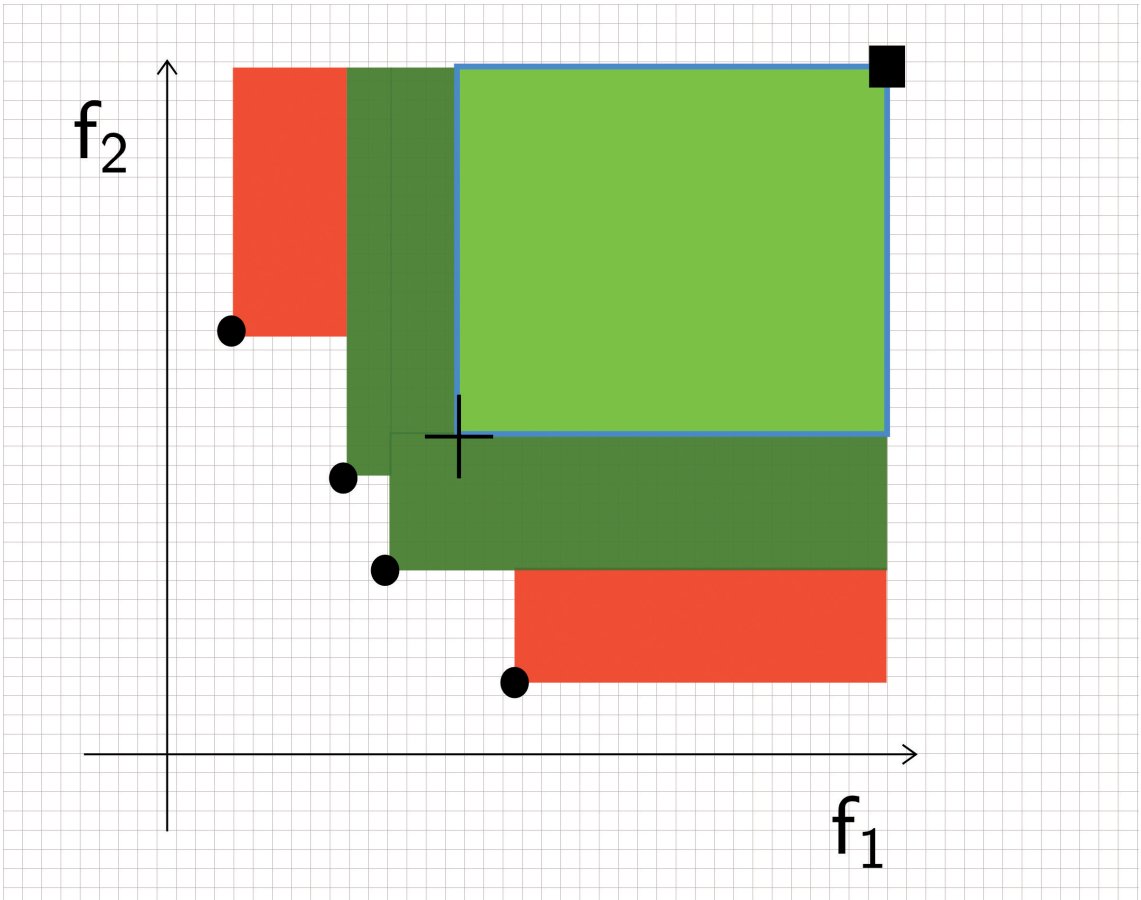
JYU DISSERTATIONS 576

Pouya Aghaei Pour

Preference-based Evolutionary Multiobjective Optimization: Methods, Performance Indicators, and Applications



UNIVERSITY OF JYVÄSKYLÄ
FACULTY OF INFORMATION TECHNOLOGY

JYU DISSERTATIONS 576

Pouya Aghaei Pour

Preference-based Evolutionary Multiobjective Optimization: Methods, Performance Indicators, and Applications

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksen auditoriossa 2
joulukuun 2. päivänä 2022 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Agora, auditorium 2, on December 2, 2022 at 12 o'clock noon.

JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2022

Editors
Marja-Leena Rantalainen
Faculty of Information Technology, University of Jyväskylä
Ville Korkiakangas
Open Science Centre, University of Jyväskylä

Cover picture by Pouya Aghaei Pour.

ISBN 978-951-39-9233-0 (PDF)
URN:ISBN:978-951-39-9233-0
ISSN 2489-9003

Permanent link to this publication: http://urn.fi/URN:ISBN:978-951-39-9233-0

ABSTRACT

Aghaei Pour, Pouya
Preference-based Evolutionary Multiobjective Optimization: Methods, Performance Indicators, and Applications
Jyväskylä: University of Jyväskylä, 2022, 86 p. (+included articles)
(JYU Dissertations
ISSN 2489-9003; 576)
ISBN 978-951-39-9233-0 (PDF)

Multiobjective optimization problems (MOPs) involve optimizing multiple conflicting objective functions simultaneously. As a result of this conflict, we have several mathematically incomparable solutions called *Pareto optimal solutions* with different trade-offs. Typically. in real-world MOPs, a *decision maker* (DM) is needed to choose one of these solutions based on her/his preferences for implementation. In this thesis, we work with methods incorporating DM's preferences during the solution process. We call these methods *preference-based methods*. In this thesis, we develop preference-based evolutionary multiobjective optimization methods and means for assessing their performance.

Real-world MOPs come with several challenges. For example, they can have some objectives and constraints with long computation time. In such problems, we can use some *surrogate models* to replace the expensive functions. However, by utilizing these models, we introduce new challenges: how to incorporate the DM's preferences during the solution process? How can we satisfy constraints if we have used surrogates? How do we manage the surrogate models?

Another challenge we address in this thesis is: how to systematically compare preference-based evolutionary methods? Such comparisons would require quantitative assessments utilizing performance indicators. A handful of performance indicators have been proposed for a priori methods, but no performance indicator has been explicitly designed for interactive methods.

This thesis addresses the challenges mentioned above. We propose an a preference-based method called *KAEA-C*, which is suitable for MOPs involving computationally expensive constraints. It has a novel model management that considers both the DM's preferences and the feasibility of solutions. We identify 13 desirable properties of indicators designed for interactive evolutionary methods. Based on this foundation, we propose a novel performance indicator called *PHI*, which we can utilize to assess the performance of interactive evolutionary methods. Finally, we introduce a novel surrogate-assisted interactive method called *interactive K-RVEA* suitable for computationally expensive problems. We also apply this method to real-world problems.

Keywords: Interactive evolutionary multiobjective optimization, Quality indicators, Computationally expensive problems, Decision making, preference information

TIIVISTELMÄ (ABSTRACT IN FINNISH)

Aghaei Pour, Pouya
Preferenssipohjainen evolutiivinen monitavoiteoptimointi: Menetelmät, suoritusindikaattorit ja sovellukset
Jyväskylä: University of Jyväskylä, 2022, 86 s. (+artikkelit)
(JYU Dissertations
ISSN 2489-9003; 576)
ISBN 978-951-39-9233-0 (PDF)

Monitavoiteoptimointiongelmissa optimoidaan useita ristiriitaisia tavoitefunktioita samanaikaisesti. Ristiriitaisuuden vuoksi niillä on useita ns. Pareto-optimaalisia ratkaisuja, jotka edustavat erilaisia vaihtosuhteita. Pareto-optimaalisia ratkaisuja ei voida matemaattisesti vertailla keskenään ilman lisäinformaatiota. Tyypillisesti reaalielämän monitavoiteoptimointiongelmia ratkaistaessa tarvitaan päätöksentekijä valitsemaan hänen mieltymyksiään ja preferenssejään parhaiten vastaava Pareto-optimaalinen ratkaisu vietäväksi käytäntöön.

Reaalielämän monitavoiteoptimointiongelmissa on erilaisia haasteita. Niissä voi olla tavoitefunktioita tai rajoitteita, joiden arvojen laskeminen vie paljon aikaa. Tällöin voidaan käyttää sijaismalleja korvaamaan aikaa vievät funktiot. Sijaismallien käyttäminen tuo kuitenkin mukanaan uusia haasteita: miten huomioidaan päätöksentekijän preferenssit ratkaisuprosessin aikana, miten huomioidaan rajoitteet sijaismalleja käytettäessä ja miten sijaismalleja tulee päivittää? Toinen tärkeä teema on preferenssipohjaisten evoluutiomenetelmien systemaattinen vertaileminen. Tähän tarvitaan menetelmien toiminnan arviointiin sopivia mittareita. Muutamia mittareita on kehitetty menetelmille, jotka huomioivat preferenssejä ratkaisuprosessin alussa, mutta interaktiivisille menetelmille suunnattuja mittareita ei ole ollut tarjolla. Interaktiivisissa menetelmissä päätöksentekijä antaa preferenssi-informaatiota iteratiivisesti ratkaisuprosessin aikana.

Vastauksena näihin haasteisiin tässä väitöskirjassa esitellään uusi preferenssipohjainen evoluutioalgoritmeihin perustuva menetelmä KAEA-C. Se sopii monitavoiteoptimointiongelmille, joissa on laskennallisesti aikaa vieviä rajoitteita. Siinä on uudenlainen sijaismallien hallintatapa, joka huomioi päätöksentekijän preferenssit ja sen, että ratkaisut noudattavat rajoitteita. Väitöskirjassa tunnistetaan 13 toivottavaa ominaisuutta interaktiivisille evoluutiopohjaisille menetelmille soveltuville mittareille. Tämän pohjalta esitellään uusi mittari PHI, joka soveltuu interaktiivisten evoluutiopohjaisten menetelmien vertailemiseen ja niiden suoriutumisen arviointiin. Lopuksi esitellään uusi sijaismalleja hyödyntävä interaktiivinen menetelmä K-RVEA laskennallisesti aikaa vieville ongelmille. Tätä menetelmää sovelletaan myös reaalielämän ongelmiin.

Avainsanat: Interaktiivinen evoluutiopohjainen monitavoiteoptimointi, laatumittarit, laskennallisesti kalliit ongelmat, päätöksenteko, preferenssi-informaatio

Author

Pouya Aghaei Pour
Faculty of Information Technology
University of Jyväskylä
Finland

Supervisors

Professor Kaisa Miettinen
Faculty of Information Technology
University of Jyväskylä
Finland

Dr. Jussi Hakanen
Faculty of Information Technology
University of Jyväskylä
Finland

Associate Professor Michael Emmerich
Leiden Institute of Advanced Computer Science
University of Lieden
The Netherlands

Reviewers

Professor Sanaz Mostaghim
Faculty of Computer Science
Otto von Guericke University
Germany

Professor Andrzej Jaszkiewicz
Faculty of Computing and Telecommunications
Poznan University of Technology
Poland

Opponent

Professor Luís Paquete
Department of Informatics Engineering
University of Coimbra
Portugal

ACKNOWLEDGEMENTS

I want to express my appreciation to my supervisors Prof. Kaisa Miettinen, Dr. Jussi Hakanen, and Associate Professor Michael Emmerich, for their advice during my doctoral studies. They helped me in various ways, from learning how to execute my ideas to realizing where my flaws are. At every stage of my studies, they believed in me and encouraged me to believe in myself. I believe I could not have asked for a better group of supervisors to guide me through my path to becoming a doctor.

I also want to thank Dr. Sunith Bandaru, who helped me to explore new ideas, which led to interesting research directions. Moreover, I want to thank Dr. Tobias Rodemann, who introduced me to the applications of multiobjective optimization in my very first year of doctoral studies. Finally, I want to thank all of my colleagues and co-authors, Dr. Atanu Mazumdar, Dr. Bhupinder Saini, Dr. Babooshka Shavazipour, Dr. Giomara Larraga Maldonado, Giovani Mistano, Dr. Jana Burkotova, Dr. Tomas Karatky, Risto Heikkinen, and last but not least Adhe Kania.

I also want to thank Honda Research Institute, The Faculty of Information technology of the University of Jyvaskyla, and the Academy of Finland, who provided different funding opportunities for me. I greatly appreciate my partner Veera Rantala, who helped me during my studies, encouraged me when I faced difficult times, and believed in me when I doubted my abilities. Moreover, I would like to thank my parents, who supported me in every stage of life and pushed me to be the best version of myself. I also want to thank my friends Pouya Iazadirad, Mohammad Saberi, and Janne Stefanovski.

LIST OF FIGURES

CONTENTS

LIST OF INCLUDED ARTICLES

PI Aghaei Pour, P., Hakanen, J., Miettinen, K. A surrogate-assisted a priori multiobjective evolutionary algorithm for constrained multiobjective optimization problems. *under review*.

PII Aghaei Pour, P., Bandaru, S., Afsar, B., Miettinen, K. Desirable properties of performance indicators for assessing interactive evolutionary multiobjective optimization methods. *in Proceedings of the Genetic and Evolutionary Computation Conference Companion. Edited by J. E. Fieldsend, The Association of Computing Machinery, New York, 1803–1811*, 2022.

PIII Aghaei Pour, P., Bandaru, S., Afsar, B., Emmerich, M., Miettinen, K. A Performance indicator for interactive evolutionary multiobjective optimization methods. *under review*.

PIV Aghaei Pour, P., Rodemann, T., Hakanen, J., Miettinen, K. Surrogate assisted interactive multiobjective optimization in energy system design of buildings. *Optimization and Engineering*, 23, 303–327 (2022).

1 INTRODUCTION

Real-world optimization problems usually come with several challenges. One of the challenges is dealing with multiple conflicting objective functions (subject to some constraints) that must be optimized simultaneously. We refer to such problems as *multiobjective optimization problem* (MOPs). For example, when buying a cellphone, one may consider different objectives such as cost, battery consumption, and camera quality. Cellphones with low battery consumption and high camera quality are often more expensive than other cell phones. Obviously, a cellphone with a low price, low battery consumption, and high camera quality are preferable, but it is unlikely to find such a cellphone.

As shown in the above example, one can have several trade-offs between cost, battery consumption, and camera quality. In MOPs, we refer to the solutions that represent these trade-offs as *Pareto optimal solutions*[1]. In multiobjective optimization, the evaluation of a candidate solution is a vector in a space (known as the objective space) with a number of dimensions equal to the number of objectives. The set of Pareto optimal solutions is called a *Pareto front* in the objective space. Pareto optimal solutions are mathematically incomparable if we do not have any preference information about the objectives. In practice, one of the Pareto optimal solutions, called *the most preferred solution* must be selected for implementation. Here, we usually need help from a domain expert, also known as a *decision maker* (DM), to provide some preferences to find the most preferred solution.

We can involve the DM's preferences for finding the most preferred solution after or during the solution process [81]. We refer to methods that invoke the DM's preferences after the solution process as *a posteriori methods* and methods that incorporate the DM's preferences during the solution process as *preference-based methods* (see e.g, [45]). In the a posteriori methods, first, a set of Pareto optimal solutions, which represents the entire Pareto front, is presented to the DM. Then, the DM compares different solutions and selects the most preferred one. Some challenges arise by using a posteriori method:

[1] Named after Vilfredo Pareto, who was an Italian economist [91]. See [43] for a detailed history.

1. Comparing several solutions can become challenging when dealing with problems with many objectives;

2. Computing a set of solutions representing the whole Pareto front can be costly, and the DM may not even be interested in the entire Pareto front.

Preference-based methods address the two challenges mentioned above by incorporating the DM's preferences during the solution process. These methods aim to generate solutions that best reflect the preferences of the DM. By doing so, the DM is presented with solutions that are interesting to her/him, and at the same time, we save computation time since it is not needed to generate solutions in the whole Pareto front. Preference-based methods can be divided into two main categories: a priori and interactive ones. In *a priori methods*, the DM provides preference information only once before the solution process. These methods are suitable when the DM is busy and cannot be involved more actively in the solution process. On the other hand, the downside of a priori methods is that the DM may have unrealistic expectations since the DM may be unfamiliar with the reachability of solutions that reflect her/his preferences (e.g., because of complicated interdependencies). For more details on a priori methods, see e.g, [10, 25]

In *interactive methods*, the DM is actively involved in the solution process and updates her/his preferences to guide the search direction toward the most preferred solution. In interactive methods, the DM has the chance to learn about the trade-offs and the feasibility of her/his preferences and update them iteratively. More precisely, we can often observe that when applying interactive methods, DMs have two phases, called a *learning phase* and a *decision phases* [83]. During the learning phase, the DM explores different solutions and increases her/his knowledge about the problem until she/he identifies a so-called *region of interest* (ROI). Then, in the decision phase, the DM fine-tunes the solutions within the region of interest until she/he finds the most preferred solution. For more details on interactive methods, see e.g, [5, 82, 112]

So far, we have discussed how the DM can be involved in the solution process when solving an MOP. Over the years, many a posteriori and preference-based methods have been developed [32, 63, 81]. Among them, evolutionary multiobjective optimization methods (evolutionary methods for short) are one of the most popular methods. The popularity is due to their ability to handle e.g., discontinuous functions and different types of variables. However, evolutionary methods have some drawbacks, too. For example, they require many function evaluations to approximate the Pareto front. The number of function evaluations becomes more problematic when MOPs have an objective or constraint function that its evaluation needs a lot of computational resources, like examples e.g., in [21, 40, 59, 72, 89, 97]. We refer to such functions as *expensive functions* and such problems as *computationally expensive MOPs*.

To reduce the time required for solving computationally expensive MOPs with evolutionary methods, we can use surrogate-assisted evolutionary methods [24, 66]. In these methods, we train machine learning predictive models known as *surrogate models* [67]. Surrogate models mimic the behavior of the ex-

pensive functions during the solution process in an inexpensive manner. Several surrogate-assisted a priori and a posteriori evolutionary methods have been developed in the literature, see, e.g., [8, 29, 53, 73, 98]. However, to the best of our knowledge, there is no surrogate-assisted a priori evolutionary method that can solve constrained MOPs with at least one computationally expensive constraint.

Even though using surrogate models reduces the computation time significantly, it introduces new challenges due to the *uncertainty* (inaccuracy) of the predictions (of the objective and constraint functions) values. Examples of these challenges include:

S1: How to identify feasible solutions when working with surrogate models?

S2: How to identify feasible solutions that reflect the DM's preferences when working with surrogate models?

S3: How to incorporate the DM's preferences iteratively?

Improving the accuracy of surrogate models is a common way to tackle the above-mentioned challenges. By doing so, the prediction values of surrogate models become more trustworthy, and the uncertainty within the predicted values will decrease. To improve the accuracy of the surrogate models, we can iteratively choose some solutions to *update* (re-train) the surrogate models. We refer to this process as *model management*. Various model management ideas have been proposed in the literature, e.g., [24, 66, 67, 98], that consider different criteria. We usually have a limited budget for expensive function evaluations. Therefore, choosing (or developing) appropriate model management approaches is crucial to surrogate-assisted evolutionary methods.

Thus far, we have discussed how to solve MOPs. However, over the years, various evolutionary methods have been developed and choosing the most suitable one is challenging. Moreover, since the DM is a domain expert but not in multiobjective optimization, we cannot expect her/him to select the most suitable evolutionary method. Here, an *analyst* with knowledge of multiobjective optimization assists the DM and selects the suitable method for her/him.

Because of the trade-offs among objective values, assessing the performance of evolutionary methods designed for MOPs is not straightforward. There have been various studies on how to assess the performance of a posteriori evolutionary methods [126], and many *performance indicators* (indicators for short) have been developed, see, e.g., [6, 95]. Moreover, some indicators have been developed for a priori evolutionary multiobjective optimization methods, like [9, 58, 75, 85, 118]. Furthermore, there have been some attempts to compare different interactive evolutionary methods [4, 113]. For instance, by conducting experiments where different DM's test different interactive evolutionary methods [4] or using artificial decision makers (ADM) to replace humans [3, 5]. However, no indicator has been explicitly proposed for interactive evolutionary methods that could quantify performance in a meaningful way. To develop an indicator for measuring the performance of any method, first, we need to understand the desirable

properties the method and, therefore, the indicator needs to possess. For example, the desirable properties of indicators developed for a posteriori and a priori evolutionary methods are described in [122] and [85], respectively. However, there has not been much study on interactive evolutionary methods. Challenges of designing an indicator for interactive evolutionary methods include the following:

IN1: What desirable properties should an indicator designed for interactive evolutionary methods possess?

IN2: How to incorporate the DM's preferences in a meaningful way to assess the performance of interactive evolutionary methods?

IN3: How to assess the performance of interactive evolutionary methods as a whole process?

This thesis is a collection of four articles published in or submitted to scientific journals (Articles PI - PIV) that address the challenges mentioned (S1-S3 and IN1-IN3). In Chapter 3, we discuss a novel Kriging-assisted a priori evolutionary algorithm for constrained problems, called *KAEA-C*, proposed in Article PI, for MOPs with three or more objectives. KAEA-C can handle MOPs with at least one computationally expensive objective and constraint while incorporating the DM's preferences provided a priori. To our knowledge, KAEA-C is the first surrogate-assisted a priori evolutionary method designed for MOPs with at least one computationally expensive constraint and the ability to incorporate the DM's preferences. KAEA-C utilizes some elements from an evolutionary method called *reference vector guided evolutionary algorithm* (RVEA) [20] to decompose the objective space and Kriging models [50, 99] to approximate the expensive functions. KAEA-C handles computationally expensive constraints and the DM's preferences in the solution process by replacing expensive functions with the Kriging models. Furthermore, KAEA-C has an appropriate model management approach, which has been developed for computationally expensive constrained problems. We assess the performance of KAEA-C on some benchmark and engineering problems and show why it is important to develop more methods for such problems.

Despite the excellent performance of KAEA-C, it may not be sufficient for solving real-world problems if a DM wants to update one's preferences. Therefore, we needed a surrogate-assisted interactive method. However, as mentioned earlier, no indicators have been proposed in the literature to assess interactive evolutionary methods. Because of this, in Chapter 4, we discuss the desirable properties that an ideal indicator should possess for assessing the performance of interactive evolutionary methods. One common way to assess interactive evolutionary methods is to see them as a series of a priori steps and then consecutively use indicators suitable for such methods to assess the performance. In Article PII[2], we demonstrate why this approach is not ideal for interactive methods

[2] the supplementary materials can be found here:
https://github.com/ppouyaa/desirable-properties-master

and how it can be misleading. In fact, we argue that we need different indicators for assessing the performance of the learning and the decision phases since they have different characteristics [5].

The hypervolume indicator [125] has been originally developed for a posteriori methods, but we propose a hypervolume-based indicator for assessing the performance of interactive evolutionary methods. We refer to this indicator as *preference-based hypervolume indicator* (PHI) in Article PIII[3] and introduce it in Chapter 5. PHI can identify solutions that reflect the DM's preferences (in the form of a reference point) and those that do not. Then, this indicator uses the hypervolume indicator for rewarding the solutions that reflect the DM's preferences (similar to [115]) and punishes the performance assessment based on the solutions that do not reflect the preferences. Moreover, we propose two separate ways to utilize PHI for assessing the performance of the learning and the decision phases.

In Chapter 6, we go back to developing an interactive evolutionary method for a real-world MOP since we now have the indicator PHI for performance assessment. The problem at hand (see [97]) has four conflicting objectives with regard to configuring energy consumption in large buildings. Three of the objectives are computationally expensive. The Kriging-assisted interactive evolutionary method we developed for this problem is called *interactive K-RVEA*, proposed in Article PIV. It uses elements of interactive RVEA [56] to direct the search direction toward the DM's preferences. Additionally, interactive K-RVEA uses a unique model management to update the Kriging models with the best solutions that reflect the DM's preferences.

Moreover, in Chapter 6 we apply interactive K-RVEA to another real-world problem introduced in [17]. This problem is about designing a pump and maximizing the fluid flow in different parts of it. This problem has three conflicting objectives, and evaluating each function requires running an extremely computationally expensive simulator. More precisely, each simulation run takes between 16-20 hours. Another challenge with this problem is that sometimes the simulations may fail, and we cannot get the objective values for the given decision variables. We show how interactive K-RVEA can help to reduce the computation time, and by incorporating the domain expertise of the DM, we can avoid failed simulations.

The rest of this thesis is organized as follows. In Chapter 2, we discuss the main terminologies and background information used in this thesis. Then, we discuss the surrogate-assisted a priori method KAEA-C (introduced in Chapter 3). In Chapter 4, we provide the desirable properties that indicators for interactive evolutionary methods should possess and give some examples justifying the need for such indicators. We develop a novel indicator for interactive evolutionary methods called PHI in Chapter 5 and use it on an engineering benchmark problem to demonstrate how we can assess interactive evolutionary methods. We introduce a surrogate-assisted interactive evolutionary method in Chapter 6 and

[3] the supplementary materials can be found here: https://github.com/ppouyaa/PHI

demonstrate it with two applications. The conclusions, future research directions, and author's contributions are elaborated in Chapter 7.

2 BACKGROUND

In this chapter, we present a short description of the general form of MOPs considered, overviews of constraint handling techniques and interactive evolutionary methods as well as indicators to be utilized in the following chapters. In addition, we discuss several evolutionary methods and indicators for a priori evolutionary methods that are relevant to this thesis.

2.1 Multiobjective Optimization

We define an MOP as:

$$\begin{aligned} &\text{minimize } f(x) = (f_1(x), \ldots, f_k(x)) \\ &\text{subject to } g_i(x) \geq 0, && i = 1, \ldots, m \\ &\qquad x_j^l \leq x_j \leq x_j^u && j = 1, \ldots, n, \end{aligned} \tag{1}$$

where $f(x)$ denotes an objective vector which consists of the values of $k(\geq 2)$ conflicting objective functions f_i at $x = (x_1, \ldots, x_n)^T$, an n-dimensional decision variable vector (for short, *decision vector*). The decision vector belongs to an n-dimensional space called the decision space, and the objective vector belongs in a k-dimensional space called the objective space. Note that in problem 1, the objectives may require to be maximized.

We say a decision vector x is feasible in the decision space, if it satisfies the inequality $g_i(x) \geq 0$ for all indices i and all the box constraints. The set of all feasible decision vectors is denoted by $\mathbb{F}$. In contrast, we call a decision vector infeasible if it violates the inequality $g_i(x) \geq 0$ for at least one index i. We call an objective vector feasible if the corresponding decision vector is feasible. In what follows, we refer to objective vectors as solutions.

We say that $f(x^1)$ dominates $f(x^2)$ ($f(x^1) \succ f(x^2)$) if $f_i(x^1) \leq f_i(x^2)$ and $f_j(x^1) < f_j(x^2)$ for at least one index j. We refer to the solutions that do not dominate each other as *nondominated solutions*. Moreover a feasible solution $f(x)$

is Pareto optimal if there does not exist another feasible solution $f(x^\star)$ such that $f(x^\star)$ dominates $f(x)$. Note that nondominated solutions are not always Pareto optimal, but all Pareto optimal solutions are nondominated.

Some solutions in correlation to the Pareto front are of particular importance. An *ideal point* $z^\star$ is a point that is attained by optimizing the objective functions individually. Typically, the ideal point is not feasible because of the trade-offs between objectives. However, if the objectives are not conflicting, the MOP at hand has only one optimal solution, which is the ideal point. The components of a *nadir point* represent the worst objective values attained in the Pareto front. There is no exact method for finding the nadir point except for bi-objective problems [42]. However, a few methods can approximate the nadir point (e.g., [11,35, 36,81]). In real-world problems, it is usually hard to calculate the ideal and nadir points. In this thesis, we use the population of evolutionary methods to calculate them. Finally, we define a new concept, called a *dystopian point* as a point slightly worse than the nadir point. In other words, we add a small positive number to each component of the nadir point.

Various preference-based methods assume the DM to provide preferences in different ways. Examples of different preference types can be seen in [1,20,71,74]. Among them, providing desirable values for each objective that the DM wishes to see is regarded as understandable to her/him [10, 111] since both the desires and solutions are in the same space. These values form a point, which we refer to as a *reference point*. Throughout this thesis and the Articles PI, PII, PIII, PIV, we have used reference points as the type of preference information that the DM provides.

2.2 Evolutionary Multiobjective Optimization Methods

Several evolutionary methods have been developed to solve MOPs with different characteristics[1], see, e.g., [2, 32]. Evolutionary methods are nature-inspired and based on the concepts of Darwinian evolution [28]. To put the idea briefly, a population of species is evaluated by its environment, and the "fittest" species have the best chance to breed and pass their "good" genes to their offspring. Here, genes are recombined and mixed, and the next generation will have more of the genetic characteristics considered good within the population's species.

Figure 1 illustrates the general framework of evolutionary methods. Evolutionary methods view a set of decision vectors as a population. a mathematical function to describe the fitness of the decision vectors is created and referred to this function as a *fitness function*. Fitness functions vary in evolutionary methods. We refer to assessing the fitness of decision vectors and selecting the most fitted one as a selection strategy. After the selection strategy, we use crossover and mutation operators [65] to recombine the decision vectors and generate the offspring. We refer to the acts of selection and recombination as one generation.

[1] The very first evolutionary method for solving MOPs was proposed in [102].

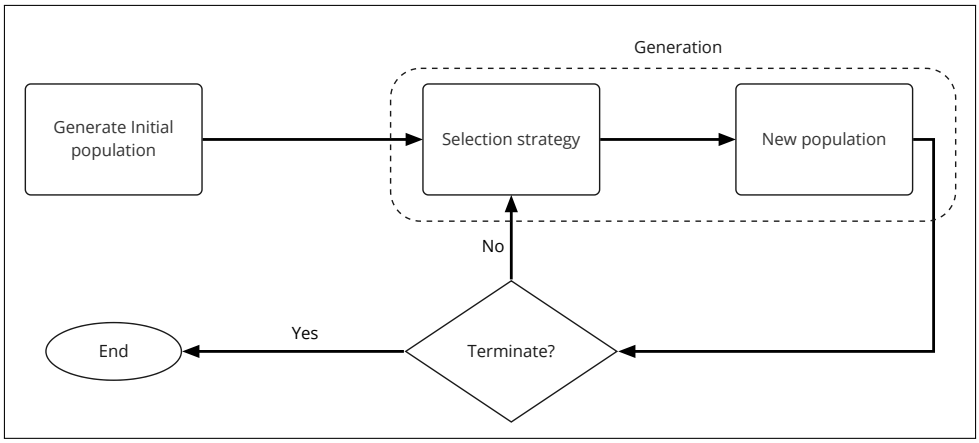


FIGURE 1 General framework of evolutionary methods.

We continue the generations until some stopping criterion is met.

2.2.1 Different Types Of Evolutionary Methods

Evolutionary methods can be divided into three main classes [45]:

1. dominance-based [18];
2. indicator-based [46] and;
3. decomposition-based methods [101, 107, 114].

Dominance-based methods use the concept of dominance to assess a population's fitness and select the offspring. These methods perform well for problems with up to three objectives [101]. However, they seem to have some issues in computation time and converging toward the Pareto front when we deal with more than four objectives [52]. The very first dominance-based method was proposed in [49]. Additionally, NSGA-II [37] and SPEA2 [123] are famous dominance-based methods. Moreover, there are some preference-based dominance-based methods in the literature as well [48, 117].

Indicator-based methods [46] use an indicator such as hypervolume indicator [125](see, e.g., [13]), or R2 indicator [57] (see, e.g., [92]) to assess the performance of a set of solutions. These methods use the indicator assessment to convert the MOP into a new optimization problem with a single objective by optimizing the improvement of the performance assessment. Indicator-based evolutionary methods can become computationally expensive as the number of objectives increases. According to a recent survey [46], SMS-EMOA [13] is the most representative indicator-based method. For a set of nondominated solutions, the ones that contribute the most to the hypervolume indicator are selected for the next population. In the literature, there are preference-based versions of indicator-based methods. For example, see [68, 106]

Decomposition-based evolutionary methods have mainly been developed for MOPs with more than three objectives since the performance of the methods in the other two classes drops for such problems [14, 20]. These methods use

evenly distributed reference vectors (or weight vectors) to divide the objective space into many *subspaces*. In each subspace, we have to solve a *subproblem*. Then for each subproblem, one solution gets selected based on a fitness function that assesses the fitness of the solutions of each subproblem. Among decomposition-based evolutionary methods, NSGAIII [34], RVEA [20], and MOEA/D [120] are famous ones.

In this thesis, we focus on preference-based decomposition-based evolutionary methods since they offer a natural possibility for incorporating a DM's preferences. To be more specific, they can adjust the positions of the reference vectors according to the preferences provided by a DM. For example, in [56], the method can incorporate four different types of preferences. Another recent example is an interactive version of MOEA/D [74], which adjusts weight vectors.

2.2.2 RVEA

The reference vector-guided evolutionary algorithm (RVEA) was proposed to handle problems with a high number of objectives [20]. Moreover, a function called *angle penalized distance* (APD) is used for the selection strategy that can balance convergence and diversity. According to the experiments in [20], the performance of RVEA is competitive with other evolutionary methods such as NSGA-III and MOEA/D. In this thesis, we use some components of RVEA in developing the new methods KAEA-C and interactive K-RVEA.

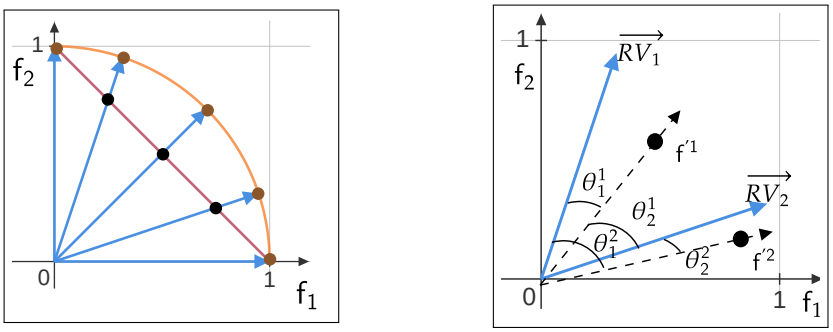


FIGURE 2 An example of (a) decomposition of the objective space by using reference vectors and (b) assigning solutions to reference vectors

Figure 2a illustrates how RVEA uses reference vectors to decompose the objective space into subspaces. First, a set of uniformly distributed points (black circles in the figure) is generated using the canonical simplex-lattice design method [19] on a unit hyperplane (illustrated as a red line). Then, the reference vectors (illustrated as blue vectors) are obtained by projecting the generated points from the hyperplane to a hypersphere (illustrated as an orange arch).

After decomposing the objective space, for each generation t, we translate the objective vectors as $f'^i(x) = f^i - z_i^\star$, where $i = 1, \ldots, |P_t|$, and $|P_t|$ is the size of population P_t at generation t so that the origin of the reference vector is always the origin (see Figure 2b). Then, we assign the translated objective vectors to the

reference vectors. Here, we assign $f'^i(x)$ to the reference vector with the smallest angle. Figure 2b demonstrate how we can assign the solutions in P_t to different reference vectors. Assume P_t only contains translated objective vectors f'^1 and f'^2 (denoted as black filled circles). we calculate the angles (θ_1^1, θ_1^2, θ_2^1, and θ_2^2) between these solutions and the reference vectors v_1 and v_2 (denoted as black vectors) and assign them to the reference vector that they have the smallest angle with. For example, in Figure 2b, f^1 is assigned to v_1 and f^2 is assigned to v_2. Then we use APD to select the best solution of each subspace to create the next population P_{t+1}. APD assesses each solution based on two criteria: the distance of solutions to the ideal point $z^\star$ and the angle between each solution and the reference vector they are assigned to. The first criterion is responsible for leading the search direction toward the Pareto front, and the latter one is responsible for maintaining diversity.

Preference-based versions of RVEA [20, 56] have also been proposed. An a priori RVEA was proposed in [20], where reference vectors are generated around the DM's preferences (given in the form of a reference point) by using the following equation:

$$\bar{v}^i = \frac{r \cdot v^i + (1-r) \cdot v^c}{\|r \cdot v^i + (1-r) \cdot v^c\|}, \tag{2}$$

where $v_j^c = \frac{\hat{z}_j}{\|\hat{z}\|}$, and $\|\hat{z}\| \geq 0$ is the Euclidean norm of the reference point. Note that if $\|\hat{z}\| = 0$, then we set v^c to be the unit vector. The parameter $r \in (0, 1)$ controls how the reference vectors are adjusted towards the reference point. If r is close to 1, then the reference point has less effect on the reference vectors, and if it is close to 0, they will get closer to the reference point. According to [67, 114] RVEA has the most straightforward way of incorporating the DM's preferences among decomposition-based evolutionary methods. In Articles PI,PIV we use (2) for generating the reference vectors.

2.3 Surrogate Assisted Evolutionary Multiobjective Optimization Methods

As mentioned in Chapter 1, real-world MOPs may possess computationally expensive functions. This makes evolutionary methods practically inapplicable to such problems, since they need many generations and function evaluations. As mentioned in the previous chapter, surrogate-assisted evolutionary methods provide a way to reduce the computation time required by evolutionary methods [24, 66].

In Chapters 3 and 6, we assume that at least one of the functions (either objective or constraint) is computationally expensive to evaluate. This means that during the optimization process we might deal with both computationally expensive and inexpensive functions. Therefore, we define the following terms for more clarity:

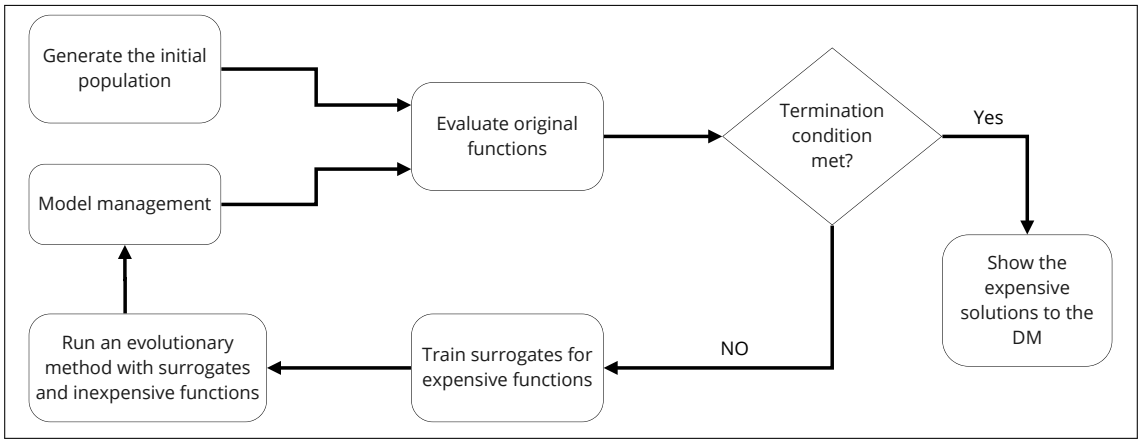


FIGURE 3 General framework of surrogate-assisted methods

Term 1 ***Original function:*** *An MOP may have computationally expensive or inexpensive functions. We refer to an objective or constraint function of an MOP as an original function before training a surrogate model for an expensive one.*

Term 2 ***Expensive function:*** *We refer to a computationally expensive function that we need to train a surrogate model for as an expensive function.*

Term 3 ***Expensive evaluation:*** *We refer to evaluating an expensive function at a given decision vector as an expensive evaluation.*

Term 4 ***Expensive solution:*** *We refer to a solution $f(x)$ whose corresponding decision vector x has been used for evaluating an expensive function as an expensive solution.*

Term 5 ***Surrogate evaluation:*** *We refer to evaluating a surrogate function at a decision vector as a surrogate evaluation.*

Term 6 ***Surrogate solution:*** *We refer to a solution whose corresponding decision vector has been used only for evaluating a surrogate function as a surrogate solution.*

Term 7 ***Iteration:*** *Every time we use model management to update surrogate models, we say an iteration has happened. In this thesis, each iteration is a fixed set of generations.*

Term 8 ***Interaction:*** *An interaction is a fixed number of generations (or sometimes iterations), where the DM provides her/his new preferences.*

Figure 3 illustrates the main steps of surrogate-assisted evolutionary methods. After generating the initial population, we need to evaluate the decision vectors with the original functions. Then, we train the surrogate models for the expensive functions. Next, we run an evolutionary method that optimizes the surrogate models and inexpensive functions for a fixed number of generations. After the optimization process, we should apply model management to select one or more solutions for updating the surrogate models. We repeat this process until we meet a stopping criterion, such as the limit for expensive evaluations.

Different surrogate models in the literature can be utilized to replace expensive functions, that is, mimic their behavior [41, 78, 99]. Among them, Kriging models [50] are widely used surrogate models [24, 66]. The popularity of Kriging models is due to their ability to provide uncertainty information with the predicted values for the objectives and constraints. The uncertainty information can be used to design new model management approaches that can improve the accuracy of the models or find new (expensive) solutions that are interesting to the DM. In Articles PI and PIV, we have used Kriging as the surrogate model.

2.3.1 Kriging

Kriging models (also known as Gaussian process) have been used in different fields such as Bayesian optimization [103], geo-statistics [27], hydraulic pump design [72] and even in the field of surrogate-assisted multiobjective optimization methods [70, 121]. A Kriging model can be expressed as a multivariate normal distribution with a mean μ and a covariance matrix C:

$$y \sim \mathcal{N}(\mu, C). \tag{3}$$

In Articles PI, PII, we assumed that the mean is zero. The covariance matrix C uses a kernel function $\kappa(\vec{x}, \vec{x'})$ that performs a pairwise comparison between two decision vectors $\vec{x}$ and $\vec{x'}$. Different kernels can be used in Kriging models [93]. Based on some preliminary tests on different kernels, in Articles PI, PII we used the Matern 5/2 kernel, that can be expressed as:

$$\begin{aligned}\kappa(\vec{x}, \vec{x'}) = \sigma_{\mathrm{f}}^2 \left(1 + \sqrt{5} \sum_{j=1}^{n} \frac{\|x_j - x'_j\|}{l_j} + \frac{5}{3} \sum_{j=1}^{n} \frac{\|x_j - x'_j\|^2}{{l_j}^2}\right) \\ \exp\left(-\sqrt{5} \sum_{j=1}^{n} \frac{\|x_j - x'_j\|}{l_j}\right) + \sigma_{\mathrm{t}}^2 \delta_{\vec{x}\vec{x'}},\end{aligned} \tag{4}$$

where the σ_f and l_j are the amplitude and length scale of the jth decision variable, $\delta_{\vec{x}\vec{x'}}$ is the Kronecker delta function [54], and σ_t is the noise parameters. In Articles PI and PIV, where we deal with computationally expensive problems, we define the uncertainty of a decision vector $x^\star$ as:

$$unc(x^\star) = \kappa(\vec{x^\star}, \vec{x^\star}) - \kappa(\vec{x^\star}, X)^T C^{-1} \kappa(\vec{X}, \vec{x^\star}). \tag{5}$$

We use the uncertainty information during the model management for updating the Kriging models.

2.3.2 Model Management

As mentioned earlier, model management selects some of the solutions evaluated by the expensive functions to update the surrogate models with. In the literature, different model management approaches have been proposed [67]. For instance,

in Bayesian optimization, the solution that maximizes the acquisition function gets selected [115], or in a method called K-RVEA [22] the number of solutions selected should be set as a parameter, and the solutions are selected that have the highest uncertainty.

Moreover in some model management approaches, at the end of each iteration[2], we use the whole final population to update the surrogate models. However, this approach may not be ideal if a limited, expensive evaluation budget is given, or if the problem is extremely expensive [17].

2.3.3 Handling Computationally Expensive Constraints

According to [77], although many real-world MOPs possess some computationally expensive constraints [26, 40, 59, 79, 90, 119], handling them has not been explored enough. However, recently some new methods have been developed for handling computationally expensive constraints [8, 29–31, 98].

Most methods for handling computationally expensive constraints use an acquisition function that measures the expected improvement in a performance quality indicator [8, 80]. For example, most methods suitable for handling computationally expensive constraints use an acquisition function that measures the expected improvement in a performance quality indicator. The solution with the best acquisition function value is chosen for updating the surrogate models.

2.3.4 Surrogate-assisted Preference-based Evolutionary Methods

Surrogate-assisted preference-based evolutionary methods are quite rare to find. A handful of methods can incorporate a DM's preferences when solving computationally expensive problems [23, 53, 108, 116]. For example, in a priori K-RVEA [23] preferences are only considered during the optimization process, and the model management does not consider them. In [116], the DM provides her/his preferences as a reference point, and a desired region is formed around it. Then an acquisition function is developed that measures the expected hypervolume improvement within this desired region. Additionally, in [53], the DM is able to provide multiple reference points and then an acquisition function is used that can incorporate all of them along with the expected hypervolume improvement. Then, the solution that maximizes the acquisition function is selected for updating the surrogate models. In [108] a utility function is used to find the most preferred solution at each iteration and use that solution to update the surrogate models. Then the method tries to generate new solutions that are similar to the most preferred solution of the previous iteration.

In computationally expensive problems we usually have limited expensive evaluation budget [24, 66], and finding the most preferred solution is exacerbated by this limitation. Additionally, the existence of expensive constraints makes it harder to find feasible solutions. Even though using DM's preferences in MOPs with computationally expensive constraints seems complementary, to the best

[2] Reminder: see term 7 for the definition of iteration.

of our knowledge, no method in the literature can both handle computationally expensive constraints and incorporates a DM's preferences.

2.4 Indicators

In single objective optimization, it is quite clear how to assess the performance of different methods. We can refer to the optimal objective function values found, and the method with the lowest objective value (assuming we are minimizing) has the best performance. Or a method that finds a good enough solution fastest. However, the performance assessment is not a straightforward task in multiobjective optimization. Because of the existence of different trade-offs, it becomes tricky to compare different methods against one another. There are many studies regarding different types of indicators [7, 76, 95]. In this thesis, for compactness, we refer to the indicators designed for a posteriori methods as *a posteriori indicators*, indicators designed for a priori methods as *a priori indicator*, and indicators designed for interactive methods as *interactive indicators.*

In the following subsections, we provide a short review of indicators relevant to this thesis. For example, according to [126] a good a posteriori evolutionary method should be able to generate a set of solutions that is as close as possible to the Pareto front (convergence), and can present a good distribution of solutions that represents the whole shape of the Pareto front (diversity). Thus, an indicator should measure both convergence and diversity. But in preference-based evolutionary methods, solutions need to reflect also the preferences and, thus, convergence and diversity around the preferences.

2.4.1 A posteriori indicators

As mentioned, according to [126] an a posteriori indicator should be able to assess the following:

1. The distance between the Pareto front and the outcome of the method, that is, the approximated front;

2. The diversity of the approximated front;

3. The extent of the approximated front, meaning a wide range of values covered for each objective.

According to [6, 95, 104] the hypervolume indicator is one of the most famous indicators widely used for a posteriori evolutionary methods.

The hypervolume indicator was first proposed in [124], and it was referred to as the "Size of the space covered". For minimization problems like problem (1), we can define the hypervolume indicator as follow:

The hypervolume indicator maps a point set in $\mathbb{R}^k$ to the measure of the region dominated by that set and bounded above by a given *dystopian point*[3] $z^{dy} = \{z_1^{dy}, \ldots, z_k^{dy}\}$.

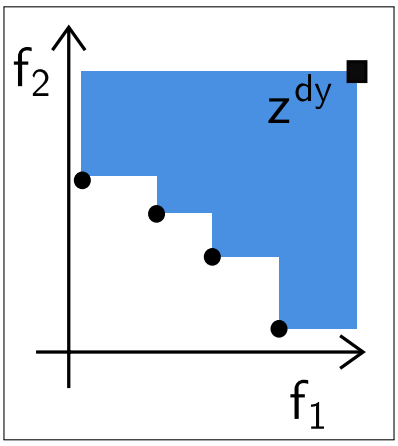


FIGURE 4 Hypervolume in a bi-objective case. The dystopian point is denoted as the black rectangle. The blue area represents the hypervolume of the solution set consisting of black circles.

In Figure 4 the blue area is the part of objective space that is calculated by hypervolume for a set of solutions in a bi-objective problem. Here, the hypervolume is the dominated area defined by these three solutions and the dystopian point (the blue area).

Besides the popularity of the hypervolume indicator, it has some disadvantages. For example, it is computationally expensive to calculate [39]. The computation time indicator increases exponentially as the number of objectives increases [12, 16, 55]. Choosing the dystopian point can also adversely affect the value of the hypervolume indicator performance assessment [60, 61]. Some attempts have been made to provide a framework to specify the dystopian point [60] for some specific problems. However, there is no general guideline for specifying the dystopian point. In this thesis, we assume that the nadir point is provided, and we specify the dystopian point as $z^{dy} = \{z_1^{nad} + \epsilon, \ldots, z_k^{nad} + \epsilon\}$, where epsilon is a small positive constant. In this thesis we denote the function that calculates the hypervolume of a set of nondominated solutions as:

$$hypervolume = HV(P, z^{dy}). \tag{6}$$

2.4.2 A Priori Indicators

As mentioned earlier, some indicators have been designed to evaluate a priori evolutionary methods. In [85], a list of desirable properties for a priori indicators has been proposed. They are the following:

[3] In the literature dystopian point is also referred to as reference point. However, to avoid confusion, in this thesis we refer to it as dystopian point.

1. Form a desired region by incorporating the DM's preferences;
2. Measure both convergence and diversity of the solutions with respect to the preferred region;
3. Be independent of knowledge of Pareto front;
4. Scale well as the number of objectives increases.

The authors mentioned that the a priori indicators (e.g., [109, 110, 126]) that existed at the time of publishing their list, did not possess these desirable properties. Moreover, the authors in [85] developed a new a priori indicator called *user preference metric based on composite front* (UPCF). The main steps of UPCF are as follows:

1. Merge the solution set of all methods to be compared into a set and select all nondominated solutions from this set. This set is referred to as a *composite front*;
2. Choose the solution with the smallest Euclidean distance to the reference point. This solution is known as the *mid-point*;
3. Use any a posteriori indicator to assess the performance of solutions within a cubic centered at the mid-point with a side length of a parameter Δ (determines the size of the desired region).

The main drawback of UPCF is that sometimes all the solutions generated by some method may be outside of the desired region. In such a case, UPCF cannot assess the performance of this method.

A new indicator called R-metric was introduced in [75] to address the drawback of UPCF mentioned above. The main difference between R-metric and UPCF is that after generating the composite front, a point called a *pivot point* is found instead of finding the mid-point. The pivot point for each solution set is the solution that has the lowest achievement scalarization function (ASF) [111] value in a solution set. Once the pivot point is found, a parameter Δ is used to determine the size of the desired region to remove solutions that are too far away from the reference point. Next, the pivot point is used to transfer the solutions into a so-called virtual space. Finally, an a posteriori indicator is used to assess the performance of the transferred solutions. Because of how R-metric transfers the solutions, the desired region is never empty for any solutions. Therefore, the solution sets are always comparable. In this thesis, whenever we use R-metric we will utilize hypervolume in it and we refer to it as R-HV.

UPCF and R-metric only consider the solutions that are inside the desired region. This way of assessing the performance can be misleading if the DM sees all of the solutions, but only some of them play a role in the performance assessment. Indicators proposed in [58, 118] penalize the performance of a priori methods in case some solutions exist outside of the desired region. In PMDA [118], the desired region is identified based on a reference point and a parameter that

determines its size. Then, only their distance to the ideal point is calculated for solutions inside the desired region. However, if a solution lies outside the desired region, its performance is penalized based on how far it is from the desired region. Another a priori indicator called PMOD [58] has almost the same steps as PMDA, but the penalty function is different.

All the indicators mentioned so far rely on one or several parameters, like the size of the desired region or a penalty coefficient. This has two main issues. First, these parameters' role in the final performance assessment has not been studied. Second, setting these parameter values is challenging due to their unintuitive nature. Therefore, a parameterless indicator called EH-metric has been proposed in [9]. It uses the concept of an *expanding hypercube*, which starts as a point at the reference point and expands (with the reference point at its center) until it envelops all solutions. The EH-metric value for an a priori method is calculated as the area under the curve generated by plotting the fraction of solutions enveloped by the hypercube as it expands versus the size of the hypercube. The EH-metric can measure convergence and diversity without relying on the knowledge of the Pareto front.

In Articles PII,PIII, we use the indicators mentioned above to compare some preference-based methods. To the best of our knowledge, no interactive indicator has been proposed in the literature. In Article PIII, we demonstrate why we cannot utilize a series of a priori indicators for assessing the performance of interactive evolutionary methods.

3 A PRIORI EVOLUTIONARY METHOD WITH COMPUTATIONALLY EXPENSIVE CONSTRAINTS

As mentioned, one of the advantages of considering the DM's preferences while solving MOPs is to save computation time, which is particularly essential in computationally expensive MOPs. However, it is not straightforward how to handle DM's preferences in this setting, and only a handful of methods can do this. In addition, if the computationally expensive MOPs have some expensive constraints, it becomes even more challenging to incorporate the DM's preferences. In fact, there is no preference-based method in the literature that is designed for computationally expensive constrained MOPs. KAEA-C[1] (proposed in Article PI) uses Kriging models to replace the computationally expensive functions and incorporates the DM's preferences provided in the form of a reference point.

For the selection strategy, where we optimize surrogate models and original functions (as defined in Section 2.3), KAEA-C decomposes the objective space by using equation (2) (similar to a priori RVEA [20]). KAEA-C selects a set of feasible nondominated surrogate solutions based on two fitness functions. The first fitness function measures the distance to the ideal point (calculated based on the current population), and the second one is the angle between the surrogate solutions and the reference point. To handle the constraints during the selection strategy, KAEA-C prioritizes feasible surrogate solutions over infeasible ones. In addition, if there is no feasible surrogate solution, KAEA-C uses a ranking system based on the normalized constraint violation of all constraints and the number of constraints violated.

For the model management, KAEA-C takes advantage of the uncertainty information that Kriging models provide to select two types of surrogate solutions to evaluate the expensive functions (the terms have been defined in Section 2.3):

1. Surrogate solutions with the most potential to improve the accuracy of the Kriging models the most (Surrogate solutions that have a high uncertainty in their predictions).

[1] As mentioned earlier, the acronym stands for "Kriging assisted a priori evolutionary algorithm for constrained problems"

2. Feasible surrogate solutions that reflect the DM's preferences the best with low uncertainty.

Note that for the first type of surrogate solutions, we do not consider their feasibility because we are only interested in improving the accuracy of the surrogate models. In the second type, however, we choose feasible surrogate solutions with low uncertainty because we are looking for promising solutions to be shown to the DM at the end of the solution process. In what follows, we describe the selection strategy and the model management of KAEA-C in more detail.

3.1 Selection Strategy

For each subspace of the decomposed objective space, three situations can happen:

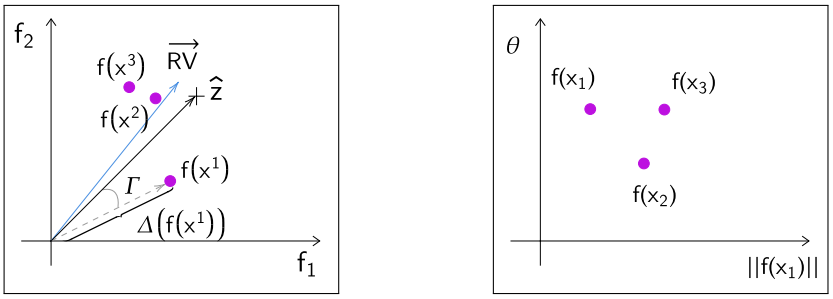
1. Only one feasible surrogate solution exists;
2. There is more than one feasible surrogate solution;
3. There is no feasible surrogate solution.

In the first situation, it is quite obvious that we have to select the only existing feasible surrogate solution for the next population. The values of the two fitness functions for each feasible surrogate solution we mentioned earlier are calculated in the second situation. These fitness functions can be expressed as:

1. $\delta(f(x), z^\star)$ which calculates the distance between $f(x)$ and the ideal point $z^\star$ of the current population;
2. $\Gamma(f(x), \hat{z})$ which calculates the angle between $f(x)$ and the reference point $\hat{z}$.

Then, instead of selecting a single surrogate solution for each subspace, we select the nondominated surrogate solutions based on these two fitness function values. The role of the first fitness function is to find surrogate solutions close to the Pareto front, while the second fitness function finds surrogate solutions that reflect the DM's preferences.

Figure 5 demonstrates a simple example of how the selection strategy works in KAEA-C in the second situation. Assume we have generated three feasible nondominated surrogate solutions (purple circles in Figure 5) that are assigned to the reference vector *RV* (see Figure 5a). Figure 5b shows how we can select a set of nondominated surrogate solutions based on the two fitness functions. In this example, we can observe that $f(x^1)$ is dominated by $f(x^2)$ because its distance to the ideal point and angle to the reference vector $\hat{z}$ is higher than those of $f(x^2)$. However, $f(x^1)$ and $f(x^2)$ are mutually nondominated and therefore are selected for the next population.



(a) (b)

FIGURE 5 An example of the selection strategy in KAEA-C. Here, the reference vector *RV* is denoted by a gray vector and the reference point $\hat{z}$ by a black vector. In addition, Γ is the angle between surrogate solution $f(x^1)$ and the reference point.

The last situation is when we have no feasible surrogate solution in a subspace. In this case, KAEA-C first calculates the constraint violation of each constraint. Then it counts the number of constraints that have been violated. Finally, KAEA-C ranks the surrogate solutions based on these two criteria, and the one with the lowest rank will get selected for the next population.

TABLE 1 An example of ranking surrogate solutions based on their constraint violation and the number of violated constraints.

	g_1	g_2	g_3	CV	R_{N_V}	R_{CV}	R_T
x^1	0	3	0	3	0	2	2
x^2	1	0.3	0	1.3	1	0	**1**
x^3	0.25	0	4	4.25	1	3	4
x^4	0.2	0.2	1	1.4	3	1	4

As an example, consider Table 1 with four decision vectors x^1, x^2, x^3, and x^4 for a subspace, which all are infeasible given three constraints $\{g_1, g_2, g_3\}$. In the table 1, the column CV represents the sum of constraint violations, the R_{CV} column shows the ranking of CV for each decision vector, R_{N_V} shows the ranking of the number of violated constraints for each decision variable vector, and R_T shows the total rank of each decision variable vector. Here, we can observe that x^2 has the lowest total rank so that it will be selected for the next population.

KAEA-C's selection strategy generates the next population by finding trade-offs between how well the surrogate solutions reflect the DM's preferences ($\Gamma(f(x), \hat{z})$) and how close they are to the ideal point ($\delta(f(x), z^\star)$). Therefore, during model management, we have more flexibility to select the most beneficial solution for our surrogate models.

3.2 Model Management

KAEA-C uses a novel model management technique to update the surrogate models. It is important to note that if we only consider improving the accuracy of the surrogate models, then the expensive solutions we generate do not necessarily reflect the DM's preferences. Therefore, besides the surrogate solutions with high uncertainty, we also need feasible surrogate solutions with the highest potential of reflecting the DM's preferences. In KAEA-C, we use the following ASF to identify the surrogate solutions that follow the DM's preferences:

$$\max_{i=1,\ldots,k}[w_i(f_i(x)-\hat{z}_i)]+\rho\sum_{i=1}^{k}w_i(f_i(x)-\hat{z}_i), \tag{7}$$

where $w \in R^k$ is a weight vector with positive fixed values, and $\rho\sum_{i=1}^{k} w_i(f_i(x) - \hat{z}_i)$ with $\rho > 0$ is an augmentation term to assure Pareto optimality [81].

We assume that the DM provides a priori the maximum number of expensive solutions N_U that she/he wants to see at the end of the solution process, and we have an archive A for storing expensive solutions. The following steps constitute the general framework of KAEA-C's model management for every iteration:

1. For improving accuracy:
   - 1.1. Calculate the uncertainty of every surrogate solution in population P_{it} at the end of each iteration;
   - 1.2. Select $\frac{N_U}{2}$ surrogate solutions that have the highest uncertainty;
   - 1.3. Evaluate the expensive functions with these solutions and store the values in the archive A.
2. For finding expensive solutions that reflect the DM's preferences:
   - 2.1. Calculate the ASF values of every feasible surrogate solution in population P_{it} at the end of each iteration;
   - 2.2. Select N_u surrogate solutions that have the lowest ASF values;
   - 2.3. Select $\frac{N_U}{2}$ surrogate solutions that have the lowest uncertainty;
   - 2.4. Evaluate the expensive functions with these surrogate solutions and store the values in the archive A.
3. Use the expensive solutions in A to update the Kriging models.

The first step in KAEA-C's model management is responsible for improving the accuracy of the Kriging models. Here the procedure is quite straightforward. As mentioned earlier, we select the surrogate solution with the highest uncertainty in the objectives and constraints. The second step is responsible for finding surrogate solutions that are likely to reflect the DM's preferences by using 7. In Article PI we assumed that both of these steps have equal importance, and that

is why we select $\frac{N_u}{2}$ solutions in each phase. However, one can change this ratio based on the MOP that is to be solved.

3.3 Discussion and Results

To compare the performance of KAEA-C, we needed to find other surrogate-assisted a priori evolutionary methods. Unfortunately, as mentioned earlier, no method in the literature has been designed to handle computationally expensive constraints and incorporate DM's preferences. Because of this, to enable comparison, we used different selection techniques and model management approaches in Article PI to assemble three new methods.

TABLE 2 Assembled methods by using different selection strategies and model management. The sign ✓ indicates the selected component for the corresponding method.

	Selection strategy		Model management	
	KAEA-C	RVEA [20]	KAEA-C	*BMOO* [47]
AM1		✓	✓	
AM2	✓			✓
AM3		✓		✓

We used the selection strategy of RVEA since KAEA-C selection strategy's main idea comes from APD. In addition, we used the model management of a Bayesian method called BMOO (stands for "Bayesian multi-objective optimization") since it uses expected hypervolume imptovement [44] and considers the feasibility of solutions in the acquisition function in a straightforward way. Table 2 shows the components of each assembled method. Note that there are a variety of selection strategies and model management, and our main purpose in Article PI was to show the importance of developing surrogate-assisted preference-based evolutionary methods for constrained MOPs.

There is a handful of constrained benchmark MOPs in the literature that have more than three objectives. We chose some of these problems as an example to emphasize the gap in the literature and show we need methods like KAEA-C that can handle computationally expensive constraints and DM's preferences simultaneously. In Article PI, We assess the performance of KAEA-C and the assembled methods on a constrained benchmark problem C3DTLZ4 [62] with three and seven objectives, the car-side impact problem [62], the water resource problem [94], and the multiple-disk clutch brake design problem [88]. 15 random reference points were generated in a way that they are dominated by the ideal point, and they dominate the nadir point. Due to the fact that EH-metric is a parameterless indicator, we used it to assess the performance of the assembled methods and KAEA-C. Then a paired color map was used to create the EH-metric results shown in Figure 6 as a heatmap. Here, we denote ranks 1 and 2 by dark blue and purple, respectively, rank 3 by orange, and rank 4 by yellow color.

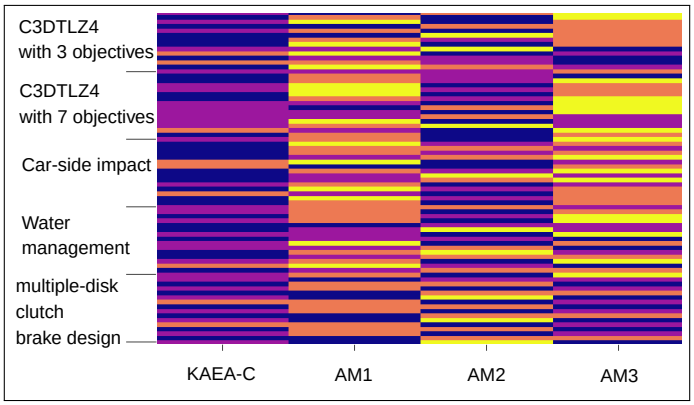


FIGURE 6 Heatmap for EH-metric value of expensive solutions obtained by KAEA-C, AM1, AM2, and AM3

We observe in Figure 6 that KAEA-C mostly achieved ranks 1 and 2, except for some exceptions where it achieved rank 3. Among the assemble methods, we can observe that AM2 had the best performance, and after KAEA-C had the most ranks 1 and 2. The selection strategy used in AM2 is the same as KAEA-C. However, the model management of BMOO is used for AM2. Even though the test problems were not complicated, the results show that the assembled methods cannot perform as well as KAEA-C. This means we need explicitly designed methods for MOPs with computationally expensive constraints and that can incorporate DM's preferences.

KAEA-C's selection strategy selects a set of surrogate solutions for each subspace (if possible) instead of only one solution, which decomposition-based evolutionary methods typically do. Obviously, it increases the computation time of the solution process for KAEA-C. However, since we deal with inexpensive functions during the solution process, we argue that it is worth spending some more time during the selection strategy, but in return, we have more options to choose from in the model management.

As for the other assembled methods, we can observe that AM2 had the best performance and AM3 had the worse performance among them. Moreover, AM1 was using KAEA-C's model management, and AM2 was using KAEA-C's selection strategy. Thus, a conclusion could be drawn that the selection strategy of KAEA-C has a more significant effect on its performance than the model management.

4 DESIRABLE PROPERTIES OF INDICATORS FOR EVOLUTIONARY METHODS

We must be able to assess the performance of different interactive methods to choose a proper one for a given MOP. As mentioned in Chapter 2, no indicator has been explicitly designed explicitly to assess the performance of interactive evolutionary methods. To develop such indicators, first, we need to understand the properties that an indicator should possess.

4.1 Desirable Properties

We defined the learning and decision phases of interactive solution processes in Chapter 1. They have shared and separate characteristics that should be considered when assessing the performance of methods. In Article PII, we introduced 13 desirable properties for interactive indicators. Nine of them are applicable to both the learning and the decision phases. We refer to them as *general properties (GPs)*. On the other hand, two of the desirable properties are explicitly designed for the learning phase. We refer to them as *learning phase properties* (LPs). The last two desirable properties are explicitly designed for the decision phase, and we refer to them as *decision phase properties* (DPs).

In what follows, we outline the desirable properties briefly. An indicator should be able to

GP1: assess the convergence of solutions in those regions of the approximated Pareto front that reflect the DM's preferences the best (local convergence);

GP2: assess the diversity of solutions in those regions of the approximated Pareto front that reflect the DM's preferences the best (local diversity);

GP3: assess the performance irrespective of the number of objective functions (scalability);

GP4: assess the performance without knowledge of the Pareto front;

GP5: assess the performance by incorporating preferences that are provided in different ways;

GP6: assess the performance in a computationally inexpensive manner;

GP7: assess the performance in a manner that is independent of the interactive methods being compared;

GP8: assess the performance without introducing parameters that have an unclear effect on the performance or are unintuitive to set;

GP9: assess the performance as a whole process and not as a series of independent a priori steps;

LP1: assess how much of the Pareto front has been studied (expedition);

LP2: assess how well/fast the method can adapt to new (even very different) preferences (responsiveness);

DP1: assess the capability of fine-tuning solutions inside the ROI and

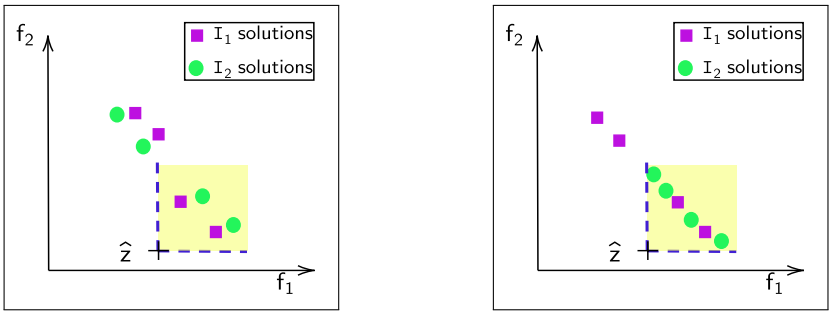
DP2: assess the decision phase by considering the amount of information shown to the DM at each interaction.

Before discussing the list above further, we must point out that the list of desirable properties is expandable, and one can add more or redefine them. Nevertheless, it was our aim to cover the most important properties in a concise manner. In what follows, we discuss them in more detail and justify why we find them important. In addition, we provide simple examples to demonstrate the importance. In the examples, we use illustrative solutions generated by two interactive evolutionary methods I_1 (visualized by purple squares) and I_2 (visualized by green circles). In addition, the reference point $\hat{z}$ is denoted by a black cross and the desired region by a blue dashed cone.

GP1 and GP2

As mentioned, in a posteriori methods, solutions must be as close as possible to the Pareto front (convergence) and represent an approximation of the whole Pareto front (diversity), while in preference-based methods, we are only interested in parts of the Pareto front that reflect the DM's preferences. Therefore, we use the concept of local convergence and local diversity [85]. According to [4], if a set of solutions is locally converged and diverse, the DM feels in control and can choose the most preferred solution more confidently.

Figure 7 shows an example to illustrate what we mean by local convergence and local diversity. In Figure 7a we can observe that methods I_1 and I_2 each have generated two solutions inside the desired region and two outside of it. For solutions outside of the desired region, solutions generated by I_2 dominate the ones generated by I_1. On the other hand, within the desired region, solutions



(a) Example for illustrating local convergence.

(b) Example for illustrating local diversity.

FIGURE 7 An example to illustrate local convergence and local diversity around the DM's reference point.

generated by I_1 dominate the ones generated by I_2, which means that I_1 has better local convergence than I_2.

Figure 7b illustrates a situation when I_2 has better local diversity than I_1. Here we can observe that I_1 is globally more diverse than I_2. However, within the desired region, this is not the case, and I_2 solutions have a better local diversity than I_1.

GP3

Whenever we are dealing with problems with less than four objectives, we can visualize solutions in the objective space and visually validate the assessment of an indicator for the generated solutions. However, as the number of objectives increases, this gets harder. Therefore, an indicator should be applicable to any problem with any number of objectives, and an analyst should be able to trust the indicator's performance assessment.

GP4

In most real-world MOPs, we do not know where the Pareto front lies. Therefore, an indicator designed for interactive evolutionary methods must not rely on such information.

GP5

As mentioned in Chapter 2, different interactive methods can incorporate different types of preferences. Therefore, appropriate indicators are needed. This does not mean that one indicator should be able to handle all different ways of providing preferences.

GP6

An indicator should not require a lot of computational resources. In different studies on evolutionary methods, we may need to use an indicator for perfor-

mance assessment every few generations. For example, in [15] an indicator is used at every generation to determine whether an evolutionary method has converged or not. Therefore, if the performance assessment is computationally expensive, then the indicator becomes inapplicable, e.g., for real-world problems.

GP7

In some indicators designed for a priori methods like [9, 75, 85], two (or more) sets of solutions are compared against one another. This means that the performance assessment of these indicators does not have an independent meaning. In other words, if we add a new set of solutions to the performance assessment, we need to start the whole performance assessment from the beginning. We find such behavior undesirable. Therefore, an indicator should be able to assess the performance of an interactive evolutionary method so that the assessment is not affected by the other methods being compared.

GP8

Evolutionary methods have several parameters that need to be specified, such as the number of generations, iterations, and so on. Moreover, if we use surrogate models, the number of parameters increases even more. Therefore, an indicator should not require setting many parameters. But if it needs parameters, we should be able to understand their effect on the performance assessment.

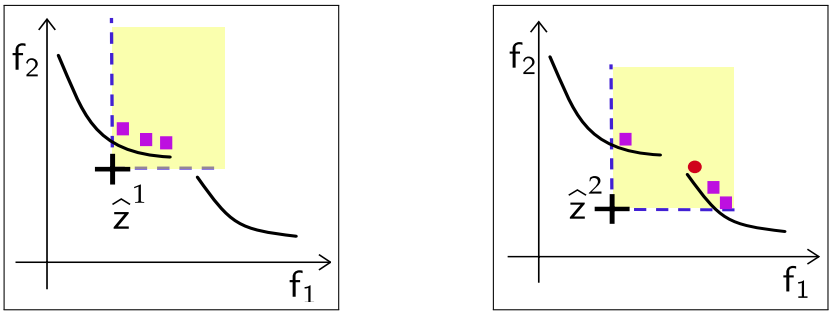
GP9

An indicator should consider the interactive solution process as a whole and assess the performance of this complete process taking, e.g., into account the learning that takes place. We can design different indicators for the learning and the decision phases since they are different by nature, but each of these phases should be seen as a whole process and not as a series of independent steps.

LP1

During the learning phase, it is important that the DM feels comfortable about the problem and knows which preferences are achievable, and feels confident that he has found the ROI. Therefore, it is desirable that an interactive indicator can assess how much the DM has studied of the interesting regions on the Pareto front (We refer to it as expedition).

Note that LP1 and GP2 (local diversity) are different. The desirable property GP2 is related to every preference the DM provides. However, LP1 is about the whole learning phase and whether the Pareto front's interesting regions have been studied enough. For example, consider Figure 8. Assume that after the first interaction (Figure 8a), the DM is interested in improving f_1. Therefore, for the second interaction (Figure 8b), she/he provides the reference point $\hat{z}_2$. Even though all the generated solutions in the second interaction are within the de-



(a) Example for illustrating local convergence.

(b) Example for illustrating local diversity.

FIGURE 8 An example when the desired region has not been studied well enough by an interactive method. The solutions generated by an interactive method are denoted by purple squares. The solution that was not found by the method is denoted by a red circle. The Pareto front is represented by black lines.

sired region and have good local diversity, the method could not find the solution shown as a red circle. Here, if the DM decides to move on with the solution process, she/he will not know such a solution existed. An indicator should be able to assess how well the Pareto front has been covered.

LP2

In the learning phase, the DM usually wants to learn about the feasibility of her/his preferences and learn more about different trade-offs between the objectives. In addition, between interactions, the preferences may be drastically different. We should not keep the DM waiting for a long time for the interactive evolutionary method to generate new solutions. Therefore, it is essential that the method can respond to these changes by representing new parts of the Pareto front and generate these solutions reflecting the new preferences in a reasonable time.

DP1

As mentioned in Chapter 2, the DM fine-tunes the solutions in the decision phase and provides preferences to study the ROI closer until she/he is satisfied with one of the generated solutions. In this phase, an indicator should be able to identify the concordance between preferences and measure how well the interactive method allows the DM to fine-tune solutions. However, the definition of concordance between preferences is quite vague. For example, in Article PII, we defined the following term for concordance between a set of reference points:

Term 9 *Reference points inside the ROI have concordance with each other, and the ones outside of the ROI have no concordance.*

We suggest that an interactive indicator should identify the concordance between preferences at each interaction during the decision phase. Then, the indicator

should use this information to assess how well solutions are fine-tuned with regard to the preferences.

DP2

In interactive evolutionary methods, we show a limited number of solutions to the DM. The idea is not to put too much cognitive load set on her/him. The number of solutions to be shown should be set by the DM based on her/him cognitive capacity. During the learning phase, when the DM is learning about the shape of the Pareto front, the number of solutions presented to her/him can vary, considering the acceptable cognitive load. In contrast, in the decision phase, the interactive evolutionary method should respect the DM's wishes and provide as many solutions as she/he asked for within the ROI (if possible). Hence, an indicator designed for interactive evolutionary methods must consider the number of solutions shown to the DM in its performance assessment of the decision phase.

4.2 Numerical Results

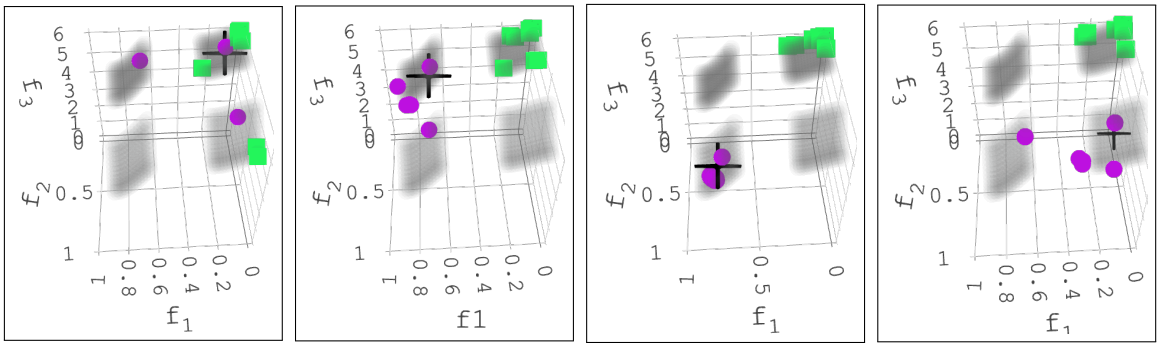
As mentioned earlier, among the existing indicators, the ones designed for a priori evolutionary methods can be used for assessing interactive evolutionary methods if each interaction is treated as a separate step and the learning between interactions is ignored. In Article PII, we utilized some of these indicators to assess the performance of two interactive evolutionary methods, interactive RVEA [56] (iRVEA) and an interactive variant of NSGA-III (called iNSGA-III) [5]. Our goal here is not to have a deep comparison of these two methods but only to show that applying a priori indicators like this is not desirable. In this section, we present the highlights of the numerical results we obtained in Article PII.

4.2.1 Learning Phase

As an example of the learning phase, we have chosen a benchmark problem DTLZ7 [38] with three objectives and 12 decision variables. The reason we chose DTLZ7 is because of its unique Pareto front shape. It has four disconnected regions (shown in grey in Figure 9)

We manually provided four reference points for each region of the Pareto front (one reference point per region). These reference points are as follows:

- Interaction 1: [0.11, 0.10, 5.4],
- Interaction 2: [0.70, 0.14, 4.5],
- Interaction 3: [0.76, 0.76, 3.5],
- Interaction 4: [0.14, 0.70, 4.5].



(a) Interaction 1. (b) Interaction 2. (c) Interaction 3. (d) Interaction 4.

FIGURE 9 DTLZ7 results in the learning phase. Each reference point (the black plus sign) has been generated randomly in each region of the Pareto front (the grey area).

As we can see in Figure 9, for the first interaction, both methods have generated some solutions around the first reference point (Figure 9a). However, as we change the reference point to other regions, we can see that iRVEA can respond to these changes (see Figures 9b, 9c, and 9d), but iNSGA-III was stuck in the initial region. Importantly, if the analyst was using iNSGA-III, she/he would never know that these other regions exist. Therefore, for LP1 and LP2, iRVEA performs better than iNSGA-III.

TABLE 3 Learning phase results of iRVEA and iNSGA-III for DTLZ7 with three objectives. The symbol ↑ next to the indicator name indicates that the higher values of the indicator are better, and the symbol ↓ indicates that lower values are better.

	R-HV ↑		EH-metric ↑		PMOD ↓		PMDA ↓		UPCF-HV ↑	
i	iRVEA	iNSGA-III	iRVEA	iNSGA-III	iRVEA	iNSGA-III	iRVEA	iNSGA-III	iRVEA	iNSGA-III
1	5.927	**6.669**	0.276	**0.561**	**6.543**	7.888	**4.632**	5.050	3.437	**4.310**
2	**5.859**	5.401	**0.569**	0.419	**5.713**	6.626	**4.248**	5.092	**3.985**	2.546
3	**7.301**	6.489	**0.688**	0.121	**5.418**	5.833	**3.226**	4.940	**4.015**	2.135
4	**5.909**	2.930	**0.482**	0.196	**5.690**	6.392	**4.245**	4.893	**3.874**	2.497

In Table 3, we show the performance assessment of a priori indicators for each interaction individually. Based on the indicator values for each interaction, it shows that except for the first interaction, iRVEA has a better performance than iNSGA-III. However, based on these values, we cannot conclude that iNSGA-III solutions are stuck in one region. For example, if an analyst uses PMOD or PMDA to assess the performance of methods, she/he may not think that iNSGA-III was much worse than iRVEA.

4.2.2 Decision Phase

Here we show the decision phase of DTLZ3 with five objectives and 13 decision variables. We used an artificial decision maker (ADM) proposed in [5] to identify the ROI and provide the preferences as a reference point in the decision phase. The reference points for four interactions are as follows:

- Interaction 1: [0.000, 0.000, 0.000,0.000, 3.072],

TABLE 4 Decision phase results of iRVEA and iNSGA-III for DTLZ3 with five objectives. The symbol ↑ next to the indicator name indicates that higher values of the indicator are better and the symbol ↓ indicates that lower values are better.

	R-HV ↑		EH-metric ↑		PMOD ↓		PMDA ↓		UPCF-HV ↑	
i	iRVEA	iNSGA-III	iRVEA	iNSGA-III	iRVEA	iNSGA-III	iRVEA	iNSGA-III	iRVEA	iNSGA-III
1	14.145	**32.212**	0.475	**0.590**	**5.835**	8.533	**2.888**	3.329	**0.185**	0.195
2	24.263	**32.253**	0.619	**0.643**	**6.405**	9.265	**1.430**	3.329	**0.185**	0.262
3	26.126	**31.956**	**0.766**	0.744	13.813	**12.160**	**0.704**	3.329	**0.311**	0.347
4	28.699	**31.957**	**0.768**	0.741	14.950	**12.160**	**0.687**	3.329	**0.412**	0.458

- Interaction 2: [0.000, 0.000, 0.000, 0.000, 1.951],
- Interaction 3: [0.000, 0.000, 0.000, 0.000, 1.010],
- Interaction 4: [0.000, 0.000, 0.000, 0.000, 1.010].

We ran the interactive solution process (with the same reference points) 10 times with iRVEA and iNSGA-III. Then we calculated the average performance assessment of each of the indicators mentioned in Table 3 and reported the results in Table 4.

The results are unanimous for some of the indicators in Table 4. For example, based on PMDA, iRVEA's performance was better for all four interactions, and R-metric declared iNSGA-III as the winner. However, for EH-metric and PMOD, this is not the case. The EH-metric results indicate that for the first two interactions, iNSGA-III was the winner, and for the last two interactions, iRVEA was the winner. In the literature, when a priori indicators are used to compare interactive methods, all interactions in the decision phase have the same importance for performance assessment [3, 74]. However, in the description of the desirable property DP1, we mentioned that this should not be the case. In fact, we should measure the concordance between the reference points and corresponding solutions and consider it in the performance assessment of the decision phase.

4.2.3 Guidelines for Designing a New Interactive Indicator

As shown in this chapter, there are many different desirable properties that an interactive indicator should possess. In addition, we showed why it is not enough to use a priori indicators to assess the performance of interactive evolutionary methods. Moreover, because of the complex nature of interactive evolutionary methods, there are many desirable properties for an interactive indicator compared to a posteriori or a priori ones. Therefore, when designing a new indicator, one should realize the followings:

1. The desired region and ROI should be defined clearly;
2. One interactive indicator does not need to have all of the desirable properties;
3. Every solution that we show to the DM should play a role in the performance assessment, even if it is outside of the desired region.

Typically, the definition of desired region and ROI is quite vague and there is no unified way to define them. Moreover, the way the preference-based evolutionary methods and indicators define the desired region is usually different and this can lead to misleading results in the performance assessment. For example, a preference-based evolutionary method may use the distance between solutions and a reference point to identify the desired region, but the indicator utilized for assessing the performance, uses some scalarization function to determine if solutions reflect the reference point. Therefore, the desired region that is identified by the indicator may not be the same as the one identified with the preference-based evolutionary method. Identifying ROI is even more challenging than the desired region since we have to understand what parts of the Pareto front are interesting to the DM and how he is fine-tuning the solutions.

An example of the second item of the list above is when we are dealing with a computationally expensive problem. Developing an indicator that requires a lot of computational resources is probably not a good idea in this case since we should not introduce a lot of additional computational overhead. Therefore, the new indicator used to assess the performance of methods for solving this problem should possess at least property GP6.

Another example is when the DM asks for a certain number of solutions to be analyzed during the decision phase. As we have discussed in Article PII, if the interactive method generates fewer solutions than the DM has asked for (assuming the true Pareto front contains as many solutions as the DM wants), the solution process can be delayed, and if the number of solutions is more than what the DM has asked, it may put too much cognitive load set on her/him. In this case, the indicator must consider the number of solutions an interactive method shows to the DM.

As for the last item on the list above, sometimes some of the solutions shown to the DM are outside of the desired region, and, therefore, she/he may not be interested in them. We should realize that these solutions may distract the DM or make her/him lose confidence in the interactive evolutionary method she/he is using. Therefore, we should consider these solutions in the performance assessment. In other words, they should have a negative effect on the performance assessment.

5 PREFERENCE-BASED HYPERVOLUME INDICATOR FOR INTERACTIVE EVOLUTIONARY MULTIOBJECTIVE OPTIMIZATION METHODS

Many aspects may explain the absence of interactive indicators in the literature. Interactive methods have many components that make it challenging to assess their performance. For instance, the DM plays an important role, and the learning and decision phases have different characteristics [4]. Additionally, preferences get updated iteratively, and an indicator needs to adjust accordingly. In Chapter 4, we discussed the desirable properties of interactive indicators considered in Article PII. In this chapter, we introduce the interactive indicator PHI proposed in Article PIII that can incorporate DM's preferences in the form of a reference point. Moreover, in Article PIII, we discuss how to utilize PHI for assessing the learning and decision phases by capturing their characteristics that we mentioned in Chapter 4. In Article PIII, we used a bi-objective problem to show how to utilize PHI. In this chapter, we provide a case study with nine objectives to show how we can use PHI in practice and demonstrate why a priori indicators are unsuitable for assessing interactive evolutionary methods.

5.1 PHI Description

In the performance assessment of an interactive evolutionary method, the main idea of PHI is to reward the solutions that reflect the reference point of the DM and punish if the solutions do not reflect preferences. To know when to reward, we first need to define a desired region and identify solutions that reflect the DM's reference point. For a reference point $\hat{z}$ and a dystopian point z^{dy}, we defined the desired region in Article PIII as: "a region of the objective space enclosed by a hyperrectangle with corners $\hat{z}$ and z^{dy}."

In other words, the desired region is the area of the objective space $\hat{z}$ dominates and is limited by z^{dy}. In Chapter 2 we introduced the hypervolume indica-

tor (see (6)). The size of the desired region can be measured as $\mathrm{HV}(\hat{z}, z^{dy})$, which we call a hypervolume measure of $\hat{z}$.

Next, we identify solutions that reflect the DM's reference point. We divide a set of nondominated solutions P generated by an interactive evolutionary method into three subsets:

1. The subset $P^{\succ}$ which contains solutions that dominate $\hat{z}$;
2. The subset $P^{\prec}$ which contains solutions that are dominated by $\hat{z}$;
3. The subset $P^{=}$ which contains solutions that are not dominated by $\hat{z}$ nor dominate it.

We say that solutions in $P^{\prec}$ and $P^{\succ}$ reflect the DM's reference point, and they have a positive contribution to calculating the PHI value. In contrast, solutions that belong to $P^{=}$ are not of immediate interest to the DM and have both positive and negative contributions to calculating the PHI value based on how far they are from the desired region.

To continue to the description of PHI, we need to define the following terms:

Term 10 *$v^{\prec}$: a part of the hypervolume measure* $\mathrm{HV}(P, z^{dy})$ *inside the desired region (the light green areas in Figure 10).*

Term 11 *$v^{\succ}$: a part of the hypervolume measure* $\mathrm{HV}(P^{\succ}, z^{dy})$ *that is outside the desired region (the dark green areas in Figure 10b).*

Term 12 *v^{-} : a part of the hypervolume measure* $\mathrm{HV}(P^{=}, z^{dy})$ *that is outside the desired region (the orange areas in Figure 10).*

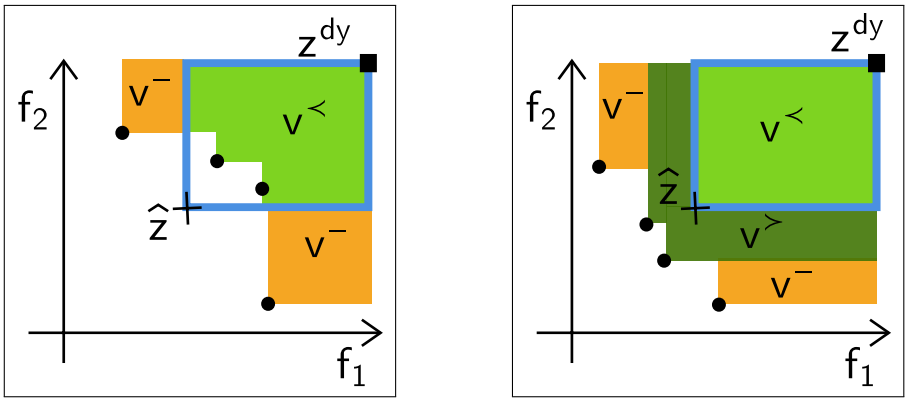
Term 13 ***positive and negative contributions:*** *for a set of solutions P we refer to the volume of $v^{\prec} + v^{\succ}$ as the positive contribution of P. In addition, we refer to volume v^{-} as the negative contribution of P.*

In short, the PHI value for the solution set P that has been generated by an interactive evolutionary method as:

$$PHI(P, \hat{z}, z^{dy}) := \frac{v^{\prec}}{\mathrm{HV}(\hat{z}, z^{dy})} + \frac{v^{\succ}}{\mathrm{HV}(P, z^{dy})}, \tag{8}$$

where $\mathrm{HV}(\hat{z}, z^{dy})$ and $\mathrm{HV}(P, z^{dy})$ are used to normalize the values of $v^{\prec}$ and $v^{\succ}$, respectively. Algorithm 1 shows the main steps of assessing the performance of an interactive method that has generated the set of solutions P.

As we mentioned earlier, the first step is to create the subsets $P^{\succ}$, $P^{\prec}$, and $P^{=}$. We know that solutions in $P^{\succ}$ are better than what the DM asked for, and we are interested in calculating how many good solutions are beyond the given reference point. The following subset is $p^{\prec}$, which contains solutions that $\hat{z}$ dominates. Here, our goal is to reward the method's performance as the solutions in $P^{\prec}$ get closer to $\hat{z}$. In other words, if $p^{\prec} \neq \varnothing$, we would like the solutions to be as



(a) (b)

FIGURE 10 A simple illustration of how to calculate PHI for a set of solutions (black circles): (a) No solution dominates the reference point; (b) The reference point is dominated by at least one solution. The dystopian point is denoted by a black square, and the reference point by a plus sign. A dashed purple box illustrates the desired region. The light and dark green areas represent the positive contribution of solutions, and the orange areas the negative contribution of solutions.

close as possible to the Pareto front (local convergence) and be locally diverse in $p^{\prec}$. Finally, the subset $p^{=}$ contains solutions that do not dominate $\hat{z}$ nor are dominated by it. Because we define the desired region, we know solutions in $p^{=}$ are outside of it. Therefore, they are not of immediate interest to the DM. However, if they are very close to the desired region, they may become interesting, and the DM may choose one of these solutions. Consequently, their role in assessing the performance of the method that generated P is based on how far they are from the desired region.

Next, we calculate the positive contribution of solutions in P to the desired region. Here we have two possibilities: $P^{\succ} \neq \varnothing$ or $P^{\succ} = \varnothing$. In the first case, because P contains nondominated solutions, $P^{\prec} = \varnothing$. In other words, it is impossible to have solutions in both subsets $p^{\succ}$ and $p^{\prec}$. Because the reference point is dominated by at least one solution ($P^{\succ} \neq \varnothing$), we can say $v^{\prec} = \mathrm{HV}(\hat{z}, z^{dy})$ (see Figure 10b). Next, we measure $v^{\succ}$ (step 5 of Algorithm 1). As for the second case, because no solution dominates the reference point ($v^{\succ} = 0$), we can calculate $v^{\prec}$ as the positive contribution of P (step 7).

Now that we calculated the solutions' positive ($v^{\succ} + v^{\prec}$) and negative (v^{-}) contributions, we can calculate the final value of the indicator PHI(step 9). In (8), we normalize the values of $v^{\prec}$ and $v^{\succ}$ because by doing so, we can provide additional information. If at least one solution in P dominates the reference point, the final indicator value (in (8)) can be written as:

$$PHI(P, \hat{z}, z^{dy}) = 1 + \frac{v^{\succ}}{\mathrm{HV}(P, z^{dy})}, \tag{11}$$

because $v^{\prec} = \mathrm{HV}(\hat{z}, d^{dy})$ and since $\frac{v^{\succ}}{\mathrm{HV}(P, z^{dy})} \leq 1$, $PHI(P, \hat{z}, z^{dy})$ always belongs

Algorithm 1 Computation of $PHI(P, \hat{z}, z^{dy})$

Input: A set of solutions P, reference point$\hat{z}$, and dystopian point z^{dy}
Output: Performance assessment of an interactive evolutionary method

1: Divide P into
   a: $P^{\succ}$: subset of P where solutions dominate $\hat{z}$.
   b: $P^{\prec}$: subset of P where solutions are dominated by $\hat{z}$.
   c: $P^{=}$: subset of P where solutions neither dominate $\hat{z}$ nor are dominated by $\hat{z}$.
2: Calculate v^{-} (orange areas in Figures 10a and 10b):

$$v^{-} := HV(P \cup \{\hat{z}\}, z^{dy}) - HV(P^{\succ} \cup \{\hat{z}\}, z^{dy}). \tag{9}$$

3: **if** $P^{\succ} \neq \varnothing$ **then**
4: Calculate $v^{\prec}$ (light green area in Figure 10b):

$$v^{\prec} := HV(\hat{z}, z^{dy}). \tag{10}$$

5: By using the value of $v^{\prec}$ (dark green area in Figure 10b) calculate $v^{\succ}$ as:

$$v^{\succ} := HV(P^{\succ}, z^{dy}) - v^{\prec}.$$

6: **else**
7: Use (9) to calculate $v^{\prec}$(light green area in Figure 10a) as:

$$v^{\prec} := HV(P, z^{dy}) - v^{-}.$$

8: Put $v^{\succ} := 0$. // Since $P^{\succ} := 0$.
9: **end if**
10: Calculate final value of PHI by using (8).

to the interval $(1, 2]$.

Moreover, if no solution dominates the reference point, the final indicator value can be expressed as:

$$PHI(P, \hat{z}, z^{dy}) = \frac{v^{\prec}}{\mathrm{HV}(P, d^{dy})}. \tag{12}$$

Here, because $\frac{v^{\prec}}{\mathrm{HV}(P, d^{dy})} \leq 1$, the final value of PHI always belongs to the interval $(0, 1]$. Therefore, in problems with a high number of objectives, where it is hard to assess how well the solutions reflect a given reference point, we can determine if some of the solutions dominate the reference point or not by only looking at the PHI values.

5.2 Assessing the Learning Phase

As we mentioned in Chapter 4, the list of desirable properties proposed in Article PII can be improved. In Article PIII, we decided to redefine responsiveness which is the second desirable property regarding the learning phase (LP2) as: " assess the ability to adapt to new (even very different) preferences (responsiveness) and maintain the best-so-for solutions within interactions (stability)."

To assess the performance of an interactive evolutionary method during the learning phase, we take the following steps:

1. In each generation t of the interactive evolutionary method, select the non-dominated solutions P_t.

2. Calculate $\text{PHI}_t = \text{PHI}(P_t, \hat{z}, z^{dy})$.

3. Calculate the indicator value for the learning phase as:

$$\text{RS} = \sum_{0}^{t_m} PHI_t, \tag{13}$$

where t_m is the number of generations in the learning phase, and RS is the responsiveness and stability of an interactive evolutionary method during the learning phase. In other words, if we plot each generation's PHI value for an interactive evolutionary method, the area underneath the line would be the RS value.

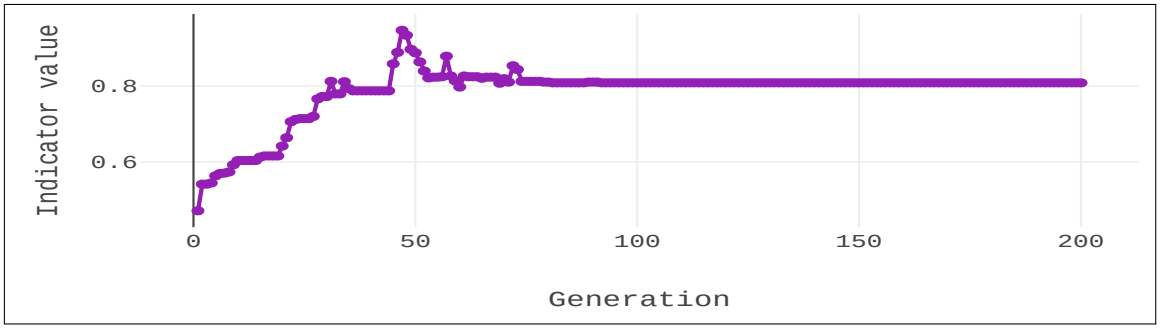


FIGURE 11 Tracking the values of an indicator throughout generations of an interaction. The learning phase can be calculated as the area underneath the purple line.

Figure 11 demonstrates an example of one interaction of an interactive evolutionary method. Here, we can see that around generation 50, the method reaches its top indicator value. However, the performance drops slightly and then converges to the PHI value of 0.8. Calculating the area underneath the purple line can capture the method's responsiveness and steadiness.

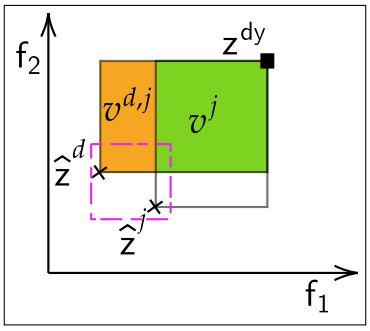


FIGURE 12 Similarities of two desired regions corresponding to reference points $\hat{z}^j$ and $\hat{z}^d$ in a bi-objective example. A black square denotes the dystopian point. The ROI is denoted as a purple dashed box.

5.3 Assessing the Decision Phase

At the beginning of the decision phase, the DM already has the ROI and is looking for the most preferred solution within it. We assume the DM tries to provide her/his reference point within the identified ROI. However, sometimes due to human error or other reasons, the reference points may not have concordance with each other, and some of them are outside the ROI. To assess the performance of an interactive evolutionary method for the decision phase, we show in this section how to measure the concordance between the reference points that the DM has provided. Then, we demonstrate how to use these measurements to assess the method's performance for this phase.

Assume we have d interactions during the decision phase, and $\hat{z}^d$ is the reference point provided in the last interaction. In this thesis, we assume that the DM chooses the most preferred solution from the solution set corresponding to the last reference point $\hat{z}^d$. Therefore, the d-th interaction should have the most importance. We measure the concordance between $\hat{z}$ and the rest of the reference points provided in the decision phase. We assign a coefficient λ to each interaction that determines the concordance (similarity) between $\hat{z}^j$ and $\hat{z}^d$ where $j = 1, \ldots, d$. The higher λ_j is $\hat{z}^j$ and has more similarity with $\hat{z}^d$. In Algorithm 2, we show the main steps of calculating the coefficients λ_j. For each reference point $\hat{z}^j$ we create a set $\zeta = \{\hat{z}^j, \hat{z}^d\}$. Then, we calculate the individual hypervolume contribution of each reference point to $\mathrm{HV}(\zeta_j, z^{dy})$. Next, we calculate the coefficients in step 6. Note that $\lambda_d = 1$ since v^d represents $\mathrm{HV}(\hat{z}^d, z^{dy})$. Once all the coefficients are calculated, we can express the indicator value of the decision phase as:

$$\mathrm{FD} = \frac{\sum_{j=1}^{d} \lambda_j PHI(P_j, \hat{z}, z^{dy})}{d}, \tag{16}$$

where FD measures the ability of the method to fine-tune solutions.

We assume that we utilize PHI to assess the performance of the learning and decision phases after the interactive solution process. The reason for such an assumption is that usually, after the interactive solution process, we have a decent

Algorithm 2 Calculating coefficients λ_j

Input: Set of reference points used in the decision phase $\{\hat{z}^1, \ldots, \hat{z}^d\}$
Output: set of coefficients $\{\lambda_1, \ldots, \lambda_d\}$ determining the concordance between reference points.

1: set counter $j = 1$
2: **while** j<=d **do**
3: $\zeta_j := \{\hat{z}^j, \hat{z}^d\}$
4: Calculate the individual hypervolume contribution of $\hat{z}^j$ to $HV(\zeta, z^{dy})$ denoted as $v^{d,j}$ (orange area in Figure 12) by the following equation:

$$v^{d,j} := HV(\zeta_j, z^{dy}) - HV(\hat{z}^j, z^{dy}). \tag{14}$$

5: Calculate the shared hypervolume contribution of $\hat{z}^j$ and $\hat{z}^d$ to $HV(\zeta, z^{dy})$ denoted as v^j (green area in Figure 12) by the following equation:

$$v^j := HV(\hat{z}^d, z^{dy}) - v^{d,j}.$$

6: Calculate λ_j as:

$$\lambda_j := \frac{v^j}{HV(\hat{z}^d, z^{dy})}. \tag{15}$$

7: Set $j = j + 1$
8: **end while**

idea of the Pareto front, and it helps us in estimating the nadir point, which we need for setting the dystopian point. However, the analyst can use PHI during the interactive solution process to guide the DM. For instance, the analyst can suggest using different methods for the learning and decision phase by analyzing the results during the interactive solution process. Another example is that the analyst can help the DM to understand if the provided preferences are reachable or not.

Furthermore, sometimes there is no clear line between the learning and decision phases. In this case, we suggest assessing the performance of interactive methods as if all of the interactions belong to the learning phase since we can visualize the results, which may be easier to digest for the analyst. It is also worth mentioning that even though hypervolume is typically considered to be computationally expensive when the number of objectives increases, there have been some efforts in developing fast algorithms for hypervolume calculation [64], which one should consider when using a hypervolume-based indicator such as PHI.

5.4 Results

In Article PIII, we showed that existing a priori indicators are not as suitable as PHI to assess the performance of interactive evolutionary methods. (This is understandable since they have been developed for a different purpose.) We also demonstrated how we could utilize PHI to assess the performance of the learning and decision phases by capturing some of their desirable properties. Moreover, we demonstrated that PHI could provide additional information to the analyst to assist him/her in choosing the most appropriate method. For instance, the analyst can analyze the positive and negative contributions of the solutions set that a method generates to see how converged/diverse the solutions are within the desired region or how far away they are from it.

In this section, we provide an example and show how PHI can be used to assess the performance of interactive evolutionary methods. We deal with MOPs that have many objectives (more than three objectives). In Article PIII, we used PHI to assess the performance of iRVEA and another method called *interactive optimization using preference incorporated space* (IOPIS) [100]. The test problem was a bi-objective engineering problem known as "RE21" [105]. In Article PIII, we showed that IOPIS could follow the reference point better, while iRVEA, on the other hand, always generated solutions outside of the desired region during the learning and decision phases. In this case, we can visualize all solutions and the desired region with two objectives. However, visualization becomes an issue for problems with more than three objectives. We chose a problem known as "RE91" [105], which has nine conflicting objectives and seven decision variables to be solved by IOPIS and iRVEA. For more details of the problem and mathematical formulations, see [33].

The author acted as both an analyst and a DM in the following experiment. We used 200 generations per interaction. Note that our goal is not to find the right number of generations before the methods converge toward the Pareto front but only to assess their performance in the same environment. Note that we show all the solutions to the DM during the interactive solution process.

The DM stopped the interactive solution process after eight interactions. The first three interactions constituted a learning phase, and then the DM started a decision phase and took five more interactions. In this section, we are more interested in how the analyst can use PHI to compare interactive methods. Therefore, we do not include detailed interactive solution processes. However, to be able to reproduce the results, we have gathered the reference points that the DM provided for the interactive methods in Table 5.

5.4.1 Performance Assessment of the Learning Phase

As mentioned in Section 5.2, we calculate the PHI values for every generation during the learning phase and then use (13) to get the final indicator value. Figure 13 illustrates the PHI values we tracked for every generation of IOPIS (the

TABLE 5 The reference points used during the interactive solution process of RE91.

$\hat{z}^1$	33.30	1.50	40.06	0.80	1.90	1.80	1.78	1.84	1.89
$\hat{z}^2$	27.16	0.92	22.24	0.80	1.33	1.41	1.23	0.97	0.92
$\hat{z}^3$	21.54	1.52	10.31	0.64	1.73	1.01	1.21	1.32	1.13
$\hat{z}^4$	18.50	0.85	0.15	0.75	1.35	1.15	1.15	1.05	1.05
$\hat{z}^5$	18.32	0.81	0.22	0.54	1.52	1.10	0.87	0.98	1.10
$\hat{z}^6$	16.25	0.83	0.14	0.52	1.13	0.88	0.86	1.06	0.82
$\hat{z}^7$	17.19	0.91	0.17	0.72	1.11	0.91	0.94	1.16	0.93
$\hat{z}^8$	17.15	0.73	0.12	0.62	1.23	0.95	0.93	0.86	0.93

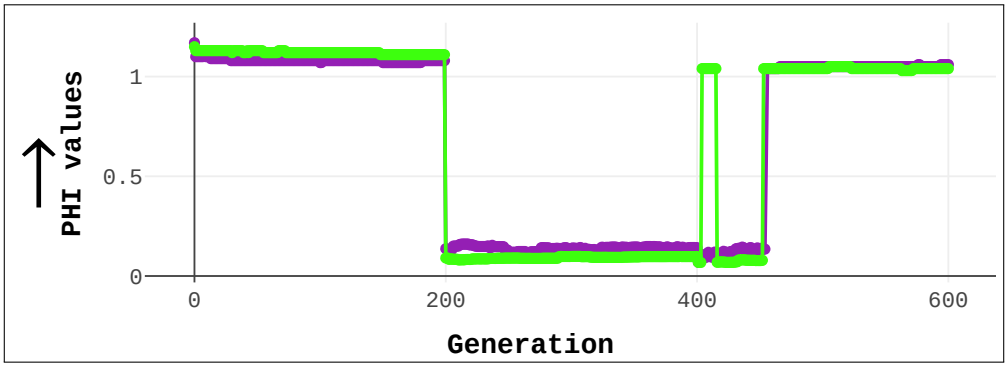


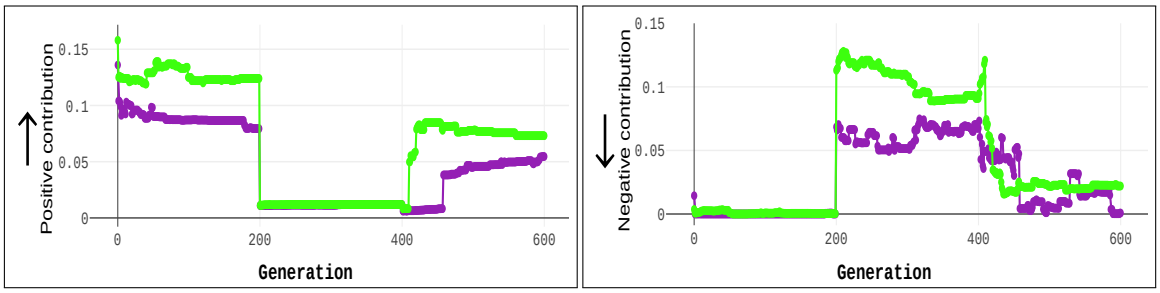
FIGURE 13 PHI values in the learning phase in each generation. The purple and green lines correspond to IOPIS and iRVEA, respectively.

purple line) and iRVEA (the green line). We can observe that for the first interaction, iRVEA had a slightly better performance than IOPIS. In addition, since the value of PHI is above 1, we know that at least one solution dominated the provided reference point (for both methods). The PHI values of IOPIS for the second interaction were slightly better than that of iRVEA. Here, we can observe that both methods' PHI values were close to 0. This means that the DM's reference point ($\hat{z}^2$) was not reachable, and none of the methods could find good solutions to reflect it very well. Finally, for the third interaction, both methods almost had the same PHI values for the first few generations, where iRVEA performed better than IOPIS. For this interaction, we can see that the third reference point $\hat{z}^3$ was dominated by at least one solution. However, for both methods, the values of PHI are closer to 1 than in the first interaction. This means that $\hat{z}^3$ is almost on the Pareto front, and if this value is shown to the analyst, he gets some idea of where some parts of the Pareto front are. The overall performance assessments of the learning phase (RS values) obtained by IOPIS and iRVEA are shown in Table 6. iRVEA had a slightly better performance than IOPIS here. However, the difference was too small to make a firm judgment.

Thanks to PHI, we can dig deeper into PHI results by analyzing solutions' positive and negative contributions for each generation. In Figure 14 we can observe these values. Here, In Figure 14a, we illustrated that the positive contribution of solutions of iRVEA is better for the first interaction. The positive contribution in the second interaction is almost the same for IOPIS and iRVEA. In the

TABLE 6 Performance assessment of IOPIS and iRVEA for the learning phase.

	RS
iRVEA	4.10E2
IOPIS	4.03E2



(a) (b)

FIGURE 14 The values of solutions' (a) positive and (b) negative contribution at each generation. The purple and green lines represent these values for IOPIS and iRVEA, respectively.

third interaction, solutions generated by iRVEA have a better positive contribution than IOPIS.

For the negative contribution (Figure 14b), we can observe that solutions generated by IOPIS have almost the same values as iRVEA. However, for the last two interactions, solutions generated by IOPIS have a better negative contribution than the ones of iRVEA. This means the solutions IOPIS generates reflect the DM's reference point better than those of iRVEA. This is a critical piece of information for the analyst. Even though the overall performance (RS value) of iRVEA is slightly better than IOPIS, he thinks IOPIS has a better performance than iRVEA for the learning phase since it can find solutions that reflect the DM's preferences better than those of IOPIS.

5.4.2 Performance Assessment of the Decision Phase

To assess the decision phase, first, we use Algorithm (2) to measure the coefficients $\{\lambda_4, \ldots, \lambda_8\}$, which represent the similarities between the reference points of interactions four to eight (see Table 7). Here we can observe that the fourth reference point $\hat{z}^4$ has the lowest similarity ($\lambda_4 = 0.20$) to the last reference point $\hat{z}^8$ and the sixth reference point $\hat{z}^6$ has the highest similarity ($\lambda_6 = 0.76$) since the DM chooses the final solution from the 8-th interaction, $\lambda_8 = 1$.

Next, we assess the performance of the two interactive methods considered for each interaction in the decision phase (see Table 7). Now that we have the values of $\{\lambda_4, \ldots, \lambda_8\}$, and PHI value of each interaction of the methods in the decision phase, we can assess the performance of IOPIS and iRVEA for the decision phase by using (16). The indicator value for the decision phase with IOPIS is 0.26 and with iRVEA 0.34, which means that overall, iRVEA could fine-tune solutions better than IOPIS.

TABLE 7 Coefficients $\{\lambda_4, \ldots, \lambda_8\}$, PHI, positive contribution, and negative contribution values of solutions during the decision phase for problem RE91. The best values are in bold font. The sign $\downarrow$ means lower values are better, and $\uparrow$ means the higher the values are, the better. The better values are in boldface.

Interaction	λ	**PHI** $\uparrow$		**Positive contribution** $\uparrow$		**Negative contribution** $\downarrow$	
		IOPIS	iRVEA	IOPIS	iRVEA	IOPIS	iRVEA
4	0.20	1.03	1.03	0.04	**0.05**	0.04	**0.03**
5	0.45	0.70	**0.83**	0.28	**0.62**	0.60	**0.24**
6	0.76	**0.60**	0.56	0.39	**0.53**	**0.35**	0.45
7	0.47	0.77	**0.88**	0.36	**0.47**	0.53	**0.46**
8	1	0.68	**0.83**	0.51	**0.70**	0.25	**0.15**

In addition, we provide solutions' positive and negative contributions during the decision phase in Table 7. Here, we can observe that the positive contribution of solutions generated by iRVEA is never less than the ones of IOPIS. In addition, except for the 6-th interaction, the negative contribution for solutions generated by iRVEA is smaller than that of IOPIS. In fact, because the negative contribution of solutions generated by iRVEA was much higher than the ones of IOPIS, we can observe that IOPIS had a better overall performance for this interaction. However, since iRVEA was the winner in the other four interactions, its decision phase performance assessment is superior to IOPIS. These observations align with the fact that the performance of iRVEA for the decision phase is better than IOPIS.

6 SURROGATE-ASSISTED INTERACTIVE EVOLUTIONARY MULTIOBJECTIVE OPTIMIZATION METHOD

In real-world MOPs, a DM is not necessarily aware of the shape of the Pareto front before solving the problem or whether her/his preferences are reachable or unreachable (see, e.g., [84]). Because of this challenge, interactive methods are often an ideal tool to be used since the DM can learn and update her/his preferences iteratively. Another challenge of real-world MOPs is that they may contain expensive functions[1]. A method called interactive K-RVEA was proposed in Article PIV for solving such problems.

Interactive K-RVEA uses Kriging models to speed up calculations to replace expensive functions. The uncertainty information in the predictions of the Kriging models is used in the model management to select some of the surrogate solutions[2] for updating the Kriging models.

6.1 Description of Interactive K-RVEA

In this section, we briefly introduce the main elements of interactive K-RVEA. Moreover, we discuss the model management of interactive K-RVEA.

Figure 15 shows the main steps of interactive K-RVEA for one interaction. First, we generate an initial population randomly. Then we use the initial population to train Kriging models and ask for the DM's preferences. We assume here that the DM provides a reference point. Next, we use iRVEA [56] to solve a multiobjective optimization problem, where the inexpensive Kriging models have replaced expensive functions. This means that the problem consists of the Kriging models and the inexpensive original functions (for which no surrogate models are needed). Afterward, we use model management to select some of the

[1] Reminder: see term 2 for the definition of expensive functions.

[2] Reminder: see term 6 for the definition of surrogate solutions.

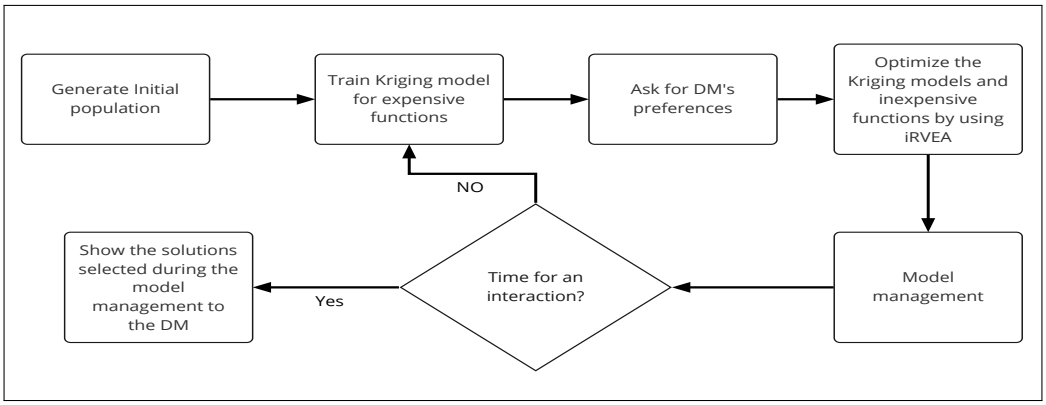


FIGURE 15 General steps of interactive K-RVEA for one interaction.

solutions and update the Kriging models. We refer to the last two steps as an iteration[3]. We repeat iterations a fixed number of times (interactive K-RVEA takes this number as a parameter). Then, we show the solutions we selected to the DM and evaluate the expensive functions at them. The DM continues the interactions until she/he finds the most preferred solution or other stopping criteria are met.

The behavior of interactive K-RVEA was initially demonstrated with a real-world problem [97]. In Article PIV, we tested different surrogate models on the initial population of the problem, and Kriging showed promising results. In fact, the Kriging models' accuracy was so good that we decided that the model management only needs to consider the DM's preferences.

We denote the number of solutions we update the Kriging models with by N_U. The model management of interactive K-RVEA can be summarized in the following steps:

1. Select $2 * N_U$ solutions that follow the DM's preferences the best based on an ASF (see (7)).

2. Out of the selected solutions in the previous step, select N_U that have the lowest uncertainty in their Kriging predictions and use them to update the Kriging models.

In the following section, we discuss two real-world problems introduced in Article PIV, 17. We describe the interactive solution processes for these problems and analyze the performance of interactive K-RVEA.

6.2 Two Real-World Problems and Their Solution Processes

The two problems we solve with interactive K-RVEA in this section are both computationally expensive. The first problem we discuss is an energy configuration problem for large buildings where each expensive evaluation[4] takes about 14 sec-

[3] Reminder: see term 7 for the definition of an iteration.

[4] See Term 3 for the definition of expensive evaluation.

onds. The second problem is finding the optimal pump design, where each expensive evaluation takes 18-22 hours.

6.2.1 Building Energy Configuration Problem

The building energy configuration problem was initially proposed in [96]. Our goal here is to find the optimal configuration of the energy system of a heterogeneous business building complex. Building management usually looks at how to optimally invest in building energy systems where different energy sources can be utilized. For instance, *photo voltaic* (PV) systems (also known as solar power systems) or battery storage capacity are some of the options that the building manager may invest in. Typically, there are several different options that the building manager can choose from. Because of this, the complexity of the building energy system can increase, and finding the optimal usage of each energy source becomes challenging.

Some works have addressed this problem as a single objective optimization problem [69,86], where a linear formulation of the problem has been used. However, the linear formulation cannot capture all aspects of the problem, such as the aging of the battery, dynamic user behavior, and so on. Therefore, a detailed building simulator [51] was proposed in [96]. For the simulation-based problem, the objectives to be considered are:

1. initial investment cost (to be minimized),
2. annual operation cost (to be minimized),
3. annual CO_2 emissions (to be minimized) and
4. resilience (to be maximized).

The decision variables of the problem are as follows:

- Three decision variables for configuring the PV system:
  - inclination angle,
  - orientation angle and
  - peak output power.
- Four decision variables for configuration of the battery system:
  - mix battery state-of-charge level,
  - max battery state-of-charge level,
  - charging threshold and
  - discharging threshold.
- Total volume of heat storage in m^3.
- Two decision variables for controlling co-generator for heat and power:

– upper threshold and

– lower threshold.

All decision variables have values within the interval $[0, 1]$. For more details on the objectives and decision variables, see [97].

For this problem, the author acted as the analyst and a domain expert as the DM. The DM mentioned that he does not want to wait more than three minutes between interactions. Moreover, the analyst had decided to use five solutions for updating the Kriging models since it was the number used in [22]. Since each simulation call took about 10 seconds, it meant that each iteration took about one minute, and therefore, we could have three iterations between interactions. This meant that interactive K-RVEA generated 15 solutions per interaction. The DM showed interest in analyzing all 15 solutions. (Hence, there was no need to select a subset of the generated solutions to be shown.) In Figures 16 - 21, the generated solutions are illustrated in a parallel coordinate plot in purple, the reference point in black, and the most preferred solution in orange.

6.2.1.1 Interactive Solution Process

In Article PIV, we provided the detailed interactive solution process for solving this problem. The initial population created randomly is illustrated in Figure 16. Figures 17 - 20 illustrate the solutions obtained by interactive K-RVEA. To emphasize the importance of model management, we went through the same interactive solution process (providing the same reference points) without updating the surrogate models. In other words, we optimized the surrogate models by using iRVEA. Figure 21 demonstrates the solutions generated by iRVEA.

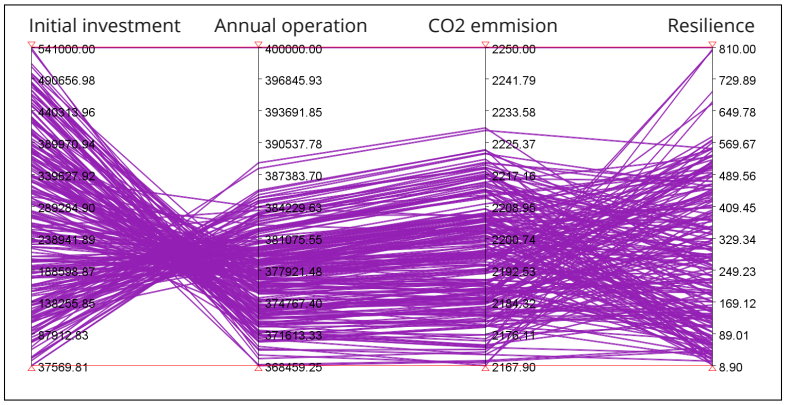


FIGURE 16 Initial population for the building energy configuration problem.

We can observe that, except for the first interaction, interactive K-RVEA found some solutions that are very similar to the reference point. Moreover, interactive K-RVEA found 15 solutions to show to the DM for every interaction, and the DM felt that his preferences were taken into consideration by the method.

In Figure 20 we illustrate the solutions generated in the fourth interaction. The DM chooses the (149886, 380764, 2211, 561) as the final solution (illustrated

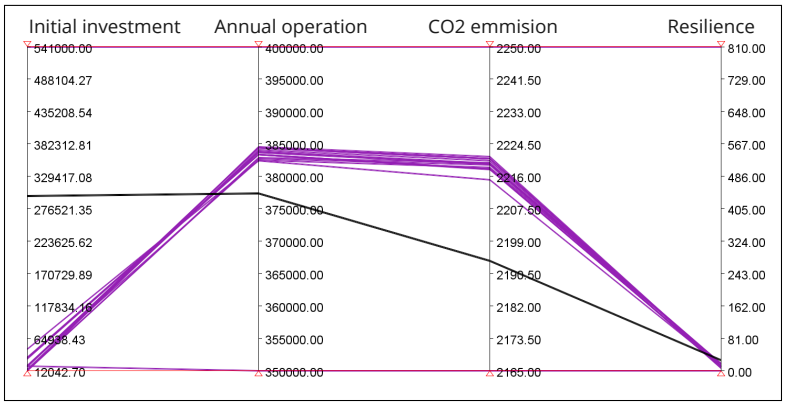


FIGURE 17 First interaction of interactive K-RVEA for the building energy configuration problem.

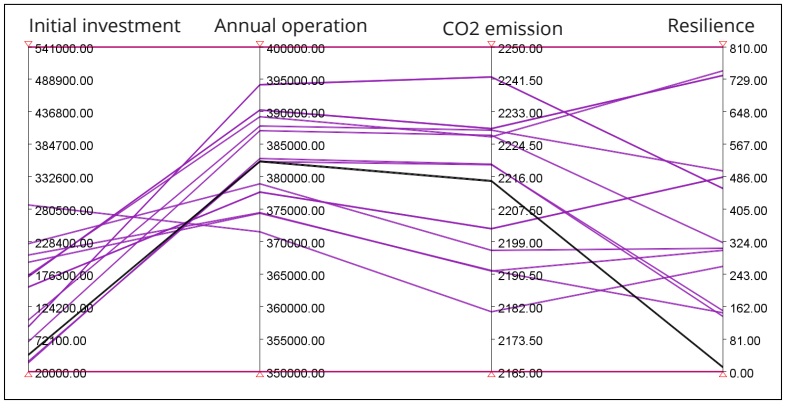


FIGURE 18 Second interaction of interactive K-RVEA for the building energy configuration problem.

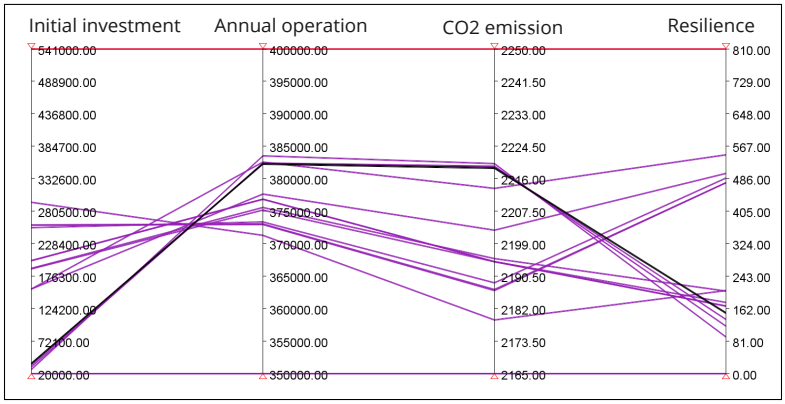


FIGURE 19 Third interaction of interactive K-RVEA for the building energy configuration problem.

as an orange line) since it had a good compromise between the first and third objectives.

Figure 21 illustrates the solutions that were generated by iRVEA. Here we can observe that the solutions iRVEA found are more scattered than those of in-

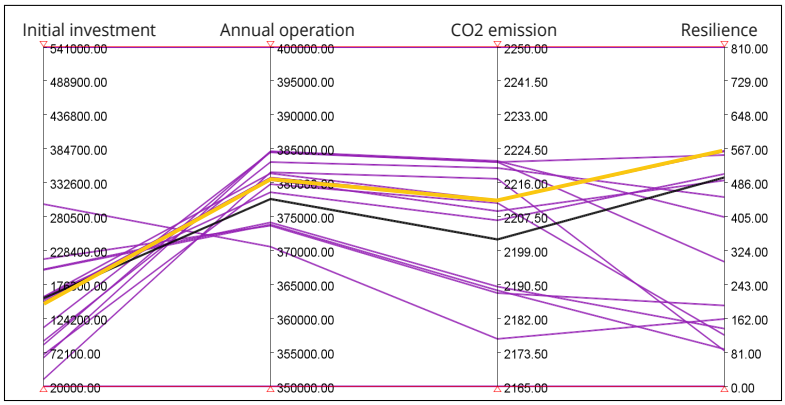


FIGURE 20 Fourth interaction of interactive K-RVEA for the building energy configuration problem.

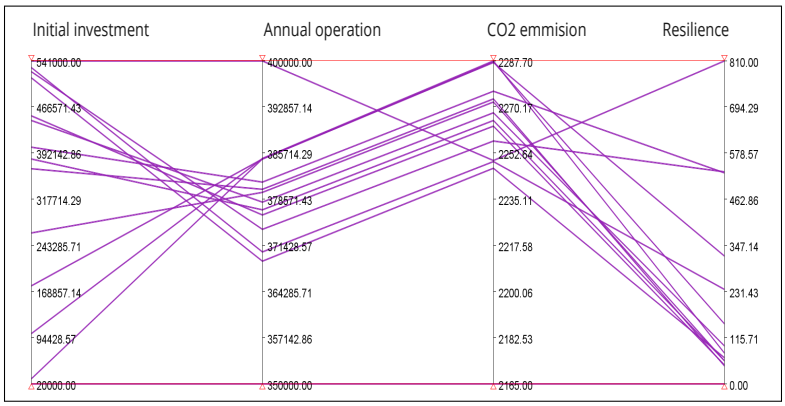


FIGURE 21 Final solutions generated by iRVEA.

teractive K-RVEA (see Figure 20), especially for the first objective. In Article PIV, we used the ASF values to assess how well the solutions of interactive K-RVEA and iRVEA reflected the DM's preferences. Additionally, we used the R-metric to measure the local convergence and local diversity of the solutions. R-metric can measure the diversity and convergence of solutions within the desired region and does not consider the solutions outside of it. For both ASF and R-metric, interactive K-RVEA was superior to iRVEA.

6.2.1.2 Performance Assessment

The indicator PHI had not been developed when Article PIV was written. Hence, we could not use it for the performance assessment of interactive K-RVEA. Moreover, with iRVEA, we did not have any iterations to update the surrogate models; therefore, we can only compare the final solutions of iRVEA with interactive K-RVEA. The performance assessment of interactive K-RVEA and iRVEA using PHI is shown in Table 8. Here we can observe that the PHI value of interactive K-RVEA is very close to 1, which means that almost all of the desired region is covered by the solutions. On the other hand, the PHI value for iRVEA is 0.18, demonstrating the method's poor performance compared to interactive K-RVEA.

TABLE 8 PHI values for the final population of interactive K-RVEA and iRVEA. The best values are in boldface.

	$PHI \uparrow$	**Positive contribution** $\uparrow$	**Negative contribution** $\downarrow$
Interactive K-RVEA	**0.98**	**0.78**	**0.22**
iRVEA	0.18	0.45	0.65

Furthermore, the positive/negative contributions of solutions generated by these methods can give us more insight into their performance. Here, we can observe that the positive/negative contributions of solutions generated by interactive K-RVEA are better than those of iRVEA. This means that interactive K-RVEA was able to:

1. generate solutions with better local convergence and local diversity than iRVEA, and

2. generates solutions that reflect the DM's preferences better than iRVEA.

In addition to the performance assessment of PHI, the analyst can use PHI as a tool during the interactive solution process to assist the DM. The interactive solution process did not have an explicit learning and decision phase. Therefore, as we suggested in Chapter 5, we treated the interactions as if we were in the learning phase. However, instead of calculating the value of PHI for every generation, we conducted it at every iteration in which we used expensive function evaluations during the model management (see Figure 22. This is justified since we only show expensive solutions to the DM. Moreover, We have provided the positive contribution (illustrated as a green line) and negative contribution (illustrated as a red line) in Figure 23.

Before getting to how to use PHI as the analyst to assist the DM, we would like to point out an important observation in Figure 22. We can observe that the values of PHI are increasing slowly (with some fluctuation). This means two things: Firstly, the surrogate models are getting more accurate due to the model management in interactive K-RVEA. Secondly, the DM is getting a sense of where the Pareto front may lie.

Figure 23 demonstrates the positive/negative contributions of solutions generated by interactive K-RVEA at each iteration. Suppose the analyst had access to this information (Figures 22 and 23) during the interactive solution process, he could advise the DM to keep the second reference point for one more interaction since the values are fluctuating. The method has not probably converged toward that region of the Pareto front yet.

For the fourth interaction, we can observe in Figure 23 that the positive contribution of solutions in the last iteration is increasing and the negative contribution is decreasing, which leads to improvements to the PHI values (see the fourth interaction in Figure 23). This means that the solutions are moving toward the desired region. Here, the analyst would have advised the DM to continue for one more interaction (if there is no time limitation) since the value of PHI is in-

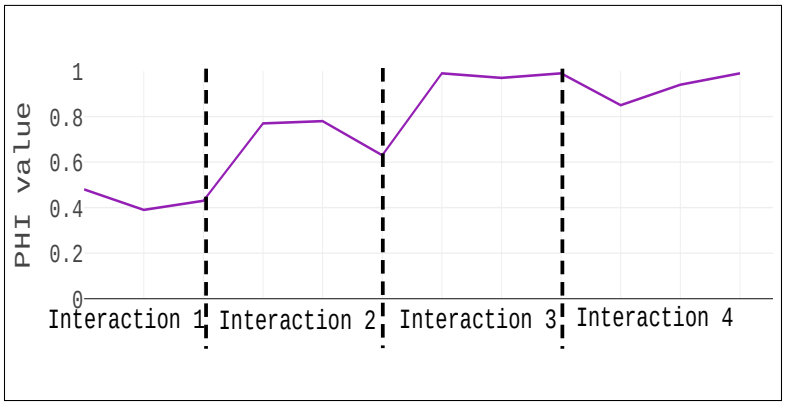


FIGURE 22 PHI values for each iteration of interactive K-RVEA The black dashed lines indicate when new interactions occurred. The purple line indicates the value of PHI at each iteration.

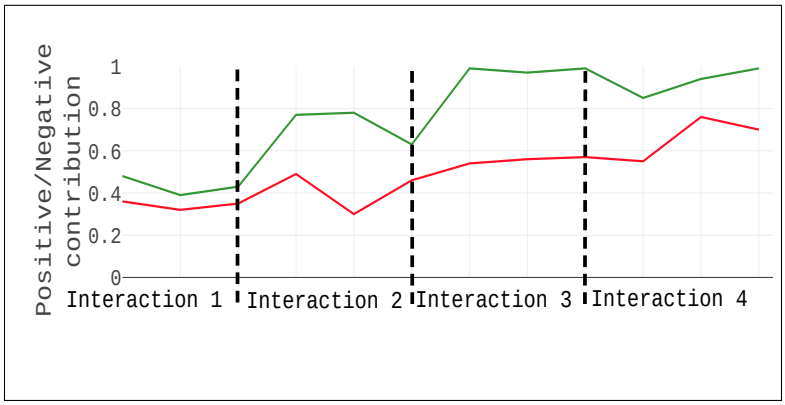


FIGURE 23 The positive and negative contribution of solutions at each iteration for the energy configuration problem. The black dashed lines show when new interactions occur. The green line is the positive contribution of solutions generated by interactive K-RVEA at each iteration. The red line is the negative contribution of solutions generated by interactive K-RVEA at each iteration.

creasing, and there may be a solution that he likes even more than the one he has already chosen.

6.2.2 Pump Design Problem

The pump design problem was introduced in [72], and the task was to design a pump with the maximum fluid flow rate in different parts of the pump. The problem has 22 decision variables leading to a pump stator design. The pump's flow rate is optimized for three different efficiencies 76%, 100%, and 120% of the pump design point.

A computational fluid dynamics simulator was used to analyze the design of the pump. The simulator is extremely computationally expensive, so every simulation call takes about 16–20 hours to complete. Hence, using surrogate-assisted methods is necessary. Previously, several surrogate-assisted methods

(e.g., K-RVEA [22]) have been used to solve this problem [72]. Even though some promising solutions have been found, there remain several challenges to be addressed:

1. The computation time is very long, and it takes weeks to solve the problem with methods like K-RVEA;

2. For some decision variables, the simulation fails, and no objective function values are obtained. It is not possible to avoid failed simulations by introducing constraints, but it is still desirable to avoid wasting time;

3. Not all nondominated solutions are interesting to the DM, but there is a specific trade-off between the objectives that are of interest.

In the literature, Kriging methods have shown good performance for the pump design problem. Moreover, K-RVEA had performed in a promising way for this problem in [72]. Therefore, interactive K-RVEA was a natural choice to address the challenges mentioned above. In what follows, we describe the interactive solution process reported in [17]. In the parallel coordinate plots in Figures 24-30, the generated solutions are illustrated in purple, the reference point in black, and the most preferred solution in orange.

Interactive Solution Process

The DM wanted to see 15 solutions at each interaction, and based on this information, we set the parameters of interactive K-RVEA. This means that in the model management, we should select 15 solutions to update the surrogate models before each interaction. Figure 24 shows the initial solutions generated randomly. The DM was a domain expert and quite familiar with the problem. Therefore, he started from the decision phase (he had, for example, seen solutions generated by K-RVEA).

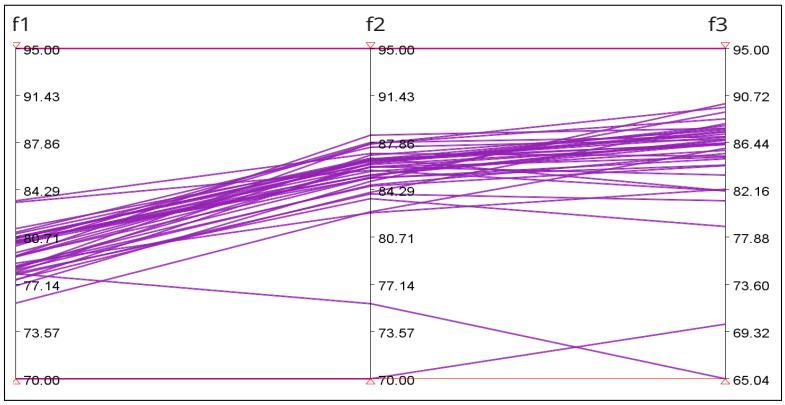


FIGURE 24 The initial population of the pump design problem

For the first interaction, the DM provided the reference point $\hat{z}^1 = (84, 90, 86)$. Figure 25 illustrates the solutions after the first interaction. The DM thought that

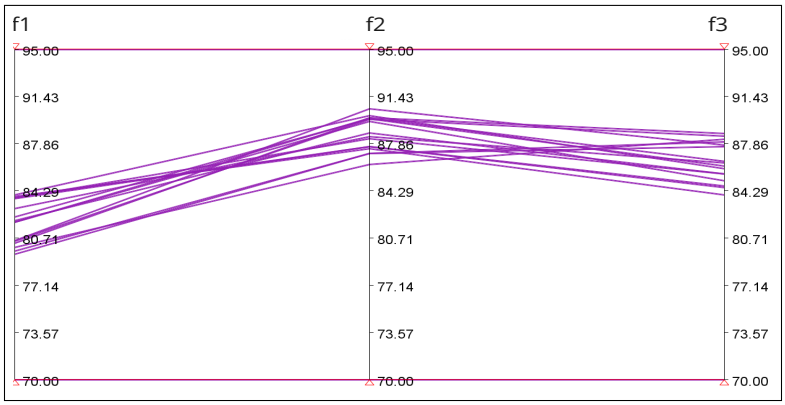


FIGURE 25 Solutions generated by interactive K-RVEA for the pump design problem after the first interaction.

the solutions could still improve and decided to keep the same reference point ($\hat{z}^2 = \hat{z}^1$) for the second interaction.

In Figure 26, we see the solutions for the second interaction. However, again, the DM asked to keep the reference point the same for the next interaction ($\hat{z}^3 = \hat{z}^2$) since he thought there was still room for improvement in all three objectives.

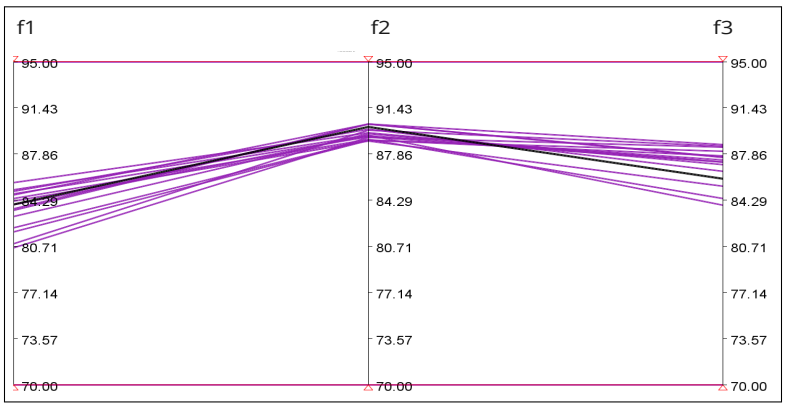


FIGURE 26 Solutions generated by interactive K-RVEA for the pump design problem after the second interaction.

Figure 27 illustrates the solutions generated by interactive K-RVEA after the third interaction. Here, the DM found some solutions that dominated the reference point. Therefore, he modified the reference point by increasing the aspiration level for the first objective and provided the fourth reference point as $\hat{z}^4 = (86, 90, 86)$. This was because the DM hoped to improve the first objective without sacrificing the second and third objectives. In Figure 28, we see the generated solutions after the fourth interaction. Again, the DM was not satisfied with any of them and decided to provide the same reference point ($\hat{z}^5 = \hat{z}^4$) for the fifth interaction. The resulting solutions are shown in Figure 29. Here we can observe that one of the solutions dominated $\hat{z}^5$. For the sixth interaction, the DM hoped to improve the first objective even more without sacrificing in the other two objectives. He provided the sixth reference point as $\hat{z}^6 = (88, 90, 84)$. In Figure 30, we

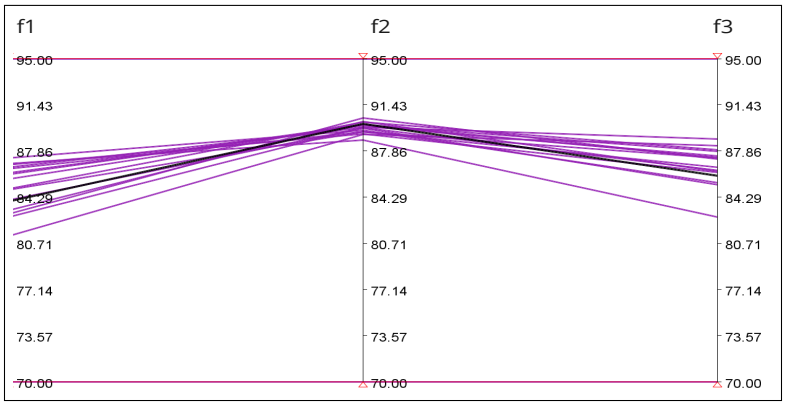


FIGURE 27 Solutions generated by interactive K-RVEA for the pump design problem after the third interaction.

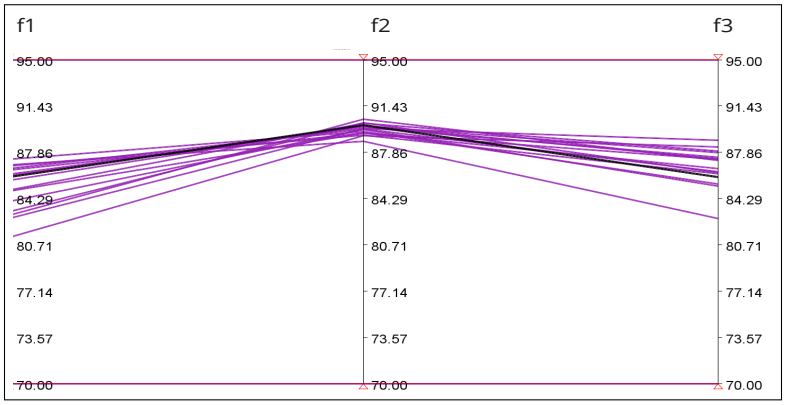


FIGURE 28 Solutions generated interactive K-RVEA for the pump design problem after the fourth interaction.

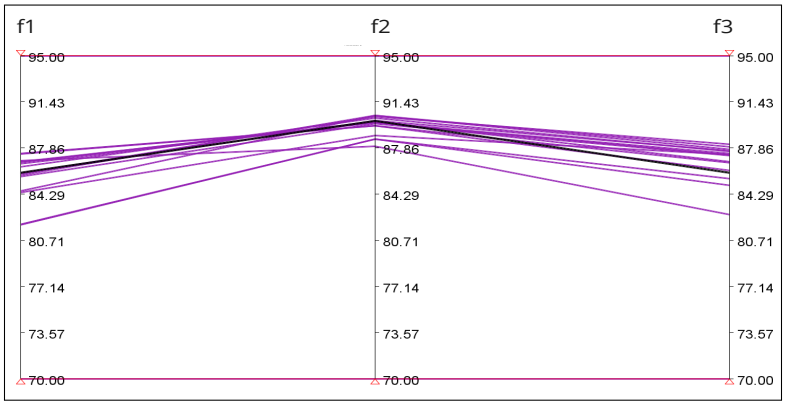


FIGURE 29 Solutions generated interactive K-RVEA for the pump design problem after the fifth interaction.

show the solutions generated based on $\hat{z}^6$. The DM was finally satisfied with one of the solutions and chose $(87.3, 90.1, 87.5)$ as the most preferred solution. The DM believed the chosen solution represented a good trade-off between the three objectives, and he could fine-tune solutions successfully.

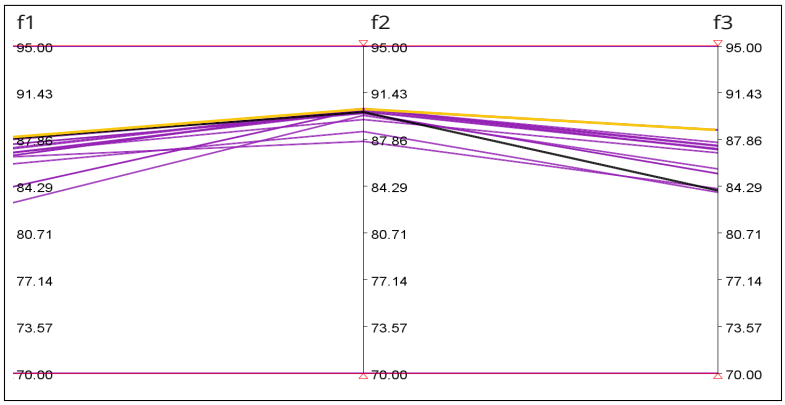


FIGURE 30 Solutions generated by interactive K-RVEA for the pump design problem after the sixth interaction. The most preferred solution is denoted by an orange line.

The results reported in the literature involved some failed expensive simulations. For example, K-RVEA had 36 failed expensive simulations. On the other hand, interactive K-RVEA involved no failed expensive simulations. Thus, it could avoid the decision vectors that lead to such a result. We believe this behavior was because of the DM's expertise and the fact that he knew which part of the Pareto front should be explored. In addition, interactive K-RVEA obeyed the DM's preferences well and provided solutions that reflected the provided reference points.

When it comes to problems such as the pump design problem, where function evaluations are extremely expensive, even saving a small number of evaluations can lead to substantial computational savings. With interactive K-RVEA, the DM could find the most preferred solution with 30 iterations (6 interactions), which means that it saved 11 days in computational time compared to K-RVEA.

7 CONCLUSIONS AND AUTHOR'S CONTRIBUTION

In the final chapter of this thesis, we provide the conclusion and future research directions. Furthermore, we provide the authors' contributions to individuals who had a role in writing the articles mentioned in this thesis.

7.1 Conclusions

In this thesis, we have addressed several challenges that arise when solving real-world MOPs. We can roughly summarize them as follow:

1. Handling computationally expensive constraints;

2. Incorporating the DM's preferences in solving computationally expensive problems;

3. Assessing the performance of preference-based methods, especially interactive ones.

Real-world problems may have computationally expensive constraints. However, there were no preference-based evolutionary methods that could handle them. In Article PI, we proposed the KAEA-C that is capable of addressing MOPs with computationally expensive constraints. We borrowed components from different methods to assemble new ones, compared their performance against KAEA-C, and demonstrated the strengths of the proposed method.

As our next step, our goal was to further develop KAEA-C into an interactive evolutionary method. However, one of the main obstacles was that there were no performance indicators to assess the performance of interactive methods. Therefore, in Article PII we characterized properties that interactive indicators need to possess. We argue that a single indicator cannot necessarily possess all desirable properties, but we may need different interactive indicators for different purposes. For example, the learning and decision phases have very different

characteristics, and probably a single indicator cannot assess the performance of these phases simultaneously.

We then extended the research in Article PIII where we developed PHI as the first performance indicator for interactive evolutionary methods. Here, we used the concept of a hypervolume indicator but modified it to reward solutions covering a desired region and to punish others. PHI can provide the overall performance assessment of a method, how far the solutions are from the desired region (negative contribution), and how diverse and well converged the solutions inside the desired region are (positive contribution). We showed how to utilize PHI to assess the performance of a method in the learning and decision phases. In addition, we demonstrated how PHI could help the analyst to identify the most suitable interactive evolutionary method for a given MOP.

Interactive K-RVEA was developed in Article PIV for real-world problems with computationally expensive objective functions. In the paper, we were able to reduce the computation time of solving the building energy configuration problem from 23 hours to around only one hour. In addition, the waiting time for each interaction was only three minutes which was what the DM requested from the author. This means we were able to save a massive amount of computation time. As for the performance, we used R-metric and an ASF to compare the final populations of interactive K-RVEA and interactive RVEA (with surrogates but without model management). For both indicators, interactive K-RVEA had a better performance. More importantly, when we applied interactive K-RVEA, the DM was satisfied with the solutions found and thought the method was useful to them and they could use it for their internal use. Additionally, we used PHI to show how it gives more insight to the analyst so he/she can assist the DM in such problems and analyze the behavior of interactive K-RVEA.

We applied interactive K-RVEA also in a pump design problem with three objectives. Evaluating the objective function values required a simulator which was computationally extremely expensive (16-20 hours for each simulation). Interactive K-RVEA showed very promising results and could satisfy the DM within 90 function evaluations. Compared to the previous methods applied to solve this problem, interactive K-RVEA saved 15 function evaluations which in this case means almost 11 days in computation costs.

The proposed methods and indicator in this thesis can be utilized to make solving real-world multiobjective optimization problems more manageable. We only showed two applications of interactive K-RVEA, but many more applications can benefit from the methods developed in this thesis. All methods developed in this thesis have been implemented in Python as a part of an open-source software frameworkDESDEO [87], designed for solving MOPs with preference-based methods (especially interactive methods) and therefore easily accessible to everyone to use.

The methods and indicators proposed in this thesis are one of a kind in the sense that there were no immediate competitors in the literature. Besides the challenges they addressed in this thesis, they open the doors to new research paradigms. We demonstrated a clear need for more surrogate-assisted preference-

based methods and performance indicators for interactive evolutionary methods.

In short, this thesis sheds light on the importance of surrogate-assisted preference-based methods and how they can help us solve computationally expensive problems. The proposed approaches include incorporating the DM's preferences in the model management, handling computationally expensive constraints, and assessing the performance of interactive methods. In Chapter 1 we mentioned six challenges (S1-S3 and IN1-IN3). In this thesis, we have addressed all of them.

There are still several challenges for future research. For example, in both KAEA-C and interactive K-RVEA, we used a linear formulation to generate the reference vectors for decomposing the objective space. One could explore different and perhaps new ways of generating reference vectors. In addition, utilizing different types of model management is a research direction that can be explored more. The methods assumed in this thesis were only able to incorporate DM's preferences in the form of a reference point. Developing more preference-based surrogate-assisted methods that can incorporate different types of preferences would be quite useful. Moreover, surrogate-assisted interactive evolutionary methods have many parameters that need to be set. Creating an artificial intelligent-based method that can set or change these parameters during the solution process or model management is an interesting future research topic.

As for the performance indicators, the list of desirable properties can possibly be improved, and more research is needed to study desired regions and regions of interest. Moreover, the dystopian point used in PHI for calculating the hypervolume requires further research on how to set it in interactive methods. For example, we may need to set it differently for the learning and decision phases. Furthermore, PHI uses hypervolume calculation many times, which makes it computationally expensive as the number of objectives increases. Therefore, exploring other interactive indicators where the calculation is inexpensive would be desirable. Next, we can use the concept of PHI to develop an indicator-based interactive method to see if PHI can also guide the search direction toward the desired region. Finally, every preference-based indicator developed so far assumes the DM provides preferences in the form of a reference point. However, there are other kinds of interactive methods. Therefore, future indicators should be able to incorporate different types of preferences.

7.2 Author's Contribution

The author's supervisors suggested working with interactive evolutionary multiobjective optimization methods, surrogate models, and performance indicators. Initially, the author conducted a literature survey on computationally expensive multiobjective optimization problems. For example, he found out that preference-based methods that can handle computationally expensive constraints have not been explored enough. He started testing different ideas toward developing an interactive method. However, because of the complexity of the matter, the au-

thor decided to first develop an a priori evolutionary method, where the DM's preferences are to be considered only once.

The author tested different selection strategies, model management techniques, and surrogate models. The initial idea of KAEA-C in using reference vectors to decompose the objective space came from Dr. Jussi Hakanen. Moreover, Prof. Kaisa Miettinen provided rich literature information about the advantages and disadvantages of different approaches. The main idea of using two fitness functions, one for converging solutions toward the Pareto front and one for incorporating the DM's reference point, came from the author. In addition, he had the idea that the DM's preferences should also be considered during the model management and came up with a novel model management approach introduced in KAEA-C. The author implemented KAEA-C in Python and tested its performance on different benchmark problems. Finally, as a result of this research, the author wrote major parts of Article PI while comments of the supervisors helped in improving it.

The idea of desirable properties for interactive indicators proposed in Article PII was initiated when the author was on a research visit at the University of Skövde and met Dr. Sunith Bandaru, who has worked with preference-based methods and indicators. During the visit, the author and Dr. Sunith Bandaru started collecting ideas as desirable properties of indicators designed for interactive evolutionary multiobjective optimization methods. Dr. Bekir Afsar and Prof. Kaisa Miettinen brought their expertise and experiences in comparing interactive methods. The Article PII with the list of desirable properties and a detailed discussion of them was mostly written by the author and iterated and further polished together with the co-authors.

A deeper understanding of desirable properties was necessary before developing indicators for interactive methods. It was then natural to continue collaboration with Dr. Sunith Bandaru and develop the first indicator for interactive evolutionary methods. The author had the idea of PHI when he was studying different indicators and noticed that for most a priori indicators, solutions that are outside of the desired region are not considered in the performance evaluation. Thus, the main idea of PHI was developed by the author, and Dr. Bandaru helped with some fine-tuning in calculation and in utilizing PHI for assessing the learning phase. Moreover, the author found a way to find similarities between the reference points that the DM provides during the interactive solution process and use this information to assess the ability of interactive methods to refine solutions during the decision phase. Prof. Kaisa Miettinen and Assoc. prof. Michael Emmerich helped identify the proposed method's potential drawbacks. Dr. Bekir Afsar provided the code for the ADM he developed and provided valuable insights of interactive solution processes. The author implemented PHI and most of other a priori indicators in Python and gathered the numerical results. The author wrote most of Article PIII and the co-authors provided valuable comments to mature the ideas.

Dr. Tobias Rodemann provided the building simulator used in Article PIV, where the interactive K-RVEA method was introduced. He expressed a need for

an interactive method where the DM can explore different building energy configurations. One of the main issues at the time was the computationally expensive nature of the problem. The author searched the literature and realized that there was no surrogate-assisted interactive evolutionary method for computationally expensive multiobjective optimization problems available. Additionally, the author tested several different surrogate models to find the most suitable one for this problem. The idea of incorporating the DM's preferences in the model management was from the author. Prof. Kaisa Miettinen and Dr. Jussi Hakanen provided further information on using the achievement scalarizing functions to determine solutions that reflect the DM's preferences. Moreover, Dr. Tobias Rodemann acted as the DM. The author wrote most of the paper and iterated improving it based on the comments of the co-authors.

The author was the main source of new ideas, implemented all methods and conducted all numerical experiments. Therefore, the role of the author in each paper was significant.

YHTEENVETO (SUMMARY IN FINNISH)

Monissa reaalielämän ongelmissa on useita ristiriitaisia tavoitefunktioita ja niiden ratkaiseminen voi vaatia paljon laskentaresursseja. Tällöin sijaismalleja käyttävät preferenssipohjaiset menetelmät ovat usein käyttökelpoisia. Niiden käyttämiseen liittyy kuitenkin haasteita, joita ei ole vielä riittävästi tutkittu kirjallisuudessa. Esimerkiksi sijaismalleja käytetään korvaamaan laskennallisesti aikaavieviä tavoitefunktioita laskentaresurssin säästämiseksi, mutta nämä mallit eivät välttämättä ole riittävän tarkkoja. Toinen haaste on preferenssipohjaisten menetelmien suoriutumisen vertaileminen keskenään silloin, kun päätöksentekijän preferenssejä tulee huomioida ratkaisuprosessin aikana.

Tässä väitöskirjassa on kaksi osaa: menetelmäkehitys ja interaktiivisten menetelmien arviointi. Menetelmäkehitysosassa esitellään KAEA-C -menetelmä, joka sekä huomioi ennen ratkaisuprosessin alkua annettuja preferenssitietoja että pystyy käsittelemään laskennallisesti aikaavieviä funktioita (sekä tavoitteita että rajoitteita). Lisäksi esitellään interaktiivinen K-RVEA -menetelmä, joka on kehitetty laskennallisesti aikaavieville ongelmille yhdessä Honda Research Institute Europen tutkijoiden kanssa. Menetelmää sovelletaan reaalielämän ongelmiin ja havainnollistetaan miten päätöksentekijän preferenssejä huomioidaan ratkaisuprosessissa ja miten laskenta-aikaa voidaan säästää.

Menetelmien arviointiosassa keskitytään interaktiivisten menetelmien vertailemiseen keskenään. Tätä pohjustetaan keskittymällä ensin interaktiivisille menetelmille soveltuvien mittareiden toivottaviin ominaisuuksiin. Esimerkkien avulla havainnollistetaan sitä, miksi olemassa olevat mittarit eivät sovellu interaktiivisten menetelmien arviointiin ja tarjotaan suosituksia mittareiden kehittämiseen. Lopuksi esitellään uusi mittari PHI, joka pystyy arvioimaan interaktiivisen evoluutiopohjaisen menetelmän suoriutumista erikseen ratkaisuprosessin oppimis- ja päätösvaiheessa.

Väitöskirja koostuu menetelmistä, mittareista ja sovelluksista. Siten sitä voidaan pitää lähtökohtana jatkotutkimukselle ja preferenssipohjaisen monitavoiteoptimoinnin alan laajemmalle kehitykselle.

REFERENCES

[1] ABDOLSHAH, M., SHILTON, A., RANA, S., GUPTA, S., AND VENKATESH, S. Multi-objective Bayesian optimisation with preferences over objectives. *Advances in Neural Information Processing Systems 32* (2019), 12214–12224.

[2] ABRAHAM, A., AND JAIN, L. Evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization*. Springer, 2005, pp. 1–6.

[3] AFSAR, B., MIETTINEN, K., AND RUIZ, A. An artificial decision maker for comparing reference point based interactive evolutionary multiobjective optimization methods. In *Evolutionary Multi-Criterion Optimization, 11th International Conference, EMO 2021, Proceedings* (2021), H. Ishibuchi, Q. Zhang, R. Cheng, K. Li, H. Li, H. Wang, and A. Zhou, Eds., Springer, pp. 619–631.

[4] AFSAR, B., MIETTINEN, K., AND RUIZ., F. Assessing the performance of interactive multiobjective optimization methods: A survey. *ACM Computing Surveys 54*, 4 (2021), 1–27.

[5] AFSAR, B., RUIZ, A. B., AND MIETTINEN, K. Comparing interactive evolutionary multiobjective optimization methods with an artificial decision maker. *Complex & Intelligent Systems* (to appear).

[6] AUDET, C., BIGEON, J., CARTIER, D., LE D., S., AND SALOMON, L. Performance indicators in multiobjective optimization. *European Journal of Operational Research 292*, 2 (2021), 397–422.

[7] AUDET, C., BIGEON, J., CARTIER, D., LE DIGABEL, S., AND SALOMON, L. Performance indicators in multiobjective optimization. *European Journal of Operational Research 292*, 2 (2021), 397–422.

[8] BALANDAT, M., KARRER, B., JIANG, D., DAULTON, S., LETHAM, B., WILSON, A. G., AND BAKSHY, E. BoTorch: a framework for efficient Monte-Carlo Bayesian optimization. *Advances in Neural Information Processing Systems 33* (2020), 21524–21538.

[9] BANDARU, S., AND SMEDBERG, H. A parameterless performance metric for reference-point based multi-objective evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference* (2019), ACM, pp. 499–506.

[10] BECHIKH, S., KESSENTINI, M., BEN SAID, L., AND GHÉDIRA, K. Chapter four - preference incorporation in evolutionary multiobjective optimization: a survey of the state-of-the-art. In *Advances in Computers*, A. R. Hurson, Ed., vol. 98. Elsevier, 2015, pp. 141–207.

[11] BECHIKH, S., SAID, L. B., AND GHEDIRA, K. Estimating nadir point in multi-objective optimization using mobile reference points. In *IEEE Congress on Evolutionary Computation* (2010), IEEE, pp. 1–9.

[12] BEUME, N., FONSECA, C. M., LOPEZ-IBANEZ, M., PAQUETE, L., AND VAHRENHOLD, J. On the complexity of computing the hypervolume indicator. *IEEE Transactions on Evolutionary Computation 13*, 5 (2009), 1075–1082.

[13] BEUME, N., NAUJOKS, B., AND EMMERICH, M. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research 181*, 3 (2007), 1653–1669.

[14] BEZERRA, L. C., LÓPEZ-IBÁÑEZ, M., AND STÜTZLE, T. A large-scale experimental evaluation of high-performing multi-and many-objective evolutionary algorithms. *Evolutionary Computation 26*, 4 (2018), 621–656.

[15] BLANK, J., AND DEB, K. A running performance metric and termination criterion for evaluating evolutionary multi- and many-objective optimization algorithms. In *2020 IEEE Congress on Evolutionary Computation, Proceedings* (2020), pp. 1–8.

[16] BRINGMANN, K., AND FRIEDRICH, T. Approximating the volume of unions and intersections of high-dimensional geometric objects. *Computational Geometry 43*, 6-7 (2010), 601–610.

[17] BURKOTOVA, J., AGHAEI POUR, P., KRÁTKÝ, T., AND MIETTINEN, K. Interactive multiobjective optimization of an extremely computationally expensive pump design problem. *Under review*.

[18] CHABANE, B., BASSEUR, M., AND HAO, J.-K. Lorenz dominance based algorithms to solve a practical multiobjective problem. *Computers & Operations Research 104* (2019), 1–14.

[19] CHENG, R., JIN, Y., NARUKAWA, K., AND SENDHOFF, B. A multiobjective evolutionary algorithm using gaussian process-based inverse modeling. *IEEE Transactions on Evolutionary Computation 19*, 6 (2015), 838–856.

[20] CHENG, R., JIN, Y., OLHOFER, M., AND SENDHOFF, B. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation 20*, 5 (2016), 773–791.

[21] CHENG, R., RODEMANN, T., FISCHER, M., OLHOFER, M., AND JIN, Y. Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation. *IEEE Transactions on Emerging Topics in Computational Intelligence 1*, 2 (2017), 97–111.

[22] CHUGH, T., JIN, Y., MIETTINEN, K., HAKANEN, J., AND SINDHYA, K. A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation 22*, 1 (2018), 129–142.

[23] CHUGH, T., KRATKY, T., MIETTINEN, K., JIN, Y., AND MAKKONEN, P. Multiobjective shape design in a ventilation system with a preference-driven surrogate-assisted evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference* (New York, NY, USA, 2019), ACM, p. 1147–1155.

[24] CHUGH, T., SINDHYA, K., HAKANEN, J., AND MIETTINEN, K. A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing 23*, 9 (2019), 3137–3166.

[25] COELLO, C. C. Handling preferences in evolutionary multiobjective optimization: A survey. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)* (2000), vol. 1, IEEE, pp. 30–37.

[26] COELLO COELLO, C. A. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering 191*, 11 (2002), 1245–1287.

[27] CRESSIE, N. *Statistics for Spatial Data*. John Wiley & Sons, 2015.

[28] DARWIN, C. *On the Origin of Species, 1859*. Routledge, 2004.

[29] DATTA, R., AND REGIS, R. G. A surrogate-assisted evolution strategy for constrained multi-objective optimization. *Expert Systems with Applications 57* (2016), 270–284.

[30] DAULTON, S., BALANDAT, M., AND BAKSHY, E. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. *Advances in Neural Information Processing Systems 33* (2020), 9851–9864.

[31] DE WINTER, R., VAN STEIN, B., AND BÄCK, T. Samo-cobra: A fast surrogate assisted constrained multi-objective optimization algorithm. In *Evolutionary Multi-Criterion Optimization: 11th International Conference, EMO 2021, Shenzhen, China, March 28–31, 2021, Proceedings 11* (2021), Springer, pp. 270–282.

[32] DEB, K. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001.

[33] DEB, K., GUPTA, S., DAUM, D., BRANKE, J., MALL, A. K., AND PADMANABHAN, D. Reliability-based optimization using evolutionary algorithms. *IEEE Transactions on Evolutionary Computation 13*, 5 (2009), 1054–1074.

[34] DEB, K., AND JAIN, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation 18*, 4 (2014), 577–601.

[35] DEB, K., AND MIETTINEN, K. Nadir point estimation using evolutionary approaches: better accuracy and computational speed through focused search. In *Multiple Criteria Decision making for Sustainable Energy and Transportation Systems*, M. Ehrgott, B. Naujoks, T. J. Stewart, and J. Wallenius, Eds. Springer, 2010, pp. 339–354.

[36] DEB, K., MIETTINEN, K., AND CHAUDHURI, S. Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches. *IEEE Transactions on Evolutionary Computation 14*, 6 (2010), 821–841.

[37] DEB, K., PRATAP, A., AGARWAL, S., AND MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation 6*, 2 (2002), 182–197.

[38] DEB, K., THIELE, L., LAUMANNS, M., AND ZITZLER, E. Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation.* (2002), vol. 1, IEEE, pp. 825–830.

[39] DENG, J., AND ZHANG, Q. Approximating hypervolume and hypervolume contributions using polar coordinate. *IEEE Transactions on Evolutionary Computation 23*, 5 (2019), 913–918.

[40] DIAZ, C., ASLAM, T., NG, A. H., FLORES-GARCÍA, E., AND WIKTORSSON, M. Simulation-based multi-objective optimization for reconfigurable manufacturing system configurations analysis. In *2020 Winter Simulation Conference* (2020), pp. 1527–1538.

[41] DRUCKER, H., BURGES, C. J., KAUFMAN, L., SMOLA, A. J., AND VAPNIK, V. Support vector regression machines. In *Advances in Neural Information Processing Systems* (1997), pp. 155–161.

[42] EHRGOTT, M. *Multicriteria optimization*, vol. 491. Springer, 2005.

[43] EHRGOTT, M. Vilfredo Pareto and multi-objective optimization. *Documenta math* (2012), 447–453.

[44] EMMERICH, M. *Single-and multi-objective evolutionary design optimization assisted by Gaussian random field metamodels*. Computer Science Department, TU Dortmund, Germany, 2005.

[45] EMMERICH, M., AND DEUTZ, A. H. A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural Computing 17*, 3 (2018), 585–609.

[46] FALCÓN-CARDONA, J. G., AND COELLO, C. A. C. Indicator-based multi-objective evolutionary algorithms: A comprehensive survey. *ACM Computing Surveys 53*, 2 (2020), 1–35.

[47] FELIOT, P., BECT, J., AND VAZQUEZ, E. A Bayesian approach to constrained single-and multi-objective optimization. *Journal of Global Optimization 67*, 1-2 (2017), 97–133.

[48] FILATOVAS, E., KURASOVA, O., AND SINDHYA, K. Synchronous r-nsga-ii: an extended preference-based evolutionary algorithm for multi-objective optimization. *Informatica 26*, 1 (2015), 33–50.

[49] FONSECA, C. M., AND FLEMING, P. J. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the ICGA-93: Fifth International Conference on Genetic Algorithms* (1993), S. Forrest and M. Kaufmann, Eds., vol. 93, San Mateo, pp. 416–423.

[50] FORRESTER, A. I., AND KEANE, A. J. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences 45*, 1 (2009), 50–79.

[51] FRITZSON, P., AND BUNUS, P. Modelica-a general object-oriented language for continuous and discrete-event system modeling and simulation. In *Proceedings of the 35th Annual Simulation Symposium* (2002), IEEE, pp. 365–380.

[52] GIAGKIOZIS, I., PURSHOUSE, R. C., AND FLEMING, P. J. An overview of population-based algorithms for multi-objective optimisation. *International Journal of Systems Science 46*, 9 (2015), 1572–1599.

[53] GIBSON, F. J., EVERSON, R. M., AND FIELDSEND, J. E. Guiding surrogate-assisted multi-objective optimisation with decision maker preferences. In *Proceedings of the Genetic and Evolutionary Computation Conference* (2022), pp. 786–795.

[54] GRIFFITHS, D. J., AND SCHROETER, D. F. *Introduction to Quantum Mechanics*. Cambridge University Press, 2018.

[55] GUERREIRO, A. P., FONSECA, C. M., AND PAQUETE, L. The hypervolume indicator: Computational problems and algorithms. *ACM Computing Surveys (CSUR) 54*, 6 (2021), 1–42.

[56] HAKANEN, J., CHUGH, T., SINDHYA, K., JIN, Y., AND MIETTINEN, K. Connections of reference vectors and different types of preference information in interactive multiobjective evolutionary algorithms. In *2016 IEEE Symposium Series on Computational Intelligence, Proceedings* (2016), pp. 1–8.

[57] HANSEN, M. P., AND JASZKIEWICZ, A. *Evaluating the quality of approximations to the non-dominated set*. Technical report, Institute of Mathematical Modeling, Technical University of Denmark, 1998.

[58] HOU, Z., YANG, S., ZOU, J., ZHENG, J., YU, G., AND RUAN, G. A performance indicator for reference-point-based multiobjective evolutionary optimization. In *2018 IEEE Symposium Series on Computational Intelligence, Proceedings* (2018), IEEE, pp. 1571–1578.

[59] IRIONDO PASCUAL, A., HÖGBERG, D., SYBERFELDT, A., BROLIN, E., PEREZ LUQUE, E., HANSON, L., AND LÄMKULL, D. Multi-objective optimization of ergonomics and productivity by using an optimization framework. In *Proceedings of the 21st Congress of the International Ergonomics Association* (2022), N. L. Black, W. P. Neumann, and I. Noy, Eds., Springer, pp. 374–378.

[60] ISHIBUCHI, H., IMADA, R., SETOGUCHI, Y., AND NOJIMA, Y. How to specify a reference point in hypervolume calculation for fair performance comparison. *Evolutionary Computation 26*, 3 (2018), 411–440.

[61] ISHIBUCHI, H., PANG, L. M., AND SHANG, K. Difficulties in fair performance comparison of multi-objective evolutionary algorithms [research frontier]. *IEEE Computational Intelligence Magazine 17*, 1 (2022), 86–101.

[62] JAIN, H., AND DEB, K. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation 18*, 4 (2014), 602–622.

[63] JASZKIEWICZ, A. *Multiple objective metaheuristic algorithms for combinatorial optimization*. PhD thesis, Poznan University of Technology, Poznan, Poland, 2001.

[64] JASZKIEWICZ, A. Improved quick hypervolume algorithm. *Computers Operations Research 90* (2018), 72–83.

[65] JH, H. Adaptation in natural and artificial systems. *Ann Arbor* (1975).

[66] JIN, Y. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation 1*, 2 (2011), 61–70.

[67] JIN, Y., WANG, H., AND CHAOLI., S. *Data-Driven Evolutionary Optimization*. Springer, 2021.

[68] KADZIŃSKI, M., TOMCZYK, M. K., AND SŁOWIŃSKI, R. Preference-based cone contraction algorithms for interactive evolutionary multiple objective optimization. *Swarm and Evolutionary Computation 52* (2020), 100602.

[69] KHODR, H., VALE, Z. A., RAMOS, C., SOARES, J., MORAIS, H., AND KADAR, P. Optimal methodology for renewable energy dispatching in islanded operation. In *IEEE PES Transmission and Distribution Conference and Exposition 2010* (2010), IEEE, pp. 1–7.

[70] KNOWLES, J. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation 10*, 1 (2006), 50–66.

[71] KÖKSALAN, M., AND KARAHAN, I. An interactive territory defining evolutionary algorithm: itdea. *IEEE Transactions on Evolutionary Computation 14*, 5 (2010), 702–722.

[72] KRÁTKÝ, T. *Shape optimization of hydraulic surfaces of the impeller and stator parts of hydrodynamic pumps*. PhD thesis, Palacký University Olomouc, Faculty of Science, 2020.

[73] LAN, G., TOMCZAK, J. M., ROIJERS, D. M., AND EIBEN, A. Time efficiency in optimization with a Bayesian-evolutionary algorithm. *Swarm and Evolutionary Computation* (2021), 100970.

[74] LÁRRAGA, G., AND MIETTINEN, K. Interactive moea/d with multiple types of preference information. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (New York, NY, USA, 2022), ACM, p. 1826–1834.

[75] LI, K., DEB, K., AND YAO, X. R-metric: Evaluating the performance of preference-based evolutionary multiobjective optimization using reference points. *IEEE Transactions on Evolutionary Computation 22*, 6 (2017), 821–835.

[76] LI, M., AND YAO, X. Quality evaluation of solution sets in multiobjective optimisation: A survey. *ACM Computing Surveys 52*, 2 (2019), 1–38.

[77] LIANG, J., BAN, X., YU, K., QU, B., QIAO, K., YUE, C., CHEN, K., AND TAN, K. C. A survey on evolutionary constrained multi-objective optimization. *IEEE Transactions on Evolutionary Computation* (to appear). doi:10.1109/TEVC.2022.3155533.

[78] LIAW, A., AND WIENER, M. Classification and regression by randomforest. *R news 2*, 3 (2002), 18–22.

[79] LIU, B., ZHANG, Q., AND GIELEN, G. G. A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Transactions on Evolutionary Computation 18*, 2 (2013), 180–192.

[80] MARTÍNEZ-FRUTOS, J., AND HERRERO-PÉREZ, D. Kriging-based infill sampling criterion for constraint handling in multi-objective optimization. *Journal of Global Optimization 64*, 1 (2016), 97–115.

[81] MIETTINEN, K. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.

[82] MIETTINEN, K., HAKANEN, J., AND PODKOPAEV, D. Interactive nonlinear multiobjective optimization methods. In *Multiple Criteria Decision Analysis*, S. Greco, M. Ehrgott, and J. R. Figueira, Eds., 2nd ed. Springer, 2016, pp. 927–976.

[83] MIETTINEN, K., RUIZ, F., AND WIERZBICKI, A. P. Introduction to multiobjective optimization: Interactive approaches. In *Multiobjective Optimization: Interactive and Evolutionary Approaches*, J. Branke, K. Deb, K. Miettinen, and R. Slowinski, Eds. Springer, 2008, pp. 27–57.

[84] MIRJALILI, S., RAWLINS, T., HETTENHAUSEN, J., AND LEWIS, A. A comparison of multi-objective optimisation metaheuristics on the 2d airfoil design problem. In *Proceedings of the 16th Biennial Computational Techniques and Applications Conference, CTAC-2012* (2013), S. McCue, T. Moroney, D. Mallet, and J. Bunder, Eds., vol. 54 of *ANZIAM Journal.*, pp. C345–C360.

[85] MOHAMMADI, A., OMIDVAR, M. N., AND LI, X. A new performance metric for user-preference based multi-objective evolutionary algorithms. In *2013 IEEE Congress on Evolutionary Computation, Proceedings* (2013), IEEE, pp. 2825–2832.

[86] NAHARUDINSYAH, I., AND LIMMER, S. Optimal charging of electric vehicles with trading on the intraday electricity market. *Energies 11*, 6 (2018).

[87] OJALEHTO, V., AND MIETTINEN, K. Desdeo: An open framework for interactive multiobjective optimization. In *Multiple Criteria Decision Making and Aiding: Cases on Models and Methods with Computer Implementations*, S. Huber, M. J. Geiger, and A. T. de Almeida, Eds. Springer, 2019, pp. 67–94.

[88] OSYCZKA, A. *Computer Aided Multicriterion Optimization System(CAMOS): Software Package in Fortran.* International Software Publisher, 1992.

[89] OYAMA, A., KOHIRA, T., KEMMOTSU, H., TATSUKAWA, T., AND WATANABE, T. Simultaneous structure design optimization of multiple car models using the k computer. In *2017 IEEE Symposium Series on Computational Intelligence* (2017), pp. 1–4.

[90] PAN, J.-S., LIU, N., CHU, S.-C., AND LAI, T. An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems. *Information Sciences 561* (2021), 304–325.

[91] PARETO, V. *Manuale di economia politica: con una introduzione alla scienza sociale*, vol. 13. Società Editrice Libraria, 1919.

[92] PHAN, D. H., AND SUZUKI, J. R2-ibea: R2 indicator based evolutionary algorithm for multiobjective optimization. In *2013 IEEE Congress on Evolutionary Computation* (2013), IEEE, pp. 1836–1845.

[93] RASMUSSEN, C. E. Gaussian processes in machine learning. In *Summer School on Machine Learning* (2003), Springer, pp. 63–71.

[94] RAY, T., TAI, K., AND SEOW, K. C. Multiobjective design optimization by an evolutionary algorithm. *Engineering Optimization 33*, 4 (2001), 399–424.

[95] RIQUELME, N., VON LÜCKEN, C., AND BARAN, B. Performance metrics in multi-objective optimization. In *2015 Latin American Computing Conference, Proceedings* (2015), IEEE, pp. 1–11.

[96] RODEMANN, T. A many-objective configuration optimization for building energy management. In *2018 IEEE Congress on Evolutionary Computation* (2018), IEEE, pp. 1–8.

[97] RODEMANN, T. A comparison of different many-objective optimization algorithms for energy system optimization. In *Applications of Evolutionary Computation* (2019), P. Kaufmann and P. Castillo, Eds., Springer, pp. 1–16.

[98] ROJAS-GONZALEZ, S., AND VAN NIEUWENHUYSE, I. A survey on Kriging-based infill algorithms for multiobjective simulation optimization. *Computers & Operations Research 116* (2020), 104869.

[99] SACKS, J., SCHILLER, S. B., AND WELCH, W. J. Designs for computer experiments. *Technometrics 31*, 1 (1989), 41–47.

[100] SAINI, B. S., HAKANEN, J., AND MIETTINEN, K. A new paradigm in interactive evolutionary multiobjective optimization. In *Parallel Problem Solving from Nature – PPSN XVI, 16th International Conference, Proceedigs, Part II* (Cham, 2020), T. Bäck, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich, and H. Trautmann, Eds., Springer, pp. 243–256.

[101] SANTIAGO, A., HUACUJA, H. J. F., DORRONSORO, B., PECERO, J. E., SANTILLAN, C. G., BARBOSA, J. J. G., AND MONTERRUBIO, J. C. S. A survey of decomposition methods for multi-objective optimization. In *Recent Advances on Hybrid Approaches for Designing Intelligent Systems* (Cham, 2014), O. Castillo, P. Melin, W. Pedrycz, and J. Kacprzyk, Eds., Springer, pp. 453–465.

[102] SCHAFFER, J. D. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the First International Conference of Genetic Algorithms and Their Application* (1985), pp. 93–100.

[103] SHAHRIARI, B., SWERSKY, K., WANG, Z., ADAMS, R. P., AND DE FREITAS, N. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE 104*, 1 (2015), 148–175.

[104] SHANG, K., ISHIBUCHI, H., HE, L., AND PANG, L. M. A survey on the hypervolume indicator in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation 25*, 1 (2020), 1–20.

[105] TANABE, R., AND ISHIBUCHI, H. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing 89* (2020), 106078.

[106] THIELE, L., MIETTINEN, K., KORHONEN, P. J., AND MOLINA, J. A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation 17*, 3 (2009), 411–436.

[107] TRIVEDI, A., SRINIVASAN, D., SANYAL, K., AND GHOSH, A. A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation 21*, 3 (2016), 440–462.

[108] UNGREDDA, J., BRANKE, J., MARCHI, M., AND MONTRONE, T. One step preference elicitation in multi-objective Bayesian optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (2021), pp. 195–196.

[109] VAN VELDHUIZEN, D. A., AND LAMONT, G. B. On measuring multiobjective evolutionary algorithm performance. In *Proceedings of the 2000 Congress on Evolutionary Computation.* (2000), vol. 1, IEEE, pp. 204–211.

[110] VELDHUIZEN, V. *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. PhD thesis, Air Force Institute of Technology, 1999.

[111] WIERZBICKI, A. P. The use of reference objectives in multiobjective optimization. In *Multiple Criteria Decision Making Theory and Application*, G. Fandel and T. Gal, Eds. Springer, 1980, pp. 468–486.

[112] XIN, B., CHEN, L., CHEN, J., ISHIBUCHI, H., HIROTA, K., AND LIU, B. Interactive multiobjective optimization: A review of the state-of-the-art. *IEEE Access 6* (2018), 41256–41279.

[113] XIN, B., CHEN, L., CHEN, J., ISHIBUCHI, H., HIROTA, K., AND LIU, B. Interactive multiobjective optimization: A review of the state-of-the-art. *IEEE Access 6* (2018), 41256–41279.

[114] XU, Q., XU, Z., AND MA, T. A survey of multiobjective evolutionary algorithms based on decomposition: Variants, challenges and future directions. *IEEE Access 8* (2020), 41588–41614.

[115] YANG, K., LI, L., DEUTZ, A., BACK, T., AND EMMERICH, M. Preference-based multiobjective optimization using truncated expected hypervolume improvement. In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)* (2016), IEEE, pp. 276–281.

[116] YANG, K., LI, L., DEUTZ, A., BACK, T., AND EMMERICH, M. Preference-based multiobjective optimization using truncated expected hypervolume improvement. In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)* (2016), IEEE, pp. 276–281.

[117] YI, J., BAI, J., HE, H., PENG, J., AND TANG, D. ar-moea: A novel preference-based dominance relation for evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation 23*, 5 (2018), 788–802.

[118] YU, G., ZHENG, J., AND LI, X. An improved performance metric for multiobjective evolutionary algorithms with user preferences. In *2015 IEEE Congress on Evolutionary Computation, Proceedings* (2015), IEEE, pp. 908–915.

[119] ZHAN, D., AND XING, H. Expected improvement for expensive optimization: a review. *Journal of Global Optimization 78*, 3 (2020), 507–544.

[120] ZHANG, Q., AND LI, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation 11*, 6 (2007), 712–731.

[121] ZHANG, Q., LIU, W., TSANG, E., AND VIRGINAS, B. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation 14*, 3 (2010), 456–474.

[122] ZITZLER, E., DEB, K., AND THIELE, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation 8*, 2 (2000), 173–195.

[123] ZITZLER, E., LAUMANNS, M., AND THIELE, L. Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report 103* (2001).

[124] ZITZLER, E., AND THIELE, L. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International Conference on Parallel Problem Solving from Nature, Proceedings* (1998), Springer, pp. 292–301.

[125] ZITZLER, E., AND THIELE, L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation 3*, 4 (1999), 257–271.

[126] ZITZLER, E., THIELE, L., LAUMANNS, M., FONSECA, C. M., AND DA FONSECA, V. G. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation 7*, 2 (2003), 117–132.

ORIGINAL PAPERS

I

A SURROGATE-ASSISTED A PRIORI MULTIOBJECTIVE EVOLUTIONARY ALGORITHM FOR CONSTRAINED MULTIOBJECTIVE OPTIMIZATION PROBLEMS

by

Aghaei Pour, P., Hakanen, J., Miettinen, K.

under review

Request a copy from author.

II

DESIRABLE PROPERTIES OF PERFORMANCE INDICATORS FOR ASSESSING INTERACTIVE EVOLUTIONARY MULTIOBJECTIVE OPTIMIZATION METHODS

by

Aghaei Pour, P., Bandaru, S., Afsar, B., Miettinen, K.

in Proceedings of the Genetic and Evolutionary Computation Conference Companion. Edited by J. E. Fieldsend, The Association of Computing Machinery, New York, 1803–1811, 2022.

https://doi.org/10.1145/3520304.3533955

Desirable Properties of Performance Indicators for Assessing Interactive Evolutionary Multiobjective Optimization Methods

Pouya Aghaei Pour
University of Jyvaskyla
Faculty of Information Technology, Finland
poaghaei@jyu.fi

Sunith Bandaru
University of Skövde
School of Engineering Science, Sweden
sunith.bandaru@his.se

Bekir Afsar
University of Jyvaskyla
Faculty of Information Technology, Finland
bekir.b.afsar@jyu.fi

Kaisa Miettinen
University of Jyvaskyla
Faculty of Information Technology Finland
kaisa.miettinen@jyu.fi

ABSTRACT

Interactive methods support decision makers in finding the most preferred solution in multiobjective optimization problems. They iteratively incorporate the decision maker's preference information to find the best balance among conflicting objectives. Several interactive methods have been developed in the literature. However, choosing the most suitable interactive method for a given problem can prove challenging and appropriate indicators are needed to compare interactive methods. Some indicators exist for a priori methods, where preferences are provided at the beginning of the solution process. We present some numerical experiments that illustrate why these indicators are not suitable for interactive methods. As the main contribution of this paper, we propose a set of desirable properties of indicators for assessing interactive methods as the first step of filling a gap in the literature. We discuss each property in detail and provide simple examples to illustrate their behavior.

CCS CONCEPTS

• **General and reference** → **Metrics**; **Performance**; • **Applied computing** → **Multi-criterion optimization and decision-making**.

KEYWORDS

Multiple criteria optimization, Performance evaluation, Performance assessment, Interactive methods

ACM Reference Format:
Pouya Aghaei Pour, Sunith Bandaru, Bekir Afsar, and Kaisa Miettinen. 2022. Desirable Properties of Performance Indicators for Assessing Interactive Evolutionary Multiobjective Optimization Methods. In *Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3520304.3533955

GECCO '22 Companion, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9268-6/22/07…$15.00
https://doi.org/10.1145/3520304.3533955

1 INTRODUCTION

In multiobjective optimization problems, we usually optimize several conflicting objectives simultaneously. This leads to multiple optimal solutions (known as *Pareto optimal solutions*) that are mathematically incomparable [20]. The set of Pareto optimal solutions is referred to as the Pareto front in the objective space.

Multiobjective evolutionary algorithms (MOEAs) are well-known methods for solving multiobjective optimization problems due to their ability to provide an approximation of the Pareto front. In addition, they can handle problems without analytical functions, different types of decision variables, and so on [9, 15]. On the other hand, they cannot guarantee Pareto optimality, but generate an approximation of the Pareto front. However, in most real-world problems, only a single Pareto optimal solution needs to be selected for implementation. Typically, we use the knowledge of a domain expert, also known as a *decision maker* (DM), to provide some kind of preference information. Then, based on the DM's preference information, the most preferred solution is selected.

We can incorporate the DM's preference information for MOEAs in three main ways [13, 20]: 1) *a posteriori* methods, where the DM first sees an approximation of the Pareto front, and then chooses one or more solutions based on her/his preferences, 2) *a priori* methods, where the DM provides the preference information before the solution process, and then, a suitable MOEA tries to generate solutions that reflect the DM's preferences as well as possible and 3) *interactive* methods, where the DM provides her/his preferences iteratively during the solution process and guides the search to find one's most preferred solution in the approximated Pareto front.

A posteriori methods enable the DM to better understand existing trade-offs before expressing preferences. However, generating an approximation of the entire Pareto front is computationally expensive. In addition, it may be overwhelming for the DM to compare many solutions, especially, if we deal with a high number of objectives. A priori methods are usually computationally less expensive than a posteriori methods. However, it may be hard to provide preference information without knowing what kind of trade-offs are feasible. Moreover, besides preference information, most a priori methods require a parameter to identify a parameterized *region of interest* (ROI) which is a part of the approximated Pareto front that the DM is interested in. It is worth mentioning that according to [17], the definition of parameterized ROI is vague, and it can be identified in many different ways.

In interactive methods, the DM has the chance to learn about the trade-offs between objectives during the solution process and identify her/his most preferred solution in the ROI. The ROI is a part of the approximated Pareto front, where the DM likes to fine-tune the preferences and refine solutions. Moreover, unlike a posteriori methods, the DM has to process only a limited amount of information based on her/his preference information, which reduces the cognitive load set on her/him. There are different ways to provide preferences [7, 21]. For instance, in [11], the DM is able to provide his/her preferences in four different ways. Specifying aspiration levels representing desirable objective function values (constituting a so-called *reference point*) is a well-known way of providing preference information. The reference point is a popular way to provide preference information since it has been proven to be understandable to the DM [7, 25].

Many *performance indicators* (or indicators for simplicity) have been developed for a posteriori methods to be able to compare them [5, 23]. They assess the performance in approximating the whole Pareto front. In addition, some indicators have been dedicated to a priori methods [6, 12, 18, 22, 26]. They assess the performance in representing specific parts of the Pareto front identified by preference information provided by a DM.

However, comparing interactive methods has been studied less. Typically, before the DM uses an interactive method, an analyst, who knows the behavior of interactive methods, should choose the most appropriate one. However, there are many aspects that the analyst should consider to be able to choose. To the best of our knowledge, no indicators have specifically been designed for assessing interactive methods.

As the first step towards developing indicators for interactive methods, we must identify desirable properties for such indicators. As the main contribution of this paper, we identify such desirable properties. It is important to note that a single indicator is unlikely to possess all desirable properties. In fact, we suggest that several indicators should be developed for assessing different aspects of interactive methods. To support our motivation, we show that the indicators designed for a priori methods are not suitable to assess interactive methods. However, we do not claim that the list of properties presented in this paper is exhaustive. Our objective is to initiate research in this direction.

In this paper, we first briefly review existing indicators in Section 2. Then in Section 3, we propose the desirable properties that indicators designed for interactive methods should possess, and describe each property in detail. In Section 4, we assess existing indicators against our proposal. Section 5 includes numerical experiments to support our arguments. Finally, we conclude the paper and mention future research directions in Section 6.

2 BACKGROUND

Different indicators have been developed for assessing a priori methods. Their desirable properties are discussed in [22] and the indicators are stated to possess most of the desirable properties. However, some of the indicators require the knowledge of the Pareto front [24, 29] but, according to [22], an indicator should not rely on the knowledge of the Pareto front. In what follows, we briefly describe some recent indicators designed for a priori methods.

In *R-metric* [18], a reference point is incorporated to identify the parameterized ROI. Then, based on an achievement scalarizing function [25], one of the solutions is selected as a pivot point. Next, all solutions are transferred into a virtual position using the pivot point. Finally, the hypervolume [28], or the IGD [8] of the solutions inside the parameterized ROI is calculated as the assessment of an a priori method. In this paper, we use R-metric by calculating the hypervolume (we refer to it as R-HV) and higher values of R-HV represent better performance.

PMOD [12] is a distance-based indicator. The main idea is to map solutions onto a hyperplane generated based on the DM's reference point. Then, three different distances are calculated. First, the distance between the solutions and the reference point, second, the standard deviation of each mapped point to the nearest point (for measuring diversity), and third, the distance of each solution and the origin point is calculated, but if the solutions are outside of parameterized ROI, this value is multiplied by a penalty coefficient. Finally, the PMOD value is calculated by using these three distances (for more details see [12]). For PMOD, lower values represent better performance both in convergence and diversity.

The *preference metric based on distances* [26] (PMDA) indicator is based on light beam search [14] and decomposition-based multi-objective evolutionary algorithm [27]. PMDA has four main steps. First, the reference point is decomposed into k+1 light beams, where k is the number of objectives. Then a preference-based hyperplane is constructed by means of the light beams. Next, the Euclidean distances of solutions to the ideal point are calculated as the main assessment. Following this, angles between solutions outside the parameterized ROI and the reference point are calculated to form a penalty function by multiplying them by a constant coefficient. Finally, the mean of distances and angles for all solutions generate the PMDA assessment of a set of solutions. The lower the value of PMDA, the better it is.

In *the user-preference composite front* (UPCF) indicator [22], first, all the solution sets are merged. Then, all of the nondominated solutions are selected. Next, the closest solution to the reference point is identified, and a parameterized ROI is formed around it by acquiring a parameter that determines the size of the parameterized ROI. Finally, the hypervolume or IGD values for the solutions inside the parameterized ROI are calculated as the final assessment. In this paper, we use only the hypervolume version of UPCF (we refer to it as UPCF-HV). Higher values in UPCF-HV indicate better performance.

The *EH-metric* [6] is a parameterless indicator designed to eliminate the problem of defining parameterized ROI required by the indicators above. Instead of asking the user to define the size of the parameterized ROI through a parameter, this indicator uses the concept of an *expanding hypercube*, which starts as a point at the reference point and expands (with the reference point at its center) until it envelops all solutions. The EH-metric value for an a priori method is calculated as the area under the curve generated by plotting the fraction of solutions enveloped by the hypercube as it expands versus the size of the hypercube. The former is considered to be a measure of diversity around the reference point, while the latter is a measure of convergence to the reference point. Thus, higher EH-metric values indicate good convergence and diversity of preferred solutions.

As mentioned earlier, to the best of our knowledge, there are no indicators for assessing interactive methods. So far, researchers have applied indicators developed for a priori methods (with some adjustments) as the best viable option. For example, in [1, 3] the R-metric has been used in this way.

According to [21], we can often observe two phases in interactive solution processes: a *learning phase* and a *decision phase*. In both of these phases, the provided preferences direct the search to a *desired region*, where interactive methods try to generate solutions. In the learning phase, the DM studies different parts of the Pareto front to increase her/his knowledge about the problem, how well different preferences can be reflected, and learn more about the achievable values for objectives. At the end of the learning phase, the DM is more confident about which part of the Pareto front she/he is interested in and has identified her/his ROI. Here, the DM enters the decision phase, where she/he fine-tunes the search within the ROI until she/he is satisfied with one of the solutions.

One should note that we use the concept ROI with different meanings in different contexts. The ROI in a priori indicators is based on the preferences that the DM provides before the optimization process, whereas in interactive methods, the ROI is identified at the end of the learning phase to be further studied in the decision phase. In addition, we refer here to the act of providing new preference information by the DM as an *interaction*. It happens after every method-specific number of generations.

3 DESIRABLE PROPERTIES

In this section, first, we provide a list of desirable properties for designing indicators suitable for interactive methods. Then, we discuss and describe them in detail. Since these properties are meant to assess different aspects of interactive methods, a good starting point is the list of desirable properties identified for interactive methods, provided in [2]. In that study, the authors divided the desirable properties of interactive methods into three categories. The first category consists of properties that should be considered during the whole solution process, that is, both in the learning and decision phases. These properties are referred to as general properties (GPs). The second set of desirable properties, referred to as LPs, relates to the learning phase. In this phase, the method is supposed to assist the DM in studying the objective space and learning about the different trade-offs to identify a ROI. The third and final set of desirable properties, DPs, relates to the decision phase, where the interactive method is intended to assist the DM in identifying the most preferred solution in the ROI. For more details about the three phases, see [2].

In the same way, we divide desirable properties of indicators for interactive methods into the corresponding categories. Ideally, indicators for interactive methods must be able to:

- GP1: Assess the convergence of solutions in those regions of the approximated Pareto front that reflect the DM's preferences the best (local convergence).
- GP2: Assess the diversity of solutions in those regions of the approximated Pareto front that reflect the DM's preferences the best (local diversity).
- GP3: Assess the performance irrespective of the number of objective functions (scalability).
- GP4: Assess the performance without knowledge of the Pareto front.
- GP5: Assess the performance by incorporating preferences that are provided in different ways.
- GP6: Assess the performance in a computationally inexpensive manner.
- GP7: Assess the performance in a manner that is independent of other interactive methods being compared.
- GP8: Assess the performance without introducing parameters that have an unclear effect on the performance or are unintuitive to set.
- GP9: Assess the performance as a whole process and not as a series of independent a priori steps.
- LP1: Assess how much of the Pareto front has been studied (expedition).
- LP2: Assess how well/fast the method can adapt to new (even very different) preferences (responsiveness).
- DP1: Assess the capability of fine-tuning solutions inside the ROI.
- DP2: Assess the decision phase by considering the amount of information shown to the DM at each interaction.

Next, we discuss each desirable property in more detail. Moreover, for some of the desirable properties, we provide hypothetical examples illustrating their role in designing indicators for assessing interactive methods. We consider two hypothetical interactive methods, I_1, and I_2, and visualize their solutions as red rectangles and orange circles, respectively.

In the provided examples, we use a reference point $\hat{z}$ as preference information. Furthermore, we assume that the DM begins with a learning phase, which is the case in many practical scenarios. After having identified an ROI, the DM moves to the decision phase.

Moreover, the desired region and ROI may be identified differently in each indicator. For simplicity, in the provided examples, we use a cone (green dashed lines) to represent the desired region. In addition, the ROI is represented by a purple box, where we expect the DM would provide her/his reference points in the decision phase. Actually, the ROI is a subset of Pareto optimal solutions.

Figure 1 illustrates an example of how reference points are typically provided in learning and decision phases to reflect different needs. Reference points in the learning phase (denoted by $+$) are often scattered as the DM goes on an expedition to learn more about the Pareto front. On the other hand, reference points in the decision phase (denoted by $\oplus$) have some conformance among them as the DM refines solutions in the ROI identified at the end of the learning phase.

3.1 General Properties

We have nine general desirable properties that should be considered when designing new indicators. These general properties are valid for both learning and decision phases. According to [22], indicators for a priori methods should have four desired properties. We can extend these desirable properties to be applicable in the case of interactive methods. Thus, the first four general properties correspond to those in [22]. In addition, we have formulated five more general properties for designing indicators for interactive methods. These general properties do not depend on specific desires of the DM regarding the learning or decision phases.

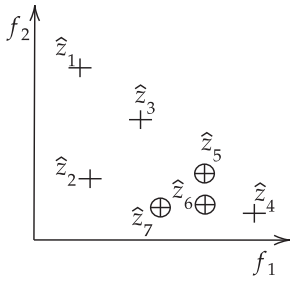
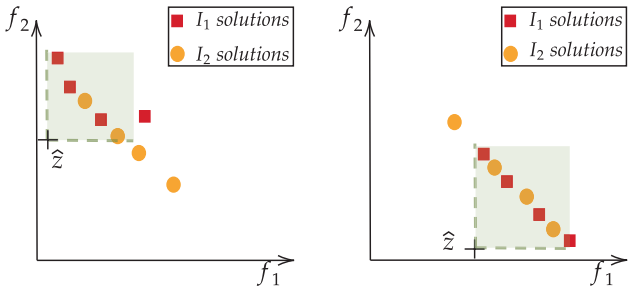


Figure 1: Example of reference points in the learning phase (denoted by +) and the decision phase (denoted by ⊕)

GP1. Convergence following the preferences, which we refer to as *local convergence*, is an important desirable property because with each interaction, whether in the learning phase or the decision phase, the DM expects to see solutions that reflect the preferences. According to [2], this gives the DM the feeling of being in control of the solution process.



(a) Example for showing local convergence.

(b) Example for showing local diversity.

Figure 2: Solutions generated with two hypothetical interactive methods I_1 (red rectangles) and I_2 (orange circles) for the reference point $\hat{z}$ (denoted by +). The desired region is shown by a green dashed cone.

Figure 2a illustrates local convergence. Here, both interactive methods I_1 and I_2 have generated four solutions. If we consider the union of the solutions and eliminate dominated ones, we can observe that I_2 retains all of its solutions, while I_1 loses one. However, since the remaining solutions of I_1 reflect $\hat{z}$ better, an indicator should be able to identify I_1 as a better method.

GP2. An indicator should be able to measure the diversity of solutions reflecting the DM's preference, which we refer to as *local diversity* [22]. This is important because, at each interaction, the DM must have "discernibly distinct" solutions to choose from. A good balance between local convergence and local diversity is required so that the solutions are not too diverse to make the DM feel that the preferences are not being reflected by the interactive method (c.f. GP1).

Figure 2b illustrates local diversity. It is clear that the solutions generated by I_2 are more diverse than those of I_1. We can observe that all solutions of I_1 are in the desired region, while I_2 has generated a solution outside it. We assume that interactive methods show all these solutions to the DM. Therefore, solutions that are outside of the desired region should not be disregarded but should influence the indicator's assessment in a negative way to reflect differences between methods compared.

GP3. Scalability is a desirable property of indicators. For example, when the number of objectives grows, and we cannot even visualize the solutions properly, it will be imperative for the analyst to be able to rely on the indicator when comparing interactive methods.

GP4. If an indicator needs knowledge of the Pareto front, the applicability of the indicator is limited. This is important to keep in mind since the main purpose of interactive methods is solving real-world problems, where we do not know the Pareto front in most cases. Therefore, it is essential that indicators do not depend on this information.

GP5. As mentioned in Section 1, different interactive methods assume preference information to be provided in different ways. Therefore, appropriate indicators are needed. This does not mean that one indicator should be able handle all different ways of providing preferences.

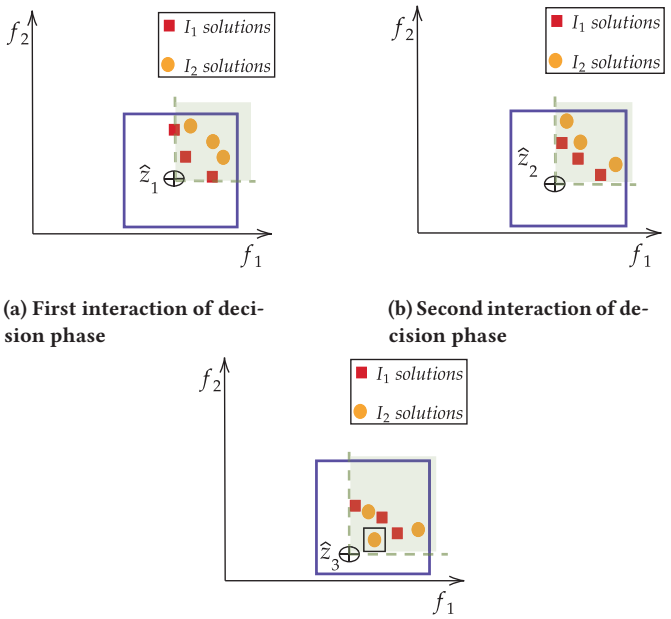
GP6. In general, the calculation of indicator values should be computationally inexpensive. This enables their more versatile usage. For example, one may want to calculate them at regular intervals during the solution process to monitor the progress of an interactive method or compare progress of different methods. For example, if an indicator is based on an inherently expensive computation, such as the hypervolume, the computation time increases exponentially as the number of objectives grows. It is impractical to use such indicators often (e.g., in regular intervals).

GP7. It is desirable that the indicator value for a given interactive method be independent of the other interactive methods being compared. This avoids the problem of recomputing the indicator when a new interactive method needs to be included in the comparison.

GP8. An indicator should be easy to use, not having parameters whose effect on assessing the performance is unclear. For example, many interactive methods have a parameter that identifies the desired region based on the DM's preferences (see e.g., [11]). Here, if the indicator asks for a new parameter to redefine the desired region, the analyst can get confused since she/he has to provide this information twice in different ways.

GP9. In some studies, the performance of interactive methods has been assessed by considering each interaction as a distinct a priori step [1, 3, 19], and indicators for a priori methods have been used to assess the median performance of interactions of each phase. This allows the use of existing (a priori) indicators in the absence of those designed for interactive methods. However, this can mislead assessments since the solution process as a whole and different roles of learning and decision phases are not supported.

Figure 3 illustrates why interactive methods should not be assessed as a series of a priori steps. Figures 3a, 3b, and 3c show three interactions of the decision phase with the two interactive methods I_1 and I_2 in a biobjective minimization problem. It is shown that



(a) First interaction of decision phase

(b) Second interaction of decision phase

(c) Third interaction of decision phase

Figure 3: Three interactions of interactive methods I_1 and I_2 with the reference points $\hat{z}_1$, $\hat{z}_2$, $\hat{z}_3$. The desired region is shown by a green dashed cone, and the ROI is shown by a purple box.

the solutions generated by I_1 in Figures 3a and 3b are better than those of I_2. However, in the third interaction, I_2 manages to find a solution that the DM prefers (orange circle inside a black rectangle). Here, if we calculate the mean of the performances of I_1 and I_2 with most of the indicators developed for a priori methods, I_1 will have a better performance than I_2. However, the most preferred solution was generated by I_2. Therefore, it is important for indicators to consider the interactive methods as more than a series of a priori steps.

Before moving to desirable properties specific for decision and learning phases, it is worth pointing out that whether the DM is in the learning phase or the decision phase is not always obvious. Ideally, the indicator should have a mechanism to detect this transition based on the sequence of DM's preferences and modify its calculations to either suit the learning phase or the decision phase. Alternatively, one can design separate indicators for the two phases.

3.2 Learning Phase

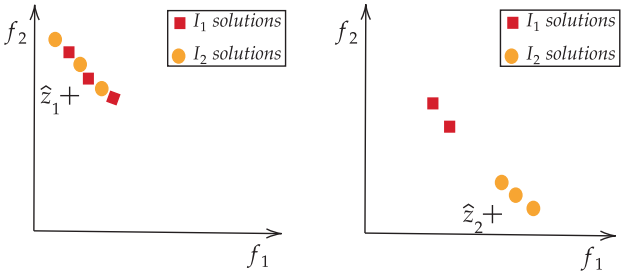
In addition to the general properties, indicators should have specific desirable properties for the learning phase. In this phase, the DM wants to study the objective space to finally identify her/his ROI. Therefore, when designing indicators, we should consider the unique characteristics of the learning phase.

LP1. Measuring the expedition of an interactive method in the learning phase can help the analyst to figure out whether the interactive method has covered the approximation of the Pareto front well enough. Typically in this phase, the DM is not aware of the shape of the Pareto front. So, it is difficult for the DM to say how much expedition she/he has done. This is particularly true in many-objective problems. Moreover, measuring the expedition does not need to be exact since we do not want to rely on the knowledge of the Pareto front (GP4) and it is enough if the indicator can identify different regions of the Pareto front and communicates this information to the analyst.

It is worth mentioning that expedition is not the same as local diversity. The solutions shown to the DM should be diverse within the desired region so that they still reflect the DM's preferences (GP2). On the other hand, expedition is more about an approximation of how much of the approximated Pareto front has been covered by the generated solutions through the learning phase.

LP2. In the learning phase, the DM is still studying the objective space, and therefore her/his preferences may change drastically. As mentioned in [2], responsiveness to these changes is a desirable property for interactive methods in the learning phase. Therefore, in the learning phase, it is desirable for indicators to assess how well an interactive method can adapt to the changes in the preferences.

Moreover, as mentioned in Section 2, one of the main advantages of interactive methods is that they do not need as many function evaluations as a posteriori methods. Besides the responsiveness of an interactive method, measuring how fast it can converge toward new preferences is essential as well because usually the DM has limited time to wait for new solutions to be generated. Therefore, it is important that the interactive method can respond to the new preferences as fast as possible to minimize the waiting time of the DM. For example, we can track how many function evaluations it takes to generate solutions that reflect the new preferences.



(a) First interaction

(b) Second interaction

Figure 4: Two interactions of methods I_1 and I_2. Here, I_2 responds better to the change of preferences (from $\hat{z}_1$ to $\hat{z}_2$).

Figure 4 illustrates an example of the importance of responsiveness. Here, we assume that both interactive methods I_1 and I_2 had the same budget for function evaluations for each interaction. We can observe that in the first interaction (Figure 4a) with the reference point $\hat{z}_1$, both interactive methods had almost similar results. However, when the DM provided the second reference point (Figure 4b), $\hat{z}_2$, the solutions generated by I_2 are closer to $\hat{z}_2$ than what I_1 has generated. In other words, we could say that I_2 is more responsive to the changes in reference points than I_1.

3.3 Decision Phase

As the DM begins the decision phase, new desirable properties dedicated to this phase should be considered. Now, she/he is more interested in fine-tuning solutions in the ROI identified at the end of the learning phase.

DP1. Since the DM refines solutions by providing her/his preferences within the ROI, preferences are likely to be concordant.

However, it is not easy to confirm whether the provided preferences have concordance with each other in problems with more objectives. For example, when we are using reference points in the decision phase, if the new reference point at each interaction in the ROI dominates the old reference point (which is also in the ROI), they are concordant. Otherwise, even if the new reference point dominates the previous one, they are not concordant if the new one is provided outside of the ROI. An indicator must identify the preferences with concordance and increase the role of corresponding solutions in assessing the decision phase.

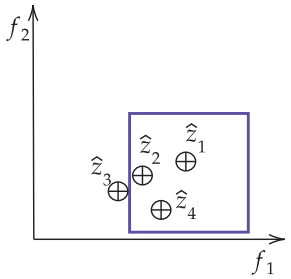


Figure 5: Hypothetical fine-tuning in the decision phase during four interactions. The ROI is represented by a purple box.

Figure 5 shows a simple example of the fine-tuning capability of an interactive method. Here, $\hat{z}_1$, $\hat{z}_2$, $\hat{z}_3$ and $\hat{z}_4$ are the reference points that the DM has provided in successive decision phase interactions. Assume that the most preferred solution is chosen in the last interaction (from solutions corresponding to $\hat{z}_4$). In this example, we can easily observe that $\hat{z}_1$, $\hat{z}_2$ and $\hat{z}_4$ are all inside the ROI (denoted by a purple box), but $\hat{z}_3$ is not. Here, $\hat{z}_4$ dominates $\hat{z}_1$ and therefore, they are concordant. However, there is no concordance between $\hat{z}_4$, $\hat{z}_2$, and $\hat{z}_3$ because $\hat{z}_2$ is not dominated by $\hat{z}_4$, and $\hat{z}_3$ is outside the ROI.

Moreover, since the most preferred solution is chosen from the fourth interaction, $\hat{z}_4$ must play the most significant role in assessing the decision phase. An indicator should be able to measure the concordance between the DM's preferences in the decision phase. Here, each interaction does not influence the assessment equally, Thus, this approach is different from seeing interactive methods as a series of a priori steps where all interactions have an equal effect on the assessment of the performance.

DP2. Another important aspect of interactive methods is the amount of information shown to the DM at each interaction. In relation to GP5, the amount and nature of information depends on the way preference information is provided. In the learning phase, the amount of information required by the DM to learn about the shape of the Pareto front can vary as long as the cognitive load is acceptable to the DM. However, in the decision phase, a typical requirement from a DM is the number of solutions that she/he wishes to analyze within the ROI [20]. An interactive method that generates fewer solutions than what the DM desires may delay the solution process, whereas one that generates more solutions may increase the cognitive load on the DM. In essence, the desires of the DM should be respected. An indicator should be able to take both these aspects into account.

4 APPLICABILITY OF EXISTING INDICATORS

As mentioned earlier, since there are no indicators in the literature designed specifically for comparing interactive methods, some studies [1, 3, 4] have resorted to using indicators developed for a priori methods. In this section, we assess the five a priori indicators presented in Section 2 with respect to the desirable properties discussed in the previous section.

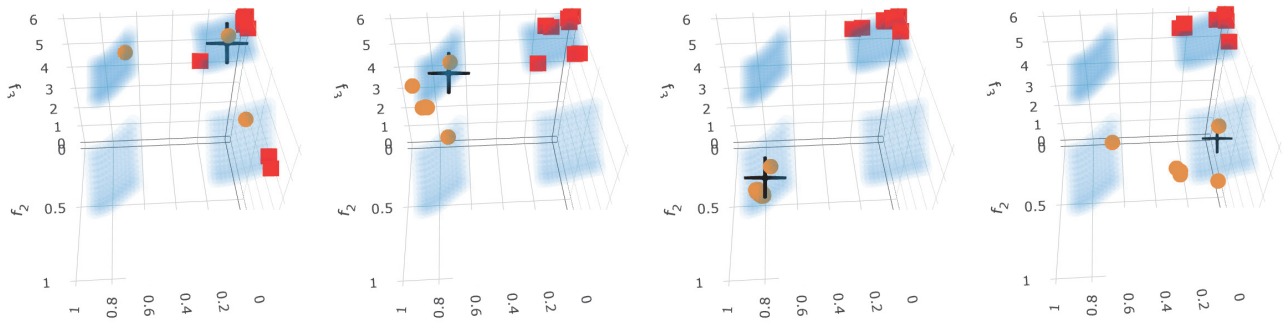
Table 2 shows how well the five indicators stack against the desirable properties. All of the indicators satisfy the first four desirable properties concerning local convergence and local diversity, scalability and knowledge of the Pareto front (GP1, GP2, GP3 and GP4). However, for GP1 and GP2, some of the indicators like R-HV and UPCF-HV remove solutions outside the desired region before calculating GP1 and GP2. This may be misleading since the DM sees these solutions. In other words, if the interactive method presents some solutions outside the desired region (or the ROI), then these solutions should have a negative effect in assessing the method (instead of being deleted).

All the indicators consider reference point(s) as the preference information. Hence, none of the indicators satisfy the desirable property GP5. Moreover, R-HV and UPCF-HV do not satisfy GP6, since they are based on computationally expensive calculations (hypervolume). Here, as the number of objectives increases, the computation time of these indicators grows exponentially, which is not desirable in interactive methods.

Among the indicators we listed in Table 2, R-HV, EH-metric and UPCF-HV do not satisfy GP7. They employ a prescreening step which combines and sorts the final solutions from all methods being compared. Thus, the values of these indicators depend on the methods being compared. If a new method is to be compared, these values may need to be recomputed (for more details see [6, 18, 22]).

Except for EH-metric, the rest of the indicators need an analyst to set at least one parameter. For example, R-HV, PMOD, PMDA and UPCF-H require the size of parameterized ROI. In addition, PMOD and PMDA require a penalty coefficient for the solutions outside the parameterized ROI. It may be confusing for the analyst to provide these parameters values. None of the studies have analyzed the effect of these parameters on their assessment of the performance.

The rest of the desirable properties are understandably not satisfied by any of the indicators, because they were not designed to consider the learning and decision phases. We have already mentioned that there could be separate indicators dedicated to assess learning and decision phases. In addition, it may be even too difficult to design an indicator that satisfies all the desirable properties in the learning or decision phases. Hence, we may need several indicators for different purposes in each phase.



(a) interaction 1 (b) interaction 2 (c) interaction 3 (d) interaction 4

Figure 6: DTLZ7 in the learning phase where we provided a reference point in each part of the Pareto front manually.

Table 1: Indicator values for iRVEA and iNSGA for 3-objective DTLZ7 problem in the learning phase. Here, ↑ means that higher values are better for the corresponding indicator, and ↓ means that lower values are better. Bold values indicate that the corresponding interactive method has a better performance.

	iRVEA					iNSGA				
	R-HV ↑	EH-metric ↑	PMOD ↓	PMDA ↓	UPCF-HV ↑	R-HV ↑	EH-metric ↑	PMOD ↓	PMDA ↓	UPCF-HV ↑
interaction 1	5.927	0.276	**6.543**	**4.632**	3.437	**6.669**	**0.561**	7.888	5.050	**4.310**
interaction 2	**5.859**	**0.569**	**5.713**	**4.248**	**3.985**	5.401	0.419	6.626	5.092	2.546
interaction 3	**7.301**	**0.688**	**5.418**	**3.226**	**4.015**	6.489	0.121	5.833	4.940	2.135
interaction 4	**5.909**	**0.482**	**5.690**	**4.245**	**3.874**	2.930	0.196	6.392	4.893	2.497

Table 2: Proposed desirable properties and their presence in existing indicators for a priori methods.

Properties	R-HV	EH-metric	UPCF-HV	PMOD	PMDA
GP1	✓	✓	✓	✓	✓
GP2	✓	✓	✓	✓	✓
GP3	✓	✓	✓	✓	✓
GP4	✓	✓	✓	✓	✓
GP5	✗	✗	✗	✗	✗
GP6	✗	✓	✗	✓	✓
GP7	✗	✗	✗	✓	✓
GP8	✗	✓	✗	✗	✗
GP9	✗	✗	✗	✗	✗
LP1	✗	✗	✗	✗	✗
LP2	✗	✗	✗	✗	✗
DP1	✗	✗	✗	✗	✗
DP2	✗	✗	✗	✗	✗

5 NUMERICAL EXPERIMENTS

In this section, we present two numerical examples to emphasize the need for developing new indicators specifically designed for interactive methods. We demonstrate that the current practice of assessing interactive methods as a series of a priori steps is inappropriate. The first example shows the importance of desirable properties for the learning phase (LP1 and LP2). The second example focuses on the importance of DP1 in the decision phase and GP9 of the general properties. We compare interactive NSGAIII (iNSGA) [3] and interactive RVEA (iRVEA) [11] using DTLZ benchmark problems [10]. For both examples, the number of generations is limited to 100 per interaction, while the number of function evaluations for both methods is 100000. These numbers are examples as our goal here is not to find the best method but to study the behavior of the indicators.

Since using a priori indicators for comparing interactive methods is not the main focus of this study, we have provided the details of this part in the Supplementary Material.[1].

5.1 An Example for the Learning Phase

Here, we use the 3-objective 11-variable (number of variables is based on [16]) DTLZ7 problem due to the unique shape of its Pareto front, which has four disconnected regions as shown by the blue areas in Figure 6. These distinct regions enable demonstrating the importance of measuring the expedition (LP1) and responsiveness (LP2) in the learning phase. The following reference points corresponding to each region were used in successive interactions to test the expedition capability of iRVEA and iNSGA (a) [0.11, 0.10, 5.4], (b) [0.70, 0.14, 4.50], (c) [0.76, 0.76, 3.5], and (d) [0.14, 0.70, 4.5].

Figure 6 shows the solutions that iNSGA (red rectangles) and iRVEA (orange circles) have generated corresponding to different reference points. We can observe that iNSGA could not respond to the changes of the reference point and stayed in one region. On the other hand, iRVEA could provide solutions in the same region with the reference point. Therefore, for expedition (LP1) and responsiveness (LP2), iRVEA was better than iNSGA in this example.

The assessments of indicators for a priori methods for each interaction have been gathered in Table 1. We can observe that most of the indicators declare that iRVEA is better than iNSGA

[1]Link to the implementation: https://github.com/ppouyaa/desirable-properties-master

Table 3: Mean indicator values of iRVEA and iNSGA for 5-objective DTLZ3 problem in the decision phase. As before, ↑ means that higher values are better for the corresponding indicator, and ↓ means that lower values are better. Bold values indicate that the corresponding interactive method has a better performance.

	iRVEA					iNSGA				
	R-HV ↑	EH-metric ↑	PMOD ↓	PMDA ↓	UPCF-HV ↑	R-HV ↑	EH-metric ↑	PMOD ↓	PMDA ↓	UPCF-HV ↑
interaction 1	14.145	0.475	**5.835**	**2.888**	0.185	**32.212**	**0.590**	8.533	3.329	**0.195**
interaction 2	24.263	0.619	**6.405**	**1.430**	0.203	**32.253**	**0.643**	9.265	3.329	**0.262**
interaction 3	26.126	**0.766**	13.813	**0.704**	0.311	**31.956**	0.744	**12.160**	3.329	**0.347**
interaction 4	28.699	**0.768**	14.950	**0.687**	0.412	**31.957**	0.741	**12.160**	3.329	**0.458**

in the last three interactions. However, based on these values, we cannot get the essential information that iNSGA was stuck in the initial region and could not cover the Pareto front well. In fact, by looking at these values, the analyst may be misled to think that the performance of these interactive methods is not that different (e.g., see PMOD or PMDA values). Therefore, it is essential that the indicators can communicate such important insight to the analyst.

5.2 An Example for the Decision Phase

In this section, we compare iRVEA and iNSGA using the 5-objective DTLZ3 problem in the decision phase. Thus, we assume that the ROI has already been identified. To generate reference points, we used an artificial decision maker [3]. These reference points are: (a) [0.000, 0.000, 0.000, 3.072], (b) [0.000, 0.000, 0.000, 0.000, 1.951], (c) [0.000, 0.000, 0.000, 0.000, 1.010], and (d) [0.000, 0.000, 0.000, 0.000, 1.010]. We can observe that the artificial decision maker changed the reference point for the first three interactions, but at the last interaction used the same reference point as the previous one. After generating the reference points, we ran each interactive method ten independent times. Then, we calculated the average of results for each interaction (see Table 3).

According to Table 3, EH-metric indicates that for the first two interactions, iNSGA was better than iRVEA, while for the third and fourth interactions, iRVEA was better than iNSGA. Earlier, we mentioned that in the current literature, the mean of indicators values for each interaction is typically calculated to find the best interactive methods. Here, if we calculate the mean of the EH-metric values, iRVEA has the value of 0.656, and iNSGA has the value of 0.679. Therefore, based on this way of calculation, iNSGA is better than iRVEA.

However, earlier, we noted that one of the desirable properties in the decision phase is fine-tuning solutions in the ROI (DP1). This involves information about the concordance of reference points. Moreover, the reference points did not change in the last two interactions (where iRVEA had a better performance). If we consider the concordance of reference points and let solutions corresponding to the third and fourth interaction influence the results more, iRVEA could probably be regarded to have a better performance than iNSGA. This shows that it is important to have specifically designed indicators for the decision phase (or the learning phase). Besides, this example shows why it is important not to assess the interactive methods as a series of a priori steps (GP9).

Finally, we can observe in Table 3 that the indicators are not similar for each interaction. For example, based on R-HV, iNSGA was better than iRVEA at every interaction. However, based on PMDA, iRVEA was better than iNSGA at every interaction. In addition, EH-metric indicates that at the first two interactions, iNSGA was better, and for the third and fourth interactions, iRVEA was better. On the other hand, PMOD gave opposite results to EH-metric. This is interesting since most of these indicators were designed to calculate local convergence and local diversity, and still, the results are so different from one another. Thus, if these indicators are used to assess interactive methods, the results may be sensitive to the choice of the indicator. This supports the need of indicators designed specifically for assessing interactive methods.

6 CONCLUSIONS

In this paper, we identified the desirable properties for designing indicators suitable for interactive methods and discussed them in detail. There are three main categories for these desirable properties. The general properties that should be considered in both the learning and decision phase, the desirable properties regarding the learning phase, and the decision phase properties that focus on aspects of interactive methods that help the DM refine a solution. Together, we suggested 13 desirable properties that indicators designed for interactive methods should possess. However, one indicator cannot satisfy all the desirable properties, and there should be different indicators for different purposes.

We also demonstrated why indicators developed for a priori methods should not be applied for interactive methods. We showed that these indicators do not satisfy most of the desirable properties that we presented. Furthermore, we provided two numerical examples to support the claim that there is a need for indicators specifically designed for assessing interactive methods.

By assessing interactive methods, we can analyze their characteristics and choose the appropriate method for different real-world problems. Therefore, as a future research direction, we plan to develop indicators that satisfy at least most of the desirable properties we presented in this paper. Moreover, these desirable properties only consider algorithmic aspects of interactive methods. It is also important to study interactive methods from human perspectives such as cognitive load set on the DM.

ACKNOWLEDGMENTS

This research was partly supported by the Academy of Finland (grant no 311877 and 322221) and is related to the thematic research area DEMO (Decision Analytics utilizing Causal Models and Multiobjective Optimization, jyu.fi/demo) of the University of Jyväskylä. The authors like to thank Dr. Michael Emmerich for providing valuable comments.

REFERENCES

[1] Afsar, B., Miettinen, K., and Ruiz, A. (2021a). An artificial decision maker for comparing reference point based interactive evolutionary multiobjective optimization methods. In Ishibuchi, H., Zhang, Q., Cheng, R., Li, K., Li, H., Wang, H., and Zhou, A., editors, *Evolutionary Multi-Criterion Optimization, 11th International Conference, EMO 2021, Proceedings*, pages 619–631, Cham. Springer.

[2] Afsar, B., Miettinen, K., and Ruiz, F. (2021b). Assessing the performance of interactive multiobjective optimization methods: A survey. *ACM Computing Surveys (CSUR)*, 54(4):1–27.

[3] Afsar, B., Ruiz, A., and Miettinen, K. (2021). Comparing interactive evolutionary multiobjective optimization methods with an artificial decision maker. *Complex & Intelligent Systems*. to appear.

[4] Aghaei Pour, P., Rodemann, T., Hakanen, J., and Miettinen, K. (2022). Surrogate assisted interactive multiobjective optimization in energy system design of buildings. *Optimization and Engineering*, 23:303–327.

[5] Audet, C., Bigeon, J., Cartier, D., Le Digabel, S., and Salomon, L. (2021). Performance indicators in multiobjective optimization. *European Journal of Operational Research*, 292(2):397–422.

[6] Bandaru, S. and Smedberg, H. (2019). A parameterless performance metric for reference-point based multi-objective evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 499–506. ACM.

[7] Bechikh, S., Kessentini, M., Ben Said, L., and Ghédira, K. (2015). Chapter four - preference incorporation in evolutionary multiobjective optimization: a survey of the state-of-the-art. In Hurson, A. R., editor, *Advances in Computers*, volume 98, pages 141–207. Elsevier.

[8] Bosman, P. A. and Thierens, D. (2003). The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188.

[9] Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons.

[10] Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2002). Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02*, volume 1, pages 825–830. IEEE.

[11] Hakanen, J., Chugh, T., Sindhya, K., Jin, Y., and Miettinen, K. (2016). Connections of reference vectors and different types of preference information in interactive multiobjective evolutionary algorithms. In *Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8.

[12] Hou, Z., Yang, S., Zou, J., Zheng, J., Yu, G., and Ruan, G. (2018). A performance indicator for reference-point-based multiobjective evolutionary optimization. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI), Proceedings*, pages 1571–1578. IEEE.

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

[14] Jaszkiewicz, A. and Słowiński, R. (1999). The 'light beam search'approach–an overview of methodology applications. *European Journal of Operational Research*, 113(2):300–314.

[15] Jin, Y., Wang, H., and Chaoli., S. (2021). *Data-Driven Evolutionary Optimization*. Springer.

[16] Knowles, J. (2006). ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66.

[17] Li, K., Chen, R., Min, G., and Yao, X. (2018). Integration of preferences in decomposition multiobjective optimization. *IEEE Transactions on Cybernetics*, 48(12):3359–3370.

[18] Li, K., Deb, K., and Yao, X. (2017). R-metric: Evaluating the performance of preference-based evolutionary multiobjective optimization using reference points. *IEEE Transactions on Evolutionary Computation*, 22(6):821–835.

[19] Mazumdar, A. (2021). *Novel Approaches for Offline Data-Driven Evolutionary Multiobjective Optimization*. JYU Dissertations 456, University of Jyväskylä.

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

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

[22] Mohammadi, A., Omidvar, M. N., and Li, X. (2013). A new performance metric for user-preference based multi-objective evolutionary algorithms. In *2013 IEEE Congress on Evolutionary Computation, Proceedings*, pages 2825–2832. IEEE.

[23] Riquelme, N., Von Lücken, C., and Baran, B. (2015). Performance metrics in multi-objective optimization. In *2015 Latin American Computing Conference, Proceedings*, pages 1–11. IEEE.

[24] Van Veldhuizen, D. A. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Air Force Institute of Technology.

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

[26] Yu, G., Zheng, J., and Li, X. (2015). An improved performance metric for multi-objective evolutionary algorithms with user preferences. In *2015 IEEE Congress on Evolutionary Computation, Proceedings*, pages 908–915. IEEE.

[27] Zhang, Q. and Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731.

[28] Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.

[29] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132.

III

A PERFORMANCE INDICATOR FOR INTERACTIVE EVOLUTIONARY MULTIOBJECTIVE OPTIMIZATION METHODS

by

Aghaei Pour, P., Bandaru, S., Afsar, B., Emmerich, M., Miettinen, K.

under review

Request a copy from author.

IV

SURROGATE ASSISTED INTERACTIVE MULTIOBJECTIVE OPTIMIZATION IN ENERGY SYSTEM DESIGN OF BUILDINGS

by

Aghaei Pour, P., Rodemann, T., Hakanen, J., Miettinen, K.

Optimization and Engineering, 23, 303–327 (2022)

https://doi.org/10.1007/s11081-020-09587-8

Optimization and Engineering
https://doi.org/10.1007/s11081-020-09587-8

REVIEW ARTICLE

Surrogate assisted interactive multiobjective optimization in energy system design of buildings

Pouya Aghaei Pour[1] · Tobias Rodemann[2] · Jussi Hakanen[1] · Kaisa Miettinen[1]

Received: 20 August 2020 / Revised: 4 December 2020 / Accepted: 4 December 2020

Abstract

In this paper, we develop a novel evolutionary interactive method called interactive K-RVEA, which is suitable for computationally expensive problems. We use surrogate models to replace the original expensive objective functions to reduce the computation time. Typically, in interactive methods, a decision maker provides some preferences iteratively and the optimization algorithm narrows the search according to those preferences. However, working with surrogate models will introduce some inaccuracy to the preferences, and therefore, it would be desirable that the decision maker can work with the solutions that are evaluated with the original objective functions. Therefore, we propose a novel model management strategy to incorporate the decision maker's preferences to select some of the solutions for both updating the surrogate models (to improve their accuracy) and to show them to the decision maker. Moreover, we solve a simulation-based computationally expensive optimization problem by finding an optimal configuration for an energy system of a heterogeneous business building complex. We demonstrate how a decision maker can interact with the method and how the most preferred solution is chosen. Finally, we compare our method with another interactive method, which does not have any model management strategy, and shows how our model management strategy can help the algorithm to follow the decision maker's preferences.

Keywords Model management · Evolutionary interactive methods · Surrogate-assisted optimization · Multiobjective optimization · Computationally expensive problems

✉ Pouya Aghaei Pour
pouya.p.aghaei-pour@student.jyu.fi

Extended author information available on the last page of the article

Published online: 05 January 2021

1 Introduction

Real-world optimization problems often contain multiple conflicting objective functions, and we call them multiobjective optimization problems (MOPs). In MOPs, instead of having one optimal solution, we have many so-called Pareto optimal solutions with different trade-offs. Mathematically, all of the Pareto optimal solutions are equally good if no additional information is available since vectors cannot be ordered completely. However, one of the Pareto optimal solutions needs to be selected as the outcome of the optimization process to be implemented. Here, we need an expert known as the decision maker (DM) who knows the properties of the problem and can provide preferences and compare different Pareto optimal solutions.

Based on the literature (see, e.g., Miettinen 1999; Hwang and Masud 1979), the DM can participate in solving MOPs in three different ways. In a priori methods, the DM expresses one's preferences before the solution process. In the second category, a posteriori methods, the DM selects the final solution after the method provides a set of solutions representing different trade-offs. In the third category, the DM actively interacts with the algorithm and provides preferences during an iterative solution process. In the literature, the last type is referred to as interactive methods.

By using interactive multiobjective optimization methods that involve a DM's preference information, the DM directs the solution process to the regions that one is interested in. A solution pattern is repeated iteratively, and information is provided to the DM at each iteration, who then needs to provide preferences in order to improve solutions from the current iteration. There are many interactive methods in the literature that use different types of preferences (see, e.g., Miettinen 1999; Miettinen et al. 2016). Using interactive methods can be beneficial in the process of problem-solving because as mentioned by Miettinen (1999):

1. The DM learns about the interdependencies between the conflicting objectives and the feasibility of one's preferences.
2. The algorithm focuses on those parts of the objective space that are interesting to the DM.

Moreover, since the DM's understanding of the problem grows during the optimization process, one will have more confidence in the final selection.

There exist several types of methods to solve a MOP (see e.g., Miettinen 1999; Deb 2001). One of the well-known methods is evolutionary multiobjective optimization (EMO) algorithms. EMO algorithms are population-based a posteriori methods where a set of solutions approximating the actual Pareto optimal solutions, is found (Deb 2001).

Over the years, EMO algorithms have become popular due to certain advantages. For example, they can provide a set of representative solutions in one run, they can handle different kinds of decision variables (Deb 2001), and they can be applied to objective functions or constraints that are discontinuous or non-differentiable. Many EMO algorithms have been proposed (see, e.g, Deb 2001;

Branke et al. 2008). However, usually, evolutionary algorithms need a considerable number of function evaluations. Recently, some interactive EMO algorithms have been developed, where the DM provides preferences iteratively during the solution process to get a set of solutions that is the most preferable (for reviews, see Wang et al. 2017; Xin et al. 2018; Purshouse et al. 2014).

Real-world multiobjective optimization problems may involve functions that do not have any analytic formulation. For instance, when we are dealing with simulation-based problems (Rodemann 2019; Cheng et al. 2017), one only gets output for a given input. Then, in some cases, we can use the output directly as the values of the objective functions, and sometimes some post-processing analysis on the output data is needed to calculate the values of the objective functions. Calculating the output may be time-consuming, and such problems are known as computationally expensive multiobjective optimization problems. EMO algorithms are viable for simulation-based problems since we do not necessarily know the properties of the functions involved, but their need for many function evaluations makes solution processes time-consuming.

In this paper, we focus on finding an optimal configuration for the energy system design of buildings, as formulated by Rodemann (2019). The usage of local energy production and storage facilities has become increasingly interesting both in terms of energy costs and CO_2 emissions. Facility management is, therefore, looking at how to invest in extensions to the current building energy system optimally. Here a simulator is used that has a time-consuming process to generate the outcome (Rodemann 2019).

Even though interactive methods have desirable properties, applying them in computationally expensive problems is not straightforward since the DM must wait for solutions corresponding to one's preferences to be generated, which can take hours. Waiting too long may become exhausting for the DM, and this is why it is desirable to speed up the calculation in such problems. One way to reduce the computing time is to approximate the objective functions by analytic functions. In the literature, this is known as surrogate (meta-model)-assisted optimization (see e.g., Jin 2011; Chugh et al. 2019).

As far as we know, there has been no attempt to tackle the problem addressed by Rodemann (2019) by any interactive methods. Besides, there are only few interactive evolutionary methods in the literature that are suited for computationally expensive problems. Therefore, we develop an interactive method that is suitable for solving computationally expensive multiobjective optimization problems, like the one addressed by Rodemann (2019), to show how it provides decision support for the DM in computationally expensive problems. Moreover, there are some algorithms in the literature that motivated our novel interactive method. The first algorithm is the reference vector guided evolutionary algorithm (RVEA) (Cheng et al. 2016) since it has got good results in similar simulation-based problems like the one presented by Cheng et al. (2017). The second algorithm is the surrogate assisted version of RVEA (K-RVEA) presented by Chugh et al. (2018) where the Kriging models (Sacks et al. 1989) have been used to reduce the computation time. The final method that inspired us is the

interactive version of RVEA (Hakanen et al. 2016) in which RVEA is modified to be able to incorporate the DM's preferences.

Typically, in surrogate-assisted optimization problems, model management (i.e., how to select solutions to evaluate with a computationally expensive function) is used to improve the accuracy of the surrogate models with updating them. Model management is a very crucial part of surrogate-assisted optimization. For instance, solutions computed by the surrogate functions might deviate substantially from the true values, and it is desirable to find the solutions that are following the DM's preferences when they are evaluated by the original objective functions. A good model management strategy can help the surrogate models to make such selection.

The contributions of this paper are two-fold. First, we develop a novel model management strategy that has a smart selection process, where the solutions, which are generated by the surrogate models, will be examined and the ones that have the highest chance of following the DM's preferences are selected to be shown to the DM and update the surrogate models. The second contribution is to show how model management can help an interactive method to follow the DM's preferences better than when there is no model management involved. In other words, we show that by reserving some of the computational resources that we have available for updating the surrogate models, we can provide several solutions that reflect the DM's preferences well.

The rest of this paper is structured as follows. In Sect. 2, the energy system design problem is briefly described, along with relevant background information. In Sect. 3, we present a new interactive method for solving computationally expensive problems. In Sect. 4, we solve the problem presented in Sect. 2 with our new interactive method and demonstrate the importance of having a model management strategy with some comparisons. Finally, conclusions are drawn and future research directions mentioned in Sect. 5.

2 Background

Next, we provide some background about notation and terminology, the energy management problem we consider, and the supporting materials for developing our new interactive method.

2.1 Terminology and notation

The general form of a multiobjective optimization problem (for minimization) is as follows:

$$\begin{aligned} &\text{minimize} \ \ \{f_1(x), f_2(x), \ \ldots, f_k(x)\} \\ &\text{subject to } x \in S, \end{aligned} \tag{1}$$

where the set S is called the feasible region which is a subset of the decision space $\mathbb{R}^n$. We consider k($\geq$ 2) objective functions $f_i : S \to \mathbb{R}$. For every feasible decision variable vector x, there is a corresponding objective vector $f(x) = (f_1(x), \ldots, f_k(x))^T$,

and $f(S)$ is called the feasible objective region which is a subset of the objective space $\mathbb{R}^k$.

As mentioned earlier in Sect. 1, usually, the objective functions in problem (1) conflict with each other. Hence, not all the objective functions can achieve their optimal values simultaneously. A feasible solution $x^* \in S$ and the corresponding $f(x^*)$ are called Pareto optimal, if there does not exist another feasible solution $x \in S$ such that $f_i(x) \leq f_i(x^*)$ for all $i = 1, \ldots, k$, and $f_j(x) < f_j(x^*)$ for at least one index j. The set of all Pareto optimal objective vectors is called a Pareto front (PF). A feasible solution $x^* \in S$ and the corresponding $f(x^*)$ are called weakly Pareto optimal, if there does not exist another feasible solution $x \in S$ such that $f_i(x) < f_i(x^*)$ for all $i = 1, \ldots, k$.

Assume that the set $X = \{x^1, \ldots, x^m\}$ is an arbitrary subset of feasible solutions in S, and $F = \{f(x^1), \ldots, f(x^m)\}$ the corresponding objective vectors in the objective space. A solution x^i for $i = 1, \ldots, m$ that satisfies the definition of Pareto optimality within the set X, is called a nondominated solution in X (Miettinen 1999). Note that sometimes in the EMO literature, Pareto optimality and nondominance are regarded as synonyms, but this is a more precise distinction. By definition, a Pareto optimal solution is always nondominated but not necessarily vice versa.

In this paper, we have two important concepts, iteration, and interaction. By an iteration, we mean a fixed number of generations, and in this paper, we update the surrogate models at the end of each iteration. Whenever the DM provides preferences, we call it an interaction, and it happens after a fixed number of iterations. For simplicity, every time we evaluate a decision variable vector with the surrogate models, we refer to it as a surrogate evaluation, and every time we use the original expensive objective functions, we use the term function evaluation.

In the method to be proposed, we use an achievement scalarizing function (ASF) proposed by Wierzbicki (1980) to order nondominated solutions based on a given reference point $\hat{z}$. It consists of aspiration levels $\hat{z}_i$ $(i = 1, \ldots, k)$ provided by the DM. There are different ways to formulate an ASF. Here, we use the following formulation to be minimized:

$$\max_{i=1,\ldots,k} [w_i(f_i(x) - \hat{z}_i)] + \rho \sum_{i=1}^{k} w_i(f_i(x) - \hat{z}_i), \tag{2}$$

where k is the number of objective functions, w is some weighting vector with positive fixed values, and $\rho \sum_{i=1}^{k} w_i(f_i(x) - \hat{z}_i)$ with $\rho > 0$ is the augmentation term to avoid finding weakly Pareto optimal solutions (Miettinen 1999).

In this paper, we use an ASF as an indicator of how well a given solution is following the DM's preferences (given as a reference point). The lower the ASF value for a given x, the better it is following the DM's preferences (Wierzbicki 1980).

2.2 Simulation-based problem considered

Managers of large buildings are confronted with complex investment decisions concerning possible extensions of the energy system, like photovoltaics, stationary

batteries, or heat storage. They have to consider a multitude of objectives, for example, investment and annual operation costs and CO_2 emissions.

Here, we want to find an optimal configuration for an energy system of a heterogeneous business building complex. Because of the complex nature of the problem, it is possible to consider different numbers of objective functions and decision variables. For example, the problem considered by Rodemann (2019) consisted of five objective functions and ten decision variables, and a building simulator based on Modelica Fritzson and Bunus (2002); Yang and Wang (2012) was used, which is capable of modeling the most relevant real-world effects. Several EMO algorithms were applied to solve this problem (Rodemann 2019). However, no analysis of the final set of solutions was done to determine the DM's most preferred solution. This can be a difficult task since the DM has to choose a solution from a big pool of solutions with different trade-offs.

We have ten real-valued decision variables (see Appendix for more details) whose values are given to the same simulator that was used by Rodemann (2019) as input. Here, we consider four objective functions:

f_1: minimize initial investment cost (in euros),
f_2: minimize annual operation cost (in euros),
f_3: minimize annual CO_2 emissions (in tons), and
f_4: maximize resilience (in seconds),

where resilience is defined as the time the facility can run without grid power. Here, f_1 is independent of the simulator and it is computationally cheap to calculate $f_1(x)$. On the other hand, the other objective functions are computationally expensive, and we need to post-process the simulator's output to calculate them (for more details, see Rodemann 2019).

We formulate our multiobjective optimization problem as:

$$\begin{aligned} &\text{minimize } \{f_1(x), f_2(x), f_3(x)\}, \\ &\text{maximize } \{f_4(x)\} \\ &\text{subject to } 0 \leq x_i \leq 1, i = 1, \ldots, 10, \end{aligned} \tag{3}$$

where f_i for $i = 2, \ldots, 4$ are derived from the output of the simulator and x_i for $i = 1, \ldots, 10$ are the decision variables which only have box-constraints. In what follows, we consider and solve problem (3).

2.3 Related Work

As we mentioned in the previous section, our method is inspired by RVEA, K-RVEA, and interactive version of RVEA. Here, we provide some background on these algorithms.

2.3.1 RVEA

RVEA (Cheng et al. 2016) is a decomposition-based algorithm which divides the objective space into a number of subspaces using reference vectors. The reference vectors are initially generated so that they are uniformly distributed in the feasible objective space, and they are adjusted within the algorithm based on the structure of the PF. RVEA balances between the diversity of the solutions and the convergence towards Pareto optimality by using an angle penalized distance (APD) scalarization (Cheng et al. 2016) to select solutions from different subspaces for the next generation.

RVEA has three main steps. First, generating a set of uniformly distributed reference vectors to divide the objective space to a number of subspaces. Second, using a heuristic algorithm to find solutions in the created subspaces. Third, assigning the solutions found in the previous step to the reference vectors by using APD and then adjusting the positions of reference vectors based on those solutions.

2.3.2 K-RVEA

As mentioned in Sect. 1, it takes much time to solve a computationally expensive problem with EMO algorithms. A widely used approach for solving computationally expensive problems is to use surrogate functions to approximate the original ones (Jin 2011; Chugh et al. 2019). A surrogate-assisted version of RVEA called K-RVEA was proposed by Chugh et al. (2018). K-RVEA assumes that all the objective functions are computationally expensive, and uses Kriging (also known as Gaussian process regression) as a surrogate model. The main idea of Kriging is to predict the values of a function for a given decision variable vector by generating weighted coefficients of the true values of the function in the neighborhood of the decision variable vector. Typically, the computation time for training the Kriging models in population-based EMO is quite high and there might be a need for a model management strategy to limit the size of the training samples like the one mentioned by Chugh et al. (2018).

A major difference between K-RVEA and RVEA is that in RVEA, the final population is examined to measure the quality of solutions. However, in K-RVEA, an archive is used to store all the function evaluations, and in the end, the solutions that are stored in the archive are examined to determine the quality of the solutions.

K-RVEA consists of three main steps. First, in the initialization step, a sampling method is used to create a training data set in the decision space. Then, the collected samples are evaluated with the original objective functions, and the data, which is stored in an archive, is used to train a surrogate model for each objective function. Second, RVEA is run with the surrogate models instead of the original objective functions. Third, the surrogate models are updated after a certain number of generations by using both APD and uncertainty information, which is provided by the Kriging models (see Chugh et al. 2018 for more details).

2.3.3 Interactive RVEA

As mentioned earlier, in interactive methods, the DM guides the algorithm to find one's most preferred solution by providing preference information. There are many types of preferences, for example, reference points, classification, pairwise comparisons, and selecting preferred solutions, see, e.g., (Miettinen 1999; Hwang and Masud 1979). An interactive version of RVEA, to be referred to as iRVEA, was proposed by Hakanen et al. (2016). In iRVEA, the preference information given by the DM is used to adjust reference vectors $V = \{v^1, \ldots, v^m\}$ so that the search focuses on solutions reflecting the preferences. For example, if the DM provides a reference point $\hat{z} = (\hat{z}_1, \ldots, \hat{z}_k)$, an adjusted reference vector $\bar{v}^i$ is created from v^i by the following formula (Hakanen et al. 2016):

$$\bar{v}^i = \frac{r \cdot v^i + (1 - r) \cdot v^c}{||r \cdot v^i + (1 - r) \cdot v^c||}, \tag{4}$$

where $i = 1, \ldots, k$, $||\hat{z}|| \geq 0$ is the Euclidean norm of the reference point which is used for normalization, and $v^c_j = \frac{\hat{z}_j}{||\hat{z}||}$. If $||\hat{z}|| = 0$, then it means that all the objective functions have the same amount of desirability, and we can use the unit vector as the reference vector. The parameter $r \in (0, 1)$ controls how much the reference vectors are adjusted towards the reference point. If r is close to 1, then the reference point has less effect on the reference vectors, and if it is close to 0, they will get closer to the reference point.

3 Interactive K-RVEA

We selected RVEA as the EMO algorithm that we use in our interactive method (called interactive K-RVEA) because it had reasonable results in similar problems (Rodemann 2019; Cheng et al. 2017). Moreover, we used Kriging models because they provide uncertainty information that is useful for our model management strategy. Kriging models have been used with a priori EMO algorithms before (Chugh et al. 2018) to approximate the whole PF. However, to the best of our knowledge, they have never been used to incorporate the DM's preferences to focus on particular regions of the objective space. To consider Kriging models when applying interactive methods, we must incorporate DM's preferences in model management, which has some challenges. Here, the main point of our model management strategy is that it improves the ability of the method to follow the preferences with respect to (2).

Figure 1 presents a flowchart of the main steps of interactive K-RVEA. First, we generate the initial population, evaluate it using the original objective functions, and train a Kriging model for each expensive objective function. Next, the DM provides preferences, and we solve a multiobjective optimization problem (by incorporating the preferences) by replacing original objective functions with the Kriging models. After generating an approximation of a part of the Pareto optimal set reflecting preferences, the accuracy of the Kriging models must be improved to get a better approximation. We propose a model management strategy based on the DM's

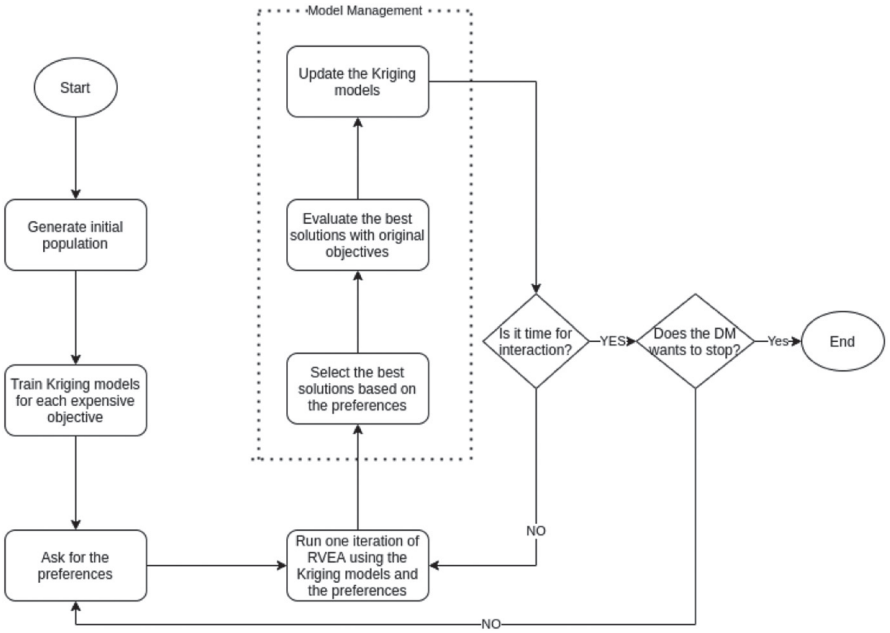


Fig. 1 Flowchart of interactive K-RVEA

preferences to update the Kriging models, which is done by selecting solutions that follow the DM's preferences best. The solutions for updating the Kriging models must be evaluated with the original objective functions. Based on how many solutions the DM wants to see at a time, we show to the DM the corresponding number of solutions reflecting the preferences among those evaluated by the original objective functions. Finally, if the DM is satisfied, he/she selects the most preferred solution and the algorithm stops.

As mentioned earlier, there are only few interactive methods that are suited for computationally expensive problems. In this section, we use Kriging models to reduce the computation time and RVEA as an EMO algorithm to build the basis of a new interactive method called interactive K-RVEA. The main contribution to developing interactive K-RVEA is a model management strategy to incorporate the DM's preferences while using the Kriging models.

We have two main steps in developing interactive K-RVEA. First, we must select the type of preferences that the DM is expected to provide, and second, we must select some of the solutions that are found by using Kriging models in a way that when they are evaluated by the original objective functions, they follow best the DM's preferences (at least they are following the DM's preferences better than other available solutions). For the first task, we mentioned in Sect. 2 that there exist different ways to express one's preferences for interactive methods. After consulting with experts, who deal with problem (3) regularly, we decided to use a reference point to develop our model management strategy because it is intuitive, and they were comfortable with this kind of preference information. Reference vectors could

be adapted based on other types of preference information as done by Hakanen et al. (2016), if so desired.

As for the second step, we have to select the solutions that have the highest chance of following the DM's preferences when they are evaluated with the original objective functions. When a solution is evaluated with the original objective functions, it may have different values than with the surrogate models because surrogates tend to contain some approximation error. Besides, evaluating all the solutions that the Kriging models find is not computationally efficient, especially in cases that some of these solutions are not following the DM's preferences. For example, due to the error of surrogate models, a surrogate evaluation of a given decision variable vector could follow the DM's preferences much better (lower ASF value) than when it is evaluated by the original objective functions. Therefore, these kinds of solutions may not be interesting to the DM, and it is ideal to avoid them. Furthermore, in problems like (3), we usually have a particular budget for the number of function evaluations, and it should be spent carefully on the solutions that have a higher probability of following the DM's preferences.

To increase our chances of selecting the best possible solutions for updating the Kriging models, we use two criteria. First, we use ASF to calculate how close each of the nondominated solutions, which are found by using the Kriging models, are to the DM's reference point. Then, we sort the solutions based on the ASF values, and we select $2 * N_U$ solutions (N_U is the number of solutions to update the Kriging models) that are the closest to the DM's preferences. In other words, we select the solutions that have the lowest values in ASF.

So far, we have selected some solutions which have the lowest ASF value. However, since Kriging models provide uncertainty information, we use this additional information as our second criterion. Typically, when the uncertainty information of generated solutions is available, those which have the highest uncertainty are chosen to improve the accuracy of the Kriging model globally (Chugh et al. 2018). However, in interactive methods, we are looking to search specific parts of the objective space that the DM has shown interest in. Therefore, after selecting the solutions that have lower ASF value, we select N_U solutions among those that have the lowest uncertainty values to update the Kriging models. By incorporating the DM's preferences in the model management strategy along with the uncertainty information, we increase our chances to select the solutions that are following the DM's preferences, both with the Kriging models and the original objective functions. Algorithm 3 shows the main steps of the interactive K-RVEA algorithm, which are discussed in more detail in the following subsections.

Algorithm 1: Interactive K-RVEA

Input: N_V = number of reference vectors; N_0 = the population size; t_{max} = maximum number of generations; N_U = number of solutions to update the Kriging models with; FE_{max} = maximum number of function evaluations; N_{update}= number of updates between each interaction; N_S = number of solutions to be shown to the DM at each interaction (provided by the DM); $\bar{N}_S$ = number of solutions from N_S that are closest to the reference point (provided by the DM);

Output: Most preferred solution selected by the DM

/* Initialization * /;

1 Generate initial population of size N_0 and store them in A. Initialize an empty archive U to store the solutions and decision variables for updating the Kriging models. Number of solutions selected by ASF, $N_{ASF} = 2 \times N_U$;

2 Evaluate initial population with the original objective functions and store the objective values in A and set number of function evaluations $FE = N_0$;

3 Train a Kriging model for each expensive objective function by using A;

4 Ask the DM if he/she wants to see all the nondominated solutions in A, just the ranges, or continue the algorithm without any additional information;

5 Ask DM to provide a reference point;

6 Generate N_V uniformly distributed unit reference vectors V and adjust them using (4) ;

/* outer loop */;

7 **while** $FE < FE_{max}$ **do**

/* middle loop */;

8 Initialize a counter $c_u = 0$ for the number of updates;

9 **while** $c_u < N_{update}$ **do**

10 Initialize t = 0;

/*inner loop */;

11 **while** $t < t_{max}$ **do**

12 Run steps 3-6 of Algorithm 1 with Kriging models and update $t = t + 1$;

13 Select N_{ASF} solutions from the final population with the lowest ASF value;

14 Choose N_U solutions with the lowest uncertainty from previous step, evaluate the selected solutions with the original objective functions, and store them (and their corresponding decision variables) in A and U;

15 Re-train the Kriging models by using the samples in A;

16 Update $FE = FE + N_U$ and $c_u = c_u + 1$;

17 Select N_S solutions from U with the lowest uncertainty values and show them to the DM;

18 Indicate $\bar{N}_S$ closest solutions based on ASF if the DM wants;

19 if the DM is able to select the most preferred solution, go to step 25;

20 Ask the DM to provide a reference point, adjust V using (4), and set $U = \emptyset$;

21 **if** *DM increased* FE_{max} **then**

22 Go to step 7;

23 **else**

24 Show the nondominated solutions in archive A to the DM, and ask the DM to indicate the most preferred solution;

25 END;

3.1 Inputs

The first input for interactive K-RVEA is the number of reference vectors N_V. In RVEA, the method called simplex-lattice design method (Cornell 2011) is used to generate a given number of reference vectors. In RVEA, as the number of objective functions increases, the number of reference vectors increases as well. For instance, for a problem with three objective functions, 105 reference vectors were used by Cheng et al. (2016). In iRVEA, on the other hand, a lower number of reference

vectors was used compared to RVEA (Hakanen et al. 2016). For example, for a problem with five objective functions, only 15 reference vectors were used. The reason for choosing a low number of reference vectors in iRVEA is that there is no model management to select the solutions that the algorithm finds, and all of them are shown to the DM. Therefore, if the number of reference vectors increases, the number of solutions that the DM sees will increase as well, and the cognitive load set on the DM grows.

In interactive K-RVEA, we develop a model management strategy that enables the algorithm to choose the solutions that the DM is most interested in. Here, we are not limited to a low number of reference vectors. In fact, we are more interested in increasing the size of N_V because we will have more solutions to choose from, and the chance of finding solutions that follow the DM's preferences increases. Besides, surrogate evaluations are computationally cheap, and therefore, we do not need to worry about the number of solutions that are found by using the Kriging models.

The number of generations (t_{max}) and the number of solutions to update Kriging models (N_U) can be set based on the sensitivity analysis by Chugh et al. (2018). The number of updates between each interaction (N_{update}) can be set based on how much time it takes to evaluate N_U solutions with original objective functions. Since FE_{max} is based on N_{update} and N_U, we can use the following formulas to calculate an estimation of FE_{max}

$$FE_{int} = N_U * N_{update}, \tag{5}$$

and

$$FE_{max} = N_0 + \mu * FE_{int}, \tag{6}$$

where FE_{int} is the number of function evaluations that we need for one interaction, and μ is the estimation of the number of interactions that the DM wants to have.

3.2 Initialization

Before the DM starts interacting with the algorithm, the Kriging models should be trained with an initial population. The size of the initial population (N_0) should be set based on the type of problem that we are dealing with and the function evaluation budget that we have. Moreover, since the algorithm has no preferences at the beginning, the Kriging models should be trained globally. Therefore, the initial population (P_0) is generated by using a method (e.g., using Latin hypercube sampling used by McKay et al. 2000). These samples are evaluated by the original objective functions, and then they are stored in the archive A (along with their corresponding decision variables). Then, the samples in A are used to train independent Kriging models for each expensive objective function.

After training the Kriging models, it is time for the DM to set the first reference point. If the DM does not have information about the problem to be confident about her/his preferences, then, we provide three alternatives to the DM. First, to see all the nondominated solutions in the initial population. Second, to see only the ranges of each objective function for the nondominated solutions in the initial population.

Third, to proceed without any further information. The purpose of the first two alternatives is to give some idea to the DM of the feasible solutions and speed up the learning process. However, one should note that no optimization has been done in this stage, and this information is not accurate enough to represent the trade-offs between different objective functions. Finally, after the DM provides the first reference point, the reference vectors are adjusted by using (4) to focus on the regions that the DM is interested in.

3.3 Loops

In Algorithm 3, we have three main loops. The inner loop runs RVEA, the middle loop updates the Kriging models after each iteration, and the outer loop interacts with the DM after each interaction.

In the middle and outer loops, we mostly focus on the model management strategy that was mentioned earlier in this section. As it was mentioned earlier, because Kriging models are not completely accurate, it is possible that some of the solutions that are found are not appealing to the DM. In these two loops, we identify and select the solutions which have the highest chance of following the DM's preferences when they are evaluated with the original objective functions. Then, we use the selected solutions to update the Kriging models.

3.3.1 Inner loop

In the inner loop, we use Kriging models to replace original objective functions. We run RVEA with the Kriging models for a fixed number of generations (t_{max}), and this parameter should be set high enough so that RVEA can perform a sufficient search of the Pareto optimal set.

3.3.2 Middle loop

In the middle loop, we select the solutions that we want to evaluate with the original objective functions to update the Kriging models. The selected solutions should improve the Kriging models in regions that the DM is interested in. Here, we manage the solutions that are found by the Kriging models in two phases. In the first phase, we select a number of solutions (N_{ASF}) that are following the DM's preferences while using the Kriging models. If the solutions are not close to the DM's preferences even with the Kriging models, then our selection will involve too much randomness, and the model management becomes unstable. In the second phase, we use the uncertainty information that Kriging provides to select the most accurate solutions (solutions with the lowest uncertainty) from the previously selected solutions and store them in U and A to update the Kriging models. Based on our tests, Kriging models can properly approximate the objective functions of problem (3) (see Appendix). However, the surrogate models have inevitably some error and by going through the two phases mentioned, we increase the probability of selecting solutions that are following the DM's preferences.

3.3.3 Outer loop

Unlike iRVEA, where the number of solutions shown to the DM (N_S) is the same as the number of reference vectors, here N_S is an independent parameter defined by the DM. Once the Kriging models are updated, ASF is used to select N_S solutions from U, and then they are shown to the DM. Then, the DM has the option of separating the best solutions (with respect to (2)) generated in the current iteration visually ($\bar{N}_S$). Next, either the DM decides to finish the solution process by selecting the most preferred solution or set a new reference point to search for more preferred solutions. At the end of this loop, we reset U to the empty set to prepare it for the next interaction. Note that if $N_S > N_U$, then the algorithm cannot provide enough solutions to be shown to the DM, and all the solutions in N_U are shown to the DM.

These three loops keep running until the function evaluation budget runs out, or the DM terminates the algorithm by finding the most preferred solution. In the first case, if the budget of function evaluations runs out and the DM is not satisfied, he/she can either increase FE_{max}, or as the final alternative (step 24), the DM can ask to see all the nondominated solutions that have been generated so far, which are stored in archive A. Then, one can use visualization tools, such as parallel coordinate plots, to study these solutions, or to provide new value to $\bar{N}_S$ to see the closest solutions to the final reference point visually, and then select the most preferred solution from there.

In the next section, we use interactive K-RVEA to solve problem (3). Besides, we show how the model management strategy that we proposed can provide better decision support for the DM by comparing our algorithm with iRVEA.

4 Numerical results

Here we describe how we can design an energy system for buildings by using interactive K-RVEA. In what follows, we first describe how we set the parameters of interactive K-RVEA, and then how the DM can interact with this algorithm to solve problem (3). We also incorporate visualizations to support the DM in providing preferences and comparing solutions. To show the results, we used the web-based parallel coordinate plots tool https://dgoldri25.github.io/Categorical-ParallelCoordinatePlot/.

For parameters that are shared between K-RVEA and interactive K-RVEA such as the number of generations before each iteration ($t_{max} = 20$), the number of samples to update the Kriging models with ($N_U = 5$), and the number of reference vectors ($N_V = 109$), we used the same values that have been used when the K-RVEA algorithm was proposed by Chugh et al. (2018). Furthermore, determining the number of iterations before each interaction is one of the important parameters. According to private discussions with experts in the domain of problem (3), DMs should not wait more than three minutes before each interaction. Each time we call the simulator, it takes about ten seconds, and since we update the models with five new solutions (c.f. N_U above), each update takes about one minute (including the training time). Consequently, to have at most three minutes waiting time before each interaction, we can

update the models three times ($N_{update} = 3$). Based on Step 21 of Algorithm 3, the DM can increase the maximum number of function evaluations (FE_{MAX}) or terminate the algorithm at any time. Here, we need 109 function evaluations to generate the initial population ($FE_{init} = 109$), and based on equation (5), we set $FE_{min} = 15$. Due to the time limitation that we had, we decided to have six interactions ($\mu = 6$), and hence, based on Eq. (6), we set $FE_{MAX} = 199$.

The number of solutions that the DM wants to see at each interaction (N_S) is the next parameter that must be set. As we mentioned above, we update the Kriging models three times before we ask for a new reference point, and it means that we can show a maximum of 15 solutions to the DM in one interaction. Here, the DM decided to see all of the solutions that interactive K-RVEA finds in each interaction ($N_S = 15$).

As mentioned in Sect. 2, in problem (3), calculating the outcome of the first objective function (initial investment cost) is not computationally expensive. Therefore, we use Kriging models only for the other three objective functions. Note that based on discussions with real DMs, one of the authors (TR) provided feedback on presented solutions similar to what we would expect from a real DM.

4.1 Interactive solution process

To get started, we generated the initial population randomly and trained Kriging models for expensive objective functions. Then, the DM was asked to provide the first reference point. To support the DM in providing the first reference point, interactive K-RVEA has different options (c.f. step 4). First, he asked to visually see nondominated solutions of the initial population (see Fig. 2). Note that the solutions provided in Fig. 2 are nondominated solutions from the random initial population, which have not yet been optimized, and they can only give a rough idea of feasible solutions. In addition to the visualization, the DM can naturally always see the numerical values of the selected solutions (N_S) in the form of a table at each interaction. However, in this paper, we only show the parallel coordinate plots during the interactive solution process for compactness. Note that the figures in this section have different scales so that the changes between the solutions can be better seen.

Here, based on the objective functions' ranges shown in Fig. 2, and the prior knowledge that f_1 and f_3 (initial investment cost and CO_2 emission) are regarded as the most important objective functions, the DM sets $RP_1 = (298806, 377430, 2194, 28)$ as the first reference point since he believes it

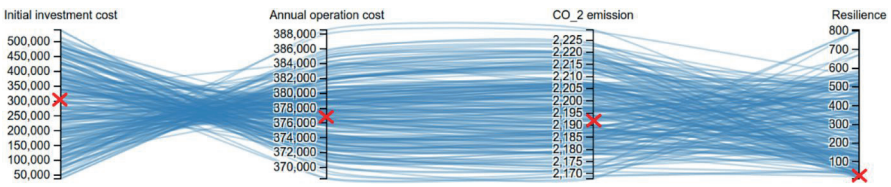


Fig. 2 The nondominated solutions in the initial population. Red crosses are the aspiration levels forming the first reference point

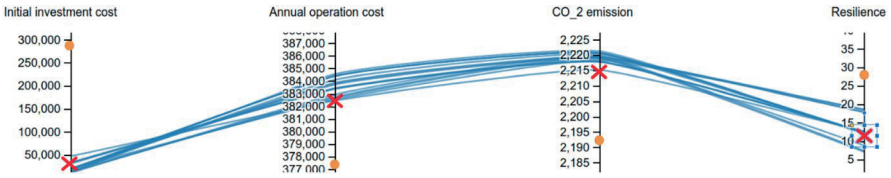


Fig. 3 Solutions after the first interaction of interactive K-RVEA. The orange dots are the aspiration levels forming the first reference point, and the red crosses are the aspiration levels forming the second reference point.

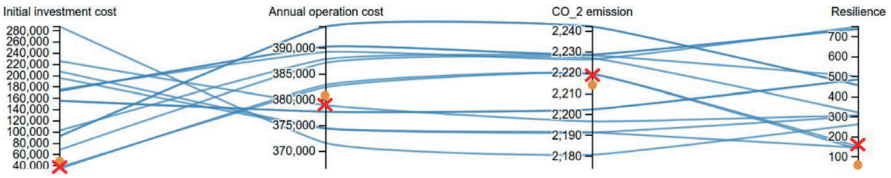


Fig. 4 Solutions after the second interaction of interactive K-RVEA. The orange dots are the aspiration levels forming the second reference point, and the red crosses are the aspiration levels forming the third reference point

is a good compromise for f_1. Components of the reference point are indicated by red crosses in Fig. 2. Based on the solutions that were generated after providing RP_1 (see Fig. 3), the DM provides $RP_2 = (47950, 382509, 2215, 12)$ as the second reference point because the values of f_1 for the generated solutions are all in this range and he also wants to improve the trade-offs between f_1 and the rest of objective functions.The corresponding aspiration levels are depicted in Figure 3 with red crosses and the previous aspiration levels with orange dots.

Next, the solutions in Fig. 4 were generated and presented to the DM. This time, the generated solutions are well spread at around RP_2. However, the trade-offs between f_1 and the rest of the objective functions still are not satisfying. The DM decides not to make a significant change in the reference point to continue searching this region of objective space. He chooses $RP_3 = (37192, 382426, 2219, 152)$ as the third reference point (denoted by red crosses in Fig. 4) because based on the generated solutions he knows such a solution is achievable, and it is quite cheaper (it has smaller value for f_1) than RP_2 and it only produces a little more CO_2 than RP_2.

Figure 5 shows the solution set that was generated after the third interaction. Now, the DM finds out that the aspiration level for f_1 in RP_2 and RP_3 is too small, and therefore, the trade-offs cannot improve significantly. As for the fourth reference point, the DM makes a compromise and sets $RP_4 = (156067, 377696, 2202, 500)$ to find a more balanced solution.

Figure 6 shows the the results corresponding to RP_4. Here, the DM was satisfied with the trade-offs and selects (149886, 380764, 2211, 561) as the most preferred solution since it has the same trade-offs as RP_4 but with lower value for f_1.

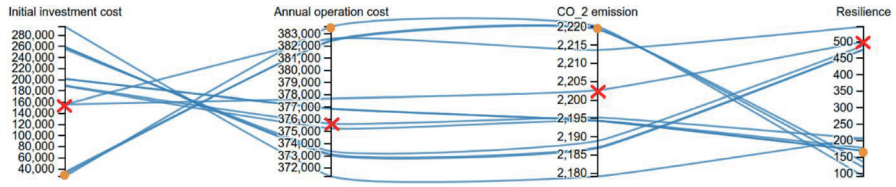


Fig. 5 Solutions after the third interaction of interactive K-RVEA. The orange dots are the aspiration levels forming the third reference point, and the red crosses are the aspiration levels forming the third reference point

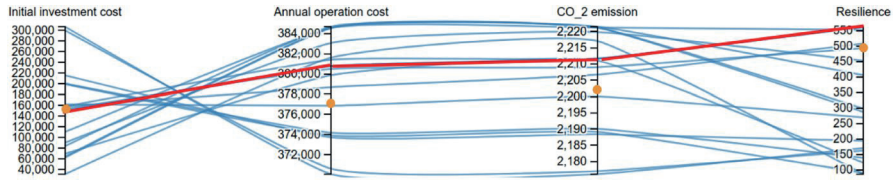


Fig. 6 Solutions after the fourth interaction of interactive K-RVEA. The orange dots are the aspiration levels forming the fourth reference point, and the red line is the most preferred solution selected by the DM

4.2 Performance evaluation

As mentioned earlier, in this paper, we show how using model management in surrogate models can incorporate DM's preferences in an interactive method to get satisfactory solutions. To show the importance of model management strategies used in this paper, we applied iRVEA with the Kriging models as objective functions and compared the results. However, comparing interactive methods is not a trivial task in the field of multiobjective optimization, and there is no widely accepted way for this.

We used the same reference points (RP_1, RP_2, RP_3, and RP_4) that were used in interactive K-RVEA. Note that interactive K-RVEA used 60 function evaluations to update the Kriging models, and since iRVEA does not update them, we increased the size of the initial population by 60 to have the same number of function evaluations as interactive K-RVEA. Next, we evaluated the final solutions that iRVEA generated with the original objective functions and present the nondominated ones in Fig. 7. The final set of solutions generated by iRVEA are more scatter than interactive K-RVEA around the final reference point (RP_3) in Fig. 5. Finally, the DM chooses (367142, 380138, 2273, 45) as the final solution since it has the best compromise between the objective functions.

None of the final solutions dominate each other. However, the final solution for interactive K-RVEA has better values than iRVEA for f_1, f_3, and f_4 objective function and only slightly worst value for f_2.

To compare interactive K-RVEA and iRVEA in terms of following the DM's preferences, we ran both algorithms with the same configuration ten times and used three different ways (ASF, domination and R-metric Li et al. 2017) to evaluate their

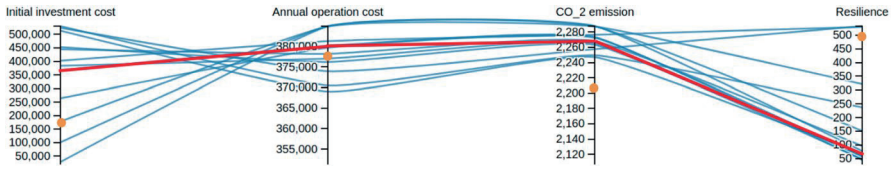


Fig. 7 The final solutions of iRVEA. The orange dots are the aspiration levels forming the fourth reference point, and the red line is the most preferred solution selected by the DM

performance. Experiments were run on a laptop with core i7 CPU, using 16 GB of RAM, and the running OS was Linux (Ubuntu).

4.2.1 Computation time

In Table 1, we present the total computation times for both algorithms without considering the decision making time. Interactive K-RVEA and iRVEA included the same number of function evaluations. However, interactive K-RVEA had the model management, where Kriging models were updated. On the other hand, iRVEA used all the function evaluations for the initial population and the solution process only used the surrogate evaluations. Therefore, the computation time for interactive K-RVEA was a bit higher than for iRVEA.

As far as waiting time is concerned, we updated the Kriging models iteratively in interactive K-RVEA. On the other hand, there was no update for iRVEA, and therefore, the waiting time of iRVEA was shorter. However, the waiting time for both methods was under three minutes, that met the DM's time limitation.

4.2.2 ASF

We recorded the ASF values for the final set of solutions (see Table 2) to measure how close they were to the final reference point. In all of the independent runs, interactive K-RVEA had lower ASF values than iRVEA, which means that interactive K-RVEA had a better convergence towards DM's preferences than iRVEA.

Table 1 Average of computation time of interactive K-RVEA and iRVEA between interactions (in seconds). The best results are highlighted in boldface

Algorithm	min	mean	max
Interactive K-RVEA	520	552	575
iRVEA	**511**	**535**	**567**

Table 2 ASF values for the 10 independent runs with interactive K-RVEA and iRVEA. The best results are highlighted

	Best	Mean	Worst
Interactive K-RVEA	**0.41**	**0.53**	**0.59**
iRVEA	0.71	0.77	0.82

4.2.3 Domination

Here, we checked to see if iRVEA solutions dominate the final set of solutions generated by interactive K-RVEA. In all ten runs, none of the solutions provided by iRVEA dominated any of the solutions that were generated by interactive K-RVEA. However, this was not the case when we checked the inverse situation. In other words, in all ten runs, we could find at least one solution generated by iRVEA that was dominated by one or multiple solutions that interactive K-RVEA generated. In Table 3, we show how many of the final solutions of iRVEA were dominated by the final solutions of interactive K-RVEA for ten independent runs.

Moreover, we merged all the solutions generated in the ten independent runs for both methods and checked how many nondominated solutions were generated with each method. Furthermore, iRVEA had 108 nondominated solutions and dominated only seven solutions generated by interactive K-RVEA. On the other hand, interactive K-RVEA had 117 nondominated solutions and dominated 31 solutions that were generated by iRVEA. The number of nondominated solutions generated by interactive K-RVEA is still more significant than iRVEA, which shows that the model management strategy used in interactive K-RVEA helps the method provide more nondominated solutions than iRVEA.

As we showed in Fig. 7 and Table 2, the solutions generated by iRVEA were more scattered than by interactive K-RVEA, which means that interactive K-RVEA followed the DM's preferences better than iRVEA. Besides, when the DM interacts with interactive K-RVEA, all the solutions that he works with are evaluated with the original objective functions, but when the DM interacts with iRVEA, the solutions are evaluated by the Kriging models. Hence, the DM cannot be sure that when the final set of solutions (generated with iRVEA) is evaluated with the original objective functions, it will follow the DM's preferences and before (when it was evaluated with surrogate functions).

4.2.4 R-metric

Finally, we used a well-known R-metric indicator, which evaluates the quality of a set of solutions with respect to a reference point. Originally, R-metric was developed for a priori methods to compare different sets of solutions, but since it includes a reference point, we apply it for the final set of solutions of interactive K-RVEA and iRVEA. To compare two sets of solutions, R-metric takes four main steps. First, we remove the common solutions between the two sets. Second,

Table 3 iRVEA final solutions that are dominated by interactive K-RVEA

	Number of dominated solutions by the other algorithm		
	Best	Mean	Worst
Interactive K-RVEA	**0**	**0**	**0**
iRVEA	7	4.7	2

Table 4 Results of R-metric for interactive K-RVEA and iRVEA. The best results are highlighted

Δ	Interactive K-RVEA				iRVEA		
	Best	Mean	Worst		Best	Mean	Worst
$1.10 * RP_4$	**0.81**	**0.71**	**0.65**	↑	0.23	0.11	0.00
$1.15 * RP_4$	**0.88**	**0.79**	**0.72**	↑	0.28	0.19	0.12
$1.20 * RP_4$	**0.98**	**0.90**	**0.82**	↑	0.42	0.35	0.27

based on the closeness of solutions to DM's reference point (Δ), we remove some of the solutions that do not represent the region of interest in the objective space. Third, we transfer the solutions into a virtual position concerning the reference point using ASF, and finally, we use an indicator like hypervolume to evaluate the quality of the solutions. For details, see Li et al. (2017).

For the second step of R-metric, we must set a value for Δ. Initially, the value of Δ is set as an arbitrary number by Li et al. (2017). However, since there does not exist a widely accepted way to set this value, we decided to analyze the results with three different values of Δ with respect to the last reference point (RP_4), and create a vector for Δ, representing separate exploration rates for each objective function. Here, we add 10, 15, and 20 percent to the aspiration levels of RP_4 to create the vector Δ. Note that we remove the solutions that are exceeding Δ in at least one objective function. We calculated the R-metric by using the hypervolume indicator for each method's ten independent runs, normalized the hypervolume values, and present the results in Table 4. Moreover, a pairwise two-tailed t-test (Derrac et al. 2011) was conducted between the two interactive methods for the R-metric results. The significance level of our testing was set at %5. In Table 4, ↑ indicates that the statistical significance of the pairwise comparison between interactive K-RVEA and iRVEA is significant in favor of interactive K-RVEA.

As it is shown in Table 4, interactive K-RVEA is performing better than iRVEA. Table 4 shows that for the first value of Δ, iRVEA might generate zero solutions (for the worst case), which means none of the solutions generated by iRVEA were in the region determined by Δ. Moreover, for the first and second values of Δ, interactive K-RVEA is getting much higher R-metric values than iRVEA, which shows that more solutions are generated by interactive K-RVEA that are concentrating on the regions around RP_4. In addition, for the third value of Δ, iRVEA's performance gets much better than the previous values of Δ, which is in line with the fact that solutions are generated with iRVEA are more scattered than interactive K-RVEA. However, interactive K-RVEA is still obtaining much higher R-metric values than iRVEA. We did not continue with higher values of Δ since we wanted to analyze how each method can generate solutions close to the DM's reference point, and based on the results above, interactive K-RVEA is doing a better job than iRVEA.

5 Conclusions

In this paper, we developed a novel evolutionary interactive multiobjective optimization method, called interactive K-RVEA, that is suitable for real-world computationally expensive problems. As integral elements of the new method, we

chose the RVEA algorithm as our optimizer and the Kriging models as surrogate models. We developed a novel model management strategy that incorporates the DM's preferences (reference point in our case) in the Kriging models.

We demonstrated the performance of the developed method by solving a computationally expensive simulation-based problem where our goal was to find an optimal configuration for an energy system of a heterogeneous business building complex, and we were able to generate a reasonable solution, which had better values than the final reference point provided by the DM except for the second objective. We demonstrated how the decision maker can interact with the method and how the most preferred solution is chosen. Then, we compared the results produced by interactive RVEA that has no model management strategy. We ran both algorithms for ten independent runs and considered three different performance indicators (achievement scalarizing function, domination, and R-metric). We showed the importance of having a model management strategy for computationally expensive problems. Besides, we demonstrated that interactive K-RVEA followed the decision maker's preferences better than interactive RVEA. Thanks to interactive K-RVEA, very good results were generated without spending too much of computational resources.

In this paper, we fixed the values of most of the parameters in interactive K-RVEA, and developing an adaptive method to change these values during optimization is one of our future research directions. Another possible future research topic is to address different types of preferences and make interactive K-RVEA compatible with them. Here, the challenge is how to develop a model management strategy that can use different preferences and incorporate them within the surrogate models.

Acknowledgements This work was partly supported by Honda Research Institute Europe. This research was partly supported by the Academy of Finland (grant no 311877) and is related to the thematic research area DEMO (Decision Analytics utilizing Causal Models and Multiobjective Optimization, jyu.fi/demo) of the University of Jyvskyl. Interactive K-RVEA will be implemented in the open-source software DESDEO (https://desdeo.it.jyu.fi/) funded by the Academy of Finland.

Appendix

In this Appendix, first, we describe the decision variables of the simulation-based problem (3). Then, we show the performance of Kriging models for problem (3).

Decision variables

The simulator uses decision variables to calculate parameters of four different investment options. Then, based on the parameters and the investment options, we can calculate the objective functions values (f_2, f_3 and f_4). The investment options are as follows:

1. A photovoltaic (PV) system on the building roof or carport.
2. An extension of the internal heat storage.
3. A stationary battery.
4. Optimization of the operation of co-generator for heat and power (CHP).

The first three decision variables are related to the PV system: x_1 is the inclination angle, x_2 is the orientation angle, and x_3 is the peak output power of the PV system. The next two decision variables, x_4 and x_5, control the stationary battery's capacity and the maximum charging/discharging power. The minimum and maximum battery state of charge are maintained by x_6. Then, the battery has a charging and discharging threshold connected to the next two decision variables x_7 and x_8. Next, the decision variable x_9 is used to calculate the size of the heat storage. Finally, the CHP generator will only turn on if the ambient temperature is below a certain level, which defines the final decision variable x_{10}. For a full explanation of the decision variables, see Rodemann (2019).

Surrogate models

Here, we show that Kriging models are suitable for problem (3) with different initial population sizes. We tested different Kriging models (with different kernels) along with five other well known surrogate models (Chugh et al. 2019; Bartz-Beielstein and Zaefferer 2017). First, we used Kriging with normal, radial basis function (RBF), rational quadratic (RQ), exponential sine squared (ESS) and Matern kernels. Second, we used support vector regression (SVR) by Drucker et al. (1997) with linear, RBF and polynomial kernels. Last, we used random forest (Liaw and Wiener 2002) and Bayesian surrogates (Fornalski 2015). We tested these surrogates with different training sample sizes. The first sample size was set as 35 (Rodemann 2019). Then, we doubled the sample size. According to Knowles (2006) and Zhang et al. (2010), for n decision variables, the initial population should be $11n - 1$. So we used the same logic to choose the third sample size, which was 109.

In Tables 5, 6 and 7, one can see the results for different sample sizes for all the surrogate models that were tested (the best results are highlighted in boldface in each table). It is worth mentioning that each model was trained ten times, where each time the training sample was selected randomly (in the feasible space), and the R^2 value (Torrie 1960; Glantz and Slinker 1990) was used to evaluate how accurate the surrogates were. In the tables we show the average of these ten runs. Besides, a random assign algorithm was used to create a sample pool (for all sample sizes) for

Table 5 Average of the R^2 values for different surrogate models with sample size 35

	Annual operation cost	Annual CO_2 emissions	Resilience
Bayesian	0.711	0.856	0.618
Random forest	0.626	0.687	**0.802**
SVR-linear	0.454	0.368	0.516
SVR-RBF	0.498	0.435	0.482
SVR-polynomial	0.457	0.625	0.664
Kriging-Default	0.734	0.904	0.523
Kriging-RBF	0.756	0.908	0.540
Kriging-Matern	**0.768**	0.911	0.625
Kriging-ESS	**0.768**	**0.914**	0.600
Kriging-RQ	0.765	0.911	0.265

Table 6 Average of the R^2 values for different surrogate models with sample size 70

	Annual operation cost	Annual CO_2 emissions	Resilience
Bayesian	**0.912**	0.921	0.788
Random forest	0.869	**0.922**	0.768
SVR-linear	0.498	0.400	0.498
SVR-RBF	0.645	0.578	0.651
SVR-polynomial	0.747	0.704	0.784
Kriging-Default	0.764	0.914	0.632
Kriging-RBF	0.788	0.917	0.592
Kriging-Matern	0.883	0.901	**0.816**
Kriging-ESS	0.760	0.810	0.762
Kriging-RQ	0.775	0.921	0.545

Table 7 Average of the R^2 values for different surrogate models with sample size 109

	Annual operation cost	Annual CO_2 emissions	Resilience
Bayesian	0.827	0.887	0.788
Random forest	**0.836**	0.864	0.768
SVR-linear	0.476	0.471	0.516
SVR-RBF	0.745	0.564	0.683
SVR-polynomial	0.765	0726	0.767
Kriging-Default	0.689	0.723	0.727
Kriging-RBF	0.689	0.834	0.727
Kriging-Matern	0.825	**0.893**	**0.800**
Kriging-ESS	0.754	0.854	0.762
Kriging-RQ	0.795	0.891	0.698

training the surrogate models. Here, 70 percent of the sample size was used to train the surrogates, and the remaining 30 percent was used to test them.

As one can see, SVR surrogates did not perform as well as the others. This could be because of their hyper-parameter tuning. On the other hand, Kriging had the best performance for at least two objectives with different training sample sizes. Besides, the uncertainty information that Kriging provides can be utilized in interactive K-RVEA. Moreover, these results are only based on the initial populations, and the performance of Kriging models will improve as we update them during the solution process. Based on the results provided, we could conclude that Kriging models have competitive performance, and we selected them to be used in the interactive K-RVEA method.

References

Bartz-Beielstein T, Zaefferer M (2017) Model-based methods for continuous and discrete global optimization. Appl Soft Comput 55:154–167

Branke J, Deb K, Miettinen K, Slowiński R (eds) (2008) Multiobjective Optimization: Interactive and Evolutionary Approaches. Springer, Berlin

Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 20(5):773–791

Cheng R, Rodemann T, Fischer M, Olhofer M, Jin Y (2017) Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation. IEEE Trans Emer Topics Comput intell 1(2):97–111

Chugh T, Jin Y, Miettinen K, Hakanen J, Sindhya K (2018) A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. IEEE Trans Evol Comput 22(1):129–142

Chugh T, Sindhya K, Hakanen J, Miettinen K (2019) A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. Soft Comput 23(9):3137–3166

Cornell JA (2011) Experiments with mixtures: designs, models, and the analysis of mixture data. John Wiley & Sons, New York

Deb K (2001) Multi-objective optimization using evolutionary algorithms. John Wiley & Sons, New York

Derrac J, Garca S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18

Drucker H, Burges CJ, Kaufman L, Smola AJ, Vapnik V (1997) Support vector regression machines. In: Mozer MC, Jordan MI, Petsche T (eds) Advances in neural information processing systems. MIT press, pp 155–161

Fornalski KW (2015) Applications of the robust bayesian regression analysis. Int J Soc Syst Sci 7(4):314–333

Fritzson P, Bunus P (2002) Modelica-a general object-oriented language for continuous and discrete-event system modeling and simulation, In: Proceedings of the 35th Annual Simulation Symposium, pp. 365–380, IEEE

Glantz SA, Slinker BK (1990) Primer of applied regression and analysis of variance. McGraw-Hill, USA

Hakanen J, Chugh T, Sindhya K, Jin Y, Miettinen K (2016) Connections of reference vectors and different types of preference information in interactive multiobjective evolutionary algorithms, In: Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8

Hwang C-L, Masud ASM (1979) Multiple objective decision making methods and applications: a state-of-the-art survey. Springer, Berlin

Jin Y (2011) Surrogate-assisted evolutionary computation: recent advances and future challenges. Swarm Evolut Comput 1(2):61–70

Knowles J (2006) ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans Evol Comput 10(1):50–66
Li K, Deb K, Yao X (2017) R-metric: Evaluating the performance of preference-based evolutionary multiobjective optimization using reference points. IEEE Trans Evol Comput 22(6):821–835
Liaw A, Wiener M et al (2002) Classification and regression by randomforest. R News 2(3):18–22
McKay MD, Beckman RJ, Conover WJ (2000) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 42(1):55–61
Miettinen K (1999) Nonlinear multiobjective optimization. Kluwer Academic Publishers, Amsterdam
Miettinen K, Hakanen J, Podkopaev D (2016) Interactive nonlinear multiobjective optimization methods. In: Greco S, Ehrgott M, Figueira JR (eds) Multiple criteria decision analysis, 2nd edn. Springer, Berlin, pp 927–976
Purshouse RC, Deb K, Mansor MM, Mostaghim S, Wang R (2014) A review of hybrid evolutionary multiple criteria decision making methods, In: Proceedings of the 2014 IEEE Congress on Evolutionary Computation, pp. 1147–1154
Rodemann T (2019) A comparison of different many-objective optimization algorithms for energy system optimization. In: Kaufmann P, Castillo P (eds) Applications of evolutionary computation. Springer, Berlin, pp 1–16
Sacks J, Schiller SB, Welch WJ (1989) Designs for computer experiments. Technometrics 31(1):41–47
Torrie JH (1960) Principles and procedures of statistics: with special reference to the biological sciences. McGraw-Hill, USA
Wang H, Olhofer M, Jin Y (2017) A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges. Complex Intell Syst 3(4):233–245
Web-based parallel coordinate plot. Avalable: https://dgoldri25.github.io/Categorical-ParallelCoordinatePlot/. (accessed: 13.2.2019) [online]
Wierzbicki AP (1980) The use of reference objectives in multiobjective optimization. In: Fandel G, Gal T (eds) Multiple criteria decision making theory and application. Springer, Berlin, pp 468–486
Xin B, Chen L, Chen J, Ishibuchi H, Hirota K, Liu B (2018) Interactive multiobjective optimization: a review of the state-of-the-art. IEEE Access 6:41256–41279
Yang R, Wang L (2012) Multi-objective optimization for decision-making of energy and comfort management in building automation and control. Sustain Cities Soc 2(1):1–7
Zhang Q, Liu W, Tsang E, Virginas B (2010) Expensive multiobjective optimization by MOEA/D with Gaussian process model. IEEE Trans Evol Comput 14(3):456–474

Affiliations

Pouya Aghaei Pour[1] · Tobias Rodemann[2] · Jussi Hakanen[1] · Kaisa Miettinen[1]

Tobias Rodemann
tobias.Rodemann@honda-ri.de

Jussi Hakanen
jussi.hakanen@jyu.fi

Kaisa Miettinen
kaisa.miettinen@jyu.fi

[1] Faculty of Information Technology, University of Jyvaskyla, P.O. Box 35, 40014 University of Jyvaskyla, Finland

[2] Honda Research Institute Europe, Carl-Legien-Strasse, 30, 63073 Offenbach, Germany