NAUTILUS Navigator: Free Search Interactive Multiobjective Optimization Without Trading-off

Ana B. Ruiz and **Francisco Ruiz**

Department of Applied Economics (Mathematics), Universidad de Málaga, Calle Ejido 6, 29071 Málaga, Spain, email: abruiz@uma.es, rua@uma.es

Kaisa Miettinen

University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland, email: kaisa.miettinen@jyu.fi

Laura Delgado-Antequera

Programa de Doctorado de Economía y Empresa de la Universidad de Málaga, Campus El Ejido s/n, 29071 Málaga, Spain, email: lauda1g10@uma.es

Vesa Ojalehto

University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland, email: vesa.ojalehto@jyu.fi

Abstract

We propose a novel combination of an interactive multiobjective navigation method and a trade-off free way of asking and presenting preference information. The NAUTILUS Navigator is a method that enables the decision maker (DM) to navigate in real time from an inferior solution to the most preferred solution by gaining in all objectives simultaneously as (s)he approaches the Pareto optimal front. This means that, while the DM reaches her/his most preferred solution, (s)he avoids anchoring around the starting solution and, at the same time, sees how the ranges of the reachable objective function values shrink without trading-off. The progress of the motion towards the Pareto optimal front is also shown and, thanks to the graphical user interface, this information is available in an understandable form. The DM provides preference information to direct the movement in terms of desirable aspiration levels for the objective functions, bounds that are not to be exceeded as well as the motion speed. At any time, (s)he can change the navigation direction and even go backwards if needed. One of the major advantages of this method is its applicability to any type of problem, as long as an approximation set of the Pareto optimal front is available and, particularly, to problems with time-consuming function evaluations. Its functionality is demonstrated with an example problem.

Keywords: Decision support systems, Multicriteria decision making, Interactive methods, Graphical user interface, Trade-off free, Navigation

1 Introduction

Multiobjective optimization methods are needed whenever optimization problems have multiple conflicting objective functions. Such problems occur in various application domains [see e.g. 34]. Because of the conflicting nature of the objective functions, there does not exist a single solution where all objective functions can reach their individual optima but, instead, there exist so-called Pareto optimal solutions involving different trade-offs between the objective functions. This means that moving from one Pareto optimal solution to another one implies an impairment in, at least, one objective function in order to improve the value(s) of some other objective function(s).

Since Pareto optimal solutions cannot be ordered without additional preference information, typically coming from a human *decision maker* (DM), different multiobjective optimization methods can be classified according to the role of the DM in the solution process [12, 20]. If no DM is available, some *no-preference method* must be used, while if the DM expresses one's preferences before the solution process, *a priori methods* are applicable. The DM can also express preferences once a representative set of Pareto optimal solutions has been generated and, thus, apply some *a posteriori method*. In *interactive methods*, the DM takes part in the solution process and iteratively sees information about the solutions available, expresses and fine-tunes or even changes one's preference information. The advantages of interactive methods include that the DM can learn about what kind of solutions are attainable, with no need to handle large amounts of data at a time, and that (s)he can adjust one's preferences based on insight gained of the problem. Many interactive methods have been developed (see e.g. [19, 20] and references therein) and they typically differ from each other in the type of information shown to and asked from the DM, and also in the way of generating new solutions based on the preference information expressed.

One appealing way of interaction is navigation. As stated in [1] regarding interactive methods, "navigation is the interactive procedure of traversing through a set of points (the navigation set) in the objective space guided by a decision maker (DM). The ultimate goal of this procedure is to identify the single most preferred Pareto optimal solution". Recently, many neurobiological and behavioral experiments have shown that decision-making tasks include a learning phase and a decision phase [e.g. 39], which often are also inherent in computational models of decision-making [e.g. 28]. In the learning phase of multiobjective optimization, the DM explores different solutions in order to find the most interesting region of the Pareto optimal front. The final solution is then to be identified in the decision phase. In particular, navigation methods support the learning phase. Instead of directly jumping to the solution that best matches with the preferences of the DM, the idea of navigation methods is to continuously show to the DM how objective function values evolve when moving from the current solution along some direction. By analyzing in real time how objective function values change, the DM can gain insight of the interdependencies involved as well as (s)he can understand the feasibility of one's own preferences and modify them (i.e. change one's mind) if needed. Thus, navigation indeed supports the learning phase of decision-making. These methods are 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 [10].

Different methodologies to navigate towards the most preferred Pareto optimal solution have been introduced earlier. In some approaches, the DM is shown Pareto optimal solutions towards the direction (s)he is navigating in the Pareto optimal front, but they are aimed at working with linear multiobjective optimization problems [7, 15, 17] or with a surrogate model of the Pareto optimal front designed for convex problems [9, 24] or nonconvex problems [11]. Nevertheless, the only information seen by the DM in these methods is limited to solutions of an approximation set of the Pareto optimal front.

All of the previously mentioned methods (and in fact, most of the existing interactive methods) are based on considering Pareto optimal solutions, which implies that the DM must be willing to trade-off in order to get different Pareto optimal solutions. But, as suggested by [14] and [16], people do not react symmetrically to gains and losses, and therefore, the DM may be reluctant to sacrifice any objective function value. In fact, [13] states that "trade-off conflict is a major source of decisional stress", while [2] says that "choice sets that are high in trade-off conflict led to less accurate decision making". Besides, our past experiences affect our hopes and this may lead to the so-called anchoring effect [see, e.g. 4], where the DM fixes one's thinking on some (possible irrelevant) information. The study presented in [29] claims that trading-off among several Pareto optimal solutions is emotionally taxing and increases the withdrawal motivation, and supports that the emotional attachment to previously seen "good" solutions emphasizes the loss aversion and the endowment effect. This issue has already been taken into account in methods such as the interior primal-dual multiobjective linear programming algorithm presented in [3].

On the other hand, if the navigation took place among points which are not Pareto optimal, simultaneous gain in all the objective functions would be possible without sacrificing any of the current values. Corresponding win-win strategies are applied, e.g., in the negotiation literature [27] and ethical decision making [26]. Of course, we do not question the importance and meaning of having a Pareto optimal solution as the final solution of the navigation process, but we will just propose an alternative way of getting there. In addition, observations in real life confirm that if (s)he first sees a very unsatisfactory solution, a somewhat better solution is likely to be more satisfactory than otherwise. As formulated in [14], "the past and present context of experience defines an adaptation level, or reference point, and stimuli are perceived in relation to this reference point."

To overcome the disadvantages of trading-off, different methods in the so-called NAUTILUS family have been proposed [21, 22, 23, 31]. They all share the philosophy that the solution process starts from the worst possible objective function values and the DM iteratively takes steps towards the set of Pareto optimal solutions, directing the process with her/his preferences. Thus, at each iteration, (s)he can gain in each objective function and no trade-offs are needed, and only the last solution is Pareto optimal. This enables the DM to conduct the search without the pain of sacrificing and find the most preferred Pareto optimal solution without the fear of anchoring. As the Pareto optimal front is approached, the ranges of the objective function values which are reachable from the current iteration shrink, and for this reason, the DM is shown these reachable ranges after every iteration takes place.

In this paper, inspired by the navigation ideas of the Pareto Navigator method [9] and the NAUTILUS philosophy, we propose a trade-off free navigation method called *NAUTILUS Navigator*. It features a new way of presenting information in multiobjective optimization methods using an effective graphical user interface which supports an easy interpretation of relevant information. It allows the DM to freely navigate in real time from undesirable objective function values towards the most preferred solution of the Pareto optimal front. In our method, instead of Pareto optimal solutions, what the DM sees in real time is the evolution of the reachable ranges of each objective function. Therefore, unlike in other navigation methods, the DM gets information which goes beyond the values that the objective functions have at the solutions of an approximation set of the Pareto optimal front. In [24], the DM is provided with the ranges of reachable solutions, but the navigation is done using a single objective at a time, and not all of them at the same time, as in NAUTILUS Navigator. The new method essentially differs from previous NAUTILUS versions in the interaction style and how information is visualized.

The starting point of NAUTILUS Navigator is a pre-generated approximation of the Pareto optimal front. This discrete set is to be generated before involving the DM, using a appropriate a posteriori method. Then, the DM gets involved in the interactive navigation. To converge from the initial undesirable objective function values towards her/his most preferred solution, the DM gives preference information (usually, in the form of a reference point) which determines a movement direction. Then, the DM dynamically sees how the reachable ranges of the objective functions change when travelling towards the Pareto optimal front and, according to the information visualized, (s)he can stop the movement at any time and provide new preference information (in terms of a new reference point or bounds not to be exceeded). Besides, like in [9] or [15], the DM can also set (and change at any time) the navigation speed. One can also return to any of the previously seen solutions and re-specify preferences there in order to define a new direction. Thus, the speed of finding the final solution is not an end itself but the confidence of the DM that (s)he has found a satisfactory solution.

Visualization is a core element of any interactive method in which the DM navigates, so the design of an intuitive graphical user interface plays a crucial role in practical applications. It is important that the user interface is easy to use and supports the DM in gaining insight into the problem and the interdependencies among the objective functions. Thus, here we also propose a graphical user interface for NAUTILUS Navigator that supports the decision-making process. As mentioned before, the interface is based on the visualization of the evolution of the reachable ranges for each objective function. A video containing the solution process for the example that will be considered in Section 4 can be seen at `https://desdeo.it.jyu.fi/nautilus-navigator`. Furthermore, to extend the applicability of NAUTILUS Navigator, its source code is publicly available at `https://desdeo.it.jyu.fi` to the community of people interested in solving multiobjective optimization problems.

Another advantage of NAUTILUS Navigator is that, since it utilizes an approximation of the Pareto optimal front, any type of problem can be handled as long as an appropriate method can generate this approximation in a reliable way. This feature makes NAUTILUS Navigator well suited for problems which involve expensive function evaluations, as it does not matter for the DM if the generation takes hours or days. When the actual interaction takes place, the original multiobjective optimization problem, which may have time-consuming simulations, is not solved and thus the navigation takes place in real time without introducing any waiting times for the DM. There exist other approaches such as e.g. [37], where the DM can explore pre-generated solutions, but do not include interactive navigation aspects.

The three features described above, that is, the combination of the navigation methodology with the trade-off free philosophy, the visualization of the evolution of the reachable ranges of the objective functions in real time, and the ability to handle a broad range of multiobjective optimization problems, constitute the main contributions of the NAUTILUS Navigator method. As said, trading-off is a source of decisional stress and may cause the DMs to employ a low number of iterations. We can avoid these shortcomings by combining navigation with a trade-off free search, as we do. The way the user interface has been designed allows the DM to perceive the solution process as a continuous navigation instead of discrete iterations. All these properties should facilitate the learning process of the DM about the problem and the feasibility of one's preferences. To the best of our knowledge, our way of visualizing reachable ranges dynamically changing while navigating in a trade-off free fashion has not been considered before.

The rest of this paper is organized as follows. In Section 2, basic concepts and tools utilized in the paper are introduced and outlined. The NAUTILUS Navigator method is proposed in Section 3 and the graphical user interface is also described there. The applicability of the new method is illustrated with an example in Section 4. Finally, the paper is concluded in Section 5.

2 Formulation and background concepts

Let $S \subset \mathbb{R}^n$ be the *feasible set* of solutions or decision vectors in the decision space $\mathbb{R}^n$ of a *multiobjective optimization problem*, where we wish to minimize $k \geq 2$ objective functions $f_i : S \to \mathbb{R}$, for $i = 1, \ldots, k$, simultaneously. Without loss of generality, the multiobjective optimization problem can be defined as follows:

$$\begin{array}{ll} \min & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\} \\ \text{subject to} & \mathbf{x} \in S. \end{array} \tag{1}$$

The image of each decision vector $\mathbf{x} = (x_1, \ldots, x_n)^T \in S$, $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T$, is called an *objective vector* of the *feasible objective set* $\mathbf{f}(S)$, which is a subset of the objective space $\mathbb{R}^k$.

When dealing with a multiobjective optimization problem, in general, it is not possible to find a unique optimal solution where all the objectives can reach their individual optima at the same time. So, the concept of an optimal solution needs to be generalized to the concept of Pareto optimal solutions, which are feasible solutions at which no objective can be improved without impairing, at least, one of the others. Given two vectors in the objective space, $\mathbf{z}^1, \mathbf{z}^2 \in \mathbb{R}^k$, we say that $\mathbf{z}^1$ *dominates* $\mathbf{z}^2$ if $z_i^1 \leq z_i^2$ for every $i = 1, \ldots, k$ and $z_j^1 < z_j^2$ for, at least, one $j \in \{1, \ldots, k\}$. Otherwise, if $\mathbf{z}^1$ and $\mathbf{z}^2$ do not dominate each other, we say that $\mathbf{z}^1$ and $\mathbf{z}^2$ are *mutually nondominated*. Then, a feasible decision vector $\mathbf{x}^* \in S$ is said to be a *Pareto optimal solution* if there does not exist another decision vector $\mathbf{x} \in S$ such that $\mathbf{f}(\mathbf{x})$ dominates $\mathbf{f}(\mathbf{x}^*)$. Its corresponding image, $\mathbf{f}(\mathbf{x}^*)$, is called a *Pareto optimal objective vector*. Usually, there are many Pareto optimal solutions. The set of all the Pareto optimal solutions is the *Pareto optimal set*, denoted by E, and its image in the objective space, $\mathbf{f}(E)$, is referred to as the *Pareto optimal front*. In this paper, we refer to objective vectors which map with decision vectors as *solutions*, and to vectors in the objective space which do not necessarily correspond to any decision vector as *points*. Besides, for a point $\mathbf{z} \in \mathbb{R}^k$, we say that a solution $\mathbf{x} \in S$ is *reachable from* $\mathbf{z}$ if $\mathbf{f}(\mathbf{x})$ dominates $\mathbf{z}$. By a *reachable region in the Pareto optimal set from* $\mathbf{z}$ we refer to the subset of Pareto optimal solutions $\mathbf{x} \in E$ which are reachable from $\mathbf{z}$. Moreover, the image in $\mathbb{R}^k$ of this reachable region from $\mathbf{z}$ defines the reachable values of the objective functions from $\mathbf{z}$, which we will also call the *reachable ranges* from $\mathbf{z}$. Finally, if there exists at least one solution $\mathbf{x} \in S$ that is reachable from $\mathbf{z}$, we say that the point $\mathbf{z}$ is *achievable*.

The ranges of the values that the objective functions take in the Pareto optimal set are defined by so-called ideal and nadir points. An *ideal point* is defined as $\mathbf{z}^\star = (z_1^\star, \ldots, z_k^\star)^T$, where $z_i^\star = \min_{\mathbf{x} \in S} f_i(\mathbf{x}) = \min_{\mathbf{x} \in E} f_i(\mathbf{x})$ $(i = 1, \ldots, k)$, and its components are the best values that the objective functions can have individually in the Pareto optimal set. The worst possible values for each of the objective functions in the Pareto optimal set are the components of a *nadir point*, denoted by $\mathbf{z}^{\mathrm{nad}} = (z_1^{\mathrm{nad}}, \ldots, z_k^{\mathrm{nad}})^T$ with $z_i^{\mathrm{nad}} = \max_{\mathbf{x} \in E} f_i(\mathbf{x})$ $(i = 1, \ldots, k)$. In practice, the nadir point is difficult to calculate because, usually, the set E is unknown, and normally $\mathbf{z}^{\mathrm{nad}}$ is estimated using different approaches (see e.g. [35]). Finally, we define a *utopian point* $\mathbf{z}^{\star\star} = (z_1^{\star\star}, \ldots, z_k^{\star\star})^T$, which is strictly better than the ideal point, as $z_i^{\star\star} = z_i^\star - \varepsilon$ $(i = 1, \ldots, k)$, where ε is a small positive scalar.

In a mathematical sense, Pareto optimal solutions are incomparable without additional information about the preferences of a DM and, thus, the solution process comprises a decision making phase to identify the Pareto optimal solution which satisfies the DM the most. This solution is referred to as the *most preferred solution*. There exist many possibilities for the DM to express the preference information, such as desirability of trade-offs, reference points formed by desirable objective function values (denoted by $\mathbf{q} = (q_1, \ldots, q_k)^T \in \mathbb{R}^k$, with q_i the desirable aspiration level for the objective f_i, for $i = 1, \ldots, k$), classification of objective functions, marginal rates of substitution or choosing one among several solutions (see e.g. [20, 33]).

3 NAUTILUS Navigator: A new way for interacting with the DM

NAUTILUS Navigator is an interactive method, enhanced by a graphical visualization interface, for general multiobjective optimization problems that enables the DM to freely navigate in real time from the nadir point, or any undesirable point, towards the Pareto optimal front until her/his most preferred solution is found. On the one hand, the solution process is trading-off-free, and on the other hand, the navigation ideas are extended to search for a final solution by progressively moving through the whole feasible region in real time. The information shown to the DM basically consists on the visualization of the evolution of the reachable ranges for each objective function from the current point. In what follows, for the sake of simplicity, we will refer to "reachable ranges from the current point" just as the *reachable ranges*.

The general idea of NAUTILUS Navigator is the following. First, a discrete set approximating the Pareto optimal front is generated using an appropriate a posteriori method. The method starts the navigation from an undesirable point (either provided by the DM, or an estimation of the nadir point obtained using the approximation set). For simplicity, in what follows, we will refer to it as the nadir point in both cases. To converge from this initial point towards her/his most preferred solution, the DM gives preference information which determines a movement direction from the current point and a speed for this movement. On the one hand, the movement direction is specified by giving aspiration levels for the objective functions, which form a so-called reference point. Each aspiration level must lie within the reachable range of the corresponding objective function, which is initially defined by the ideal and nadir values. On the other hand, the DM specifies the movement speed towards the Pareto optimal front by selecting one of the following options: "very slow", "slow", "intermediate speed", "fast" and "very fast". For simplicity, we here denote them by integer numbers ranging from 1 (minimum speed) to 5 (maximum speed). The speed scale can be changed if the DM wishes so.

Additionally, at any time, the DM is offered the option to specify upper bounds for the objective functions (that is, values above which the objective function values are not regarded as acceptable). These values must lie within the current reachable ranges.

With this preference information, NAUTILUS Navigator progressively moves towards the Pareto optimal front in the direction defined by the DM as fast as (s)he has specified. As these movement goes on, the reachable region of the Pareto optimal front shrinks. Because of this, the DM is dynamically informed about the evolution of the reachable ranges of the objective function values if the movement continues towards the same reference point (i.e., in the same direction). Moreover, the progress of the motion to the Pareto optimal front is constantly shown to the DM. This progress is given as a number ranging between 0 and 100, with 0 meaning the minimum progress, and 100 meaning full progress (that is, a Pareto optimal solution has been reached).

Based on the information given, the DM evaluates whether the objective function values are progressively converging to values of her/his interest. On the way, (s)he may wish to change the preference information, and thus change the direction and/or the movement speed, add some bound, or go backwards and continue the solution process from some previously seen point. In this way, any solution in the Pareto optimal front can be reached at the end of the solution process, without the necessity of trading-off, by re-directing the search for her/his most preferred solution as desired.

In order to reduce the computation time of the solution process while interacting with the DM, as introduced, a well-spread approximation set of the Pareto optimal front must be pre-computed before the navigation starts using any a posteriori method, such as e.g. evolutionary multiobjective optimization algorithms [6, 8] or approximation methods [32]. As explained hereafter, this approximation set is internally used in NAUTILUS Navigator to find optimal solutions of several single-objective optimization problems that need to be solved during the interaction with the DM. Thus, the search for the DM's most preferred solution takes place using the approximation set, avoiding waiting times while iterating with her/him. In this way, our method is well suited for problems which involve expensive objective and constraint function evaluations.

Next, we describe how the algorithm works, and we also propose a graphical user interface which illustrates how the information is given by and presented to the DM.

3.1 The algorithm

As previously mentioned, we assume that an approximation set of the Pareto optimal front, denoted by P, is pre-generated using some a posteriori method. All the solutions of P are assumed to be mutually nondominated. Note that special effort should be devoted to ensuring that the solutions in P are not only nondominated but also constitute a good approximation of the Pareto optimal front (i.e. they are as close as possible to the Pareto optimal front and well-spread along it), but this depends on the accuracy of the a posteriori method used and not on NAUTILUS Navigator. For simplicity, in what follows, we will refer to this set also as the Pareto optimal front, and to its solutions as Pareto optimal solutions. The nadir and the ideal values of each objective are approximated using the worst and the best objective function values corresponding to the solutions in P, respectively.

Although the interface presents a continuous movement towards the Pareto optimal front (i.e. towards P), internally this movement is discretized by dividing it into a number of small steps. The total number of steps from $\mathbf{z}^{\text{nad}}$ to the Pareto optimal front is initially set to 100. The following notations will be used. Let us denote by h the current step number, rs^h the number of remaining steps (including step h) until reaching a Pareto optimal solution, $\mathbf{z}^h$ the point calculated at step h, $\mathbf{f}^{h,lo}$, $\mathbf{f}^{h,up} \in \mathbb{R}^k$ the lower and upper vector bounds of the reachable values at step h, respectively, p^h the progress of the motion at step h and P^h the subset of Pareto optimal solutions in P which can be reached from the current point $\mathbf{z}^h$ without impairing any objective function value. Note that, in fact, the reachable region is defined by the worst and the best value that each objective function can have in the subset P^h. Besides, we denote by $\mathbf{q}^h = (q_1^h, \ldots, q_k^h)^T$ the reference point given by the DM at step h. Let us refer by s^h to the movement speed set by the DM at step h, given as integers $s^h \in \{1, 2, 3, 4, 5\}$. In practice, s^h is the number of steps that are taken per second by the algorithm. This way, when the DM sees the reachable ranges changing in real time with the graphical user interface, (s)he experiences a continuous motion towards the Pareto optimal front as fast as (s)he wants. If s^h remains constant throughout the whole solution process, reaching the Pareto optimal front will take $\frac{100}{s^h}$ seconds. Obviously, this time can be increased or decreased if the DM wishes to move slower or faster, respectively, by giving a different value for s^h at any step h.

NAUTILUS Navigator is described in detail in Algorithm 1. To start with, the DM has to indicate the aspiration levels (s)he would like to achieve for each objective function, which define $\mathbf{q}^1$, and the initial speed, s^1 (step *2* of Algorithm 1). By showing the initial reachable ranges defined by $\mathbf{f}^{0,lo}$ and $\mathbf{f}^{0,up}$, the DM has a clear view of what kind of values are achievable for each objective function at this moment.

Algorithm 1 Algorithm of the NAUTILUS Navigator method

Input: A set approximating the Pareto optimal front, P, and estimates of the nadir and ideal points, $\mathbf{z}^{\text{nad}}$ and $\mathbf{z}^{\star}$.

Output: Most preferred Pareto optimal solution, $\mathbf{z}_{pref}$.

1. Initialization. Set the initial point $\mathbf{z}^0 = \mathbf{z}^{\text{nad}}$, the initial bounds of the reachable region, $\mathbf{f}^{0,up} = \mathbf{z}^{\text{nad}}$ and $\mathbf{f}^{0,lo} = \mathbf{z}^{\star}$ and the initial set of reachable solutions $P^0 = P$. Also, set $d^0 = 0$, the step number $h = 1$ and $rs^1 = 100$.

2. Initial preference information. Ask the DM to give desirable values for the objective functions and set them as the components of $\mathbf{q}^1$. Also, ask her/him to indicate an initial movement speed and denote it by an appropriate integer $s^1 \in \{1, 2, 3, 4, 5\}$.

3. Take a new main step. A number of s^h steps per second are taken towards the Pareto optimal front, until it is reached or until the DM stops the movement to give a new reference point and/or to change the movement speed. Each step h is carried out in the following way:

3.a If $h = 1$ or if the DM has given a new reference point $\mathbf{q}^h$, let $\mathbf{x}^h$ be the solution of problem (2). Otherwise, set $\mathbf{q}^h = \mathbf{q}^{h-1}$ and $\mathbf{x}^h = \mathbf{x}^{h-1}$. Let us denote $\mathbf{f}^h = \mathbf{f}(\mathbf{x}^h)$.

3.b Calculate the next point as $\mathbf{z}^h = \frac{rs^h - 1}{rs^h}\mathbf{z}^{h-1} + \frac{1}{rs^h}\mathbf{f}^h$.

3.c Update the lower and upper bounds, $\mathbf{f}^{h,lo}$ and $\mathbf{f}^{h,up}$, of the reachable objective function values from $\mathbf{z}^h$. To this end, obtain optimal solutions of the *min/max* problems (P_r^h) defined in (4), for every $r = 1, \ldots, k$. Then, set $\mathbf{f}^{h,lo} = (f_1^{h,lo}, \ldots, f_k^{h,lo})^T$ and $\mathbf{f}^{h,up} = (f_1^{h,up}, \ldots, f_k^{h,up})^T$, where $f_r^{h,lo}$ and $f_r^{h,up}$ are the optimal objective function values of the *min* and the *max* formulations of the problem (4), respectively, for every $r = 1, \ldots, k$.

3.d Compute the progress of the motion p^h at step h, according to (5).

3.e Calculate the subset P^h of reachable solutions from as follows. Initialize $P^h = \emptyset$ and, for each solution $\mathbf{x} \in P^{h-1}$ such that $f_i^{h,lo} \leq f_i(\mathbf{x}) \leq f_i^{h,up}$, for all $i = 1, \ldots, k$, update $P^h = P^h \cup \{\mathbf{x}\}$.

3.f. Going to a previous point. If the DM decides to return to a previous point, then restart the solution process from this point, set the values of h, $\mathbf{z}^h$, $\mathbf{f}^{h,lo}$, $\mathbf{f}^{h,up}$, P^h, p^h and rs^h to those of the previous solution chosen.

3.g. New preference information. If, right after step h, the DM decides to stop the movement towards the Pareto optimal front and give new preference information, then reset $\mathbf{q}^{h+1}$ and/qŋ s^{h+1} accordingly.

3.h If $rs^h = 1$, *stop* the solution process with the last solution $\mathbf{x}^h$ and the corresponding objective vector $\mathbf{f}^h$ as the final solution $\mathbf{z}_{pref}$. Otherwise, set $it^{h+1} = rs^h - 1$, $h = h + 1$ and go to *3.a*.

With the preferences defined, s^h steps are taken per second until the Pareto optimal front represented by P is reached (that is, if $rs^h = 1$), or until the DM decides to stop the movement and give new preference information (step *3.g* of Algorithm 1). At each step, a new point $\mathbf{z}^h$ is computed, which is a point lying in the segment joining the previous point $\mathbf{z}^{h-1}$ and the objective vector associated with a Pareto optimal solution denoted by $\mathbf{x}^h$. In practice, $\mathbf{f}(\mathbf{x}^h)$ is obtained by projecting the reference point $\mathbf{q}^h$ given by the DM onto the subset P^{h-1} of Pareto optimal solutions reachable from $\mathbf{z}^{h-1}$, and $\mathbf{x}^h$ is the decision vector corresponding to $\mathbf{f}(\mathbf{x}^h)$.

In NAUTILUS Navigator (step *3.a* of Algorithm 1), this projection is done by solving the following problem, which minimizes an achievement scalarizing function [38] over P^{h-1}:

$$\min_{\mathbf{x}\in P^{h-1}} \max_{i=1,\dots,k} \left\{ \frac{f_i(\mathbf{x}) - q_i^h}{z_i^{\mathrm{nad}} - z_i^{\star\star}} \right\} + \rho \sum_{i=1}^{k} \frac{f_i(\mathbf{x}) - q_i^h}{z_i^{\mathrm{nad}} - z_i^{\star\star}}, \tag{2}$$

where $\rho > 0$ is a small real number and z_i^{nad} and $z_i^{\star\star}$ are the nadir and the utopian points, respectively. By minimizing this function over the finite set P^{h-1}, the computation cost of finding the solution $\mathbf{x}^h$ at each step h is significantly small. Furthermore, it must be noted that problem (2) must be solved only at the first step (i.e. if $h = 1$) or if the DM has modified the reference point at step h. If $\mathbf{q}^h = \mathbf{q}^{h-1}$, there is no need to solve (2) again and, in this case, $\mathbf{x}^h = \mathbf{x}^{h-1}$.

Once the solution $\mathbf{x}^h$ has been obtained, the corresponding objective vector is calculated, denoted by $\mathbf{f}^h = \mathbf{f}(\mathbf{x}^h)$, and the next point $\mathbf{z}^h$ is computed as follows (step *3.b* of Algorithm 1):

$$\mathbf{z}^h = \frac{rs^h - 1}{rs^h}\mathbf{z}^{h-1} + \frac{1}{rs^h}\mathbf{f}^h. \tag{3}$$

It can be seen that $\mathbf{z}^h$ is a point in the line joining $\mathbf{z}^{h-1}$ and $\mathbf{f}^h$ and, as more steps are taken (i.e. rs^h decreases), $\mathbf{z}^h$ is further from the initial point (that is, from $\mathbf{z}^{\mathrm{nad}}$) and closer to the objective vector $\mathbf{f}^h$, i.e. to the solution $\mathbf{x}^h$ in P. One should note that these points are never shown to the DM as such, they are just internally calculated by the algorithm in order to determine the reachable region and the progress of the motion to the Pareto optimal front, which are continuously given at each moment to the DM by NAUTILUS Navigator.

Observe that if h is the last step (i.e. $rs^h = 1$), $\mathbf{z}^h = \mathbf{f}^h$ from equation (3), which implies that the Pareto optimal solution $\mathbf{x}^h$ in $P^{h-1} \subset P$ has been reached. If h is not the last step (i.e. $rs^h > 1$), the new point $\mathbf{z}^h$ is not Pareto optimal and may not be even feasible for the original problem (1). However, as demonstrated in [21, 31], $\mathbf{z}^h$ is always an achievable point when $rs^h > 1$ and this implies that, if it is not feasible, at least, there always exists a Pareto optimal objective vector which dominates it. Furthermore, it is proven that $\mathbf{z}^h$ dominates $\mathbf{z}^{h-1}$ at any step h [21, 31], which means that each point $\mathbf{z}^h$ produced dominates all the previous ones, i.e. each objective function is improved at each step taken. Therefore, a continuous improvement of the objective function values takes place until the final solution is reached.

If $rs^h > 1$, the range of reachable values that each objective function can take from $\mathbf{z}^h$ shrinks and this information is constantly shown to the DM in NAUTILUS Navigator (step *3.c* of Algorithm 1). The lower and upper bound vectors at step h, $\mathbf{f}^{h,lo}$ and $\mathbf{f}^{h,up}$, respectively, (which determine the reachable ranges from $\mathbf{z}^h$) are calculated by solving the following ε-constraint problems (P_r^h) [5]:

$$(P_r^h)\left\{\begin{array}{ll}\text{min/max} & f_r(\mathbf{x})\\ \text{subject to} & f_j(\mathbf{x}) \leq z_j^h,\ j = 1, \ldots, k, j \neq r,\\ & \mathbf{x} \in P^{h-1},\end{array}\right. \tag{4}$$

using the *min* formulation for the lower bounds, and the *max* formulation for the upper bounds. That is, for every $r = 1, \ldots, k$, if $f_r^{h,lo}$ and $f_r^{h,up}$ are the optimal objective function values of the *min* and the *max* formulations of problem (P_r^h), respectively, the lower and upper bound vectors are set as $\mathbf{f}^{h,lo} = (f_1^{h,lo}, \ldots, f_k^{h,lo})^T$ and $\mathbf{f}^{h,up} = (f_1^{h,up}, \ldots, f_k^{h,up})^T$, respectively. Observe that these problems minimize or maximize each objective function not over the whole feasible set, but over the (finite) subset of reachable solutions P^{h-1}. Problem (2) is correspondingly solved over the subset. With this, we save computational effort for solving these $2k$ problems at each step h.

Besides the upper and lower bound vectors, the DM is continuously informed about the progress of the motion to the Pareto optimal front at step h, which is in fact calculated in terms of the distances from $\mathbf{z}^h$ and $\mathbf{f}^h$ to the nadir point $\mathbf{z}^{\text{nad}}$ (step *3.d* of Algorithm 1):

$$p^h = \frac{\|\mathbf{z}^h - \mathbf{z}^{\text{nad}}\|_2}{\|\mathbf{f}^h - \mathbf{z}^{\text{nad}}\|_2} \times 100, \tag{5}$$

where $\|\cdot\|_2$ denotes the L_2-norm. This information is useful to let the DM have an idea of how fast the approach towards the approximation set P is at the current moment in the direction defined by the current reference point $\mathbf{q}^h$.

At each step h, the point $\mathbf{z}^h$ is closer to the approximation set P and, then, some of the solutions in P^{h-1} become not reachable at the next step (unless the DM decides to go backwards) and have to be discarded (step *3.e* of Algorithm 1). Then, the subset P^h with the reachable Pareto optimal solutions from $\mathbf{z}^h$ is constituted by the solutions $\mathbf{x} \in P^{h-1}$ whose objective values are within the corresponding lower and upper bounds, i.e. with $f_i^{h,lo} \leq f_i(\mathbf{x}) \leq f_i^{h,up}$, for all $i = 1, \ldots, k$.

3.2 Discussion and remarks about the NAUTILUS Navigator algorithm

Some issues must be clarified regarding NAUTILUS Navigator, which have not been indicated in Algorithm 1 for simplicity.

The solution process followed in NAUTILUS Navigator takes place in the objective space, and the connection to the decision space is temporarily lost. The points $\mathbf{z}^h$ internally calculated while the DM navigates may be even infeasible, but all of them are achievable, as explained before. Nevertheless, if so desired, it is easy to calculate and visualize ranges of decision variable values corresponding to the current reachable set in the objective space. Furthermore, it is assured that the final solution belongs to the approximation set P. Of course, the DM can be informed about the decision variable values corresponding to the final solution if desired.

The total number of steps from $\mathbf{z}^{\text{nad}}$ to the Pareto optimal front, which is set to 100 by default, can be changed at any time if the DM wishes so. In practice, the number of steps defines the size of each step taken towards the Pareto optimal front and determines the time needed to converge to a final solution. Thus, a higher number of steps than 100 means that the journey towards the Pareto optimal front would take more time since the step-size would be smaller than before, while a lower number implies a reduction of the time needed as the size of the step would be bigger.

Instead of a reference point, alternative options for defining a direction of movement can be also considered in NAUTILUS Navigator. For example, weights for the objective functions which indicate the relative importance given by the DM to the improvement of the current objective values [18] can be considered. In this option, the higher the weight, the closer the final solution will be to the best reachable value of the corresponding objective. If this is the case, step *3.a* of Algorithm 1 has to be adapted as follows. If the vector of weights given by the DM at step h is denoted by $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_k)^T$, then $\mathbf{x}^h$ is the solution of the following problem:

$$\min_{\mathbf{x} \in P^{h-1}} \max_{i=1,\ldots,k} \left\{ \frac{1}{\omega_i} \frac{f_i(\mathbf{x}) - z_i^{h-1}}{z_i^{\text{nad}} - z_i^{\star\star}} \right\} + \rho \sum_{i=1}^{k} \frac{f_i(\mathbf{x}) - z_i^{h-1}}{z_i^{\text{nad}} - z_i^{\star\star}}, \tag{6}$$

and $\mathbf{f}^h = \mathbf{f}(\mathbf{x}^h)$ as before. It must be pointed out that it is necessary to calculate a new point $\mathbf{x}^h$ only when the DM changes the search direction by giving a new vector of weights $\boldsymbol{\omega}$ at step h, otherwise $\mathbf{x}^h = \mathbf{x}^{h-1}$.

As mentioned before, the DM is given the option of specifying bounds for the objective functions as values which must not be exceeded. In case the DM indicates a bound b_p for an objective f_p $(1 \leq p \leq k)$ at any step h, with $f_p^{h,lo} \leq b_p \leq f_p^{h,up}$, an additional constraint as $f_p(\mathbf{x}) \leq b_p$ has to be added to problem (2) (or to problem (6) if weights are used for giving the direction) and to the (P_r^h) problems given in (4). As one constraint of this type must be included per each bound given, the DM must set them carefully given that, if the bounds are too restrictive, there may not be Pareto optimal solutions in P^{h-1} which satisfy all the constraints at the same time. In this case, the DM must be informed in order to re-adjust the information given.

Note that, in *step 3.e* of Algorithm 1, it may happen that the only Pareto optimal solution in P which is reachable from the current $\mathbf{z}^h$ is $\mathbf{x}^h$, i.e. $P^h = \{\mathbf{x}^h\}$, at some step h. This may happen if, e.g. the region of the Pareto optimal front that is reachable from $\mathbf{z}^h$ has not been approximated accurately in P and, before completing the 100 steps of the solution process, the DM has oriented the search towards this region. In this case, the DM must be told about the progress already made to the approximation set in order to let her/him evaluate whether the current point is still too far from the Pareto optimal front or not. According to this information, (s)he may desire to see the only remaining Pareto optimal solution $\mathbf{x}^h$ that is reachable, and its objective vector $\mathbf{f}^h$, and if it is satisfactory enough, stop the solution process with $\mathbf{x}^h$ as the final solution. But alternatively, (s)he may be interested in waiting for the generation of more Pareto optimal solutions in order to further explore the objective function values in this region. In this case, new Pareto optimal solutions must be generated within the current lower and upper bounds, $\mathbf{f}^{h,lo}$ and $\mathbf{f}^{h,up}$, in order to improve the accuracy of the approximation in the reachable region. This can be done using e.g. the Pareto fill module of the E-NAUTILUS method [31].

Finally, given that an approximation set P has been used, the final solution obtained is a nondominated solution approximating the Pareto optimal front, whose accuracy depends on the method used to generate P. In any case, if the user wishes so, we can guarantee, at least, the local Pareto optimality of the final solution by projecting it onto the Pareto optimal front. For this, problem (2) can be minimized over the feasible set S using the objective function values of the final solution as the reference point. Given that, in practice, this may imply to solve a single-objective optimization problem with the original (possibly computationally expensive) functions, this step can be skipped if it is regarded to be too time-consuming.

3.3 Graphical user interface

Because of the crucial role of a DM in interactive methods, a graphical user interface is important. Next, we propose the graphical user interface of NAUTILUS Navigator which permits an easy interaction with the DM and illustrates the functioning of the method better than the plain algorithm.

In order to visualize continuously and in real time all the information regarding the reachable objective function values, the approximated paths suggested in [36] have been used as an inspiration and elaborated and extended for the needs of NAUTILUS Navigator. We employ representations which we call *reachable range paths* (see Figure 1, which illustrates the problem with three objective functions used for the numerical example in Section 4). For each objective function f_r $(r = 1, \ldots, k)$, represented in the vertical axis of the plot, we have a reachable range path which initially ranges between the minimum and maximum values that the objective function can reach (that is, $f_r^{0,lo}$ and $f_r^{0,up}$). As the solution process proceeds, the variation of the reachable values for each objective function is shown in real time using two plot lines, one representing the lowest reachable value ($f_r^{h,lo}$) and the other one corresponding to the highest reachable value ($f_r^{h,up}$) from the current point. This means that the progress along the horizontal axis describes the evolution of the reachable ranges for the objective functions along the solution process. The DM can see the reachable ranges shrinking as the Pareto optimal front is approached and can stop the movement at any time, if desired, to provide new preferences. This way of showing the information enables the DM to see all the previous reachable ranges at once, instead of visualizing only the ones corresponding to the current moment. This implies a lower cognitive effort for the DM since, as stated in [25], "comparing something visible with memories of what was seen before is more difficult than comparing things simultaneously visible side by side". In addition, with these reachable range paths, the DM is also able to jump backwards to any previously seen value of any objective function, just by clicking on it at the corresponding reachable range path.

In the graphical user interface, the DM indicates her/his preferences as follows. On the one side, the components of the reference point (which determine the movement direction) are initially specified by the DM using the text boxes labelled as "Aspiration level" (in Figure 1, an "Aspiration level" text box is shown for each of the three objectives of the problem used). On the other side, the movement speed is set by moving the scroll bar labelled as "Speed" and placed in the graphical user interface below the reachable range paths (see Figure 1). As said, the DM can also specify bounds for the objectives, which are indicated in the graphical user interface using the "Bound" text boxes defined for each of the objective functions.

Once a reference point and a speed have been set, the "Start Navigation" button starts the movement towards the Pareto optimal front and the "Stop" button stops the movement at any time, in order to allow a further examination of the current information shown, or to change any preference information if desired. Along the navigation, the components of the reference point used are represented by horizontal dotted lines in the reachable range paths, as it can be seen in Figure 1. The DM can change these values dynamically at any time just by pressing the "Stop" button and moving the lines representing them upwards or downwards, or by indicating new values in the "Aspiration level" text boxes. Afterwards, by clicking again the "Start Navigation" button, the movement towards the Pareto optimal front continues using the new preference information.

The changes produced in the reachable range paths when several changes of direction have been specified along the solution process can be seen in the rest of figures shown in Section 4. In this way, the historical changes in the reachable ranges of the objective functions for the different reference points given are depicted all together with the reachable range paths, and they are visible for the DM at one glance. Finally, the numerical values of all elements (upper and lower bounds for the current reachable values, components of the reference point and speed), together with the progress of the motion to the Pareto optimal front, are also shown in the interface.

If alternative options for the preference information are considered, the graphical user interface of NAUTILUS Navigator can be adapted accordingly. For example, if the DM wishes to give a direction of movement by means of weights as mentioned in Section 3.2, they could be specified in the graphical user interface using e.g. sliding bars or a spider web chart.

4 Illustrative example

In this section, we demonstrate how the NAUTILUS Navigator method can be used from the DM's point of view, that is, we describe what kind of interaction takes place during the solution process. We consider the three-objective optimization problem described in [30], which aims to identify the improvements that can be carried out in the auxiliary services of a power plant in order to enhance its efficiency, taking into account energy savings and economic criteria. The objective functions are the energy saving achieved (denoted by f_1, in MWh, to be maximized), the economic investment required (denoted by f_2, in million €, to be minimized) and the Internal Rate of Return (IRR) of the investment (denoted by f_3, in %, to be maximized). Note that, although internally we convert all objectives to be minimized for calculation purposes, for the convenience of the DM, the graphical user interface displays the original values. The efficiency can be improved by means of three strategies: (a) the replacement of the current engines by more efficient ones, (b) the installation of variable speed drives, and (c) compensation for reactive power. Then, binary and continuous decision variables are considered in order to indicate whether strategies (a) and (b) are implemented or not in the different drives of the plant, and to specify the amount of reactive power (in kVAR) to be compensated for on each component in relation to strategy (c). The feasible set is defined by simple bound constraints which control that the final situation of the auxiliaries regarding strategies (a) and (b) is never worse than their initial situation. Besides, regarding strategy (c), some constraints assure that the reactive power to be compensated for on each element is never higher than the reactive power required on that element. The problem solved in [30] is based on the auxiliary services of a 1100 MW power plant and it has 13 continuous and 20 binary decision variables.

Due to the complex engineering formulas behind the model, the objective functions and constraints are modelled using a black-box simulator which is very time-consuming to execute. Therefore, given that the model involves computationally expensive function evaluations, the use of a method such as NAUTILUS Navigator is highly recommended to identify a satisfactory final solution interacting with a DM.

In [30, 31], several evolutionary multiobjective optimization algorithms were used to approximate the Pareto optimal front of this problem. The approximation set generated was formed by 2,118 mutually nondominated solutions and we use it as the set P. As described in [30], the Pareto optimal front of this problem is formed by several disconnected subsets of solutions. The ideal and the nadir points were estimated using the best and the worst objective function values corresponding to the solutions in P, respectively, and their approximations are $\mathbf{z}^{\star} = (47526.37, 0.05, 100.00)^T$ and $\mathbf{z}^{\text{nad}} = (408.49, 9.28, 22.13)^T$ (remember that the second objective is to be minimized and the others maximized).

To begin with, the upper and lower bounds of the reachable objective function values were set using the components of $\mathbf{z}^{\text{nad}}$ and $\mathbf{z}^{\star}$, respectively. They were shown to the DM in the graphical user interface. The DM wanted to use the maximum speed and, thus, $\mathbf{s}^1$ was set to 5.

Initially, the DM wanted to investigate which type of Pareto optimal solutions can be found if (s)he set the aspiration level for f_1 to 30000 MW, which was a satisfactory enough value. The DM had a limited budget of 3 million €, but he first wanted to investigate what could be achieved if he invested 4 million €. Regarding f_3, he was very optimistic and decided to set the aspiration level for the IRR to 60%. Anyway, he was aware that reaching a solution with a lower IRR than this was also profitable enough. Thus, the first reference point was $(30000.0, 4.0, 60.0)^T$.

As the navigation proceeded towards the first reference point, the DM could see how the reachable ranges for the objective functions evolved and shrank. Figure 1 shows the navigation using this reference point until the DM decided to first stop the movement and give new information to reorientate the search. For example, a significant decrease of the upper bound for the reachable IRR values (f_3) is observed in Figure 1 at the beginning of the movement. This indicates that the nondominated solutions with the highest IRR became unreachable right from the beginning. However, the reachable IRR values were still very high (above 60%) and thus, the DM knew that very profitable solutions could still be attained in this direction. Besides, the reachable values for the energy saving (f_1) and the investment (f_2) were appealing for the DM despite of the reduction of their ranges.

After several seconds, the DM could observe that, once the upper bound for the reachable investment values was close to 4 million €, the maximum energy savings that could be achieved decreased and its upper bound was getting close to his aspiration level (30000 MW), represented by the horizontal line. Furthermore, if the navigation continued in this direction, the energy savings achieved would fall down to values lower than 30000 MW. Then, he decided to stop the movement (i.e. stop the continuously changing visualization of the reachable range paths) in order to redirect the approach towards a new reference point. He indicated that the new reference point was $(30000.0, 3.0, 45.0)^T$, which implies a relaxation on the investment and the IRR values. These are the reference values shown in Figure 1 in the "Aspiration level" text boxes. Note that the horizontal dotted lines represent the reference values used for the objectives and, as the DM gave new aspiration levels for the investment and the IRR objectives, a change can be observed for the corresponding horizontal dotted lines.

At this moment, he decided to let the solution process reach the Pareto optimal front from the second reference point given in order to check how the reachable objective function values behaved in this direction. The navigation towards a final solution (belonging to P) is shown in Figure 2. He could see that the maximum reachable values for the energy saving decreased at the same time than the lower bounds for the reachable investment values increased. This fact highlighted the conflict degree among these two objective functions. The final solution reached was $\mathbf{z} = (30700.0, 3.26, 46.1)^T$ and the DM was also shown the corresponding decision variable values.

Note that the energy saving achieved at solution $\mathbf{z}$ meets the desirable value set by the DM in the second reference point, and it also reached a very profitable IRR, but it

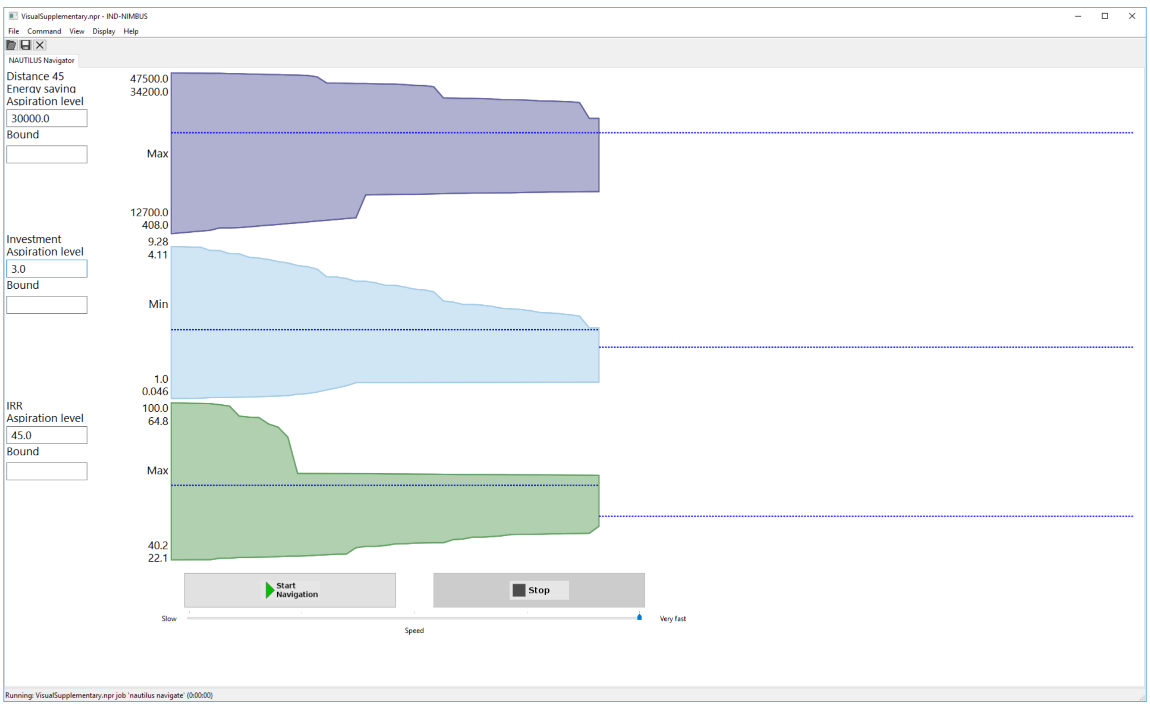

Figure 1: NAUTILUS Navigator: Interaction with the DM with the initial preferences.

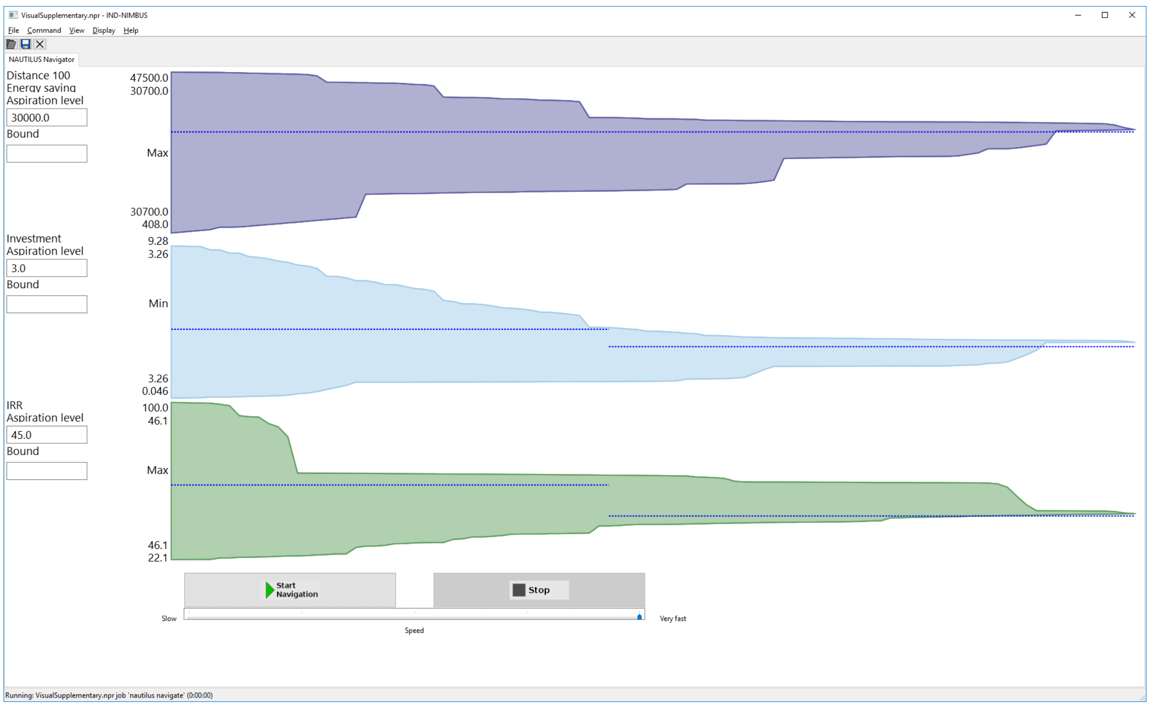

Figure 2: NAUTILUS Navigator: Interaction with the DM until a solution in P is reached.

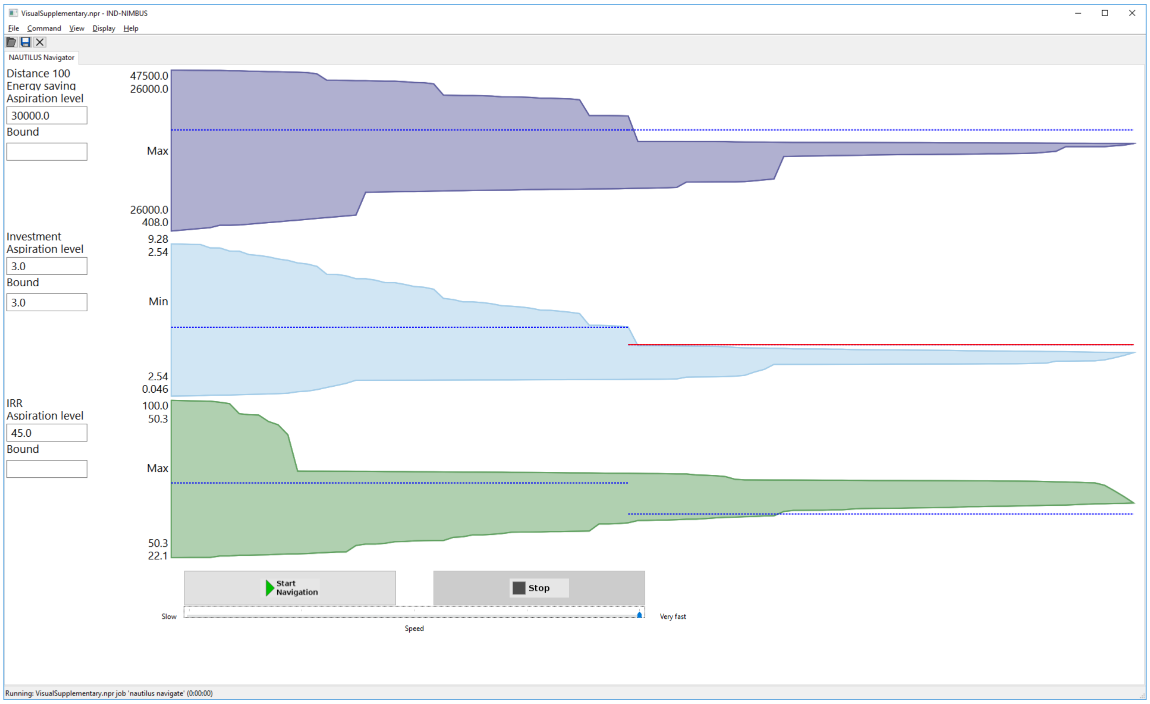

Figure 3: NAUTILUS Navigator: Interaction with the DM when travelling backward, providing new preferences.

requires an investment over the initial budget (3 million €). Because of this, the DM decided to go backwards and reorientate the search in order to investigate Pareto optimal solutions needing less than 3 million €. Thus, he returned to a previous point reached using the second reference point from which the investment objective function could have values bellow 3 million €. Furthermore, in order to assure that the final solution to be obtained did not require more than 3 million €, he introduced this amount as a bound for f_2. The new navigation towards the same reference point with the f_2 bound restriction can be seen in Figure 3. The DM did not change the direction further (i.e. the reference point) and, thus, reached a new final solution $\mathbf{z}' = (26000.0, 2.54, 50.3)^T$. In comparison to $\mathbf{z}$, $\mathbf{z}'$ saves less energy but needs a smaller investment, which is within the budget limit. However, the IRR reached at $\mathbf{z}'$ is higher than the one achieved at $\mathbf{z}$ and this indicates that $\mathbf{z}'$ represents a more profitable solution than $\mathbf{z}$ from the economic point of view. This analysis encouraged the DM to select $\mathbf{z}'$ as the final solution. A video showing the complete navigation process for this example can be seen in `https://desdeo.it.jyu.fi/nautilus-navigator`.

With this example, we have illustrated how the DM can interact with NAUTILUS Navigator until a satisfactory final solution is found. It must be mentioned that, in this illustrative example, we did not project the final solution selected by the DM to generate a Pareto optimal final solution because, on the one hand, the computational effort was too high and, on the other hand, the approximation set P generated in [30, 31] can be regarded as representative and reliable enough. This highlights the fact that the more accurate the initial approximation set of the Pareto optimal front is, the more reliable the results of NAUTILUS Navigator are, without the necessity of additional calculations to assure the Pareto optimality of the final solution.

With NAUTILUS Navigator, the DM could conveniently navigate across the objective space, conduct a free search without anchoring and find a satisfactory final solution. The search was trade-off free as he did not have to sacrifice in order to gain improvement in the objective functions. In addition, the graphical user interface enabled him to learn about the interdependencies among the objective functions and direct the solution process as desired, with information that was understandable and easy to provide, and the effect of which was directly noticeable.

Note that working with problems with discontinuous Pareto optimal fronts (as the one considered here) does not represent any major difficulty for NAUTILUS Navigator. Although this feature may not be known beforehand, the discontinuities (if any) will be evident during the interaction when looking at the information shown in the reachable range paths. As seen in Figures 1, 2 and 3, during the journey towards the final solution, some drops were observable in the reachable range paths, which indicated that the ranges of the reachable objective values decreased suddenly at some moments of the navigation. In practice, this means that the navigation procedure progressively discarded some disconnected parts of the Pareto optimal front. Note that, although we show reachable ranges for the objective functions, not all the objective values in the ranges may be feasible but they are still achievable (as explained in Section 3.2). Overall, solving a problem with a discontinuous Pareto optimal front did not represent any major inconvenience in the interaction with the DM.

5 Conclusions

In NAUTILUS Navigator, we have jointly considered the interaction style of navigation and a win-win philosophy to create a trade-off free navigation method for general multiobjective optimization problems. The method is based on a new way of presenting information to the DM, both from the analytical point of view (thanks to the use of the reachable ranges for the objective functions), and from the practical point of view (thanks to the visualization allowed by the graphical user interface). At first, an approximation of the Pareto optimal front is to be pre-generated without involving the DM. The interactive solution process starts from an undesirable point, and it evolves towards the Pareto optimal front in such a way that every point obtained dominates all the previous ones. We do not question the importance and meaning of having a Pareto optimal solution as the final solution of an interactive solution process, but we provide an alternative way of getting there. On the other hand, the graphical user interface proposed allows the DM to easily see all relevant information simultaneously and freely navigate in real time towards her/his most desired solution by dynamically adjusting the movement direction, using a reference point, and the motion speed. Unlike any other navigation method, NAUTILUS Navigator shows the ranges of the reachable objective values in real time while approaching the Pareto optimal front.

The benefits of NAUTILUS Navigator are several. First, the method allows a navigation supported by the visualization of the evolution of the reachable ranges of the objective functions, which allows the DM to freely explore and learn about the problem. This learning process is a key for the success of any interactive method in practical applications. Second, we offer a trade-off free environment for the navigation, which prevents from behaviours like anchoring effects, or reduction of the number of iterations carried out by the DM. Third, the graphical user interface developed presents the results in an intuitive and comprehensible way. The DM can see at a glance the evolution of the regions of the Pareto optimal front that are reachable according to the current preferences. This allows her/him to easily check past points and go back to any of them if desired. Finally, the fact that the whole interactive solution process takes place using a pre-generated approximation of Pareto optimal solutions, makes it capable to handle any type of problem (whenever this approximation can be sufficiently accurately computed), specially those involving expensive function evaluations.

We have provided a proof of concept and demonstrated the applicability of our method with a practical example. In there, one can see both through the description of the solution process and in the video mentioned in Section 4 how the DM can, in practice, always see gains avoiding anchoring and loss-aversion by freely navigating without trading-off. Designing an experiment framework for comparing the method with others is a research question of its own. Since our method is based on free navigation and free exploration, the behaviour of a real DM cannot be sufficiently modelled by e.g. a value function, which is sometimes used in testing interactive methods. Instead, appropriate indicators or quality measures for e.g. anchoring and loss aversion must be established and validated, which is far from trivial. Besides, a large amount of human DMs with appropriate domain expertise will be needed for testing independently different interactive methods in a different order, to compensate the fact that the DM learns about the problem while using some method(s) first and methods tested after the first one(s) will benefit from this. Therefore, designing an experiment framework to cover all these issues is our future research topic.

Acknowledgements

This work was partially financed by the Academy of Finland (project 287496) and the Spanish Ministry of Economy and Competitiveness (projects ECO2016-76567-C4-4-R and ECO2014-56397-P). Ana B. Ruiz would like to thank the University of Málaga for the financial support received (post-doctoral fellowship "Captación de Talento para la Investigación"). This research is related to the thematic research area Decision Analytics utilizing Causal Models and Multiobjective Optimization (DEMO) of the University of Jyvaskyla.

References

[1] 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*, 24:57–70, 2017.

[2] J. A. Aloysius, F. D. Davis, D. D. Wilson, A. R. Taylor, and J. E. Kottemann. User acceptance of multi-criteria decision support systems: The impact of preference elicitation techniques. *European Journal of Operational Research*, 169:273–285, 2006.

[3] A. Arbel and P. Korhonen. Using apiration levels in an interior primal-dual multiobjective linear programming algorithm. *Journal of Multi-Criteria Decision Analysis*, 5:61–71, 1996.

[4] J. Y. Buchanan and J. Corner. The effects of anchoring in interactive MCDM solution methods. *Computers and Operations Research*, 24(10):907–918, 1997.

[5] V. Chankong and Y. Y. Haimes. *Multiobjective Decision Making Theory and Methodology*. Elsevier Science Publishing Co., New York, 1983.

[6] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, 2nd edition, 2007.

[7] O. Cuate, A. Lara, and O. Schutze. A local exploration tool for linear many objective optimization problems. In *13th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pages 1–6, 2016.

[8] K. Deb. *Multi-objective Optimization using Evolutionary Algorithms*. Wiley, Chichester, 2001.

[9] P. Eskelinen, K. Miettinen, K. Klamroth, and J. Hakanen. Pareto Navigator for interactive nonlinear multiobjective optimization. *OR Spectrum*, 32(1):211–227, 2010.

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

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

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

[13] I. L. Janis and L. Mann. *Decision Making: A Psychological Analysis of Conflict, Choice and Commitment*. The Free Press, New York, 1977.

[14] D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979.

[15] P. Korhonen and J. Wallenius. A Pareto race. *Naval Research Logistics*, 35(6):615–623, 1988.

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

[17] K. M. Lin and M. Ehrgott. Multiobjective navigation of external radiotherapy plans based on clinical criteria. *Journal of Multi-Criteria Decision Analysis*, 25:31–41, 2018.

[18] M. Luque, K. Miettinen, P. Eskelinen, and F. Ruiz. Incorporating preference information in interactive reference point methods for multiobjective optimization. *Omega*, 37(2):450–462, 2009.

[19] D. Meignan, S. Knust, J. M. Frayret, G. Pesant, and N. Gaud. A review and taxonomy of interactive optimization methods in operations research. *ACM Transactions on Interactive Intelligent Systems*, 5(3):17:1–17:43, 2015.

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

[21] K. Miettinen, P. Eskelinen, F. Ruiz, and M. Luque. NAUTILUS method: an interactive technique in multiobjective optimization based on the nadir point. *European Journal of Operational Research*, 206(2):426–434, 2010.

[22] K. Miettinen, D. Podkopaev, F. Ruiz, and M. Luque. A new preference handling technique for interactive multiobjective optimization without trading-off. *Journal Global Optimization*, 63(4):633–652, 2015.

[23] K. Miettinen and F. Ruiz. NAUTILUS framework: towards trade-off-free interaction in multiobjective optimization. *Journal of Business Economics*, 86(1):5–21, 2016.

[24] M. Monz, K. H. Kufer, T. R. Bortfeld, and C. Thieke. Pareto navigation - algorithmic foundation of interactive multi-criteria IMRT planning. *Physics in Medicine and Biology*, 53(4):985–998, 2008.

[25] T. Munzner. Process and pitfalls in writing information visualization research papers. In A. Kerren, J. T. Stasko, J. D. Fekete, and C. North, editors, *Information Visualization: Human-Centered Issues and Perspectives*, pages 134–153, Berlin, Heidelberg, 2008. Springer.

[26] R. P. Nielsen. Varieties of win–win solutions to problems with ethical dimensions. *Journal of Business Ethics*, 88(2):333–349, 2009.

[27] H. Raiffa, J. Richardson, and D. Metcalfe. *Negotiation Analysis: The Science and Art of Collaborative Decision Making*. Harvard University Press, Cambridge, 2002.

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

[29] N. Ravaja, P. Korhonen, M. Köksalan, J. Lipsanen, M. Salminen, O. Somervuori, and J. Wallenius. Emotional–motivational responses predicting choices: The role of asymmetrical frontal cortical activity. *Journal of Economic Psychology*, 52:56–70, 2016.

[30] A. B. Ruiz, M. Luque, F. Ruiz, and R. Saborido. A combined interactive procedure using preference-based evolutionary multiobjective optimization. Application to the efficiency improvement of the auxiliary services of power plants. *Expert Systems with Applications*, 42(1):7466–7482, 2015.

[31] A. B. Ruiz, K. Sindhya, K. Miettinen, F. Ruiz, and M. Luque. E-NAUTILUS: A decision support system for complex multiobjective optimization problems based on the NAUTILUS method. *European Journal of Operational Research*, 246(1):218–231, 2015.

[32] S. Ruzika and M. M. Wiecek. Approximation methods in multiobjective programming. *Journal of Optimization Theory and Applications*, 126(3):473–501, 2005.

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

[34] T. Stewart, O. Bandte, H. Braun, N. Chakraborti, M. Ehrgott, M. Gobelt, Y. Jin, H. Nakayama, S. Poles, and D. Di Stefano. Real-world applications of multiobjective optimization. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Optimization. Interactive and Evolutionary Approaches*, pages 285–327, Berlin, Heidelberg, 2008. Springer.

[35] M. Szczepanski and A. P. Wierzbicki. Application of multiple criteria evolutionary algorithm to vector optimization, decision support and reference-point approaches. *Journal of Telecommunications and Information Technology*, 3(3):16–33, 2003.

[36] S. Tarkkanen, K. Miettinen, J. Hakanen, and H. Isomäki. Incremental user-interface development for interactive multiobjective optimization. *Expert Systems with Applications*, 40(8):3220–3232, 2013.

[37] H. L. Trinkaus and T. Hanne. knowCube: a visual and interactive support for multicriteria decision making. *Computers & Operations Research*, 32(5):1289–1309, 2005.

[38] A. P. Wierzbicki. The use of reference objectives in multiobjective optimization. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making, Theory and Applications*, pages 468–486. Springer, 1980.

[39] D. A. Worthy, M. A. Gorlickand, 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.