, where the components v^ℓ_{ij} of V^ℓ, $\ell = 1, 2$, satisfy $-v^1_{ii} = -v^2_{ii} = -1$ for all $i = 1, \ldots, k$ and $0 < -v^1_{ij} \leq -v^2_{ij}$ for all $i, j = 1, \ldots, k$, $i \neq j$. It immediately follows that $B^\circ_{\boldsymbol{\epsilon}^2} \subseteq B^\circ_{\boldsymbol{\epsilon}^1}$ and thus, by polarity, $B_{\boldsymbol{\epsilon}^1} \subseteq B_{\boldsymbol{\epsilon}^2}$. □

Note that the particular choice of the parameter $\boldsymbol{\epsilon}$ is important when discussing trade-off rates, and we return to this in Section 5.

Navigation set. Now the navigation set $N_{\boldsymbol{\epsilon}}$ is defined as a sum of the PAINT approximation A and the e-cones $B_{\boldsymbol{\epsilon}}$:

$$N_{\boldsymbol{\epsilon}} := A + B_{\boldsymbol{\epsilon}} = \{\mathbf{a} + \mathbf{b} : \mathbf{a} \in A,\ \mathbf{b} \in B_{\boldsymbol{\epsilon}}\}. \tag{7}$$

An example of $N_{\boldsymbol{\epsilon}}$ where the set P consists of three PO outcomes is given in Figure 3.

In practice, the parameters $\epsilon_j > 0$, $j = 1, \ldots, k$, will often be selected all equal, i.e., $\epsilon_j = \epsilon > 0$ for all $j = 1, \ldots, k$, and close to zero to ensure that the e-cones only slightly enlarge the positive orthant $\mathbb{R}^k_+$. In this way, the navigation set is connected and does not contain any weakly PO outcomes, but it also contains no points that raise expectations that can not be realized by an existing PO outcome. Note that, however, $\boldsymbol{\epsilon}$ should be chosen large enough to avoid numerical difficulties. Despite the fact that the e-cones are convex, the set $N_{\boldsymbol{\epsilon}}$ is in general nonconvex. Note moreover that the way that e-cones are used here is different from their use, e.g., in dominance relations, see, e.g., [42, 45].

4.2. Navigation Control

The core of the navigation step of the Nonconvex Pareto Navigator flowchart in Figure 1 is the module navigation control. As mentioned earlier, in the (convex) Pareto Navigator method, the DM specifies preference information as a reference point. In the Nonconvex Pareto Navigator method, we provide more options for the DM in navigation control to facilitate navigation in cases were the PO front is nonconvex or even disconnected. For this, we adapt the type of preference information used in the classification-based interactive NIMBUS method [24, 26]. The idea is that the DM can specify both aspiration levels and upper bounds for objective function values. This allows the DM to give preference information in a way that has been shown to be

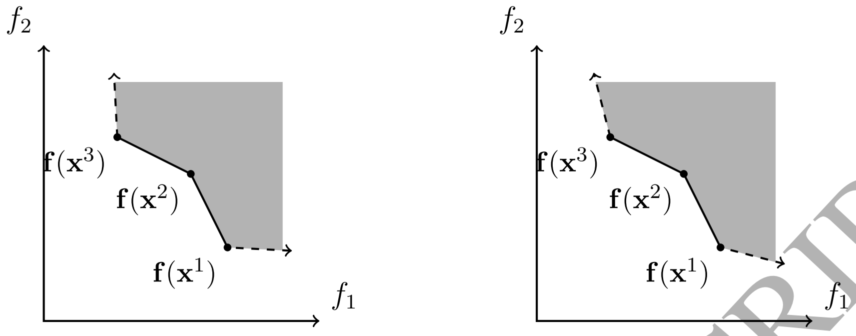


Figure 3: An example of the navigation set $N_{\boldsymbol{\epsilon}}$ in $\mathbb{R}^2$ with P consisting of three PO outcomes. The extension given by the e-cones is indicated by dashed lines. (a) On the left: $\boldsymbol{\epsilon} = (\epsilon, \epsilon)^T$ with a small value of $\epsilon > 0$; (b) on the right: $\boldsymbol{\epsilon} = (\epsilon, \epsilon)^T$ with a larger value of $\epsilon > 0$, leading to potentially misleading outcomes in the navigation set.

intuitive for a DM [22], i.e., in a way that is directly related to the values of objectives. In addition, similarly to the NIMBUS method, the DM can allow some of the objectives to change freely for the next iteration, which decreases the cognitive burden as the DM can concentrate only on a subset of the objectives at a time.

Let us assume that the current navigated point is $\mathbf{n}^1 \in N_{\boldsymbol{\epsilon}}$ and the DM has provided aspiration levels for a subset of objectives with indices in $I_1 \subseteq \{1, \ldots, k\}$, and upper bounds for a subset of objectives with indices in $I_2 \subseteq \{1, \ldots, k\}$. Note that I_1 and I_2 do not have to be disjoint. More precisely, we assume that the DM has provided aspiration levels z_i^{asp} for objectives f_i with $i \in I_1$, and upper bounds z_i^{up} ($\geq n_i^1$) for objectives f_i with $i \in I_2$. This implicitly means that the objectives with indices in $\{1, \ldots, k\} \setminus (I_1 \cup I_2)$ are allowed to change freely. Then, the navigation is technically performed by solving a series of single-objective optimization problems with iteratively updated (i.e., moving) reference points $\mathbf{z}^j$, $j = 1 \ldots, J$. The index $j = 1, 2, \ldots, J$ is the navigation parameter that is increased up to a predetermined upper bound J (i.e., J defines how far one navigates into a single direction) to simulate the movement from the current navigated point $\mathbf{n}^1$ towards (and possibly past) the aspiration levels z_i^{asp}, $i \in I_1$. With a predetermined step length $1/t$, $t \in \mathbb{N}$, the reference points $\mathbf{z}^j$, $j = 1, \ldots, J$, are defined by

$$z_i^j = n_i^1 + \frac{j}{t}(z_i^{\text{asp}} - n_i^1) \text{ for all } i \in I_1. \quad (8)$$

The components z_i^j with $i \notin I_1$ are not relevant and can be chosen arbitrarily. In our computational tests, we have used $J = 2t$.

Using these parameters, the following optimization problem is solved iteratively during the navigation phase for $j = 1, \ldots, J$, or until some other stopping condition is satisfied:

$$\min_{\mathbf{z}} \max_{i \in I_1} \left(z_i - z_i^j \right) \tag{9}$$

$$\text{s.t. } z_i \leq z_i^{\text{up}} \text{ for all } i \in I_2 \tag{10}$$

$$\mathbf{z} \in A + B_{\boldsymbol{\epsilon}}. \tag{11}$$

In the above problem, $\mathbf{z} = (z_1, \ldots, z_k)^T$ is the vector of variables. The aspiration levels z_i^j, $i \in I_1$, are parameters defined according to (8), and the upper bounds z_i^{up}, $i \in I_2$, are parameters given by the DM according to the above specifications. Note that alternative and/or additional preference information (e.g., information on a particular region of interest in the decision and/or in the objective space) may be incorporated into this model if so desired by the DM.

The optimization problem (9)–(11) is equivalent to solving (2) (without the augmentation term) in the navigation set with respect to objectives with indices in I_1 with additional hard constraints for upper bounds for objectives with indices in I_2. Note that the augmentation term is not needed as we are in the navigation set $N_{\boldsymbol{\epsilon}}$ that does not contain any weakly PO points.

The optimization problem (9)–(11) can be written as a mixed-integer linear problem because of the following observations:

1. the objective function (9) can be linearized by the standard way of linearizing min-max objectives with an additional continuous variable representing the maximum (see e.g., [24]),
2. the constraint (10) is linear and
3. the PAINT approximation can be linearly parametrized with additional binary variables as described in [11] and, thus, the constraint (11) is linear.

This means that navigation, that is, generating navigated outcomes, is computationally inexpensive as compared to the original problem.

4.3. Projection

When the current navigated outcome $\mathbf{n} \in N_{\boldsymbol{\epsilon}}$ in the navigation set is interesting or desirable for the DM, he/she can be shown the corresponding

PO outcome by projecting $\mathbf{n}$ to the PO front. This can be done by solving problem (2) with $\mathbf{n}$ as the reference point. As mentioned earlier, this yields a PO outcome. (Note that this step is identical e.g. to the projection in the Pareto Navigator method.)

5. Evaluation of Nonconvex Pareto Navigator

In this section, we evaluate how well the proposed Nonconvex Pareto Navigator method meets the desirable properties stated in Section 3.2. The criteria are considered in the same order as in Section 3.2.

5.1. Technical properties

In the following, we analyze whether or not, and if yes how well the Nonconvex Pareto Navigator method performs w.r.t. the technical properties from Section 3.2. Particularly completeness of navigation requires a rigorous analysis of the properties of the navigation set, the navigation control and the projection method.

Navigation is complete. In the following, we prove that any PO outcome can be obtained by navigating with the Nonconvex Pareto Navigator method. This means that the navigation is complete as defined in Section 3.2. Note that the e-cones B_ϵ play an important role for the completeness of navigation since they allow to span the complete objective space, also in the case of disconnected approximations of the PO front. The e-cones are thus a versatile tool that can be used in combination with approximations or representations of the PO front in general.

We therefore keep the analysis in this section general by assuming that A is a closed set that approximates the PO front (for example, given by the PAINT approximation). Questions regarding the approximation quality and the relation to trade-off rates will be discussed later in Section 6.

Using the general properties of e-cones derived in Section 4.1, we can prove that the boundary of the navigation set is exactly its PO front. This is a very important result for the functioning of the Nonconvex Pareto Navigator method, because it gives a simple geometric description of the PO front of the navigation set, which is exactly where the navigation takes place.

Theorem 3. *If ϵ is admissible, then* $\mathrm{ND}(A + B_\epsilon) = \partial(A + B_\epsilon)$.

Proof. Because of the definition of dominance, $\mathrm{ND}(A+B_{\boldsymbol{\epsilon}}) \subset \partial(A+B_{\boldsymbol{\epsilon}})$ and we have to show that $\partial(A+B_{\boldsymbol{\epsilon}}) \subset \mathrm{ND}(A+B_{\boldsymbol{\epsilon}})$. Let $\mathbf{z} \in \partial(A+B_{\boldsymbol{\epsilon}})$ and suppose that there exists a vector $\bar{\mathbf{z}} \in (A+B_{\boldsymbol{\epsilon}})$ such that $\bar{\mathbf{z}}$ dominates $\mathbf{z}$. Hence, $\bar{\mathbf{z}} = \bar{\mathbf{a}} + \bar{\mathbf{b}}$ with $\bar{\mathbf{a}} \in A$ and $\bar{\mathbf{b}} \in \mathbb{R}^k_+ \setminus \{\mathbf{0}\}$, and using Theorem 1,

$$\begin{aligned} \mathbf{z} &\in \bar{\mathbf{z}} + (\mathbb{R}^k_+ \setminus \{\mathbf{0}\}) = (\bar{\mathbf{a}} + \bar{\mathbf{b}}) + (\mathbb{R}^k_+ \setminus \{\mathbf{0}\}) = \bar{\mathbf{a}} + (\mathbb{R}^k_+ \setminus \{\mathbf{0}\}) \\ &\subset \bar{\mathbf{a}} + \mathrm{int}(B_{\boldsymbol{\epsilon}}) \subset \mathrm{int}(A+B_{\boldsymbol{\epsilon}}), \end{aligned}$$

in contradiction to $\mathbf{z} \in \partial(A+B_{\boldsymbol{\epsilon}})$. □

With the following lemma, we are almost finished with proving the completeness of the navigation. The proof is simple, because we can use Theorem 3.

Lemma 1. *Let $\mathbf{z} \in \mathbb{R}^k$, let $\boldsymbol{\epsilon} > \mathbf{0}$ be admissible, and let A be closed. Then there exist $t \in \mathbb{R}$ and $\mathbf{n} \in \mathrm{ND}(A+B_{\boldsymbol{\epsilon}})$ such that $\mathbf{z} + t\mathbf{1} = \mathbf{n}$, i.e., $z_i + t = n_i$ for all $i = 1, \ldots, k$.*

Proof. Let $\mathbf{z} \in \mathbb{R}^k$ be arbitrary, and choose any $\mathbf{z}' \in A + B_{\boldsymbol{\epsilon}}$ and $\mathbf{z}'' \in (A+B_{\boldsymbol{\epsilon}})^c = \mathbb{R}^k \setminus (A+B_{\boldsymbol{\epsilon}})$. Now $\mathbf{z}+t\mathbf{1} \in A+B_{\boldsymbol{\epsilon}}$, when $t \geq \max_{i=1,\ldots,k}(z'_i - z_i)$ and $\mathbf{z} + t\mathbf{1} \in (A+B_{\boldsymbol{\epsilon}})^c$, when $t \leq -\max_{i=1,\ldots,k}(z_i - z''_i)$. (Note that, if $\mathbf{z} + \bar{t}\mathbf{1} \in (A+B_{\boldsymbol{\epsilon}})$ for some $\bar{t} \in \mathbb{R}$, then $\mathbf{z} + t\mathbf{1} \in (A+B_{\boldsymbol{\epsilon}})$ for all $t \geq \bar{t}$, and conversely, if $\mathbf{z} + \tilde{t}\mathbf{1} \in (A+B_{\boldsymbol{\epsilon}})^c$ for some $\tilde{t} \in \mathbb{R}$, then $\mathbf{z} + t\mathbf{1} \in (A+B_{\boldsymbol{\epsilon}})^c$ for all $t \leq \tilde{t}$.) Since $A + B_{\boldsymbol{\epsilon}}$ is closed, there must exist $t \in \mathbb{R}$ such that $\mathbf{z} + t\mathbf{1} \in \partial(A+B_{\boldsymbol{\epsilon}})$. The claim follows from this. □

Finally we can prove the main result of this section. The following theorem shows that the navigation in the Nonconvex Pareto Navigator method is complete. To keep the notation simple, we assume in the following that $z_i^{\mathrm{nad}} - z_i^{\mathrm{utopia}} = 1$ for all $i = 1, \ldots, k$, and thus (2) simplifies to

$$\min_{\mathbf{x} \in S} \max_{i=1,\ldots,k} \left[f_i(\mathbf{x}) - z_i^{\mathrm{ref}}\right] + \rho \sum_{i=1}^{k} f_i(\mathbf{x}).$$

Theorem 4. *Assume that $\boldsymbol{\epsilon} > \mathbf{0}$ is admissible and that A is closed. Let $\mathbf{n}' \in A + B_{\boldsymbol{\epsilon}}$ be a current navigated point, and consider an arbitrary PO outcome $\mathbf{f}(\mathbf{x}') \in \mathrm{ND}(\mathbf{f}(S))$. Then there exist aspiration levels $z_i^{\mathrm{asp}} \in \mathbb{R}$, $i = 1, \ldots, k$, a step length $t \in \mathbb{R}_+$, and a sufficiently small augmentation parameter $\rho > 0$ such that the navigation leads to a point*

$$\mathbf{n}'' = \arg\min_{\mathbf{n} \in A+B_{\boldsymbol{\epsilon}}} \max_{i=1,\ldots,k} n_i - ((1-t)n'_i + t z_i^{\mathrm{asp}}),$$

and the projection of $\mathbf{n}''$ *onto* $\mathrm{ND}(\mathbf{f}(S))$ *is the PO outcome* $\mathbf{f}(\mathbf{x}')$*, i.e.,*

$$\mathbf{x}' = \arg\min_{\mathbf{x}\in S} \max_{i=1,\dots,k} [f_i(\mathbf{x}) - n_i''] + \rho \sum_{i=1}^{k} f_i(\mathbf{x}).$$

Proof. By Lemma 1, we may choose $\mathbf{z}^{\mathrm{asp}}$ such that $\mathbf{z}^{\mathrm{asp}} \in \mathrm{ND}(A + B_{\boldsymbol{\epsilon}})$ and $\mathbf{z}^{\mathrm{asp}} = \mathbf{f}(\mathbf{x}') + s\mathbf{1}$ for some $s \in \mathbb{R}$, i.e., for all $i = 1,\dots,k$ it holds that $\mathbf{z}_i^{\mathrm{asp}} = f_i(\mathbf{x}') + s$. Furthermore, choose $t = 1$. Then $\mathbf{n}'' = \mathbf{z}^{\mathrm{asp}}$, and the projection of $\mathbf{n}''$ onto $\mathrm{ND}(\mathbf{f}(S))$ is obtained as $\arg\min_{\mathbf{x}\in S} \max_{i=1,\dots,k}[f_i(\mathbf{x}) - (f_i(\mathbf{x}') + s)] + \rho \sum_{i=1}^{k} f_i(\mathbf{x})$. Since $\mathbf{x}'$ is PO, the minimum is attained at $\mathbf{x}'$ if $\rho > 0$ is sufficiently small. □

The navigation is computationally efficient. As described in Section 4.2, the navigation can be realized by solving mixed integer linear optimization problems. Solving these problems is fast, when compared to solving e.g., simulation-based optimization problems.

The construction of the navigation set is computationally efficient. The computational cost of constructing the navigation set is due to constructing the PAINT approximation. As shown in [11], PAINT approximations can be constructed for large numbers of PO outcomes in a relatively short time.

The accuracy of the navigation set can be measured. As described in [10], the accuracy of the PAINT approximation A at a point $\mathbf{a} \in A$ can be measured by an *error vector* $\mathbf{d}(\mathbf{a}) \in \mathbb{R}^k$ that can be estimated from the structure of the PAINT approximation. The error vector has the following two properties:

1. there exists an outcome $\mathbf{z} \in \mathbf{f}(S)$ so that $z_i \leq a_i + d_i(\mathbf{a})$ for all $i = 1,\dots,k$ and
2. there does not exist an outcome $\mathbf{z} \in \mathbf{f}(S)$ so that $z_i \leq a_i - d_i(\mathbf{a})$ for all $i = 1,\dots,k$.

These error estimates should have a clear meaning to the DM: 1. implies that there exists an PO outcome that is at least as good as $\mathbf{a} + \mathbf{d}(\mathbf{a})$ in all objectives and 2. implies that there is no outcome that dominates $\mathbf{a} - \mathbf{d}(\mathbf{a})$.

The above implies that at least when the navigation takes place on the PAINT approximation the accuracy can be measured. In addition, since the navigation set $N_{\boldsymbol{\epsilon}}$ is a sum of the PAINT approximation and the e-cone $B_{\boldsymbol{\epsilon}}$, the accuracy measure on the PAINT approximation also implies an accuracy measure for the accuracy of the complete navigation set.

The accuracy of the navigation set can be improved. By computing more PO outcomes, it is indeed possible to make the PAINT approximation and, thus, the navigation set more accurate. Especially, the PAINT approximation always contains all the given PO outcomes. Thus, the accuracy of the navigation set can be increased by computing more PO outcomes. When computing the approximation again, there, however, may be global changes to the approximation, so, theoretically, the approximation could become worse in some areas. However, since the new point will be part of the updated approximation, it can be guaranteed that the approximation becomes more accurate in the area that the DM was interested in.

5.2. Properties Related to the User Experience

The DM can control the navigation and low cognitive load is set on the DM. The DM can use aspiration levels and upper bounds when controlling navigation, that is, when indicating what kind of objective values are more preferred than the current navigated point. Both of these have been found intuitive in [22]. In addition, the DM can let some of the objectives change freely for a while, which has been found useful in real-life problems solved with the interactive NIMBUS method (see e.g., [8]). This decreases the cognitive load in particular when dealing with problems with a large number of objectives.

The DM is allowed to learn. The DM can go backwards i.e., the DM can choose any of the previously navigated outcomes as the current navigated point. After this, the DM can give different preferences than previously and, thus, change his/her mind and navigate to a different direction. In a more general level, changing one's mind is made possible, because the outcomes generated in the navigation set depend only on the current navigated point and the current preferences given by the DM, that is, current navigation control, and not preferences previously given.

The DM can get additional information of the navigation set. Guidance depends highly on the graphical user interface of the implementation. The implementation of the Nonconvex Pareto Navigator method mentioned in Section 7 shows all the time the set P to the DM. Thus, if the DM gets lost, he/she can restart the search from any of the outcomes in P.

To conclude this section, we can see that the Nonconvex Pareto Navigator method meets the desirable properties of navigation methods well. This is true from both technical and user experience points of view. The latter is further considered in Section 7 when solving an example problem.

6. Connection Between e-Cones and Trade-Off Rates

As mentioned in Section 4.1, the parameter vector $\boldsymbol{\epsilon}$ plays an important role in defining an intuitive extrapolation $N_{\boldsymbol{\epsilon}} = A + B_{\boldsymbol{\epsilon}}$ of the PAINT approximation A that facilitates navigation in nonconvex problems and, if chosen appropriately, guarantees a complete representation (c.f. Theorem 4). To realize this goal, the parameter $\boldsymbol{\epsilon}$ has to be admissible, i.e., satisfy inequality (5). On the other hand, the choice of the parameter $\boldsymbol{\epsilon}$ affects the navigated points that the DM can see. This may be overly optimistic for large values of $\boldsymbol{\epsilon}$ and the real structure of the nondominated set may be hidden by extrapolating with a large e-cone $B_{\boldsymbol{\epsilon}}$. Figure 4a shows an example where a very large value of $\boldsymbol{\epsilon}$ may cause the DM to navigate on points that are clearly infeasible, while in Figure 4b the parameter $\boldsymbol{\epsilon}$ has been selected appropriately for this example problem. Since the PAINT approximation has been shown to interpolate well between the given PO outcomes [10, 11], a reasonable requirement, avoiding the problem shown in Figure 4a, is that the PAINT approximation is included in the PO front of the navigation set.

We show how the parameter $\boldsymbol{\epsilon}$ is connected to the trade-off rates on the navigation set. Thus, information about acceptable trade-offs can be used to set the value of $\boldsymbol{\epsilon}$. The result guarantees that with an appropriate value of the parameter $\boldsymbol{\epsilon}$, the PAINT approximation is indeed included in the PO front of the navigation set. In setting the parameter $\boldsymbol{\epsilon}$, there is an analogy to setting the augmentation constant in problem (2), which is discussed, e.g., in [15, 43].

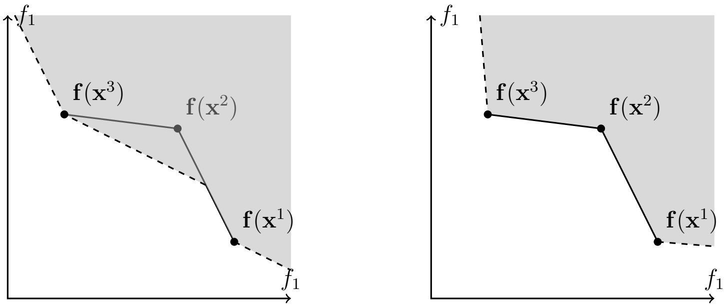
The following results show that within an e-cone $B_{\boldsymbol{\epsilon}}$ the vector parameter $\boldsymbol{\epsilon}$ controls how small the components of a vector can be in relation to its other components. This implies a trade-off type of inequality.

Theorem 5. *Let $\boldsymbol{\epsilon}$ be admissible. Then for all $\mathbf{z} \in B_{\boldsymbol{\epsilon}}$ it holds that*

$$z_i \geq -\frac{1}{1 - \sum_{j=1, j\neq i}^{k} \frac{\epsilon_j}{1+\epsilon_j}} \sum_{j=1, j\neq i}^{k} \frac{\epsilon_j}{1+\epsilon_j} z_j \quad \text{for all } i = 1, \dots, k.$$

Proof. Let $\mathbf{z} \in B_{\boldsymbol{\epsilon}}$. The inequality description $B_{\boldsymbol{\epsilon}} = \{\mathbf{z} \in \mathbb{R}^k : (V^{-1})^T \mathbf{z} \geq \mathbf{0}\}$ with V^{-1} given by (6) implies that

$$\xi_i z_i + \frac{1}{1 - \sum_{j=1}^{k} \xi_j \epsilon_j} \sum_{j=1}^{k} \xi_i \xi_j \epsilon_j z_j \geq 0$$



(a) Undesirable navigation set with a large $\boldsymbol{\epsilon} = (0.5, 0.5)^T$; the PAINT approximation is partly in the interior of the navigation set and thus not in the PO front of the navigation set.

(b) A desirable navigation set with small enough parameter $\boldsymbol{\epsilon} = (0.08, 0.08)^T$; the PAINT approximation is contained in the boundary of the navigation set.

Figure 4: Two navigation sets $N_{\boldsymbol{\epsilon}}$ with different parameter vectors $\boldsymbol{\epsilon}$, but with the same PAINT approximation. The PAINTapproximation is illustrated with two line segments and connects the PO outcomes $\mathbf{f}(\mathbf{x}^1) = (4, 1)^T, \mathbf{f}(\mathbf{x}^2) = (3, 3)^T$ and $\mathbf{f}(\mathbf{x}^3) = (1, 3.25)^T$. The navigation set is shown in grey color.

for all $i = 1, \dots, k$, where $\xi_i = \frac{1}{1+\epsilon_i} > 0$, $i = 1, \dots, k$. Multiplying by $\frac{1-\sum_{j=1}^k \xi_j \epsilon_j}{\xi_i}$ (which is nonnegative since $\boldsymbol{\epsilon}$ is admissible), we obtain

$$z_i \left(1 - \sum_{j=1, j \neq i}^{k} \xi_j \epsilon_j \right) + \sum_{j=1, j\neq i}^{k} \xi_j \epsilon_j z_j \geq 0$$

where $1 - \sum_{j=1, j\neq i}^k \xi_j \epsilon_j \geq 1 - \sum_{j=1}^k \xi_j \epsilon_j > 0$ since $\boldsymbol{\epsilon}$ is admissible, and thus

$$z_i \geq - \frac{1}{1 - \sum_{j=1, j\neq i}^k \xi_j \epsilon_j} \sum_{j=1, j\neq i}^{k} \xi_j \epsilon_j z_j.$$

□

Theorem 5 immediately implies the following slightly weaker but considerably simpler result.

Corollary 1. *Let $\boldsymbol{\epsilon}$ be admissible. Then for all $\mathbf{z} \in B_{\boldsymbol{\epsilon}} \setminus \{\mathbf{0}\}$ it holds that*

$$z_i > - \sum_{j=1, j\neq i}^{k} \epsilon_j z_j \quad \textit{for all } i = 1, \dots, k.$$

Proof. The claim follows immediately from Theorem 5 since $\boldsymbol{\epsilon}$ is admissible and thus

$$1 > \frac{1}{1-\sum_{j=1,j\neq i}^{k}\frac{\epsilon_j}{1+\epsilon_j}} \quad \text{and} \quad \epsilon_j > \frac{\epsilon_j}{1+\epsilon_j} \quad \text{for all } i,j=1,\ldots,k.$$

□

As in Section 5.1, the following results are general in the sense that they apply to any set A that is a closed set that approximates the PO front. The following theorem shows that if a trade-off type property is satisfied for a point on an approximation A, then the point is on the PO front of the navigation set. Thus, points on the approximation satisfying this trade-off type property can be navigated on. Note that Theorem 6 provides a sufficient condition, not a necessary condition. Thus, if a point on the approximation violates (12), it may nevertheless be on the PO front of the navigation set.

Theorem 6. *Let $\boldsymbol{\epsilon}$ be admissible and let $N_{\boldsymbol{\epsilon}} = A + B_{\boldsymbol{\epsilon}}$. If $\mathbf{a} \in A$ such that for all $\mathbf{a}' \in A\backslash\{\mathbf{a}\}$ there exists an index $i \in \{1,\ldots,k\}$ with*

$$a_i' - a_i > \sum_{j=1,j\neq i}^{k} \epsilon_j (a_j - a_j') \tag{12}$$

then $\mathbf{a} \in \mathrm{ND}(N_{\boldsymbol{\epsilon}})$.

Proof. Let us assume that, to the contrary, that there exist $\mathbf{a}' \in A\backslash\{\mathbf{a}\}$ and $\mathbf{b} \in B_{\boldsymbol{\epsilon}}$ such that $\mathbf{a}' + \mathbf{b}$ dominates $\mathbf{a}$, i.e., $a_i' + b_i \leq a_i$ for all $i = 1,\ldots,k$, where one of the inequalities is strict. Using Corollary 1 for $\mathbf{b} \in B_{\boldsymbol{\epsilon}}$ and the fact that $b_j \leq a_j - a_j'$ for all $j = 1,\ldots,k$ by assumption, we get directly

$$a_i' - a_i \leq -b_i \leq \sum_{j=1,j\neq i}^{k} \epsilon_j b_j \leq \sum_{j=1,j\neq i}^{k} \epsilon_j (a_j - a_j')$$

for all $i \in \{1,\ldots,k\}$, where for at least one $i \in \{1,\ldots,k\}$ the inequality is strict. This is a contradiction to the assumption, and the result follows. □

Note that condition (12) of Theorem 6 simplifies if $\boldsymbol{\epsilon} = (\epsilon,\ldots,\epsilon)^T$, i.e., if $\epsilon_i = \epsilon_j = \epsilon$ for all $i,j \in \{1,\ldots,k\}$. Indeed, in this case (12) can be rewritten as $a'_i - a_i > \epsilon \sum_{j=1, j\neq i}^{k}(a_j - a'_j)$. If in this case, for example, $a'_i > a_i$ and $\sum_{j\neq i}(a_j - a'_j) > 0$, then (12) can be rewritten as

$$\frac{a'_i - a_i}{\sum_{j=1, j\neq i}^{k}(a_j - a'_j)} > \epsilon$$

which has to be satisfied for at least one index $i \in \{1, \ldots, k\}$. In the biobjective case $k = 2$, this is even clearer. Assuming (without loss of generality) that $a'_1 > a_1$ and $a_2 > a'_2$, then (12) can be restated as

$$\frac{a'_1 - a_1}{a_2 - a'_2} > \epsilon \quad \text{and equivalently} \quad \frac{a_2 - a'_2}{a'_1 - a_1} < \frac{1}{\epsilon}.$$

This observation directly relates the parameter vector $\boldsymbol{\epsilon} = (\epsilon, \ldots, \epsilon)^T$ with the trade-off of the PAINT approximation at a point $\mathbf{a} \in A$, see, e.g. [6] for the related concept of proper Pareto optimality.

Thus, when we have constructed the PAINT approximation A, we can estimate an upper bound for the trade-offs on the approximation and then choose $\boldsymbol{\epsilon}$ as described above. This guarantees that the PAINT approximation is included in the PO front of the navigation set.

7. An Example in Wastewater Treatment Plant Operation

7.1. Solution Process with Nonconvex Pareto Navigator

We demonstrate how the Nonconvex Pareto Navigator method can be applied to enable learning with a problem of operating a wastewater treatment plant (studied also in [8, 12]). This is a simulation-based problem where objective function evaluations necessitate calling a GPS-X simulator [31]. One simulation takes minutes on an Intel® Core™ 2 Duo CPU P8600, both processors running at 2.40 GHz, and finding a single PO outcome takes approximately half an hour. The problem has five objectives: minimize total nitrogen, minimize blower/aerator wire power consumption, minimize methanol dosage rate, minimize mass flow total suspended solids and maximize total gas flow. In what follows, we refer to the objectives in this order. Even though the method was introduced for minimization problems, for the fifth objective, upper bounds do actually refer to lower bounds. For simplicity, we refer to bounds.

The navigation set was constructed based on 195 PO outcomes that were computed using the GPS-X simulator and hybridizing the UPS-EMO evolutionary algorithm [1] and the controlled random search algorithm [34]. Constructing the navigation set took about 19 hours on Intel® Xeon™ E5410 CPU, but this was no problem since the DM was not involved.

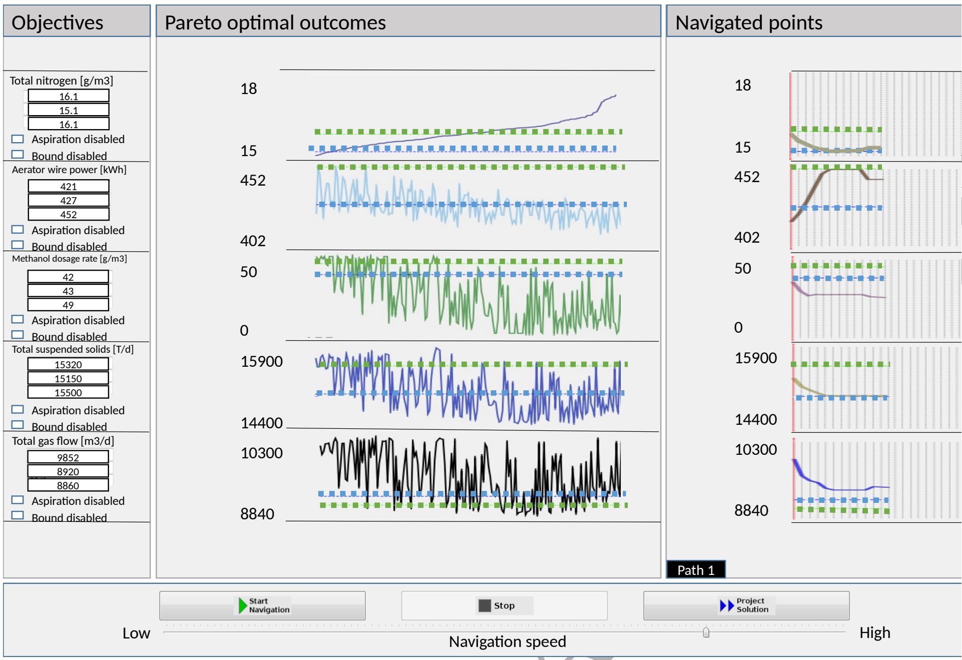
Because of the dynamic nature of the Nonconvex Pareto Navigator method, it is easier to demonstrate its performance with graphical user interfaces and visualizations. As said, the method has been implemented in the IND-NIMBUS® software framework. In Figure 5a, an example of the user interface is shown. It has been divided into three panels. In the left panel, under each objective name, the current objective value, the aspiration level and bound for that objective can be seen in that order. Below the numbers, there are two check boxes, which the DM can use if he/she does not want to provide values for aspiration levels or bounds, or alternatively disable the previously given values.

In the middle panel, the set P is shown (the values of objectives for different outcomes have been connected with a line for visual clarity). The estimated components of the nadir and the ideal objective vectors are shown on the left. The bounds and the aspiration levels are visualized with horizontal (green and blue, respectively) dotted lines. Finally, the (red) vertical dotted line visualizes the starting point of the navigation. The bounds and aspiration levels can be modified by either dragging and dropping the lines or inputting the numbers in the corresponding text boxes. The starting point can be chosen by clicking one of the points in P. Navigated points are shown in the right panel and they appear in real time. Before the actual navigation, the right panel is naturally empty in Figure 5a.

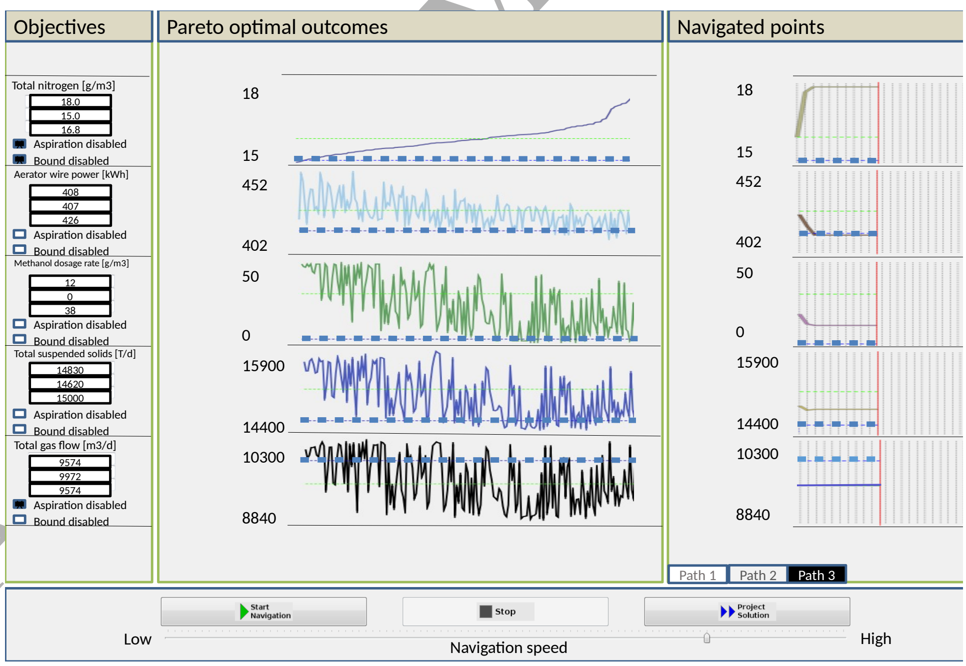
The DM can start or stop navigation and project navigated points by clicking appropriate buttons below the three panels. Finally, with a slider below the buttons, the DM can adjust the speed of navigation. Technically, this refers to setting the step length t in formula (8).

The solution process starts by selecting the starting point. The DM can sort the points in P with respect to the values of any of the objectives. In Figure 5a, the DM has sorted the points with respect to the first objective. In our case, the DM selected the point $(16.1, 421, 42, 15320, 9852)^T$ as the starting point and provided the following preference information: aspiration levels 15.1, 427, 43, 15150 and 8920 and bounds 16.1, 452, 49, 15500 and 8860. These values can be seen in the left panel of the figure.

Once the DM had provided the preference information, the navigation



(a) First navigation of the Nonconvex Pareto Navigator method.



(b) Third navigation (with some bounds and aspiration levels disabled).

Figure 5: Two navigations of Nonconvex Pareto Navigator.

was started and the navigated points are shown in the right panel of the figure. There, the DM can see how the values of the objectives tend toward the aspiration levels (when possible) and always respect the given bounds. The last point where the DM stopped the navigation is the current navigated point. The vertical line has now moved to the right panel indicating that the DM can choose any of the navigated points as the starting point for navigating in a new direction.

The DM stopped the first navigation with the values of objectives as $(16.1, 421, 42, 15320, 9652)^T$ and started a new navigation from it. For the new navigation, the DM wanted to know what happens, if he improved the amount of nitrogen (aspiration level 15.7, bound 16.6), the aerator wire power (aspiration level 404 and bound 421), and biogas production (aspiration 10052, bound 8920) by giving up on methanol dosage (aspiration level 46, bound 50) and total suspended solids (aspiration level 15640, bound 15762). With this preference information, the DM continued the navigation until the outcome with the values of objectives as $(15.7, 408, 38, 14852, 9210)^T$. After this, the DM continued the navigation to that direction and when he was happy, he then started the third navigation.

Figure 5b shows the third navigation of the DM (right panel in the figure), where the DM disabled some of the aspiration levels and bounds by checking the corresponding boxes in the left panel. To be more specific, the DM specified aspiration levels for the second, third and fourth objective and bounds for all but the first objective. As can be seen, when the bound is disabled, e.g., for total nitrogen, the navigation can go beyond the bound. In addition, since the total nitrogen did not have an aspiration level either, the objective values did impair. The latter was the case also for the last objective.

The DM continued navigating for two more rounds. Because of space limitations, we do not give the details here. Finally, the DM found a preferable navigated point with the objective values as follows: total nitrogen at 17.9 grams per a cubic meter of effluent, aerator wire power consumption at 413.0 kilowatt hours, methanol dosage rate at 11.3 grams per cubic meter of effluent, total suspended solids at 14800.0 tons per day and biogas production at 9480.0 cubic meters per day. When this point was projected, an actual PO outcome had a total nitrogen at 16.49 grams per cubic meter of effluent, aerator wire power consumption at 423.8 kilowatt hours, methanol dosage rate at 8.6 grams per cubic meter of effluent, total suspended solids at 14662.0 tons per day and biogas production at 9235.3 cubic meters per day. To the DM, the actual PO outcome was also preferable, as it complied with the navigated outcome well and the solution process was completed and the solution could be implemented in the wastewater treatment plant.

The example shows that the method complies well with the properties related to the user experience given in Section 3.2. The DM was able to find a preferable outcome by navigating. In addition, providing aspiration levels and upper bounds for the objectives was intuitive to the DM. Finally, the method did not bound the choices that the DM was able to make while navigating and, thus, the DM was allowed to learn.

7.2. Comparison

We compare the solution process with Nonconvex Pareto Navigator with the one presented in [12], where the same problem was solved by first creating a PAINT approximation to replace the original problem and then the interactive NIMBUS method [26] was applied to the approximate problem. In NIMBUS, whenever the DM has given preferences information, the DM can get one to four PO outcomes corresponding to the preferences. In [12], the DM wanted to study only one approximate outcome for given preferences at a time, and the DM had to give 19 different preferences and study the corresponding approximate PO outcomes before finding a preferable solution. In Nonconvex Pareto Navigator, the type of preference information used was similar but the DM was able to gain the necessary understanding of the interdependencies of the objectives with only five different preferences and, importantly, did not have to wait for solutions corresponding to the preferences being generated. Overall, he could find a preferable solution and get convinced of its goodness faster thanks to the navigation.

The essential difference between the approach in [12] and our approach here is that in the former, the DM only saw one PO outcome for given preferences, while in our approach the DM could see multiple outcomes tending towards and beyond the given preferences which supported learning better, as mentioned in the introduction. When objective values gradually change towards the given preferences, the DM gains more insight about the problem. These differences are essential when comparing the Nonconvex Pareto Navigator method against any other interactive reference point or classification based method that is not a navigation method as defined in this paper.

8. Conclusions

We have presented an interactive multiobjective optimization method that is based on the idea of navigation. As the term navigation has not been well defined previously even though some methods have been proposed, we first characterized the idea of navigation methods and then articulated what is intended to be achieved by navigation as a list of desirable properties. We wanted to extend navigation to nonconvex problems and, thus, paid special attention to challenges of nonconvex problems and computationally expensive problems, where function evaluations are time-consuming.

In order to enable navigation in nonconvex sets we introduced a new type of cones called an e-cone and provided guidelines as how to set its parameters. After presenting the Nonconvex Pareto Navigator method, we discussed its performance with respect to its technical properties and its properties related to user experience.

The Nonconvex Pareto Navigator method was implemented in the IND-NIMBUS® software framework and its functionality was demonstrated with a computationally expensive multiobjective optimization problem related to operating a wastewater treatment plant. With the example, it was demonstrated that the new method supports, in particular, the so-called learning phase of decision making where the DM learns about the interdependencies of the conflicting objectives and attainable outcomes as well as one's own preferences. The new method enables free search in the objective space and provides different ways for the DM to control the navigation and learning.

The preference information employed by the new method consists of aspiration levels and bounds not to be exceeded. If the DM is willing to provide this type of preference information when solving a computationally expensive problem, Nonconvex Pareto Navigator is a viable choice as it enables the DM to learn more efficiently of the interdependencies of the objectives and, thus, find a preferred solution. This was demonstrated with the example considered, which serves as a proof of concept for the new method.

Future research could involve behavioral aspects of decision making in navigation methods, see, e.g., [9]. This could lead to alternative, learning-based interactive decision support systems, possibly extending navigation methods, using an a priori computed approximation of the Pareto optimal front as suggested in this paper. Moreover, if (partial) preference information is known beforehand, e.g., by bounds in trade-offs or some (or all) objective function values, it could be used when generating the approximation.

Acknowledgments

We thank Mr. V. Ojalehto for implementing the Nonconvex Pareto Navigator method and Mr. K. Sahlstedt for providing the data and preferences for the wastewater treatment problem. We also thank Dr. H. Isomäki and Prof. P. Saariluoma for fruitful discussions on psychology of thinking and neurobiological and behavioral aspects of decision making. We express our thanks to the anonymous reviewer for valuable comments that helped to improve this paper. This research is related to the thematic research area Decision Analytics utilizing Causal Models and Multiobjective Optimization (DEMO) of the University of Jyväskylä.

References

[1] T. Aittokoski and K. Miettinen. Efficient evolutionary approach to approximate the Pareto-optimal set in multiobjective optimization, UPS-EMOA. *Optimization Methods and Software*, 25:841–858, 2010.

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

[3] P. Eskelinen, K. Miettinen, K. Klamroth, and J. Hakanen. Pareto Navigator for interactive nonlinear multiobjective optimization. *OR Spectrum*, 32:211–227, 2010.

[4] K. Fukuda. Lecture: Polyhedral Computation, spring, 2016. Lecture notes avaiable at https://www.inf.ethz.ch/personal/fukudak/lect/pclect/notes2016/PolyComp2016.pdf, downloaded on January 31, 2018.

[5] K. Fukuda and A. Prodon. Doube description method revisited. In M. Deza, R. Euler, and I. Manoussakis, editors, *Combinatorics and Computer Science*, volume 1120, pages 91–111. Springer, 1996.

[6] A. M. Geoffrion. Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications*, 22:618–630, 1968.

[7] J.I. Gold and M.N. Shadlen. The neural basis of decision making. *Annual Review of Neuroscience*, 30:535–574, 2007.

[8] J. Hakanen, K. Miettinen, and K. Sahlstedt. Wastewater treatment plant design and operation under multiple conflicting objective functions. *Environmental Modelling & Software*, 46:240–249, 2013.

[9] R. P. Hämälinen, J. Luoma, and E. Saarinen. On the importance of behavioral operational research: the case of understanding and communicating about dynamic systems. *European Journal of Operational Research*, 228:623–634, 2013.

[10] M. Hartikainen, K. Miettinen, and M. M. Wiecek. Constructing a Pareto front approximation for decision making. *Mathematical Methods of Operations Research*, 73:209–234, 2011.

[11] M. Hartikainen, K. Miettinen, and M. M. Wiecek. PAINT: Pareto front interpolation for nonlinear multiobjective optimization. *Computational Optimization and Applications*, 52:845–867, 2012.

[12] M. Hartikainen, V. Ojalehto, and K. Sahlstedt. Demonstrating the applicability of PAINT to computationally expensive real-life multiobjective optimization. *Engineering Optimization*, 47:328–346, 2014.

[13] M. Hasenjäger and B. Sendhoff. Crawling along the Pareto front: tales from the practice. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 174–181. IEEE Press, 2005.

[14] C. Hwang and A. S. M. Masud. *Multiple Objective Decision Making - Methods and Applications: a State-of- the-Art Survey*. Springer, 1979.

[15] I. Kaliszewski. *Soft Computing for Complex Multiple Criteria Decision Making*. Springer, 2006.

[16] P. Korhonen. The multiobjective linear programming decision support system vig and its applications. In C. Bana e Costa, editor, *Readings in Multiple Criteria Decision Aid*, pages 471–491. Springer, 1990.

[17] P. Korhonen. Two decision support systems for continuous and discrete multiple criteria decision making: VIG and VIMDA. In A. Lewandowski, P. Serafini, and M.G Speranza, editors, *Methodology, Implementation and Applications of Decision Support Systems*, pages 85–103. Springer, 1991.

[18] P. Korhonen. Reference direction approach to multiple objective linear programming: historical overview. In M. Karwan, J. Spronk, and J. Wallenius, editors, *Essays in Decision Making: A Volume in Honour of Stanley Zionts*, pages 74–92. Springer, 1997.

[19] P. Korhonen and J. Wallenius. A Pareto race. *Naval Research Logistics*, 35:615–623, 1988.

[20] P. Korhonen and J. Wallenius. Behavioural issues in MCDM: neglected research questions. *Journal of Multi-Criteria Decision Analysis*, 5:178–182, 1996.

[21] P. Korhonen and G. Y. Yu. A reference direction approach to multiple objective quadratic-linear programming. *European Journal of Operational Research*, 102:601–610, 1997.

[22] O. I. Larichev. Cognitive validity in design of decision-aiding techniques. *Journal of Multi-Criteria Decision Analysis*, 3:127–138, 1992.

[23] J. Linder, S. Lindkvist, and J. Sjöberg. Two-step framework for multiobjective optimization. Technical Report LiTH-ISY-R-3043, Linköping University, 2012.

[24] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, 1999.

[25] K. Miettinen, J. Hakanen, and D. Podkopaev. Interactive nonlinear multiobjective optimization methods. In S. Greco, M. Ehrgott, and J. Figueira, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 931–980. Springer, 2 edition, 2016.

[26] K. Miettinen and M. M. Mäkelä. Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research*, 170:909–922, 2006.

[27] K. Miettinen, F. Ruiz, and A. P. Wierzbicki. Introduction to multiobjective optimization: Interactive approaches. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 27–57. Springer, 2008.

[28] M. Monz. *Pareto Navigation – Algorithmic Foundation of Interactive Multi-criteria IMRT Planning*. PhD thesis, University of Kaiserslautern, 2006.

[29] M. Monz, K. H. Küfer, T. R. Bortfeld, and C. Thieke. Pareto Navigation – algorithmic foundation of interactive multi-criteria IMRT planning. *Physics in Medicine and Biology*, 53:985–998, 2008.

[30] V. Ojalehto, K. Miettinen, and T. Laukkanen. Implementation aspects of interactive multiobjective optimization for modeling environments: the case of GAMS-NIMBUS. *Computational Optimization and Applications*, 58:757–779, 2015.

[31] GPS-X Simulator Home Page, Last accessed: November 20, 2018. url = http://www.technotrade.com.pk/20/GPS-X_Waste_Water_Treatment_Simulation_Software/.

[32] V. Pareto. *Manuale di Economia Politica*. Società Editrice Libraria, 1906.

[33] A. Pascoletti and P. Serafini. Scalarizing vector optimization problems. *Journal of Optimization Theory and Applications*, 42:499–524, 1984.

[34] W. L. Price. Global optimization by controlled random search. *Journal of Optimization Theory and Applications*, 40:333–348, 1983.

[35] A. Rangel, C. Camerer, and P.R. Montague. A framework for studying the neurobiology of value-based decision making. *Nature Reviews Neuroscience*, 9:545–556, 2008.

[36] S. Ruzika and M. M. Wiecek. Approximation methods in multiobjective programming. *Journal of Optimization Theory and Applications*, 126:473–501, 2005.

[37] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of Multiobjective Optimization*. Academic Press, Orlando, 1985.

[38] J. Sherman and J. W. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Annals of Mathematical Statistics*, 21:124–127, 1950.

[39] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley & Sons, New York, 1986.

[40] M. Tabatabaei, J. Hakanen, M. Hartikainen, K. Miettinen, and K. Sindhya. A survey on handling computationally expensive multiobjective optimization problems using surrogates: Non-nature inspired methods. *Structural and Multidisciplinary Optimization*, 52:1–25, 2015.

[41] S. Tarkkanen, K. Miettinen, J. Hakanen, and H. Isomäki. Incremental user-interface development for interactive multiobjective optimization. *Expert Systems with Applications*, 40:3220–3232, 2013.

[42] M. M. Wiecek. Advances in cone-based preference modeling for decision making with multiple criteria. *Decision Making in Manufacturing and Services*, 1:153–173, 2007.

[43] A. P. Wierzbicki. On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR Spectrum*, 8:73–87, 1986.

[44] D.A. Worthy, M.A. Gorlick, J.L. Pacheco, D.M. Schnyer, and W.T. Maddox. With age comes wisdom: decision making in younger and older adults. *Psychological Science*, 22(11):1375–1380, 2011.

[45] P. L. Yu. Cone convexity, cone extreme points and nondominated solutions in decision problems with multiobjectives. *Journal of Optimization Theory and Applications*, 14:319–377, 1974.