

Collicott, Cristina et al.

Article — Published Version

Interactive navigation of multiple convex patches

Journal of Multi-Criteria Decision Analysis

Provided in Cooperation with:

John Wiley & Sons

Suggested Citation: Collicott, Cristina et al. (2021) : Interactive navigation of multiple convex patches, Journal of Multi-Criteria Decision Analysis, ISSN 1099-1360, Wiley, Hoboken, NJ, Vol. 28, Iss. 5-6, pp. 311-321,
<https://doi.org/10.1002/mcda.1768>

This Version is available at:

<https://hdl.handle.net/10419/284767>

Standard-Nutzungsbedingungen:

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

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

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

Terms of use:

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

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


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



<http://creativecommons.org/licenses/by-nc-nd/4.0/>

RESEARCH ARTICLE

Interactive navigation of multiple convex patches

Cristina Collicott | Esther Bonacker | Ina Lammel | Katrin Teichert  |
Michal Walczak | Philipp Süß

Department of Optimization, Fraunhofer
ITWM, Kaiserslautern, Germany

Correspondence

Cristina Collicott, Department of Optimization,
Fraunhofer ITWM, Fraunhofer-Platz
1, Kaiserslautern, Germany.
Email: cristina.collicott@itwm.fraunhofer.de

Abstract

Among the approaches to multi-criteria decision making, Pareto navigation is a powerful, interactive tool that has been successfully applied to a variety of real-world problems with continuous decision variables, including chemical process design, drug manufacturing, logistical vehicle routing problems, and radiotherapy treatment planning. However, many real-life problems are formulated using both continuous and binary decision variables. In this work, we introduce *patch navigation* as an algorithmic concept that extends Pareto navigation to this type of problem where the number of binary variables is relatively small. The underlying idea is the navigation across a finite set of individual, convex fronts each associated with a specific configuration of the binary variables (*patches*). We show how the user interactions employed in current Pareto front navigation, namely selection and restriction, can be adopted to handle multiple patches. These routines enable the decision maker (DM) to change the solution in small increments while controlling the related trade-offs. We also describe additional, patch-specific routines that enable the DM to consider only an individually chosen subset of patches in the navigation. To illustrate patch navigation, and to demonstrate its usefulness for real-life problems, we present numerical examples of patch navigation along with an application motivated by radiotherapy planning.

KEYWORDS

decision support, multi-criteria decision making, navigation

1 | INTRODUCTION

Problems within operations research often require the consideration of multiple conflicting criteria. In real-life problems there usually exist multiple solutions exhibiting different trade-offs between the criteria. Multi-criteria decision making (MCDM) provides methods for a decision maker (DM) to find the solution within a set of alternatives that best matches the preferences of the DM. Some approaches in literature attempt to formalise a DM's preference structure first and then

determine the ideal solution within a set of alternatives. In contrast, interactive and a posteriori approaches allow the DM to explore different solutions without explicit definition of their preferences (Miettinen, 1999 and references therein).

Pareto navigation is a powerful, interactive approach to MCDM. A variety of navigation approaches have been developed (Eskelinen et al., 2010; Hartikainen et al., 2019; Korhonen & Wallenius, 1988; Miettinen et al., 2010; Monz, 2006). While each approach is distinctively different, all allow a DM to traverse a set of points within the

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

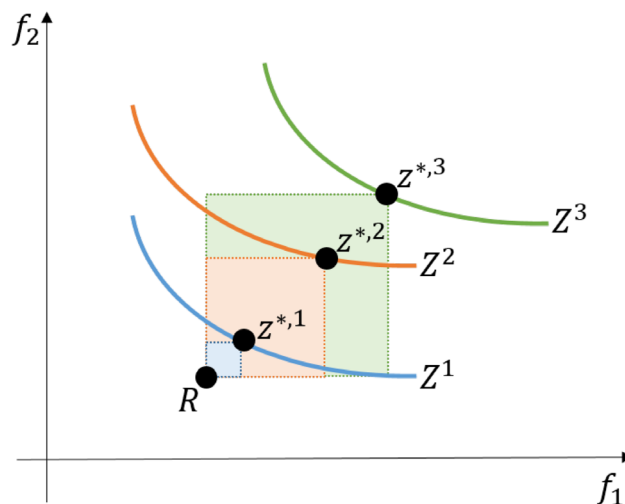
© 2021 The Authors. *Journal of Multi-Criteria Decision Analysis* published by John Wiley & Sons Ltd.

objective space with the goal of finding the DM's preferred solution within the Pareto front. As the DM navigates, they gain insight into the characteristics of the underlying optimisation problem, and through navigation can establish their preferences. Pareto navigation has been successfully applied to many real-world problems, including chemical process design (Bortz et al., 2014; Burger et al., 2014), drug discovery and manufacturing (Farid, 2007), logistical vehicle routing problems (Geiger & Wenger, 2007), and radiotherapy treatment planning (Monz et al., 2008). For a thorough review of navigation in multi-objective optimisation, see Allmendinger et al., 2017, and for a thorough review of the desirable properties of navigation methods – especially for computationally expensive problems – see Hartikainen et al., 2019.

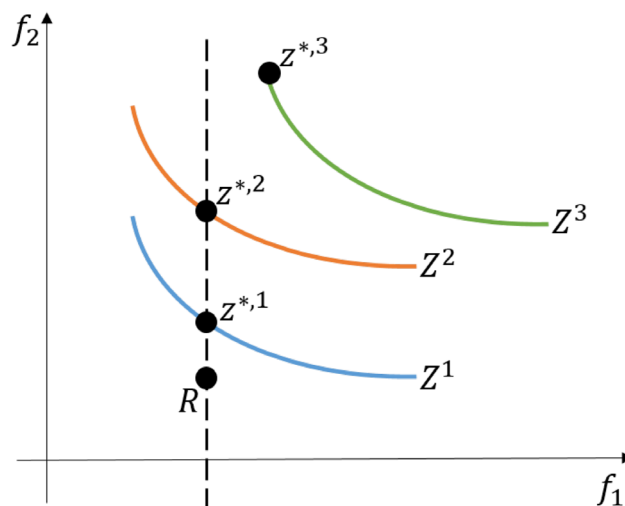
Many real-life problems are formulated using continuous variables for configuring a solution as well as binary variables to model on/off-type selections the DM has to make. In treatment planning, for example, a particular therapy variant (photon radiation vs. proton radiation) could be modeled this way. In chemical mixture problems, the DM might have to select one or several type of binders to include or exclude from the mix, to cite another example. Fixing a choice for all involved binary variables, the remaining continuous variables induce one Pareto front (defined in Section 2) for that particular choice. Consequently, the combinatorial set of all possible selections result in several Pareto fronts.

Within this work, we describe an extension to Pareto navigation as outlined by Monz, 2006 to allow for the navigation across a finite, not too large (≤ 50) set of convex Pareto fronts. With the exception of Craft & Monz, 2010 and Hartikainen et al., 2019, navigation approaches in literature have traditionally focused on navigation of a single Pareto front. The approach in Craft & Monz, 2010 is similar to ours, yet we propose in this work a more streamlined approach to handle situations where navigation takes the DM across multiple Pareto fronts. Moreover, we introduce a new way to compare solutions across different fronts. The work of Hartikainen et al., 2019 focused on the navigation of non-convex Pareto fronts and introduced the ability to navigate disconnected fronts by essentially extrapolating Pareto fronts using “e-cones”. While this extrapolation approach does not likely lead the DM to navigate to infeasible solutions during the exploration of the Pareto front, it is not strictly forbidden and a final projection step, post navigation, is included in the work of Hartikainen et al., 2019 to ensure that a final solution presented to the DM is feasible.

As in Monz, 2006 and Hartikainen et al., 2019, we solve reference point problems to determine solutions on the approximations of the Pareto fronts. Where our method differs crucially from Hartikainen et al., 2019 is that we allow precise control over one of the chosen objectives (similar to Monz, 2006): instead of general aspiration levels for all objectives which are not guaranteed to be reached in Hartikainen et al., 2019, we allow the DM to choose one objective, enforce this aspiration level exactly [the equality constraint in (11)], and produce the best approximation to the aspiration levels in the remaining dimensions. For fronts where the equality cannot be achieved, we propose a second method to determine a solution. This



(a) Solutions for each front to a reference point problem in Hartikainen et al., 2019.



(b) Our solutions to the same reference point problem with the added requirement that f_1 was selected to be achieved exactly.

FIGURE 1 Comparison of navigation solutions between the approach introduced in [?] and here. Given three fronts Z^1, Z^2 and Z^3 and a reference point R , the max-norm approximations to R represent the solutions in (1). In the work presented here, the decision maker also specifies one dimension (in this case f_1) to be achieved exactly. Our solutions $z^{*,1}$ and $z^{*,2}$ in 2 can be directly compared in the remaining dimensions because they share the same f_1 value while comparing the solutions in 1 is only based on their distance to R . $z^{*,3}$ in 2 is the point on Z^3 that comes closest to the required f_1 value (see (12) below) and is, therefore, also comparable to the other front solutions

way we are able to directly compare solutions from different Pareto fronts, allowing the DM to judge the merit of the fronts for exactly the aspiration levels they asked for. Figure 1 highlights this crucial difference for a simple example.

Our method lets the user ad hoc include or exclude different Pareto fronts in the navigation – this is of particular interest in cases

where the Pareto fronts have a real-world quality relevant to decision making. It also gives the DM more control on the navigation process in cases where one or more fronts are partially or fully dominated by another: the distance from the global solution to each feasible solution is communicated to the DM and can be considered in the decision-making process. While the global solution is optimal based on the dimensions of the Pareto front, in cases where the different fronts result from a change in a decision relevant parameter, nearby feasible solutions could still be of interest; this is especially true in cases where a DM may have a slight preference for a specific Pareto front but not want to restrict the navigation to only that front. Furthermore, in cases where a DM requests a desired value for one objective and no feasible solution exists on any patch, this is communicated directly to the DM. This ensures no false promises are made to the DM during the navigation process.

We define this approach here as patch navigation. We outline the preliminaries (Section 2) before discussing patch navigation in detail (Section 3). We present numerical examples of patch navigation (Section 4) along with a real world application motivated by radiotherapy planning (Section 4.2).

2 | PRELIMINARIES

Definition 2.1. (Multicriteria optimisation problem). We consider a multi-criteria optimisation problem

$$\min_{x \in X} F(x) \quad (1)$$

where x is the vector of decision variables, X is the feasible set, and F is a vector-valued function composed of real-valued objectives f_1, \dots, f_n .

Definition 2.2. (Pareto efficiency). A feasible solution $x \in X$ to the multi-criteria optimisation problem 2.1 is called *Pareto efficient*, if there is no $\hat{x} \in X$ with $f_i(\hat{x}) \leq f_i(x)$ for all $i \in \{1, \dots, n\}$ and $f_j(\hat{x}) < f_j(x)$ for some $j \in \{1, \dots, n\}$. We denote by X^* the set of Pareto efficient solutions.

Definition 2.3. (Pareto front). The set $F(X^*)$ we call the Pareto front, and its elements we call *non-dominated*.

For a continuous MCO problem, the set X^* is generally infinite and cannot be calculated entirely. For our a posteriori approach to be applicable, a finite set of representative solutions $X^R \subseteq X$ must be calculated first. A number of algorithms have been devised for this task, either employing specific scalarisation methods or evolutionary approaches, see Ruzika & Wiecek, 2005 for an overview.

If additionally the MCO problem is convex, the lower boundary of the convex hull of the *representation points* $Z = F(X^R)$ can serve as an approximation of the Pareto front (see Figure 2). Assuming the accuracy of the approximation is ensured by an appropriate algorithm such

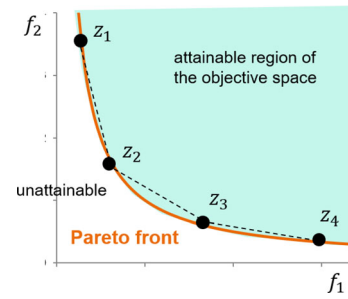


FIGURE 2 For a convex continuous MCO problem, the lower bound of the convex hull of the representation points Z (dashed line) is an attainable approximation of the Pareto front

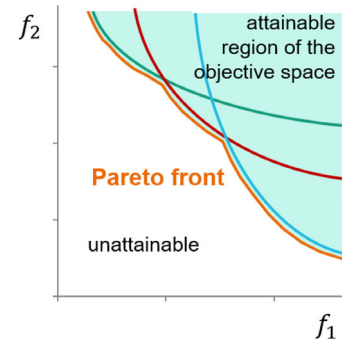


FIGURE 3 A patched Pareto front. The actual front (orange) is non-convex, but composed of convex parts originating from three individual convex fronts (green, red, blue)

as sandwiching (Bokrantz & Forsgren, 2012; Ehrgott et al., 2011; Löhne et al., 2014; Rennen et al., 2010; Serna, 2008), those additional, linearly interpolated objective space points can be included in the point set presented to the DM. In anticipation of the following, we call this the *navigation set*.

The augmentation of the navigation set for a convex continuous MCO problem discussed in the last paragraph is justified in the following sense. By way of the unique mixing coefficients λ_j associated with a chosen interpolated objective space point $F^* = \sum \lambda_j F(x^j)$, a solution $x^* = \sum \lambda_j x^j$ is defined by the corresponding mixing of the representative solutions $X^R = \{x^1, \dots, x^m\}$. Convexity ensures that $F(x^*) \leq F^*$, meaning that the quality of the implicitly chosen solution x^* meets or even exceeds those of the DM's explicitly chosen point F^* .

This paper presents algorithms that allow the navigation of a patched Pareto front. For the term *navigation* we adopt the definition from Allmendinger et al., 2017. Definition 2.5 introduces a patched Pareto front as a potentially non-convex front that is composed of a finite number of convex parts (see Figure 3).

Definition 2.4. (Navigation by Allmendinger et al., 2017). *Navigation* is the interactive procedure of traversing through a set of points (the navigation set) in the objective space guided by a DM. The ultimate goal of

this procedure is to identify the single most preferred Pareto optimal solution.

Definition 2.5. (Patched Pareto front). We call a Pareto front *patched* if it is the union of the non-dominated parts of finitely many individual convex Pareto fronts, which we call *patches*.

3 | PATCH NAVIGATION

3.1 | Features

Patch navigation is an algorithmic concept that extends the concept introduced by Monz, 2006 for navigating a single convex Pareto front to navigating a patched Pareto front (Definition 2.5). It does so by augmenting the underlying calculation routines, making them applicable for a patched Pareto front. Moreover, patch navigation provides additional routines that are specific for the handling of patches.

Patch navigation provides all functionality of the navigation by Monz, 2006. Namely, the DM can change the solution in small increments while observing the related trade-offs (*selection*), and they can set bounds to individual objectives (*restriction*) while monitoring the influence of these bounds on the obtainable ranges in other objectives.

Beyond that, in patch navigation the DM can activate and deactivate individual patches. Activating or deactivating a patch may also change the obtainable ranges, or prompt a change of the current solution. Also, the DM can observe the distance of his chosen solution to the closest solutions on the other patches. This shows if a patch is indispensable for meeting a specific preference, or if a similar solution can also be found on a different patch.

3.2 | Foundations

In the following, we outline the mathematical foundations of the patch navigation routines, starting with some definitions.

3.2.1 | Convex hull approximation, restrictions and implicit bounds

For a patch P_k and representation points $z^1, \dots, z^m \in P_k$, we define the *convex hull approximation* of P_k as

$$Z_k := \left\{ z = \sum_{j=1}^m \lambda_j z^j \mid \sum_{j=1}^m \lambda_j = 1 \right\}. \quad (2)$$

The *restrictions* are lower and upper bounds in the objective space that we denote by r^l and r^u , respectively. For each patch, the restrictions define a *feasible part* of the convex hull approximation Z_k :

$$Z_k^{\text{feas}} := \{ z \in Z_k \mid r^l \leq z \leq r^u \}. \quad (3)$$

For each objective space dimension i and patch k , we define the corresponding *obtainable range* as the interval spanned by the *lower implicit bound* $b_i^{l,k}$ and the *upper implicit bound* $b_i^{u,k}$ as given by

$$b_i^{l,k} := \min_{z \in Z_k^{\text{feas}}} z_i \quad (4)$$

and

$$b_i^{u,k} := \max_{z \in Z_k^{\text{feas}}} z_i. \quad (5)$$

Note that the obtainable range depends on the imposed restrictions r^l and r^u . We denote the vectors of lower and upper implicit bounds over all objective space dimensions $i = 1, \dots, n$ by $b^{l,k}$ and $b^{u,k}$, respectively, and collectively refer to them as the *implicit bounds*. The above Definitions (4) and (5) constitute linear programmes, allowing efficient recomputation of the obtainable ranges whenever the restrictions change.

3.2.2 | Merit function and preference relation

As a basis for determining the patch solutions and the global solution (which we will define later) as results of a navigation interaction, we utilise the *merit function* ϕ . This function is parametrised by a point $r \in \mathbb{R}^n$ which we call the *reference point*, and an objective space dimension index $i \in \{1, \dots, n\}$. These parameters reflect the current state of the navigation, as detailed in the algorithms in Section 3.3. The merit function ϕ is defined as follows:

$$\phi[r, i] : \mathbb{R}^n \rightarrow \{True, False\} \times \mathbb{R} \quad (6)$$

$$\phi[r, i](z) := [\sigma = \sigma[r_i](z), d = d[r, i, \sigma](z)] \quad (7)$$

where

$$\sigma[r_i](z) := \begin{cases} True & \text{if } z_i \leq r_i \\ False & \text{if } z_i > r_i \end{cases} \quad (8)$$

and

$$d[r, i, \sigma](z) := \begin{cases} \max_{j \neq i} \{z_j - r_j\} & \text{if } \sigma \\ z_i - r_i & \text{if } \neg \sigma. \end{cases} \quad (9)$$

Based on the components σ and d of the merit function ϕ defined above, we introduce the following preference relation between points $v, w \in \mathbb{R}^n$ (where $v \succ w$ means that v is preferred to w):

$$v \succ w : \Leftrightarrow (\sigma(v) \wedge \neg \sigma(w)) \vee (d(v) < d(w)). \quad (10)$$

The preference relation expresses that the DM always prefers a solution that fulfils the request $z_i \leq r_i$ over one that does not (this request is, for example, directly communicated via the selection mechanism). If both solutions fulfil the request, the one with the smallest maximal deviation from the reference point in the $n - 1$ other dimensions is preferred (in selection, this corresponds to the maximal trade-off.) If both solutions do not fulfil the request, the one that comes closer is preferred.

Given the above preference relation, we define the patch solution $z^{*,k}$ for patch k as the most preferable feasible point of the convex hull approximation Z_k , and we define the global solution z^* to be the most preferable point among the patch solutions $\{z^{*,k} \mid 1 \leq k \leq N, Z_k^{\text{feas}} \neq \emptyset\}$. In practice, the patch solutions can be calculated by solving linear programmes, as will be detailed next.

3.2.3 | Linear programmes to compute the patch solutions

The task to determine the patch solutions as defined above can be accomplished by solving two linear programmes (11) and (12).

The first programme (11) returns a feasible solution if and only if there is a $z \in Z_k^{\text{feas}}$ for which $\sigma[r_i]$ holds. Moreover, if (11) is feasible, the returned optimal solution fulfils $\sigma[r_i]$ and minimises $d[r, i, \text{True}]$, that is, it is indeed the patch solution $z^{*,k}$.

Whenever (11) does not have a feasible solution, implying that there is no $z \in Z_k^{\text{feas}}$ for which $\sigma[r_i]$ holds, a second linear programme (12) is solved. If (12) has a non-empty feasible set, it returns a solution that minimises $d[r, i, \text{False}]$, thus constituting the patch solution under these circumstances. If on the other hand (12) is also infeasible, then Z_k^{feas} is empty, and there is no patch solution defined for this patch.

Within (11), we denote by e^{-i} the n -dimensional vector with entries $e_j^{-i} = 1$ for $j \neq i$ and $e_i^{-i} = 0$.

$$\begin{aligned} & \min_{\alpha, z, s} \alpha \\ (\alpha, z, s) & \in \mathbb{R} \times Z_k \times \mathbb{R}_+^n \\ r + \alpha e^{-i} & = z + s \\ r^l & \leq z \leq r^u \end{aligned} \quad (11)$$

$$\begin{aligned} & \min_z z_i - r_i \\ & z \in Z_k \\ r^l & \leq z \leq r^u \end{aligned} \quad (12)$$

Note that in the second linear programme, the minimisation objective could equivalently be replaced by $\min_z z_i$. A comparison with the Definition (4) of the lower implicit bound shows that for a solution $z^{*,k}$ of the second linear programme, we have $z_i^{*,k} = b_i^{l,k}$.

It is easy to augment (11) by further linear constraints in the objective space that might be relevant for the problem at hand. Constraints in the decision space X can also be included by adding the solutions x^j to (11) as well and then imposing linear constraints on their convex combinations. Imposing upper or lower limits on input variables in X , for example, is an important feature in some applications.

3.3 | Detailed algorithms

In this section, we provide the pseudo-code algorithms for the actions the DM can perform in patch navigation. Namely, those actions are:

- *Selection*: The DM communicates a desired value for one of the objectives. As a result, they obtain a new global solution.
- *Restriction*: The DM changes the acceptable range for one of the objectives. As a result, the implicit bounds in the other objectives change as well.
- *Patch (de)activation*: The DM includes or excludes one of the patches from contributing. This potentially changes the global solution and the implicit bounds, and affects further selection and restriction.

For all algorithms, we assume that the patch navigation is in a valid state, as given by a consistent assignment of the following set of variables:

- the current global solution z^*
- the current index i^*
- the current lower and upper restrictions r^l and r^u
- the current patch solution $z^{*,k}$ for all patches with $Z_k^{\text{feas}} \neq \emptyset$
- the current patch specific implicit lower and upper bounds $b^{l,k}$ and $b^{u,k}$ for all patches P_k
- whether the patch is currently active a_k for all patches P_k
- the pairwise distances between patch solutions $d^{k,k'}$ according to some distance measure d

3.4 | UI

Concluding Section 3, we now give a user-centred overview of our User interface (UI) components, which implement the concepts described previously. A screenshot of our UI illustrating the components is given in Figure 4.

3.4.1 | Selection and restriction

As in current tools for navigation on a single Pareto front, a slider panel is employed to handle selection (Algorithm 1) and restriction (Algorithm 2). The panel comprises one slider for each objective f_i . Each slider has three moveable elements: a selector (handle), a lower and an upper restrictor (green triangular shapes).

To select a solution the DM drags the selector corresponding to an objective f_i to the desired position, which yields the desired value t in Algorithm 1. We employ the convention that in order to improve the value of a specific objective function in the global solution, the corresponding selector needs to be moved to the left.

Posing a restriction on an objective f_i is done by pulling the corresponding lower or upper restrictor to the desired position, which yields the desired value t for Algorithm 2. By restricting an objective f_i , the DM excludes some parts of the Pareto front. Apart from the explicitly excluded values for f_i , the obtainable ranges for the other

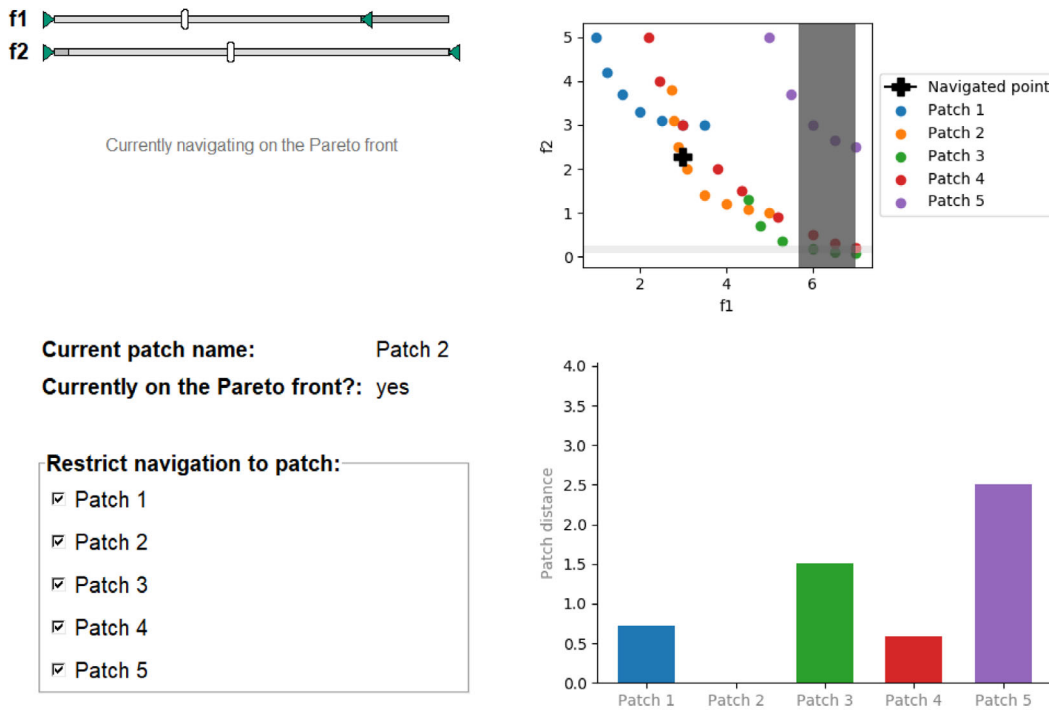


FIGURE 4 Patch navigation UI for example bi-objective problem with five patches. The global solution is highlighted as the navigated point. The UI includes control elements for selection and restriction (sliders, top left) and patch (de)activation (checkboxes, bottom left). Non-control elements include the objective space viewer (top right), along with the distance to other patches (bottom right)

Algorithm 1

Selection

Input: objective index i and desired value t

Start

Calculate reference point r by $r_j = z_j^*$ ($j \neq i$), $r_i = t$.

for $k \in \{1, \dots, N \mid Z_k^{\text{feas}} \neq \emptyset\}$ **do**

 Calculate patch solution z^{*k} based on $\phi[r, i]$.

end for

Set z^* to the most preferable patch solution among the active patches.

Set i^* to i .

End

Output: updated global solution z^* and distances $d^{k,k'}$.

objective functions may also change (as dictated by the implicit bounds). Unobtainable ranges on the sliders are grayed out.

3.4.2 | Objective space viewer

In addition to the slider panel, we offer a configurable visualisation of the representation points in the objective space. We depict both restrictions and implicit bounds in that view as light (implicit bounds) and dark (restrictions) grayed out areas. For objective spaces with

Algorithm 2

Restriction

Input: label $type \in \{lower, upper\}$, objective index i and value t

Start

Update restrictions by $r_i^l = t$ or $r_i^u = t$ depending on $type$.

for $k \in \{1, \dots, N\}$ **do**

 Calculate the implicit bounds $b^{l,k}$ and $b^{u,k}$.

end for

Calculate the global implicit bounds: $b^l = \inf_{k \text{ with } a_k} b^{l,k}$, $b^u = \sup_{k \text{ with } a_k} b^{u,k}$.

for $k \in \{1, \dots, N \mid Z_k^{\text{feas}} \neq \emptyset\}$ **do**

 Calculate patch solution z^{*k} based on $\phi[z^*, i^*]$.

end for

Set z^* to the most preferable patch solution among the active patches.

End

Output: updated global implicit bounds b^l and b^u , potentially updated global solution z^* and distances $d^{k,k'}$.

more than three dimensions, the DM can configure the objective space viewer by choosing three objectives as axes for the plot. While limited to a maximum of three-dimensions at a time, the DM may also

Algorithm 3

Activation and deactivation of patches

Input: bool a and patch index k

Start

Set a_k to a .

Set z^* to the most preferable patch solution among the active patches.

Calculate the global implicit bounds: $b^l = \inf b^{l,k}$,
 $b^u = \sup_{k \text{ with } a_k} b^{u,k}$.

for $k \in \{1, \dots, N \mid Z_k^{\text{feas}} \neq \emptyset\}$ **do**

 Calculate patch solution $z^{*,k}$ based on $\phi[z^*, i^*]$.

end for

End

Output: Potentially updated global implicit bounds b^l and b^u , global solution z^* and distances $d^{k,k'}$.

reconfigure the axes of the viewer during the navigation process to better understand the behaviour of the different patches. In practice, the objective space viewer can be a significant UI element in understanding lower-dimensional problems, but may lose value as the number of dimensions increases.

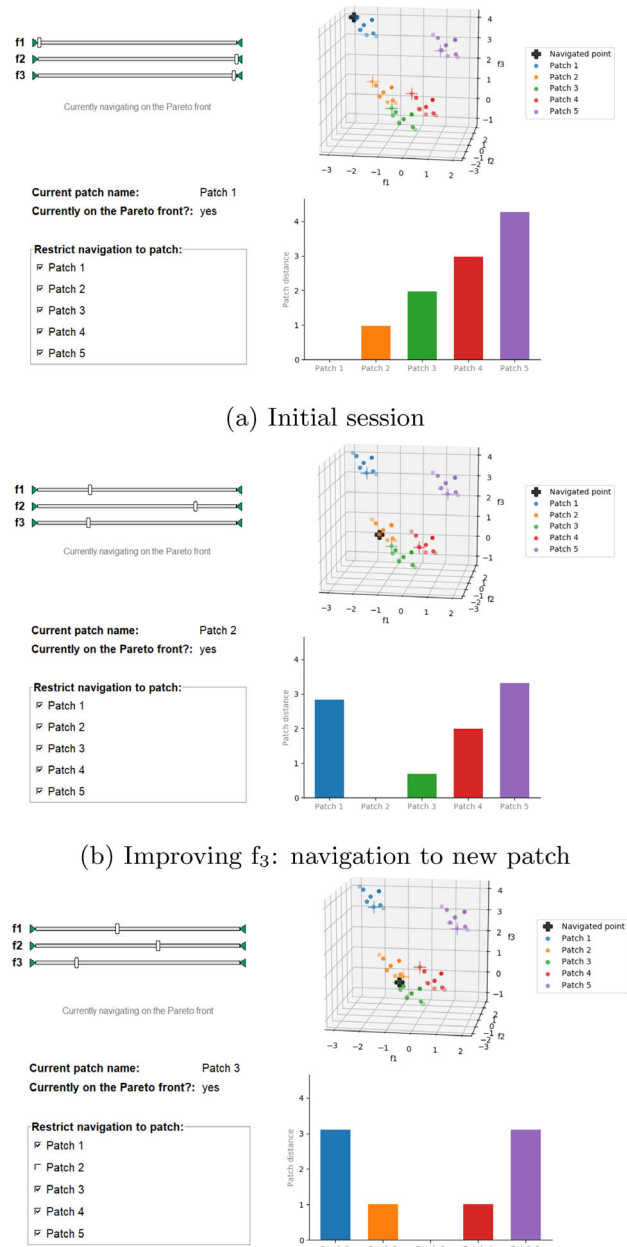
During the navigation process, the patch containing the current global solution may change. In those situations, it is possible that a jump in the selector positions occurs. By marking both the global solution and the individual patch solutions in the objective space viewer, the user can use the objective viewer to gain a better understanding of when and why patch changes or jumps in the selector positions do happen, even in high-dimensional problems.

3.4.3 | Patch (de)activation

Another new feature specific to patch navigation is the (de)activation of a patch (Algorithm 3). In the patch activation component, the DM can exclude individual patches or groups of them from the navigation, or reactivate previously excluded patches. This mechanism is particularly useful in applications where the patches have a real world meaning, for example, correspond to a certain machine, technique or modality. It enables the user to answer questions of the kind ‘‘What would I have to pay (measured in terms of objective functions) if I limit myself to using either machine A or B, but do not consider machines C and D?’’. Activating or deactivating a patch can cause a change in the obtainable ranges as well as the global solution.

3.4.4 | Distance to other patches

Individual patch solutions introduce the concept of distances between patches in a natural way. Given a metric d , the distance from a patch



(c) Patch deactivation: jump to closest available solution

FIGURE 5 Navigation session for tri-objective example problem (13). Here, the patch solutions $z^{*,k}$ are shown in the UI (colored crosses), along with the global solution z^* . Two user interactions and their impact are demonstrated. The decision maker (DM) begins with the initial session (5a) and improves f_3 resulting in a navigation from patch 1 to patch 2 (5b). The DM then deactivates Patch 2, which contains their current global solution, causing the it to jump from Patch 2 to the nearest solution on any active patch (5c), which is in this case on Patch 3

P_k to some other patch $P_{k'}$ can be defined as $d^{k,k'} := d(z^{*,k}, z^{*,k'})$. According to this definition, our patch distance component depicts the distances from the patch containing the current global solution z^* to all other patches.

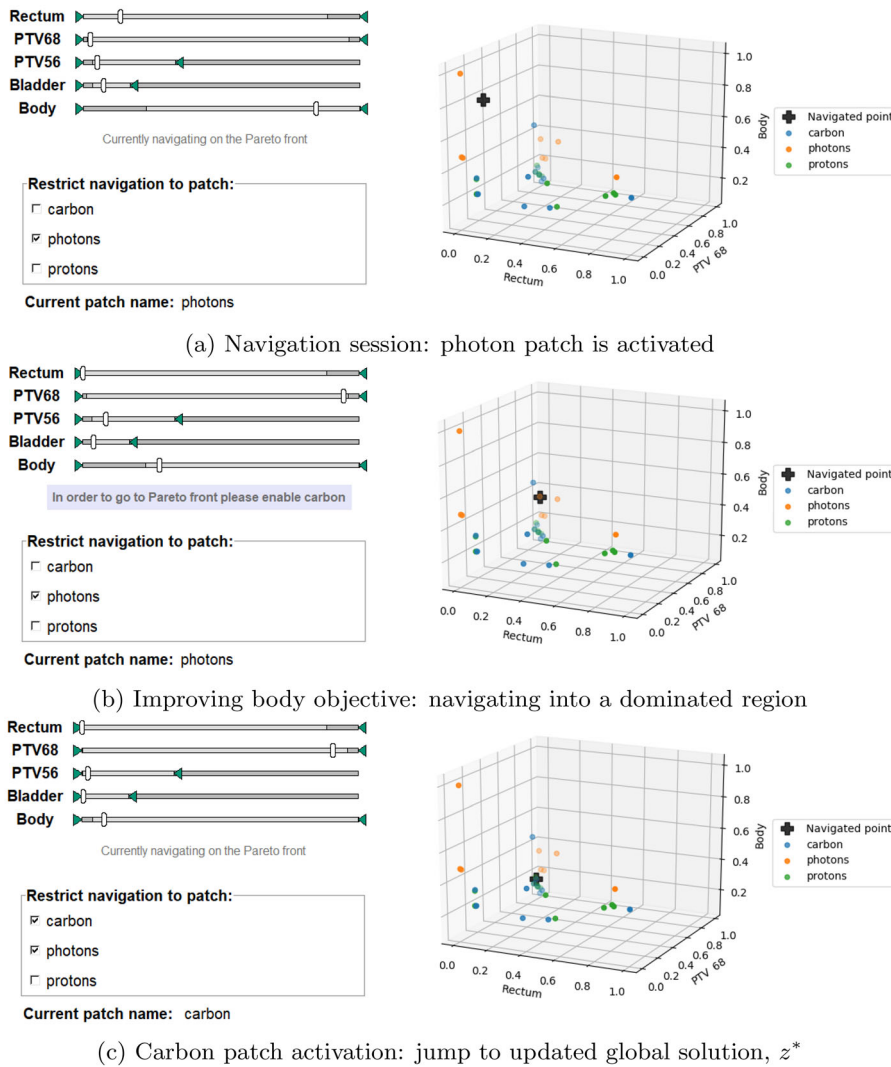


FIGURE 6 Navigation session for the radiotherapy prostate example. The process of improving the body objective is demonstrated. (a) Shows a navigation session with the photon patch activated and a set of selection and restrictions applied to the session. Following this, the decision maker (DM) improves the body objective (b) resulting in a navigation to a dominated region of the photon patch. The DM then activates the carbon patch (c), resulting in a jump of the global solution to the best available solution which is a non-dominated point on the carbon patch

When a patch solution $z^{*,k'}$ exists that is close to the current global solution $z^* = z^{*,k}$, this distance information can aid the user in assessing the benefit of keeping P_k activated. Again, this concept makes particular sense when the patches have inherent meaning, such as them corresponding to a certain machine, technique or modality.

4 | EXAMPLES

In this section, we demonstrate the use of patch navigation for a basic numerical example, along with more complex examples from the field of radiotherapy cancer treatment planning.

4.1 | Tri-objective mixed integer numerical example

We begin with a mixed integer numerical example problem from De Santis et al., 2019:

$$\begin{aligned}
 & f_1 = x_1 + x_4 \\
 \min_x & f_2 = x_2 - x_4, \\
 & f_3 = x_3 + x_4^2, \\
 & \sum_{i=1}^3 x_i^2 \leq 1 \\
 \text{s.t.} & x_i \in [-2, 2] \text{ for all } i = 1, \dots, 4 \\
 & x_4 \in \mathbb{Z}
 \end{aligned} \tag{13}$$

For this problem, each of the five feasible assignments of x_4 corresponds to a patch. For each patch, a set of representation points is determined via a sandwiching algorithm (Serna, 2008). In Figure 5, a navigation session is shown for this problem. Each objective function f_i with $i \in \{1, 2, 3\}$ has an associated slider for selection and restriction. As the DM navigates, the global solution is updated based upon the selection algorithm (Algorithm 1), and the patch distances are updated as well. In the objective space viewer, the patch solutions $z^{*,k}$ are depicted at each step in addition to the current global solution z^* . In Figure 5c, the effect of deactivating the patch containing the current global solution is shown. It results in a jump of z^* according to Algorithm 3.

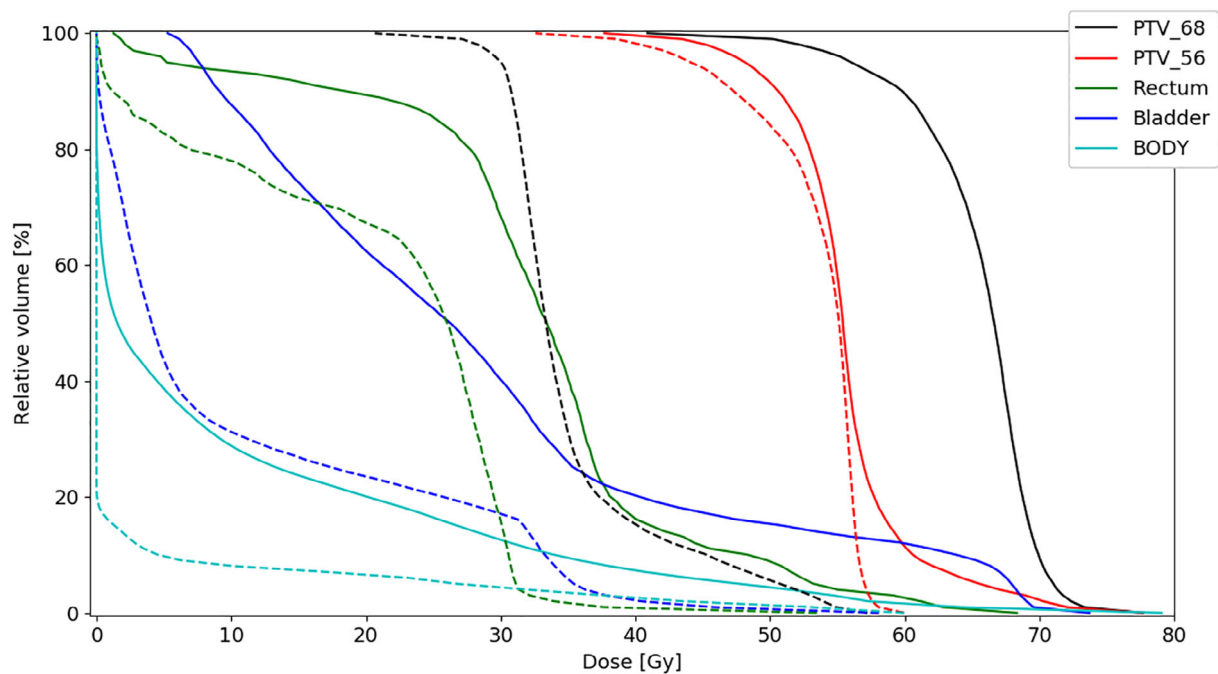


FIGURE 7 Dose-volume histograms are shown for target and OAR structures for the navigation session outlined in Figure 6a (solid) and Figure 6c (dashed). The solid and dashed lines correspond to a photon and carbon ion treatment plan respectively. During the navigation session, the DM improves the body objective and the resulting DVH for the body shows a significant dose reduction for the carbon ion plan. DM, decision maker; DVH, dose-volume histogram; OAR, organs at risk

4.2 | Multiple modality example for radiotherapy

As discussed in Section 1, radiotherapy treatment planning is naturally well suited to be handled as a multi-criteria optimisation problem. Navigation on a single Pareto front has been successfully applied to this field (Hong et al., 2008; Teichert et al., 2017), enabling even less experienced planners to create high-quality treatment plans efficiently (Kierkels et al., 2015). Here, we motivate the use of patch navigation within radiotherapy treatment planning through the exploration of multiple treatment modalities. An example planning problem for a prostate tumor was constructed within the open-source treatment planning system matRad (Wieser et al., 2017). A patch representation was generated for three treatment modalities: carbon ion, proton, and photon. In the case of both proton and carbon plans, a single external beam (angle 180°) was applied. Treatment plans for photons were produced with five equally distributed beams (angles 0° , 72° , 144° , 216° , 288°). In all cases, a couch angle of 0° was applied. A common set of five objective functions was defined for all treatment modalities. Two objectives – PTV68 and PTV56 – correspond to two volume segments of the tumor which should receive a dose of 68 and 56 Gy, respectively, where the squared deviation from the expected dose is minimised. The remaining three objectives correspond to organs at risk (OARs) – rectum, bladder and body – where the squared overdose to each organ should be minimised below a reference dose specific to each organ at risk. Within this work, rectum and bladder dose above 30 Gy, and body dose above 0 Gy, were considered an overdose. When two volumes overlapped, the overlapping voxels

were assigned to volumes based upon an overlap priority. Overlap priorities – in decreasing order of priority – were set as follows: (1) PTV68, (2) PTV56, (3) rectum and bladder and (4) body.

Figure 6 demonstrates the usefulness of patch navigation for this radiotherapy example. The patch approximations generated for each treatment modality are available for navigation, allowing the DM to explore their options across all modalities in parallel. The example navigation session shows the process of improving/reducing the dose to the body OAR. In Figure 6a, navigation is restricted to the photon patch, and the current global solution is on the Pareto front. Although the carbon and proton patches are deactivated, the patch solutions z^{*k} are still computed for all patches, but these solutions do not dominate the position on the photon patch. In Figure 6b, the DM improves the body objective (leftward slider motion), updating the solutions according to Algorithm 1. The solution on the photon patch is now dominated by the parallel solution on the carbon patch. In Figure 6c, the DM activates the carbon patch. As a result, the global solution jumps to the solution on the carbon patch, and the global solution is again Pareto optimal.

When one compares the objective values, indicated by the slider positions, in Figure 6a,c, the objective for the body has significantly improved, along with the other OARs. A small improvement is also seen in the target objective PTV56. However, the target objective PTV68 has significantly worsened.

A dose-volume histogram (DVH) is a standard tool in radiotherapy to visualise the radiation dose to a structure of interest, showing the percentage of the volume of a structure receiving a dose greater than

or equal to a specific dose. In Figure 7, the resulting DVHs are shown for the initial (Figure 6a) and final (Figure 6c) solutions of the example navigation session. Recalling that the DM's goal was to reduce the dose to the body OAR, one can see that the DVH for the body shows a significant improvement. As would be expected based on the objectives, similar improvements can be seen for the other OARs. In the case of the target objectives, PTV68 and PTV56, the objectives are a measure of the dose deviation from 68 and 56 Gy, respectively. PTV56 shows an improvement in the DVH, with a more uniform dose to the structure around 56 Gy. However, the DVH for the PTV68 shows a significant reduction in the dose and thus a significant deviation of the dose from 68 Gy.

5 | CONCLUSIONS

In this paper, we introduce patch navigation as a new method to facilitate navigation on a patched Pareto front. The method is suitable for multi-criteria problems involving binary variables, which model on/off-type selections. These problems naturally occur in practical applications, where different techniques, modalities, machines or similar can be chosen out of a finite set.

Patch navigation comprises three algorithms – selection, restriction and activation of patches – which we describe in detail. We apply the method to one numerical example and one real-world application from radiotherapy planning.

Depending on the method used to solve the linear programmes, the algorithms presented in this paper can be implemented such that they are polynomial in both the dimension and the number of patches. For all algorithms, two linear programmes per patch are solved at most to update the patch solutions. For the restriction algorithm, another two linear programmes are solved per patch to update the implicit bounds. Finding the global solution among the patch solutions is linear in the dimension and quadratic in the number of patches.

As a prerequisite for patch navigation, the patches must be approximated by a finite set of points to a reasonable accuracy. In general, obtaining such an approximation gets exponentially costlier with increased dimension. For convex fronts, dimensions up to 15 are feasible (Bokrantz & Forsgren, 2012). For a realistic application scenario with dimension ≤ 15 and a number of patches ≤ 50 , patch navigation can be expected to be close to real-time, thus facilitating its use for interactive navigation.

In this paper, we require that the patches are convex – otherwise, the patch solutions and implicit bounds cannot be calculated by solving the linear programmes as described. However, presuming an appropriate method existed to calculate the patch solutions and implicit bounds for (certain types of) non-convex patches also, the algorithms for selection, restriction and patch activation given in pseudo-code would be immediately applicable.

One practical concern of patch navigation is that, unlike in the case of a single convex front, jumps of one or more selectors during navigation can occur. If such a jump occurs unexpectedly, the DM may feel out of control, or may be unable to judge the involved

trade-off. The problem is ameliorated to some extent if the jump is foreseeable in the objective space viewer. However, this workaround does not translate well to higher dimensions. It would be worthwhile to investigate additional mechanisms that could be applied in such an instance, for example, a preview of possible selector positions.

Another concern is that the patches are only approximated up to a certain accuracy, and that the remaining inaccuracy may in some instances produce navigation behaviour that is hard to comprehend for the user. One idea to address this in the UI would be to optically “blur” both the distance display and the actual navigation point choice. In this way, when multiple patches are equal up to differences that can be attributed to approximation inaccuracy only, the user can clearly see this in the UI.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Katrin Teichert  <https://orcid.org/0000-0002-2293-407X>

REFERENCES

- Allmendinger, R., Ehrgott, M., Gandibleux, X., Geiger, M. J., Klamroth, K., & Luque, M. (2017). Navigation in multiobjective optimization methods. *Journal of Multi-Criteria Decision Analysis*, 24(1–2), 57–70. <https://doi.org/10.1002/mcda.1599>
- Bokrantz, R., & Forsgren, A. (2012). An algorithm for approximating convex pareto surfaces based on dual techniques. *INFORMS Journal on Computing*, 25(2), 377–393. <https://doi.org/10.1287/ijoc.1120.0508>
- Bortz, M., Burger, J., Asprion, N., Blagov, S., Böttcher, R., Nowak, U., Scheithauer, A., Welke, R., Küfer, K.-H., & Hasse, H. (2014). Multi-criteria optimization in chemical process design and decision support by navigation on pareto sets. *Computers & Chemical Engineering*, 60(10), 354–363.
- Burger, J., Asprion, N., Blagov, S., Böttcher, R., Nowak, U., Bortz, M., Welke, R., Küfer, K., & Hasse, H. (2014). Multi-objective optimization and decision support in process engineering – Implementation and application. *Chemie Ingenieur Technik*, 86, 1065–1072. <https://doi.org/10.1002/cite.201400008>
- Craft, D., & Monz, M. (2010). Simultaneous navigation of multiple pareto surfaces, with an application to multicriteria imrt planning with multiple beam angle configurations. *Medical Physics*, 37(2), 736–741. <https://doi.org/10.1118/1.3292636>
- De Santis, M., Eichfelder, G., Niebling, J., & Rocktäschel, S. (2019). *Solving multiobjective mixed integer convex optimization problems*. Technische Universität Ilmenau, Institut für Mathematik Retrieved from <https://books.google.de/books?id=h0euzQEACAAJ>
- Ehrgott, M., Shao, L., & Schöbel, A. (2011). An approximation algorithm for convex multi-objective programming problems. *Journal of Global Optimization*, 50, 397–416. <https://doi.org/10.1007/s10898-010-9588-7>
- Eskelinen, P., Miettinen, K., Klamroth, K., & Hakanen, J. (2010). Pareto navigator for interactive nonlinear multiobjective optimization. *OR Spectrum*, 32, 211–227. <https://doi.org/10.1007/s00291-008-0151-6>
- Farid, S. S. (2007). Process economics of industrial monoclonal antibody manufacture. *Journal of chromatography*, 848(1), 8–18. <https://doi.org/10.1016/j.jchromb.2006.07.037>
- Geiger, M. J., & Wenger, W. (2007). On the interactive resolution of multi-objective vehicle routing problems. In S. Ōbayashi (Ed.), *Evolutionary*

- multi-criterion optimization*. Springer. <https://doi.org/10.1007/978-3-540-70928-2>
- Hartikainen, M., Miettinen, K., & Klamroth, K. (2019). Interactive nonconvex pareto navigator for multiobjective optimization. *European Journal of Operational Research*, 275, 238–251. <https://doi.org/10.1016/j.ejor.2018.11.038>
- Hong, T. S., Craft, D. L., Carlsson, F., & Bortfeld, T. R. (2008). Multicriteria optimization in intensity-modulated radiation therapy treatment planning for locally advanced cancer of the pancreatic head. *International Journal of Radiation Oncology Biology Physics*, 72(4), 1208–1214. <https://doi.org/10.1016/j.ijrobp.2008.07.015>
- Kierkels, R. G., Visser, R., Bijl, H. P., Langendijk, J. A., van't Veld, A., Steenbakkers, R. J., & Korevaar, E. W. (2015). Multicriteria optimization enables less experienced planners to efficiently produce high quality treatment plans in head and neck cancer radiotherapy. *Radiation Oncology*, 10(87), 87. <https://doi.org/10.1186/s13014-015-0385-9>
- Korhonen, P., & Wallenius, J. (1988). A pareto race. *Naval Research Logistics*, 35(6), 615–623. <https://doi.org/10.1002/1520-6750>
- Löhne, A., Rudloff, B., & Ulus, F. (2014). Primal and dual approximation algorithms for convex vector optimization problems. *Journal of Global Optimization*, 60(4), 713–736. <https://doi.org/10.1007/s10898-013-0136-0>
- Miettinen, K. (1999). *Nonlinear multiobjective optimization*. Kluwer Academic Publishers. <https://doi.org/10.1007/978-1-4615-5563-6>
- Miettinen, K., Eskelinen, P., Ruiz, F., & Luque, M. (2010). Nautilus method: An interactive technique in multiobjective optimization based on the nadir point. *European Journal of Operational Research*, 206(2), 426–434. <https://doi.org/10.1016/j.ejor.2010.02.041>
- Monz, M. (2006). Pareto navigation – Interactive multiobjective optimization and its application in radiotherapy planning (Doctoral dissertation). Universität Kaiserslautern. University of Kaiserslautern - Department of Mathematics.
- Monz, M., Küfer, K., Bortfeld, T., & Thieke, C. (2008). Pareto navigation: Algorithmic foundation of interactive multi-criteria imrt planning. *Physics in Medicine and Biology*, 53(4), 985–998. <https://doi.org/10.1088/0031-9155/53/4/011>
- Rennen, G., van Dam, E. R., & den Hertog, D. (2010). Enhancement of sandwich algorithms for approximating higher-dimensional convex pareto sets. *INFORMS Journal on Computing*, 23(4), 493–517. <https://doi.org/10.1287/ijoc.1100.0419>
- Ruzika, S., & Wiecek, M. (2005). A survey of approximation methods in multiobjective programming. *Journal of Optimization Theory and Applications*, 126(3), 473–501.
- Serna, J. I. (2008). *Approximating the nondominated set of $r+$ convex bodies* (Master thesis). Technische Universität Kaiserslautern.
- Teichert, K., Currie, G., Küfer, K.-H., Miguel-Chumacero, E., Süß, P., Walczak, M., & Currie, S. (2017). Targeted multi-criteria optimisation in imrt planning supplemented by knowledge based model creation. *Operations Research for Health Care*, 23, 100185. <https://doi.org/10.1016/j.orhc.2019.04.003>
- Wieser, H.-P., Cisternas, E., Wahl, N., Ulrich, S., Stadler, A., Mescher, H., Müller, L.-R., Klinge, T., Gabrys, H., Burigo, L., Mairani, A., Ecker, S., Ackermann, B., Ellerbrock, M., Parodi, K., Jäkel, O., & Bangert, M. (2017). Development of the open-source dose calculation and optimization toolkit matrad. *Medical Physics*, 44(6), 2556–2568. <https://doi.org/10.1002/mp.12251>

How to cite this article: Collicott, C., Bonacker, E., Lammel, I., Teichert, K., Walczak, M., & Süß, P. (2021). Interactive navigation of multiple convex patches. *Journal of Multi-Criteria Decision Analysis*, 28(5–6), 311–321. <https://doi.org/10.1002/mcda.1768>