

**This is an electronic reprint of the original article.
This reprint *may differ* from the original in pagination and typographic detail.**

Author(s): Miettinen, Kaisa; Ruiz, Francisco

Title: NAUTILUS framework : towards trade-off-free interaction in multiobjective optimization

Year: 2016

Version:

Please cite the original version:

Miettinen, K., & Ruiz, F. (2016). NAUTILUS framework : towards trade-off-free interaction in multiobjective optimization. *Journal of Business Economics*, 86(1), 5-21. <https://doi.org/10.1007/s11573-015-0786-0>

All material supplied via JYX is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

NAUTILUS Framework: Towards Trade-off-Free Interaction in Multiobjective Optimization

Kaisa Miettinen¹ and Francisco Ruiz²

¹University of Jyväskylä, Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland, kaisa.miettinen@jyu.fi

²Department of Applied Economics (Mathematics), Universidad de Málaga, Calle Ejido 6, 29071 Málaga, Spain, rua@uma.es

October 9, 2015

Abstract

In this paper, we present a framework of different interactive NAUTILUS methods for multiobjective optimization. In interactive methods, the decision maker iteratively sees solution alternatives and provides one's preferences in order to find the most preferred solution. We question the widely used setting that the solutions shown to the decision maker should all be Pareto optimal which implies that improvement in any objective function necessitates allowing impairment in some others. Instead, in NAUTILUS we enable the decision maker to make a free search without having to trade-off by starting from an inferior solution and iteratively approaching the Pareto optimal set by allowing all objective functions to improve. The framework presented consists of different modules for preference elicitation and optimization. Four main NAUTILUS method variants are introduced as well as ideas of utilizing the framework in a flexible way to derive further variants.

Keywords: Interactive methods, multicriteria optimization, Pareto optimality

1 Introduction

Decision problems occur in various application areas including business, engineering and environment where several conflicting objectives must be simultaneously considered and optimized. Such problems are known as multiobjective optimization problems. Because of the conflicting nature of the objectives, instead of a single optimal solution, a set of so-called Pareto optimal or nondominated solutions can be identified. They have different trade-offs among the objectives

and can be regarded as compromise solutions. Mathematically, Pareto optimal solutions cannot be ordered, and some preference information from a human decision maker is typically needed to identify one, the most preferred, solution to be implemented.

Multiobjective optimization methods can be divided to four classes based on the role of the decision maker in the solution process (Hwang and Masud, 1979; Miettinen, 1999). If no decision maker is available, some neutral compromise solution must be found. In a priori methods, the decision maker specifies preference information before the solution process whereas in a posteriori methods a representative set of Pareto optimal solutions is first generated and the decision maker is then expected to select some of them. The class of interactive methods aims at avoiding shortcomings of the above classes: the decision maker can iteratively specify and modify one's preferences and, thus, no pre-fixed preference structure is assumed. On the other hand, the amount of information and solutions to be considered at a time is low which should decrease the cognitive load. During an interactive solution process, the decision maker can learn about the interdependencies among the objectives, what kind of solutions are attainable and about one's own preferences (see, e.g., Belton *et al.*, 2008).

Because of their desirable properties, many interactive methods have been proposed in the literature (see, e.g., Luque *et al.* (2011); Miettinen (1999); Miettinen *et al.* (2008) and references therein). They differ from each other, for example, by the type of preference information utilized, the way of incorporating the preference information in the solution process and the information that is given to the decision maker. However, what is common is that they deal with Pareto optimal solutions only, that is, throughout the solution process the decision maker is shown Pareto optimal solutions and new Pareto optimal solutions are generated based on her/his preference information. This means that when moving from a Pareto optimal solution to another, the decision maker must allow sacrifice in some objective(s) to gain improvement in others.

In this paper, we deal with interactive methods as well but question the idea that the whole solution process should focus on Pareto optimal solutions. It has been observed, e.g., in Buchanan and Corner (1997) that the starting point of an interactive method matters. This can be explained by the anchoring and adjustment bias in decision making, that is, people anchor their thinking on given information and may not be able to adjust from the anchor even if the information turns out to be irrelevant. Furthermore, the experiments conducted in Aloysius *et al.* (2006) show that if decision makers are required to make explicit trade-off judgement, they experience a higher level of decisional conflict than otherwise. According to the authors, this may lead to decision makers not finding decision support methods acceptable or continue using them. One more motivation in the background of this paper is prospect theory proposed in Kahneman and Tversky (1979). According to the prospect theory, people do not react symmetrically to gains and losses but our attitudes to losses loom larger than gains. Furthermore, our past experiences define an adaptation level and we perceive stimuli in relation to it. To overcome the above-mentioned challenges, we here introduce a NAUTILUS framework of different interactive

NAUTILUS methods which we can also call as a NAUTILUS family. All these methods start from an inferior solution instead of a Pareto optimal one.

Methods in the NAUTILUS family enable a free search without requiring the decision maker to trade off. The interactive solution process starts from the nadir objective vector or some other point from where improvement in each objective function is possible. Iteratively, the decision maker approaches the set of Pareto optimal solutions and is able to gain improvement in all objectives compared to the previous iteration. Only the final solution at the end of the solution process is Pareto optimal. In other words, we do not question the fact that the final solution should be Pareto optimal. But a necessity of trading off should not limit the decision maker's willingness to keep searching for promising solutions during the solution process.

The NAUTILUS framework is also suited for group decision making where different decision makers need to find a solution which is acceptable to them all. A setting where each decision maker can gain without a need to sacrifice is fruitful for negotiations. We have a similar motivation to the joint gains method for multi-party negotiations proposed in Ehtamo *et al.* (2001) where a method of improvement directions is suggested to be used as a mediator's tool and the mediator generates jointly preferred proposals. Not many other related methods have been proposed in the literature. One of the few exceptions is the interior primal-dual method introduced in Arbel and Korhonen (1996) for linear multiobjective optimization problems. This method leads the decision maker through the feasible region involving dominated solutions.

The NAUTILUS framework consists of different modules and combinations of module options, which result with four main NAUTILUS variants and enables developing further variants. Among others, the NAUTILUS methods that have been published before can be derived from the framework. The original NAUTILUS method was presented in Miettinen *et al.* (2010) where the decision maker provides preference information by ranking the objective functions according to the relative importance of improving their current values or distributing points indicating how much improvement is needed in the current values. However, this was regarded as a very rough approach and in Miettinen *et al.* (2015) another technique was proposed where the decision maker provides ratios of improvement and a direction of simultaneous improvement is determined from them. With this information the decision maker can directly control the solution process.

The third published method which can be derived from the NAUTILUS framework is E-NAUTILUS which has been directed, in particular, for computationally expensive problems. It consists of a pre-processing, an interactive decision making and a post-processing stage. The decision maker is involved only in the second stage where the original objective functions (which may be computationally expensive) are not dealt with. It can also be used to find the most preferred solution of the ones generated by any a posteriori method. What is different in E-NAUTILUS compared to the two above-mentioned methods is that the decision maker can consider several solutions at each iteration.

All members of the NAUTILUS family give information to the decision

maker of the proximity of the point considered to the Pareto optimal set as well as the ranges of objective function values that are still reachable from the current iteration point without any impairment. During the iterations, the ranges naturally shrink but the decision maker can go backwards if the reachable values are not acceptable.

In this paper, we propose a general NAUTILUS framework. Besides the previously published methods, the modular framework enables creating further NAUTILUS method variants and even changing the module options during the solution process, i.e., utilizing the framework in a flexible way. This gives more flexibility for the decision maker for directing the solution process.

The rest of this paper is organized as follows. In Section 2 we introduce the basic concepts and notations used and in Section 3 we outline the core that is common to all NAUTILUS methods. We present the actual NAUTILUS framework in Section 4 and, finally, conclude in Section 5.

2 Basic Concepts and Notations

We consider *multiobjective optimization problems* of the form

$$\begin{aligned} & \text{minimize} && \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^T \\ & \text{subject to} && \mathbf{x} \in S, \end{aligned} \tag{1}$$

where the decision maker wishes to simultaneously minimize k ($k \geq 2$) objective functions $f_i : S \rightarrow \mathbb{R}$. The so-called *feasible set* S , which is a nonempty compact subset of \mathbb{R}^n , is formed by feasible *decision vectors* $\mathbf{x} = (x_1, \dots, x_n)^T$. The image of the feasible set in the objective space \mathbb{R}^k is called the *feasible objective set* $\mathbf{f}(S)$ and its components $\mathbf{z} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$, which consist of objective (function) values, are called *objective vectors*.

For any two vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^k$, we say that \mathbf{v} *dominates* \mathbf{w} if $v_i \leq w_i$ for all $i = 1, \dots, k$ and $\mathbf{v} \neq \mathbf{w}$. All feasible solutions whose corresponding objective vectors are not dominated by others, form the set of Pareto optimal decision vectors:

$$E = \{\mathbf{x} \in S : \text{there is no } \mathbf{x}' \in S \text{ such that } \mathbf{f}(\mathbf{x}') \text{ dominates } \mathbf{f}(\mathbf{x})\}.$$

For the sake of simplicity, we use the term *Pareto optimal solution* to refer either to an element of E or $\mathbf{f}(E)$ (the image of E in the objective space), which will be clear from the context. The set $\mathbf{f}(E)$ is called the *Pareto optimal set*.

Solving problem (1) means finding the most preferred decision vector (also referred to as the most preferred solution), i.e., an element of S which is the most satisfying for the decision maker. It is rational to assume that if one solution dominates another solution, the decision maker always prefers the former to the latter. Therefore in multiobjective optimization, the search for the most preferred solution is constrained to the Pareto optimal solution set (Miettinen, 1999; Branke *et al.*, 2008). For this reason, it is desirable to have some

information about the ranges of the objective function values in the Pareto optimal set. The *ideal objective vector* is defined as $\mathbf{z}^* = (z_1^*, \dots, z_k^*)^T$ such that $z_i^* = \min_{\mathbf{x} \in E} f_i(\mathbf{x})$ for $i = 1, \dots, k$. Thus, its components are the best values that each objective function can achieve in the Pareto optimal set. This vector can be obtained by separately minimizing each of the objective functions in the feasible set. In calculations, a utopian objective vector \mathbf{z}^{**} with components strictly smaller than those of the ideal objective vector is often used instead of \mathbf{z}^* (see, e.g., Miettinen (1999)). On the other hand, the *nadir objective vector* is defined as $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}}, \dots, z_k^{\text{nad}})^T$ such that $z_i^{\text{nad}} = \max_{\mathbf{x} \in E} f_i(\mathbf{x})$ for $i = 1, \dots, k$, and its components are the worst values that each objective function can achieve in the Pareto optimal set. The nadir objective vector is, in general, more difficult to obtain and typically one needs to settle for approximations by using, e.g., payoff tables (see Miettinen (1999) and references therein). Lately, some approaches for more reliable nadir objective vector generation have been proposed (see, e.g., Deb *et al.* (2010); Szczepanski and Wierzbicki (2003)).

As mentioned in the introduction, in this paper, we concentrate on *interactive methods* because they are regarded as the most promising methods of multiobjective optimization due to numerous advantages (Miettinen *et al.*, 2008; Miettinen and Hakanen, 2009). Broadly speaking, an interactive method consists of iteratively generating Pareto optimal solution candidates based on preference information provided by the decision maker. In general, at each iteration the decision maker is shown one or several solution candidates and, by expressing preferences, (s)he indicates how the candidates should be changed to get more preferred solution candidates. Solution processes facilitated by such methods involve moving from one Pareto optimal solution to another and comparing solution candidates by the decision maker.

The reference point method, originally proposed by Wierzbicki (1980), which assumes that the decision maker specifies aspiration levels (or reference levels) for each objective function reflecting objective values that the decision maker regards as desirable, is one of the most widely used interactive approaches. Let $\mathbf{q} = (q_1, \dots, q_k)^T$ be a so-called reference point consisting of aspiration levels. Furthermore, \mathbf{q} is said to be an *achievable reference point*, if $\mathbf{q} \in \mathbf{f}(S) + \mathbb{R}_+^k$ (where $\mathbb{R}_+^k = \{\mathbf{y} \in \mathbb{R}^k \mid y_i \geq 0 \text{ for } i = 1, \dots, k\}$), that is, if either $\mathbf{q} \in \mathbf{f}(S)$ or \mathbf{q} is dominated by some Pareto optimal objective vector. Once a reference point has been given, a so-called achievement (scalarizing) function is minimized over the feasible set:

$$\begin{aligned} & \text{minimize} && \max_{i=1, \dots, k} \{ \mu_i (f_i(\mathbf{x}) - q_i) \} + \rho \sum_{i=1}^k \frac{f_i(\mathbf{x}) - q_i}{z_i^{\text{nad}} - z_i^{**}} \\ & \text{subject to} && \mathbf{x} \in S, \end{aligned} \quad (2)$$

where ρ is a small positive number, and μ_i ($i = 1, \dots, k$) are positive weights. In most of the interactive methods using achievement functions, weights are kept unaltered during the whole process and their purpose is mainly to normalize different ranges of objectives. However, these weights can have different roles (from the original idea of purely normalizing to fully preferential) as described

in Ruiz *et al.* (2008) and they can be varied to get different Pareto optimal solutions (see, for example, Luque *et al.* (2009)).

The optimal solution of problem (2) is assured to be Pareto optimal for any reference point (see, e.g. Miettinen (1999)). Besides, it is easy to prove that if \mathbf{q} is an achievable reference point, then either \mathbf{q} is a Pareto optimal objective vector of problem (1), or \mathbf{q} is dominated by the Pareto optimal objective vector corresponding to the optimal solution of problem (2).

3 NAUTILUS Core

The philosophy underlying the NAUTILUS framework consists of starting from a bad solution and moving towards the Pareto optimal set so that each iteration point dominates the previous one, until a Pareto optimal solution is reached at the last iteration. In this way, every objective function is improved at each iteration, and the decision maker is not compelled to trade off when moving from one iteration to the next. Therefore, we assume that our starting solution is z^{nad} , which stands for the nadir objective vector, an approximation of the nadir objective vector, or some solution regarded as clearly inferior by the decision maker.

In the following, we describe a basic NAUTILUS iteration, and its main features. This iteration is taken at every step of the NAUTILUS framework, regardless of the specific options chosen for eliciting preferences or for solving the intermediate optimization problems, and this is why we regard it as the NAUTILUS core.

First, the decision maker is asked to give the overall number of iterations (s)he wishes to carry out, itn . This parameter determines the speed of the approach to the Pareto optimal set, and the decision maker can change it at any time along the solution process if (s)he estimates that this speed is not adequate. Let us denote by h the current iteration number and let \mathbf{z}^h be the point in the objective space corresponding to the current iteration to be referred to as an iteration point. We will assume that \mathbf{z}^0 is the initial estimate of the nadir objective vector, or a starting point which the decision maker regards consisting of bad objective values. Except in trivial problems, \mathbf{z}^0 is not a Pareto optimal objective vector, and it is an achievable reference point. Let us denote by it^h the number of iterations left (including iteration h). Therefore, $it^1 = itn$. Given the previous iteration point \mathbf{z}^{h-1} , the basic iteration of NAUTILUS is based on somehow finding a Pareto optimal solution \mathbf{x}^h so that the corresponding objective vector dominates \mathbf{z}^{h-1} . Once this solution has been found, let $\mathbf{f}^h = \mathbf{f}(\mathbf{x}^h)$, the next iteration point is obtained as

$$\mathbf{z}^h = \frac{it^h - 1}{it^h} \mathbf{z}^{h-1} + \frac{1}{it^h} \mathbf{f}^h. \quad (3)$$

Therefore, a step is taken from \mathbf{z}^{h-1} towards the Pareto optimal set, in the direction given by $\mathbf{f}^h - \mathbf{z}^{h-1}$.

This basic iteration has a series of interesting properties. First, it must be noted that if h is the last iteration, then $it^h = 1$, and thus, $\mathbf{z}^h = \mathbf{f}^h$ is a Pareto optimal objective vector, which is the final solution, and \mathbf{x}^h is the final solution in the decision space. If, on the other hand, h is not the last iteration, then \mathbf{z}^h can even be an infeasible vector in the objective space, but the following results hold:

Theorem 1 (Miettinen et al., 2010)

1. If $it^h \neq 1$, then \mathbf{z}^h is not Pareto optimal but is an achievable reference point.
2. At any iteration h , \mathbf{z}^h dominates \mathbf{z}^{h-1} .

Therefore, although \mathbf{z}^h is not a Pareto optimal objective vector of problem (1) (if h is not the last iteration), and it may even be infeasible for this problem, it is assured to be an achievable reference point, that is, it is either in the feasible objective set $\mathbf{f}(S)$ for problem (1) or it is dominated by some Pareto optimal objective vector. On the other hand, each iteration point \mathbf{z}^h dominates all the previous ones. Therefore, each iteration improves the values of the objectives, without impairing any of them.

Not having to trade off among objectives is the main feature of the NAUTILUS basic iteration. However, a price has to be paid at each iteration: the range of reachable values that each objective function can take (in this and further iterations) will shrink, in general, because iteration points get closer to the Pareto optimal set. Therefore, it is interesting for the decision maker to see lower and upper bounds on these reachable values when possible. Let us denote by $\mathbf{z}^{h,lo} = (z_1^{h,lo}, \dots, z_k^{h,lo})^T$ and $\mathbf{z}^{h,up} = (z_1^{h,up}, \dots, z_k^{h,up})^T$ these lower and upper bounds, respectively, for iteration h . Initially, $\mathbf{z}^{1,lo} = \mathbf{z}^*$ and $\mathbf{z}^{1,up} = \mathbf{z}^{nad}$. In iteration h , vectors $\mathbf{z}^{h,lo}$ can be calculated as follows. Given \mathbf{z}^{h-1} , let us consider the following ε -constraint problems for $r = 1, \dots, k$:

$$(P_r^h) \begin{cases} \text{minimize} & f_r(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \leq z_i^{h-1} \quad i = 1, \dots, k, \quad i \neq r \\ & \mathbf{x} \in S. \end{cases}$$

Let us denote by $z_r^{h,lo}$ the optimal objective function value of problem (P_r^h) and $\mathbf{z}^{h,lo} = (z_1^{h,lo}, \dots, z_k^{h,lo})$ the vector formed by these values. Then, the following result holds:

Theorem 2 (Miettinen et al., 2010) $\mathbf{z}^{h,lo}$ dominates both \mathbf{z}^h and \mathbf{z}^{h-1} .

From Theorem 2, the reachable values of \mathbf{z}^h are bound in the following way:

$$z_i^h \in [z_i^{h,lo}, z_i^{h-1}] \quad (i = 1, \dots, k).$$

If we denote $\mathbf{z}^{h,up} = \mathbf{z}^{h-1}$, then we have

$$z_i^h \in [z_i^{h,lo}, z_i^{h,up}] \quad (i = 1, \dots, k). \quad (4)$$

Note that due to the iterative nature of NAUTILUS, it is obvious that for each iteration h , $[z_i^{h,lo}, z_i^{h,up}] \subseteq [z_i^{h-1,lo}, z_i^{h-1,up}]$, and thus, in general,

$$z_i^j \in [z_i^{h,lo}, z_i^{h,up}] \quad (i = 1, \dots, k), j \geq h.$$

This means that if a Pareto optimal solution is left out of the interval defined in (4), it will not be reachable again in any following iteration. On the other hand, the amount of unreachable Pareto optimal solutions increases with each iteration. For this reason, these ranges of reachable values of the objective functions for each iteration are very useful for the decision maker.

Besides, it is also possible to provide the decision maker with a measure of the closeness of the current iteration point to the Pareto optimal set:

$$d^h = \frac{\|\mathbf{z}^h - \mathbf{z}^{\text{nad}}\|_2}{\|\mathbf{f}^h - \mathbf{z}^{\text{nad}}\|_2} \times 100 \quad (5)$$

where $\|\cdot\|_2$ is the L_2 norm, that is, given a vector $\mathbf{z} = (z_1, \dots, z_k)^T \in \mathbb{R}^k$, $\|\mathbf{z}\|_2 = \sqrt{z_1^2 + \dots + z_k^2}$. Note that if $\mathbf{z}^h = \mathbf{z}^{\text{nad}}$, then $d^h = 0$ and, on the other hand, if $\mathbf{z}^h = \mathbf{f}^h$, then $d^h = 100$. Therefore, the greater d^h is, the closer the corresponding iteration point is to the Pareto optimal set. This measure allows the decision maker to determine whether the approach rhythm to the Pareto optimal set is appropriate, too fast or too slow.

Based on the information given by the intervals $[z_i^{h,lo}, z_i^{h,up}]$ and by the closeness measure d^h , instead of carrying out the standard (forward) iteration previously defined, the decision maker may wish instead to:

- redefine the number of remaining iterations, so as to change (increase or decrease) the approach pace to the Pareto optimal set or
- take a step backwards, go back to the previous iteration, and reconsider the preference information given.

In general, there are two main issues that must be taken into account in order to decide how each iteration must be carried out:

1. The type of preference information the decision maker is willing to give at each iteration. The decision maker can somehow define a direction of improvement of the objectives, which is used to determine \mathbf{x}^h , or (s)he may prefer to choose one solution among some alternatives, obtained for different vectors \mathbf{x}^h .
2. The computational complexity of the problem solved, which may allow or not to solve single objective optimization problems at each iteration.

These issues yield different options of NAUTILUS iterations, which are discussed in depth in the next section.

4 NAUTILUS Framework

In order to carry out each iteration, following the NAUTILUS core described in Section 3, we need, on the one hand, to establish a procedure for the preference elicitation from the decision maker, and on the other hand, to decide how the Pareto optimal solutions \mathbf{x}^h are determined. Therefore, the general NAUTILUS framework consists of connecting the core to two modules: the *preference elicitation module*, and the *solver module* (see Figure 1). Next, we describe the different options available for each of these modules.

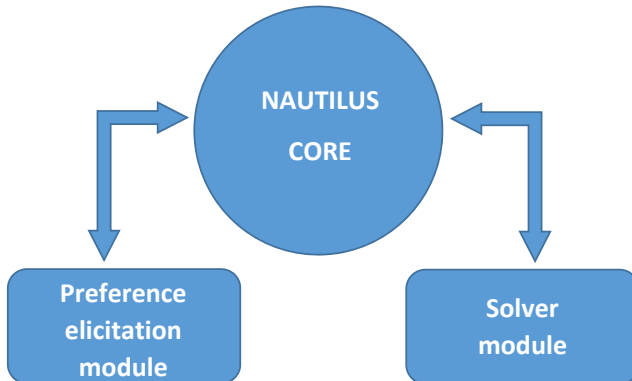


Figure 1: General NAUTILUS framework

4.1 Preference Elicitation Module

In the preference elicitation module, the decision maker is asked to provide her/his preference information, in order to evolve from the previous iteration \mathbf{z}^{h-1} to the next one \mathbf{z}^h . To this end, two different options are provided:

1. **Generation of a new iteration point following a direction of simultaneous improvement.** Given \mathbf{z}^{h-1} , the decision maker is asked to give preference information which allows us to build a *direction of simultaneous improvement* δ^h of the objective functions. Once this direction has been determined, the solver module is used to find the corresponding Pareto optimal solution \mathbf{x}^h , and to build the next iteration \mathbf{z}^h as described in formula (3). The new iteration point \mathbf{z}^h is expected to improve all the objective functions from \mathbf{z}^{h-1} , following the direction of improvement δ^h .
2. **Choice of one among several alternatives.** Instead of building a direction of improvement, we can generate a number of solutions which somehow represent what is achievable from \mathbf{z}^{h-1} , and ask the decision maker to choose one. Namely, given \mathbf{z}^{h-1} , the solver module is used to get a number of Pareto optimal solutions which provide a good representation of the part of the Pareto optimal set that is reachable from \mathbf{z}^{h-1} . This

reachable part of the Pareto optimal set is defined by the bound vectors $\mathbf{z}^{h-1,lo} = (z_1^{h-1,lo}, \dots, z_k^{h-1,lo})^T$ and $\mathbf{z}^{h-1,up} = (z_1^{h-1,up}, \dots, z_k^{h-1,up})^T$, which are calculated in the core module as described in Section 3. From each one of these Pareto optimal solutions, the corresponding alternatives as the next iteration point are calculated following formula (3). All these alternatives are presented to the decision maker, who is asked to *choose one of them*.

Next, let us describe in further detail these two preference elicitation options. The direction of simultaneous improvement δ^h expresses the way the decision maker wishes to improve each of the objectives. Roughly speaking, it means that the decision maker would like to obtain a vector lying in the half line $\mathbf{z}^{h-1} + \lambda \delta^h$, ($\lambda > 0$). Therefore, δ^h expresses improvement proportions for each objective function, from the previous iteration \mathbf{z}^{h-1} . Several ways to build this direction if improvement are described in Miettinen *et al.* (2010) and Miettinen *et al.* (2015). Let us briefly classify and summarize them as follows.

- **Direct specification** is obviously the most straightforward option. If the decision maker is well aware of the meaning of the direction of simultaneous improvement δ^h , then (s)he can provide its components directly.
- In **importance based options** the decision maker can specify preference information in terms of the importance given to the improvement of each objective function. There are two different possibilities for giving this information:

- Rank the objectives according to the relative importance of improving each current objective value. This importance evaluation allows to allocate the k objective functions into index sets J_r which represent the importance levels $r = 1, \dots, s$. Note that there can be more than one objective function in one index set. If $r < t$, then improving the current objective function values in the index set J_r is less important than improving the current objective function values in J_t . In this case, the direction of improvement is built as follows:

$$\delta_i^h = r(z_i^{nad} - z_i^{**}) \quad (i \in J_r). \quad (6)$$

- Specify percentages reflecting how the decision maker would like to improve the current objective values, by answering to the following question: *Assuming you have one hundred points available, how would you distribute them among the current objective values so that the more points you allocate, the more improvement on the corresponding current objective value is desired?* If p_i points are given to the objective function f_i (for $i = 1, \dots, k$), then $\Delta q_i = p_i/100$ and

$$\delta_i^h = \Delta q_i(z_i^{nad} - z_i^{**}). \quad (7)$$

- In **pairwise comparison options** the decision maker can define the direction of improvement by considering the objective functions in a pairwise fashion. Namely, we can define the *improvement ratio* between two different objectives f_i and f_j as $\theta_{ij} = \delta_j/\delta_i$. That is, the decision maker wishes that the improvement of f_i by one unit should be accompanied with the improvement of f_j by θ_{ij} units. Two ways of defining the direction of improvement by means of improvement ratios are the following:

- The decision maker can select a reference objective function f_i ($i = 1, \dots, k$), and for each of the other objectives f_j , (s)he sets the value θ_{ij} . Then the direction of improvement is defined by

$$\delta_i^h = 1, \quad \delta_j^h = \theta_{ij}, \quad j \neq i. \quad (8)$$

- The decision maker can freely give improvement ratios for the pairs of objectives (s)he wishes. In this case, a procedure is carried out which assists the decision maker in giving the improvement ratios that are needed in order to build δ^h . The procedure consists of building an undirected graph G where the set of nodes is the set of indices of objectives $V = \{1, \dots, k\}$, and the set of edges contains the index pairs $\{i, j\}$, for which the decision maker has set the value of θ_{ij} . The decision maker continues giving improvement ratios until G contains a spanning tree of the complete graph on k vertices of V . More details of this procedure, the way δ^h is built from these ratios, and a discussion about the advantages and disadvantages of the different ways proposed for giving δ^h can be found in Miettinen *et al.* (2015).

Once the direction of simultaneous improvement has been determined, following any of the options described above, the solver module is called to solve problem (2) with

$$q_i = z_i^{h-1} \text{ and } \mu_i = \frac{1}{\delta_i^h} \quad (i = 1, \dots, k).$$

Then, \mathbf{x}^h is set to the optimal solution obtained, we set $\mathbf{f}^h = \mathbf{f}(\mathbf{x}^h)$, and the new iteration point \mathbf{z}^h is generated as described in formula (3). The direction of simultaneous improvement option of the preference elicitation module is graphically illustrated in Figure 2. In this figure, from the previous iteration point \mathbf{z}^{h-1} , and taking into account the reachable ranges of the objectives, the decision maker gives a direction of simultaneous improvement δ^h . Here \mathbf{f}^h is the optimal solution of (2) and \mathbf{z}^h is the next iteration point.

Let us now describe the second preference elicitation option: choosing one among several solutions. The decision maker initially sets the number N_S of solutions that (s)he wishes to see at each iteration. Then, given the previous iteration \mathbf{z}^{h-1} , the solver module is called in order to generate a well-spread set P^h of Pareto optimal solutions (in the objective space) lying in the part of the

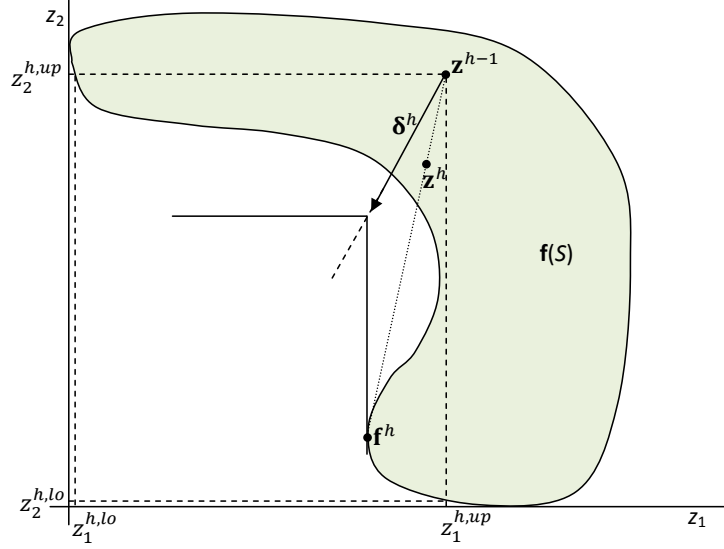


Figure 2: The generation of a new iteration point following a direction of simultaneous improvement option for a hypothetical biobjective optimization problem.

Pareto optimal set that is reachable from \mathbf{z}^{h-1} , that is, that satisfy the bounds given by $\mathbf{z}^{h,lo}$ and $\mathbf{z}^{h,up}$. Once P^h is available, a procedure described in Ruiz *et al.* (2015) is used to determine the N_S solutions that are presented to the decision maker:

- (a) The set P^h is divided into N_S subsets using some clustering technique like, for example, the average linkage method (Morse, 1980).
- (b) The centroids of these N_S subsets are determined, and they are regarded as the N_S most representative solutions of P^h .
- (c) For each centroid, formula (3) is used to determine N_S alternatives lying on the segments joining \mathbf{z}^{h-1} and each of the centroids. All these solutions dominate the previous iteration point \mathbf{z}^{h-1} .
- (d) The alternatives are presented to the decision maker, who is asked to choose one of them. The solution chosen is the next iteration point \mathbf{z}^h .

This preference elicitation option is graphically illustrated in Figure 3. In the figure, given the previous iteration \mathbf{z}^{h-1} , $\mathbf{f}^{h,i}$ are the N_S (6 in this case) solutions in the reachable part of the Pareto optimal set, and $\mathbf{z}^{h,i}$ are the alternatives presented to the decision maker for the next iteration point.

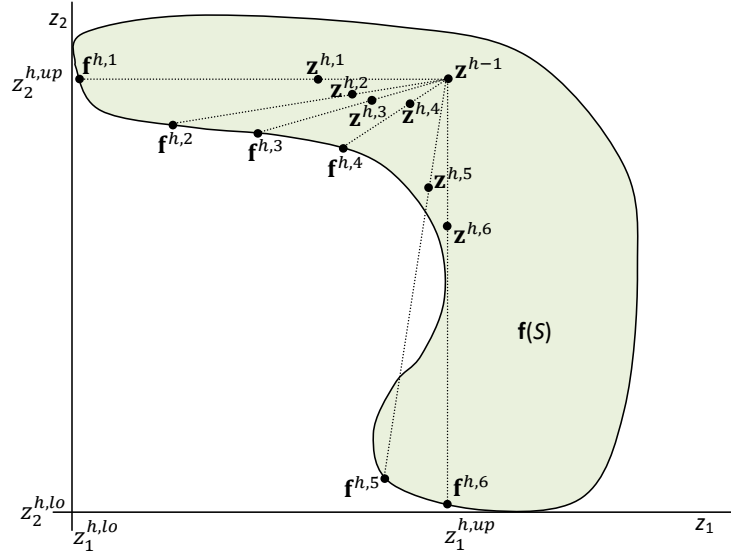


Figure 3: The choice of one among several alternative options for a hypothetical biobjective optimization problem.

4.2 Solver Module

As it has been mentioned before, the application of the interactive method NAUTILUS involves solving problem (2) repeatedly in order to get new Pareto optimal solutions, and also to solve the ε -constraint problems (P_r^h) in order to determine the vectors of bounds $z^{h,lo}$. Therefore, depending on the computational complexity of problem (1), two situations can take place: the problems can be accurately solved in a reasonable time using an appropriate single objective optimization method, or the optimization would be too time-consuming, and new iteration points cannot be produced in a reasonable time. Therefore, two options can be considered for the solver module:

1. **Optimization option.** Problems (2) and (P_r^h) are solved at each iteration using an appropriate single objective optimization method, i.e., solver. To this end, the solver module can incorporate different exact solvers for linear and convex problems, or (heuristic) global solvers for nonconvex problems.
2. **A posteriori option.** If online optimization is not viable, then one can carry out a pre-processing phase before the interactive phase, in order to generate an accurate enough representation of the Pareto optimal set with any a posteriori method. For example, evolutionary multiobjective optimization (EMO) methods can be used. The idea is to use these solutions in the interactive phase instead of solving optimization problems involving functions of the original (computationally expensive) problem.

4.3 Different Variants of NAUTILUS Methods

Next, the two options as the solver module and their adaptation to the two options as the preference elicitation module are discussed in further detail. Obviously, if the first preference elicitation option is chosen, and a direction of simultaneous improvement is built in the preference elicitation module, the first option of the solver module consists of choosing an appropriate solver to deal with problems (2) and (P_r^h) every time they are needed. The main steps of each iteration of the resulting method are:

Method A.1

1. In the preference elicitation module, given the previous iteration \mathbf{z}^{h-1} , the decision maker provides, following one of the ways described in Section 4.1, the direction of improvement δ^h .
2. In the solver module, problem (2) with $q_i = z_i^{h-1}$ and $\mu_i = \frac{1}{\delta_i^h}$, ($i = 1, \dots, k$) is solved using an appropriate solver. Let \mathbf{x}^h be the optimal solution, and $\mathbf{f}^h = \mathbf{f}(\mathbf{x}^h)$.
3. In the core module, the next iteration point \mathbf{z}^h is generated by formula (3).
4. In the solver module, problems (P_r^h) are solved to determine the components of $\mathbf{z}^{h+1, lo}$.
5. In the core module, iteration point \mathbf{z}^h , together with the bound vectors $\mathbf{z}^{h+1, lo}$ and $\mathbf{z}^{h+1, up}$, and the distance to the Pareto optimal set d^h obtained according to formula (5) are presented to the decision maker.

The first solver option (i.e., the optimization option) can also be combined with the second preference elicitation option (i.e., the choice of one among several alternatives) although this is only advisable when single objective optimization problems can be solved in very short times. To this end, a large number of vectors of weights (with weights between 0 and 1, and adding up to 1) is initially generated. This can be done, for example, using the approach defined in Steuer and Choo (1983) for the Interactive Weighted Tchebycheff Method. Then, the main steps of each iteration of the resulting method are:

Method A.2

1. In the optimization option of the solver module, problem (2) is solved for $q_i = z_i^{h-1}$, and for each of the weight vectors previously mentioned. The optimal solutions (in the objective space) form a set P^h .
2. In the preference elicitation module, P^h is divided into N_S subsets, and their centroids are calculated, as described in Section 4.1.

3. In the core module, formula (3) is used to calculate an alternative corresponding to each of the N_S centroids.
4. In the solver module, problems (P_r^h) are solved for each of the N_S alternatives.
5. In the core module, the N_S alternatives, together with their bound vectors and their estimated distances to the Pareto optimal set are presented to the decision maker.
6. In the preference elicitation module, the decision maker chooses one of these solutions, which is the next iteration point \mathbf{z}^h .

On the other hand, the a posteriori option of the solver module, i.e., representing the Pareto optimal set in a pre-processing phase, can also be combined with the two preference elicitation options. Let us denote by P the representative set of Pareto optimal solutions generated in the pre-preprocessing phase and by P^h its subset which is reachable from \mathbf{z}^{h-1} . If a direction of improvement is determined in the preference elicitation module, then the resulting method, to be called **Method B.1** is exactly like Method A.1, but problems (2) and (P_r^h) are solved in the set P^h , rather than in the feasible set S . Given that P^h contains a finite number of solutions, this can be done in very short times.

Finally, **Method B.2** combines the second preference elicitation option, i.e., the choice of one among several alternatives, with the a posteriori option of the solver module. In this case, the resulting method differs from Method A.2 in steps 1 and 4. In step 1, the set P^h is formed by the solutions in P that satisfy the bounds $\mathbf{z}^{h,lo}$ and $\mathbf{z}^{h,up}$. In step 4, problems (P_r^h) are solved over P^h . Again, these steps will now take short computing times, given that we are working with a finite number of solutions.

To summarize, the preference elicitation options and the solver options can be combined as wished, and these combinations yield different ways of iterating in NAUTILUS. Some of these combinations have already been reported in published papers, while others have not (see Figure 4).

Let us point out that the combinations of options presented are not necessarily strict, meaning that they can be combined in a flexible way during the interactive solution process. For example, if the computational complexity of the problem makes it advisable, a mixed solver option can be used, and problems (2) can be solved using the optimization option, while problems (P_r^h) can be solved over P^h , using the a posteriori option. This will reduce the computing times of each iteration while producing good quality iteration points. On the other hand, it is also possible to give the decision maker the possibility to switch among different options during the interactive solution process. For example, a decision maker might like to give directions of improvement at the beginning of the solution process, and then see several solutions when approaching the Pareto optimal set. In addition, the generation of the representative set can be done, if possible, also at some point of the process (not necessarily at the beginning), in order to generate only solutions that satisfy the current bound vectors. This

		Preference elicitation	
		Direction of improvement	Choose one solution
Solver	Optimization option	NAUTILUS (1) Miettinen et al. (2010) NAUTILUS 2 Miettinen et al. (2015)	
	A posteriori option		E-NAUTILUS Ruiz et al. (2015)

Figure 4: Combining preference elicitation and solver options

can allow the decision maker to focus on a certain part of the Pareto optimal set which is interesting for her/him. Again, the computational complexity of the problem, and the overall time available for the whole interactive solution process must be taken into account when deciding which mixed option to employ. Therefore, this modular framework and implementation of NAUTILUS provides the decision maker with a flexible decision framework that can be adapted, by combining different options, to the needs of the decision maker at each moment.

As mentioned in the introduction, NAUTILUS methods can be employed in group decision making besides solving problems with a single decision maker or a unanimous group of decision makers. When a group is concerned, the direction of simultaneous improvements may then be a preferred option for preference elicitation but this is naturally up to the decision makers.

5 Conclusions

We have presented the modular NAUTILUS framework for multiobjective optimization and introduced four different main variants of the interactive NAUTILUS methods that can be derived. Furthermore, the framework enables a flexible utilization of the modules involved. What is common in all the methods is that the decision maker does not have to trade off but one can iteratively approach the Pareto optimal set and gain improvements in all objective functions.

The NAUTILUS framework generalizes the NAUTILUS approaches published so far by presenting them in a common framework which enables the birth of new method variants. The NAUTILUS framework provides a foundation for implementation and versatile applicability of method variants.

Acknowledgements

F. Ruiz's research has been partially supported by the Regional Government of Andalucía (group PAI SEJ-445) and by the Government of Spain (project ECO2013-47129-C4-2-R).

References

- Aloysius, J. A., Davis, F., Wilson, D. D., Taylor, A. R., and Kottemann, J. E. (2006). User acceptance of multi-criteria decision support systems: The impact of preference elicitation techniques. *European Journal of Operational Research*, **169**, 273–285.
- Arbel, A. and Korhonen, P. (1996). Using aspiration levels in an interior primal-dual multiobjective linear programming algorithm. *Journal of Multi-Criteria Decision Analysis*, **5**, 61–71.
- Belton, V., Branke, J., Eskelinen, P., Greco, S., Molina, J., Ruiz, F., and Slowinski, R. (2008). Interactive multiobjective optimization from a learning perspective. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Optimization. Interactive and Evolutionary Approaches*, volume 5252 of *Lecture Notes in Computer Science*, pages 405–433.
- Branke, J., Deb, K., Miettinen, K., and Słowiński, R., editors (2008). *Multiobjective Optimization: Interactive and Evolutionary Approaches*, Berlin, Heidelberg. Springer-Verlag.
- Buchanan, J. T. and Corner, J. (1997). The effects of anchoring in interactive MCDM solution methods. *Computers & Operations Research*, **24**(10), 907–918.
- Deb, K., Miettinen, K., and Chaudhuri, S. (2010). Towards an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches. *IEEE Transactions on Evolutionary Computation*, **14**(6), 821–841.
- Ehtamo, H., Kettunen, E., and Hamalainen, R. P. (2001). Searching for joint gains in multi-party negotiations. *European Journal of Operational Research*, **130**(1), 54–69.
- Hwang, C.-L. and Masud, A. S. M. (1979). *Multiple Objective Decision Making – Methods and Applications: A State-of-the-Art Survey*. Springer-Verlag.
- Kahneman, D. and Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, **47**(2), 263–291.
- Luque, M., Miettinen, K., Eskelinen, P., and Ruiz, F. (2009). Incorporating preference information in interactive reference point methods for multiobjective optimization. *Omega*, **37**(2), 450–462.

- Luque, M., Ruiz, F., and Miettinen, K. (2011). Global formulation for interactive multiobjective optimization. *OR Spectrum*, **33**, 27–48.
- Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston.
- Miettinen, K. and Hakanen, J. (2009). Why use interactive multi-objective optimization in chemical process design? In G. Rangaiah, editor, *Multi-Objective Optimization: Techniques and Applications in Chemical Engineering*, pages 153–188. World Scientific.
- Miettinen, K., Ruiz, F., and Wierzbicki, A. (2008). Introduction to multiobjective optimization: interactive approaches. In J. Branke, K. Deb, K. Miettinen, and R. Słowiński, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 27–57, Berlin, Heidelberg. Springer-Verlag.
- Miettinen, K., Eskelinen, P., Ruiz, F., and Luque, M. (2010). NAUTILUS method: An interactive technique in multiobjective optimization based on the nadir point. *European Journal of Operational Research*, **206**, 426–434.
- Miettinen, K., Podkopaev, D., Ruiz, F., and Luque, M. (2015). A new preference handling technique for interactive multiobjective optimization without trading-off. *Journal of Global Optimization*. to appear.
- Morse, J. (1980). Reducing the size of the nondominated set: Pruning by clustering. *Computers & Operations Research*, **7**, 55–66.
- Ruiz, A. B., Sindhya, K., Miettinen, K., Ruiz, F., and Luque, M. (2015). E-NAUTILUS: A decision support system for complex multiobjective optimization problems based on the NAUTILUS method. *European Journal of Operational Research*, **246**, 281–231.
- Ruiz, F., Luque, M., and Cabello, J. M. (2008). A classification of the weighting schemes in reference point procedures for multiobjective programming. *Journal of the Operational Research Society*, **60**(4), 544–553.
- Steuer, R. and Choo, E. (1983). An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, **26**(3), 326–344.
- Szczepanski, M. and Wierzbicki, A. P. (2003). Application of multiple criterion evolutionary algorithm to vector optimization, decision support and reference point approaches. *Journal of Telecommunications and Information Technology*, **3**, 16–33.
- Wierzbicki, A. P. (1980). The use of reference objectives in multiobjective optimization. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making: Theory and Application*, pages 468–486, Berlin, Heidelberg. Springer-Verlag.