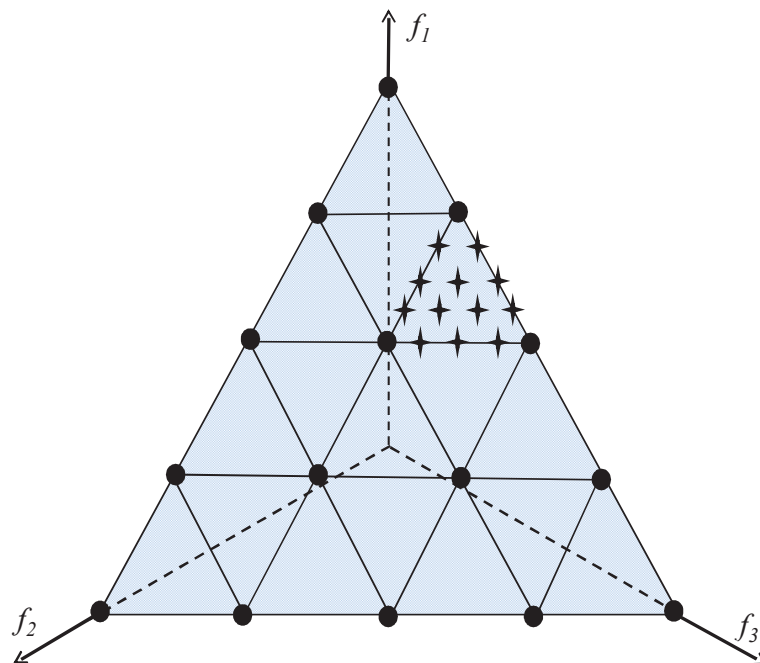


Seyed Mohammad Mehdi Tabatabaei

# On Approaches for Solving Computationally Expensive Multiobjective Optimization Problems



Seyed Mohammad Mehdi Tabatabaei

On Approaches for Solving  
Computationally Expensive  
Multiobjective Optimization  
Problems

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella  
julkisesti tarkastettavaksi yliopiston Agora-rakennuksen auditoriossa 3  
joulukuun 15. päivänä 2016 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 3, on December 15, 2016 at 12 o'clock noon.



UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2016

On Approaches for Solving  
Computationally Expensive  
Multiobjective Optimization  
Problems

JYVÄSKYLÄ STUDIES IN COMPUTING 254

Seyed Mohammad Mehdi Tabatabaei

On Approaches for Solving  
Computationally Expensive  
Multiobjective Optimization  
Problems



UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2016

Editors

Timo Männikkö

Department of Mathematical Information Technology, University of Jyväskylä

Pekka Olsbo, Ville Korhonen

Publishing Unit, University Library of Jyväskylä

URN:ISBN:978-951-39-6861-8

ISBN 978-951-39-6861-8 (PDF)

ISBN 978-951-39-6860-1 (nid.)

ISSN 1456-5390

Copyright © 2016, by University of Jyväskylä

Jyväskylä University Printing House, Jyväskylä 2016

## ABSTRACT

Tabatabaei, Seyed Mohammad Mehdi

On Approaches for Solving Computationally Expensive Multiobjective Optimization Problems

Jyväskylä: University of Jyväskylä, 2016, 68 p.(+included articles)

(Jyväskylä Studies in Computing

ISSN 1456-5390; 254)

ISBN 978-951-39-6860-1 (nid.)

ISBN 978-951-39-6861-8 (PDF)

Finnish summary

Diss.

In this thesis, we consider solving computationally expensive multiobjective optimization problems that take into account the preferences of a decision maker (DM). The aim is to support the DM in identifying the most preferred solution for problems that have several conflicting objectives and when the evaluation of the candidate solutions is time consuming. This is conducted by replacing computationally expensive functions with computationally inexpensive functions, known as surrogates. First, based on a literature survey, we introduce two frameworks, i.e., a sequential and an adaptive framework, based on which surrogate-based methods are classified and compared. We then identify relevant challenges that warrant more research efforts. In order to deal with the challenges, we develop two surrogate-based methods: SURROGATE-ASF and ANOVA-MOP.

As an interactive method, SURROGATE-ASF has two phases: initialization and decision-making. In the first phase, the decision space is decomposed into a finite number of hyper-boxes. For each hyper-box, a single-objective surrogate problem is built. By solving an appropriate surrogate problem in the latter phase, a solution corresponding to the preferences of the DM is obtained. Numerical results support that SURROGATE-ASF can solve problems with at most 12 decision variables, 5 objective functions and nonconvex and/or disconnected sets of Pareto optimal solutions.

To solve problems with high-dimensional decision and objective spaces, we develop the ANOVA-MOP method. Based on information obtained from sensitivity analysis, a problem is decomposed into a few sub-problems with low-dimensional decision and objective spaces. These sub-problems are solved, and the solutions obtained are composed to form approximated solutions for the original problem. ANOVA-MOP can be applied either as a non-interactive or an interactive method.

Finally, we discuss the potential of a new metamodeling technique, called T-splines, to be incorporated into ANOVA-MOP to solve problems including non-differentiable functions. By applying the methods developed in this thesis, we extend the applicability of interactive methods to solving computationally expensive problems.

Keywords: Multiobjective optimization, decision-making, computational cost, surrogate, decomposition.

<b>Author</b>	Seyed Mohammad Mehdi Tabatabaei Department of Mathematical Information Technology University of Jyväskylä Finland
<b>Supervisors</b>	Professor Kaisa Miettinen Department of Mathematical Information Technology University of Jyväskylä Finland  Dr. Jussi Hakanen Department of Mathematical Information Technology University of Jyväskylä Finland  Dr. Markus Hartikainen Department of Mathematical Information Technology University of Jyväskylä Finland  Dr. Karthik Sindhya Department of Mathematical Information Technology University of Jyväskylä Finland
<b>Reviewers</b>	Professor Juergen Branke Warwick Business School University of Warwick United Kingdom  Professor Hirotaka Nakayama Department of Information Science and Systems Engineering Konan University Japan
<b>Opponent</b>	Professor Dr. hab. inż. Ignacy Kaliszewski Systems Research Institute Polish Academy of Sciences Poland

## ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my supervisors, Prof. Kaisa Miettinen, Dr. Jussi Hakanen, Dr. Markus Hartikainen and Dr. Karthik Syndhya for the continuous support of my Ph.D. studies and related research, and for their patience, motivation and immense knowledge of multiobjective optimization. In particular, I would like to thank Prof. Miettinen for her openness to new ideas and discussions, for being available and her valuable encouragements, for her trust to share her priceless experiences and for assisting me to be a better person, both scientifically and socially. I am thoroughly grateful to her.

Besides my supervisors, I would like to thank my colleagues, Dr. Alberto Lovison and Dr. Matthias Tan for their valuable collaborations and insightful comments which incited me to widen my research from various perspectives.

For reviewing my dissertation, I would like to thank Prof. Juergen Branke from the Warwick Business School at the University of Warwick, United Kingdom and Prof. Hiroataka Nakayama from the Department of Information Science and Systems Engineering at the Konan University, Japan.

I would like to thank my international friends, colleagues and office staff of the Department of Mathematical Information Technology, especially, the members of the Industrial Optimization Group– Dr. Vesa Ojalehto, Mr. Tinkle Chugh and Mrs. Yue Zhou-Kangas for their fruitful supports during my Ph.D. studies.

My Ph.D. studies have been financially supported by the COMAS Doctoral Program in Computing and Mathematical Sciences at the University of Jyväskylä, the Academy of Finland project No. 287496 and the Early Career Scheme (ECS) project No. 21201414 sponsored by the Research Grants Council of Hong Kong (Dr. Matthias Tan). I am deeply grateful to their supports and generosity.

I would like to thank warmly my dear and loyal Iranian friends at the University of Jyväskylä who have been my family in Finland. I will never forget the precious moments shared by Younes, Ali and Houra, Payam and Mahsa, Ali and Maryam, Diyako and Shownem, Maziar and Zahra, Jeff and Atiyeh, as well as Hojat, Mehdi, and Azadeh.

Last but not least, my special thanks go to my parents, Mohsen and Narges, my sister, Fatemeh and my brothers, Mojtaba and Ali. They have always been priceless sources of love, hope and encouragement in my life.



## LIST OF FIGURES

FIGURE 1	A surrogate function. ....	19
FIGURE 2	Sequential framework. ....	23
FIGURE 3	Adaptive framework: type 1. ....	24
FIGURE 4	Adaptive framework: type 2. ....	24
FIGURE 5	A convex hull with 5 predetermined reference points for a bi-objective problem. (Color online) ....	31
FIGURE 6	A convex hull with 15 predetermined reference points for a three-objective problem. ....	32
FIGURE 7	Projection of a given reference point onto the convex hull. ....	33
FIGURE 8	Hyper-boxes in the decision space corresponding to the sub-regions on the convex hull in Figure 5. For instance, solutions $x^1$ and $x^2$ form $S^1$ . ....	34
FIGURE 9	An example of an effective total sensitivity indices matrix (SM). The bar shows the range of the estimated total sensitivity indices. ....	40
FIGURE 10	(a) An incidence matrix $M$ , (b) a decomposition corresponding to the matrix $M$ in (a). ....	41
FIGURE 11	The curve, circle and square represent the Pareto frontier, synthetic Pareto frontier and the knee point in problem (19), respectively. ....	45
FIGURE 12	A knot-index-coordinate system. Nodes are shown in red circles. (Color online) ....	52
FIGURE 13	(a) An initial mesh, (b) bivariate B-spline refinement, (c) T-spline refinement. (Color online) ....	52
FIGURE 14	The indices in the local knot vectors corresponding to the node (4, 3) are shown in green. (Color online) ....	54

## CONTENTS

ABSTRACT

ACKNOWLEDGEMENTS

LIST OF FIGURES

CONTENTS

LIST OF INCLUDED ARTICLES

1	INTRODUCTION .....	11
2	MULTIOBJECTIVE OPTIMIZATION .....	14
2.1	Multiobjective Optimization Problems .....	14
2.1.1	Achievement Scalarizing Function .....	16
2.2	Interactive Methods .....	17
2.3	Computationally Expensive MOPs.....	18
3	NON-NATURE-INSPIRED SURROGATE-BASED METHODS .....	21
3.1	Sequential Framework .....	22
3.2	Adaptive Framework: Types 1 and 2 .....	23
3.3	Review of Non-nature-inspired Methods .....	25
4	THE SURROGATE-ASF METHOD .....	30
4.1	Basis Elements .....	30
4.2	Discussion on the Performance of SURROGATE-ASF .....	34
5	THE ANOVA-MOP METHOD .....	37
5.1	Motivation .....	37
5.2	Basic Elements .....	39
5.3	Discussion and an Example .....	43
6	POTENTIAL FUTURE DEVELOPMENT FOR ANOVA-MOP .....	48
6.1	Motivation .....	48
6.2	Function Approximation Through Spline Functions.....	50
6.3	T-splines Review .....	51
7	AUTHOR'S CONTRIBUTION .....	55
8	CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS .....	57
	YHTEENVETO (FINNISH SUMMARY) .....	59
	REFERENCES.....	60
	INCLUDED ARTICLES	

## LIST OF INCLUDED ARTICLES

- PI M. Tabatabaei, J. Hakanen, M. Hartikainen, K. Miettinen, and K. Sindhya. A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods. *Structural and Multidisciplinary Optimization*, 2015, 52, 1-25.
- PII M. Tabatabaei, M. Hartikainen, K. Sindhya, J. Hakanen, and K. Miettinen. An interactive surrogate-based method for computationally expensive multiobjective optimization. *Submitted*.
- PIII M. Tabatabaei, A. Lovison, M. Hartikainen, M. Tan, and K. Miettinen. ANOVA-MOP: ANOVA decomposition for multiobjective optimization. *Submitted*.
- PIV M. Tabatabaei and M. Tan. Prospect of T-Splines for the ANOVA-MOP Method. *Reports of the Department of Mathematical Information Technology, Series B, Scientific Computing, No. 10, University of Jyväskylä, Jyväskylä, 2016*.

Dedicated to my parents  
Mohsen & Narges

# 1 INTRODUCTION

The desire of choosing the best alternative among all possible ones while taking limitations into account has always been part of every day life. Optimization is the art of thinking systematically to find the best alternative. Scientifically, by formulating and solving an optimization problem, we wish to find an alternative that has the most effective impact on the defined goals. Such problems arise in many application domains, such as engineering, economy and biology. When encountering optimization problems, we benefit from using precise models to realize our goals within defined constraints. The models representing our goals are called objective functions, and those representing our limitations are called constraint functions.

Everyday-life or real-world optimization problems typically consist of multiple conflicting objective functions and are known as multiobjective optimization problems. For instance, when buying a house, one may consider the price, the size, and the distance to the city center. Often, big houses near the city center are more expensive in comparison with those far away from the city center. Thus, there is a trade-off among price, size and distance. Of course, a house with a decent price and size located relatively near to the city center would be desirable, but one is not always guaranteed to find such a house.

As can be understood from the above example, instead of having one alternative, one can have several alternatives with different trade-offs in terms of price, size and location. More precisely, in multiobjective optimization problems, instead of one optimal solution, there exist several (often infinite) solutions called Pareto optimal solutions. The set of all Pareto optimal solutions can be called a Pareto frontier. The Pareto frontier can be convex, nonconvex and/or disconnected. Mathematically and in the absence of any preferences, Pareto optimal solutions are equally preferable. Thus, we require an expert known as a decision maker (DM) who has a deep insight into the problem. The task of the DM is to provide his/her preferences, compare solutions obtained and finally choose a desirable solution.

In the literature, methods developed to solve multiobjective optimization problems are divided into four categories according to the role of the DM [1].

The first one includes no-preference methods. In such methods, the preferences of the DM are not available. The second one includes a priori methods in which the DM expresses his/her preferences before the solution process. The third category involves a posteriori methods where the DM provides his/her preferences after the solution process in which solution candidates have been generated. The last category includes interactive methods. In these methods, the DM actively provides his/her preferences during the solution process. A merit of interactive methods is that the DM is allowed to explore different solutions and to guide the solution process. Thus, the DM becomes more and more aware regarding the nature of the problem during the solution process.

In reality, multiobjective optimization problems may involve functions that do not have any explicit function formulation. Such functions are called black-box functions. In black-box functions, one can only have an output for a given input. In such problems, function evaluation is conducted through computationally expensive and time-consuming experiments. In the literature, such problems are termed *computationally expensive multiobjective optimization problems*.

A common method for reducing the computational cost is approximation. In such methods, for example, statistical and probabilistic mathematical techniques are employed to approximate black-box functions. Then, these black-box functions are replaced with the approximated functions, termed *surrogate functions*. This leads to formulating computationally inexpensive problems, known as *surrogate multiobjective optimization problems*.

In the literature, methods that solve computationally expensive problems by using surrogate problems are called *surrogate-based methods*. Generally there are two kinds of surrogate-based methods, nature- and non-nature-inspired surrogate-based methods [PI]. In the former one, the mechanism of nature-inspired optimization algorithms such as evolutionary and particle swarm optimization algorithms plays a role in building surrogate functions [2, 3]. In the latter one, which is the focus of this thesis, surrogate functions are constructed independently from the optimization algorithm utilized. In this thesis, for the sake of simplicity, the term "surrogate-based methods" is used in reference to non-nature-inspired surrogate-based methods.

This thesis as a collection of articles that concentrate on developing surrogate-based methods that take into account the preferences of the DM. To do so, scientific papers written in English and published in scientific journals before 2013 are surveyed in [PI]. In this paper, we compare surrogate-based methods and identify some shortcomings and future research directions, such as 1) solving computationally expensive multiobjective optimization problems in which the Pareto frontier is nonconvex and/or disconnected, 2) developing interactive surrogate-based methods to solve problems while taking into account the preferences of the DM, 3) developing surrogate-based methods to solve problems that have high-dimensional decision and objective spaces and 4) solving problems including non-differentiable black-box functions.

In [PII], we have developed an interactive surrogate-based method called SURROGATE-ASF to study the first and second shortcomings. In SURROGATE-

ASF, the decision space is decomposed into a limited number of hyper-boxes. For each hyper-box, a corresponding surrogate function is built and a single-objective surrogate problem is formulated. These surrogate problems are employed in a systematic manner to interact with the DM to find the most preferred solution for him/her. This method performs significantly well on problems with at most 12 decision variables and 5 objective functions. However, if a given computationally expensive problem has a large number of objective functions (e.g., more than 5 objective functions), then the DM should compare solutions with a large number of components. This may impose a cognitive load on him/her. Thus, to study the third shortcoming, we have developed a method in [PIII] called ANOVA-MOP.

In ANOVA-MOP, based on information obtained from sensitivity analysis, a problem with high-dimensional decision and objective spaces is decomposed into a limited number of sub-problems with low dimensional decision and objective spaces. Then, each sub-problem is solved separately. The solutions of the sub-problems are composed to form solutions for the original high-dimensional problem. We show that ANOVA-MOP can be employed as a non-interactive or an interactive method. The method employed to elicit sensitivity analysis information and to build surrogate functions in ANOVA-MOP can approximate differentiable black-box functions.

Finally, in [PV], we discuss the potential of a new technique called T-splines to be incorporated into ANOVA-MOP for sensitivity analysis on non-differentiable computationally expensive functions. We expect that computationally expensive multiobjective optimization problems including these kinds of functions would be solved efficiently by ANOVA-MOP.

The rest of this thesis is organized as follows. In Chapter 2, we introduce multiobjective optimizations and the basic concepts used in this thesis. In Chapter 3, we discuss surrogate-based methods and the main findings of [PI]. The SURROGATE-ASF and ANOVA-MOP methods ([PII] and [PIII], respectively) are described in Chapters 4 and 5, respectively. In Chapter 6, we discuss the importance of T-splines (presented in [IV]) to be incorporated into ANOVA-MOP for solving computationally expensive problems. In Chapter 7, the author's contribution is described. We finally conclude and discuss some ideas as future research directions in Chapter 8.

## 2 MULTIOBJECTIVE OPTIMIZATION

In this chapter, we first present a general formulation of multiobjective optimization problems as well as the basic concepts to be used in the following chapters. Subsequently, we discuss several approaches to solving multiobjective objective optimization problems.

### 2.1 Multiobjective Optimization Problems

Mathematically, we consider multiobjective optimization problems of the form

$$\underset{\mathbf{x} \in S}{\text{minimize}} \{f_1(\mathbf{x}), \dots, f_k(\mathbf{x})\}, \quad (1)$$

where the set  $S$  is called the *feasible decision set* (often also called the *feasible design set*), which is a subset of the *decision space*  $\mathbb{R}^n$ . A solution  $\mathbf{x} = (x_1, \dots, x_n)^T \in S$  is called a *feasible decision (variable) vector*, where  $x_i, i = 1, \dots, n$ , are *decision variables*. We have  $k (\geq 2)$  objective functions  $f_i : S \rightarrow \mathbb{R}$ . We denote the vector of objective functions, i.e., the objective vector, by  $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$ . For example, one may consider the feasible decision set as  $S = \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) \leq 0, i = 1, \dots, m, h_j(\mathbf{x}) = 0, j = 1, \dots, p, \mathbf{x}^{lo} \leq \mathbf{x} \leq \mathbf{x}^{up}\}$ , where  $g_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, m$ , and  $h_j : \mathbb{R}^n \rightarrow \mathbb{R}, j = 1, \dots, p$ , are called *inequality* and *equality constraint functions*, respectively, and  $\mathbf{x}^{lo}$  and  $\mathbf{x}^{up}$  stand for the lower and upper bounds of the decision variables, respectively. In this thesis, we only consider the feasible decision set of the form  $S = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^{lo} \leq \mathbf{x} \leq \mathbf{x}^{up}\}$ . The image of the feasible decision set in the *objective space*  $\mathbb{R}^k$  is called the *feasible objective set* (often also called *performance set*) denoted by  $Z (= \mathbf{f}(S))$ . One should note that if some objective  $f_i$  is to be maximized, then we can minimize  $-f_i$  and therefore only minimized functions are considered in (1).

A feasible solution  $\mathbf{x}^* \in S$  and the corresponding  $\mathbf{f}(\mathbf{x}^*) \in Z$  are termed *weakly Pareto optimal* for problem (1) if there does not exist another feasible solution  $\mathbf{x} \in S$  such that  $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$  for all  $i = 1, \dots, k$ . Correspondingly, they are *Pareto optimal* for problem (1) if there does not exist another feasible solution



$\mathbf{x} \in S$  such that  $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$  for all  $i = 1, \dots, k$ , and  $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$  for at least one index  $j \in \{1, \dots, k\}$ . The set of all Pareto optimal solutions in the objective space is called a *Pareto frontier*. All Pareto optimal solutions are weakly Pareto optimal solutions but not vice versa.

Let the set  $X^\beta = \{\mathbf{x}^1, \dots, \mathbf{x}^\beta\}$  be an arbitrary subset of feasible solutions in  $S$ , and  $F^\beta = \{\mathbf{f}(\mathbf{x}^1), \dots, \mathbf{f}(\mathbf{x}^\beta)\}$ , the corresponding objective vectors in  $Z$ . A solution  $\mathbf{x}^l$  (or  $\mathbf{f}(\mathbf{x}^l)$ ),  $l = 1, \dots, \beta$ , that satisfies the definition of Pareto optimality with respect to all solutions in  $X^\beta$  (or  $F^\beta$ ), is called a *non-dominated solution* in  $X^\beta$  (or  $F^\beta$ ). A Pareto optimal solution is a non-dominated solution, but a non-dominated one is not necessarily Pareto optimal.

Consider a feasible solution  $\mathbf{x}_e^i \in \operatorname{argmin}_{\mathbf{x} \in S} \{f_i(\mathbf{x})\}$  for  $i = 1, \dots, k$ . An  $i^{\text{th}}$  *extreme point (solution)* for  $i = 1, \dots, k$ , is defined as  $\mathbf{z}_e^i = \mathbf{f}(\mathbf{x}_e^i)$ . A vector  $\mathbf{z}^{\text{ideal}} = (z_1^{\text{ideal}}, \dots, z_k^{\text{ideal}})^T$  is called the *ideal (objective) vector* in which  $z_i^{\text{ideal}} = f_i(\mathbf{x}_e^i)$  for  $i = 1, \dots, k$ . The ideal vector contains the lower bounds of the objective function values in the Pareto frontier in the objective space. We also define the *nadir (objective) vector* denoted by  $\mathbf{z}^{\text{nadir}} = (z_1^{\text{nadir}}, \dots, z_k^{\text{nadir}})^T$  to be a vector of upper bounds of the objective function values in the Pareto frontier. We also define a vector  $\mathbf{z}^{\text{utp}}$  with  $z_i^{\text{utp}} = z_i^{\text{ideal}} - \epsilon$  for all  $i = 1, \dots, k$ , and for some small positive  $\epsilon \ll 1$  as the *utopian (objective) vector*.

Mathematically, without any preferences, all Pareto optimal solutions are equally acceptable for problem (1) [1]. However, it is generally desirable to obtain the best possible solution to be implemented. Selecting one solution among all Pareto optimal solution requires information which is not included in the objective functions. Therefore, a new element is considered in multiobjective optimization [1], called a *decision maker (DM)*. The DM is a person who can provide preference information by relying on his/her deep insight into the problem. Typically, the final solution is chosen by the DM.

Mathematically, without any preferences, all Pareto optimal solutions are equally acceptable for problem (1) [1]. However, it is generally desirable to obtain the best possible solution to be implemented. Selecting one solution among all Pareto optimal solution requires information which is not included in the objective functions. Therefore, a new element is considered in multiobjective optimization [1], called a *decision maker (DM)*. The DM is a person who can provide preference information by relying on his/her deep insight into the problem. Typically, the final solution is chosen by the DM.

During the solution process, preference information is sought from the DM. Such preferences can be provided, for instance, as desirable or acceptable values for the objective functions. We define  $\bar{z}^i$ , for  $i = 1, \dots, k$ , as aspiration levels containing desirable or acceptable values for objective functions. The vector  $\bar{\mathbf{z}} \in \mathbb{R}^k$ , consisting of aspiration levels, is called a *reference point* given by the DM.

In e.g. [1, 4, 5, 6, 7], methods developed for multiobjective optimization problems have been surveyed. As discussed in [1], these methods are classified according to the role of the DM in the solution process. The classes are:

- *no-preference methods* in which preference information of the DM is not available,

- *a priori methods* in which the DM provides his/her preference information before the solution process,
- *a posteriori methods* in which the DM provides his/her preference information after the solution process, and
- *interactive methods* in which the DM provides his/her preference information during the solution process.

Cases may arise that the DM is not available or does not have any preferences. In such a case, one can apply no-preference methods. In such methods, after solving problem (1), the solutions obtained are presented to the DM. In all the other classes, the DM is assumed to take part in the solution process.

In *a priori* methods, the DM first specifies his/her preferences and then the solution process tries to find a Pareto optimal solution satisfying them as well as possible. This is a straightforward approach, but the difficulty is that the DM may not be aware of the possibilities and limitations of the problem beforehand and may have too optimistic or pessimistic expectations.

Alternatively, one may apply *a posteriori* methods and find a representation of the set of Pareto optimal solutions. Then, the DM is supposed to choose the most preferred solution among them. *A posteriori* methods provide the DM an overview of different solutions available. Nevertheless, in problems with more than three objective functions, it may be challenging for the DM to analyze a large number of solutions, since visualizing the solutions is not as easy as in a biobjective or three-objective problem.

In *interactive* methods, the goal is to find the most preferred solution for the DM. In such methods, steps are repeated and the DM specifies preference information progressively during the solution process. After every iteration, some information is provided to the DM and (s)he is supposed to analyze this information and to express his/her preferences. In this way, the solution process is guided by the DM toward his/her most preferred solution. In addition, the DM can specify and correct his/her preferences during the solution process.

Another concept used in this thesis is scalarization. By scalarizing problem (1), we mean transforming it into a single-objective optimization problem using a scalarizing function. The optimal solution of this single-objective optimization problem is a (weakly) Pareto optimal solution for problem (1). Often, the DM's preferences can be taken into account when using an appropriate scalarizing function. Through scalarization, one can take advantage of the state-of-the-art methods of single-objective optimization.

### 2.1.1 Achievement Scalarizing Function

Achievement scalarizing functions are types of functions used in reference-point-based methods. They have been investigated, for example, in [1, 8, 9]. In this thesis, we employ the following widely used *achievement scalarizing function* (ASF) [9]:

$$\begin{aligned} \text{ASF: } S \times \mathbb{R}^k &\rightarrow \mathbb{R} \\ (\mathbf{x}, \bar{\mathbf{z}}) &\mapsto \max_{i=1, \dots, k} (w_i (f_i(\mathbf{x}) - \bar{z}_i)), \end{aligned} \quad (2)$$

where  $w_i = \frac{1}{z_i^{\text{nadir}} - z_i^{\text{utp}}}$ ,  $i = 1, \dots, k$ , are non-negative fixed weights. Using (2), we formulate a single-objective optimization problem as

$$\underset{\mathbf{x} \in S}{\text{minimize}} \quad \text{ASF}(\mathbf{x}, \bar{\mathbf{z}}). \quad (3)$$

In [9], it is proved that by solving problem (3), a (weakly) Pareto optimal solution corresponding to the reference point is generated. One can add an augmentation term to (2) to only obtain Pareto optimal solutions as discussed and proved in [1]. By changing reference points, one can obtain different (weakly) Pareto optimal solutions. These solutions are obtained regardless of the position of the reference point in the objective space [PII].

## 2.2 Interactive Methods

In this thesis, by solving multiobjective optimization problems, we mean intensifying the most preferred solution as the final one for a DM. The most preferred solution refers to a Pareto optimal solution that satisfies the preferences of the DM. Typically, to find the most preferred solution, the DM (with his/her deep insight into the problem) participates in the solution process and can provide preference information relating to the objective functions that are considered in the process of generating different solutions.

In interactive methods, the phases of interaction with the DM (decision phase) and solution generation (optimization phase) alternate until the DM finds the most preferred solution. After each iteration, some solutions are provided to the DM and (s)he expresses his/her opinion about them or gives some other type of information to form her/his preferences. This information is employed to generate new solutions (which are supposed to better satisfy the DM's preferences). In this way, the DM guides the solution process and only a subset of Pareto optimal solutions is generated and evaluated. In addition, during the solution process, the DM can change his/her preferences and selections.

To put it in a nutshell, general steps in an interactive method are:

1. initialization (e.g., calculate ideal and nadir values and show them to the DM),
2. ask the DM to provide some preference information (e.g., a reference point or number of new solutions to be generated),
3. generate one or more Pareto optimal solution(s) corresponding to the given preferences and show it/them to the DM. If more than one solution has been generated, ask the DM to select the most preferred solution so far, and

4. stop, if the DM is satisfied with the solution selected. Otherwise, go to step 2.

The DM can provide his/her preferences, e.g., by classifying objective functions or in the form of a new reference point [1]. As discussed, e.g., in [51], interactive methods employ different approaches to interact with the DM regarding the solutions generated, such as, visualizations or trade-off information. In fact, by applying an interactive method, (s)he can acquire knowledge about the possibilities and limitations of the problem considered. Moreover, a subset of Pareto optimal solutions is generated that is attractive for the DM. This results in savings in computational cost and avoiding the need to compare many Pareto optimal solutions simultaneously. This may be critical for computationally expensive multiobjective optimization problems when the focus is on reducing computational burden. This highlights the strength of interactive methods over a priori and a posteriori methods.

In this thesis, the main interactive method considered is the reference point method discussed in 2.1.1. In this method, the DM provides his/her aspiration levels as a reference point. Then, problem (3) is solved. The solution obtained is shown to the DM. If (s)he is not satisfied with the solution, another reference point is given by him/her and problem (3) is solved. As soon as a solution obtained satisfies the DM's preferences, the interaction with the DM is terminated and that solution is considered the most preferred solution. This is the realization of the general steps of interactive methods mentioned above for the reference point method.

### 2.3 Computationally Expensive MOPs

In this thesis, we say that problem (1) is computationally expensive if the objective function evaluation takes a significant amount of time to solve the problem. Classical or evolutionary methods developed to solve multiobjective optimization problems require many objective function evaluations. When applying such methods to solve computationally expensive MOPs, the solution process becomes complicated due to the large number of function evaluations. This means that generating a solution may take several years. Examples of real-world computationally expensive optimization problems are given in, e.g., [10, 11].

One type of computationally expensive functions is black-box functions. In such functions, a closed form mathematical formula is not available, and the function values are typically obtained through computationally expensive simulations. Other types of computationally expensive functions can be functions with a closed form formulation, but they involve complicated elements, which make their evaluation time-consuming.

Surrogate-based methods are typically employed to solve a computationally expensive problem. In such methods, a computationally inexpensive problem termed a *surrogate problem* is built. By applying an optimization procedure,

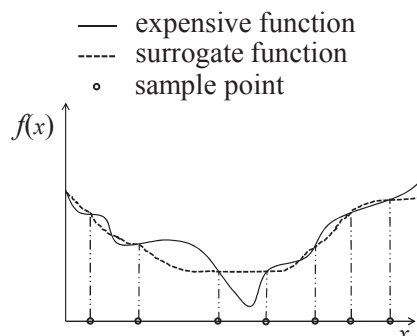


FIGURE 1 A surrogate function.

function evaluation alternates between the surrogate and the computationally expensive problems. In this manner, the number of computationally expensive evaluations can often be decreased considerably, while a high accuracy is still achieved.

Surrogate-based methods can be classified into nature- and non-nature-inspired methods [PI]. In the former, components of nature-inspired methods such as evolutionary and particle swarm algorithms play roles to build surrogate problems (see, e.g., [2, 12, 13]). In the latter, surrogate problems are built regardless of the type of the optimization algorithms employed in them.

The idea of a surrogate building method is approximation. The most widely used approximation approach is to approximate each computationally expensive function by using metamodeling techniques such as polynomial functions [14], Kriging models [15], radial basis functions (RBFs) [16], multivariate adaptive regression splines (MARS) [17], neural networks [18] and support vector regression (SVR) [19]. Approximation of a computationally expensive function requires sample points. To do so, sampling techniques such as Latin hypercube sampling (LHS) [20], central composite design (CCD) [21], orthogonal array sampling (OAS) [21] and full factorial sampling (FFS) [21] are employed to select sample points (which are solutions in the decision space). In [21, 22, 23, 24] and [21, 22, 23, 24, 25, 26, 27, 28, 29] sampling and metamodeling techniques and their characteristics are reviewed, respectively.

After sampling points, the corresponding objective vectors are calculated with the computationally expensive functions. Then, by employing a metamodeling technique, a computationally inexpensive function termed a *surrogate function* is fitted to the sample points as shown in Figure 1. When surrogate functions of all computationally expensive functions are built, a computationally inexpensive MOP known as a *surrogate problem* is formulated.

In surrogate-based multiobjective optimization, it is desirable to build an accurate surrogate problem in the region of the Pareto optimal solutions. However, the surrogate functions may not be accurate in this region. Therefore, the sets of Pareto optimal solutions of the surrogate and the original, computationally expensive MOP may not be the same. In the literature, statistical measure-

ments such as root mean square error (RMSE) [30], predicted error sum of squares (PRESS) [31] and  $R^2$  [25] are employed to evaluate the accuracy of the surrogate functions. In [23, 25, 29, 32] such criteria have been reviewed. Surrogate problems can also be built by approximating the Pareto frontier directly. In the next chapter, surrogate-based methods developed for computationally expensive MOPs are discussed.

In this thesis, for the sake of simplicity, problem (1) is called an *original problem*. The objective functions in the original and the surrogate problems are termed *computationally expensive functions* and *surrogate functions*, respectively. All objective functions in the original problem are assumed to be computationally expensive. In addition, the *approximated Pareto frontier* of the original problem refers to a set of non-dominated (or Pareto optimal) solutions of the surrogate problem which may be evaluated with the computationally expensive functions. The terms *benchmark* and *application problems* are also used in this thesis. The first one indicates the test problems in the literature, such as ZDT problems [33], which are employed to compare the performance of different methods. The second one points to the real-world problems arising from, e.g., industries.

### 3 NON-NATURE-INSPIRED SURROGATE-BASED METHODS

In surrogate-based methods, an initial surrogate problem is built. Then, based on some new evaluated points sampled, the accuracy of the surrogate problem is improved. Methods developed in the literature have devised different criteria for selecting sample points. By reviewing papers published in the literature, we have identified two frameworks to develop surrogate-based methods. However, there does not exist a unified description of major steps in such methods (see, e.g., [22, 23, 27, 34, 35]). For instance, an approach called one-stage sampling in [36] is termed two-stage sampling in [37] and a sequential approach in [23]. While survey papers have been published on surrogate-based single-objective optimization methods, e.g., [27, 32], to the best of our knowledge, no survey paper on surrogate-based multiobjective optimization methods had been published prior to 2013.

To fill the above-mentioned gap in the literature, in [PI], we have surveyed surrogate-based multiobjective optimization methods published in scientific journals before 2013. In this paper, to unify the main steps in surrogate-based methods, we have considered two general frameworks, namely sequential and adaptive frameworks, motivated by the classification in [23, 34], and depending on when to update the surrogate problem. In the former framework, discussed in Subsection 3.1, an accurate surrogate problem is built and then solved. The approximated Pareto frontier obtained is supposedly close to the Pareto frontier of the original problem. In the latter framework, considered in Subsection 3.2, an initial surrogate problem is constructed. Typically, the accuracy of this initial surrogate problem is not satisfactory over the region of the Pareto optimal solutions of the original problem. Therefore, the approximated Pareto frontier of the initial surrogate problem cannot be considered as a representation of the Pareto frontier of the original problem. To resolve this, the accuracy of the surrogate problem is iteratively improved with the aim of generating an approximated Pareto frontier and that coincides with the Pareto frontier of the original problem.

Updating the accuracy of the surrogate problem requires sampling new points (known also as infill points in [24]). Based on our observations, we di-

vide the adaptive framework into types 1 and 2. It is also possible to hybridize both frameworks and develop a new method, as can be seen in [38]. In both frameworks, surrogate problems have to be solved. In the literature, two sorts of methods are employed to solve surrogate problems, i.e., sampling-based and optimization-based ones. In the former methods, by applying a sampling process and without using any optimization algorithm, a surrogate problem is solved (see e.g. [39]). However, in the latter one, an optimization algorithm is used to solve a surrogate problem.

These frameworks have different attributes. In the sequential framework, surrogate functions are supposedly accurate enough over the entire feasible decision (objective) set. When the surrogates are accurate enough, the DM can find solutions that correspond to his/her preferences quickly during the decision-making phase thanks to the surrogate functions. This is an important matter discussed in e.g. [40].

In the adaptive framework, one may not aim at building surrogate functions accurately over the entire feasible decision (objective) set. In fact, when solving problem (1) taking into account the preferences of the DM, surrogates are assumed to be accurate in the region corresponding to the given preferences. In this way, when the DM revises his/her preferences, surrogates should be updated toward the region corresponding to the new preferences given. Due to the computational cost, the DM has to wait to see new solutions corresponding to the new preferences.

### 3.1 Sequential Framework

In the sequential framework, the emphasis is on building an accurate surrogate problem before applying an optimization algorithm. The flowchart in Figure 2 shows the main steps to be taken to develop a method in this framework. As shown in the flowchart, in Steps 1 and 2, initial sample points are selected and then evaluated with the original functions. In Step 3, a surrogate problem is built using the selected sample points. Selecting new sample points is done to improve the accuracy of the surrogate problem, a crucial step. In Step 4, the accuracy of the surrogate problem is evaluated (highlighted in Figure 2). Typically, statistical measurements such as root mean square error (RMSE) or  $R^2$  are applied to check the accuracy. If the accuracy is to be improved, new points are sampled in Step 5, and Steps 2-4 are then repeated.

Once an accurate enough surrogate problem is constructed, it is solved in Step 6. The solution process may be done with a DM, if available. In Step 7, typically, the solution(s) obtained by solving the surrogate problem are evaluated with the original functions. In Step 8, the outcome of Step 7 is treated as a set of approximated Pareto optimal solutions of the original problem and/or the most preferred approximated solution for a DM. One can apply an appropriate visualization technique to visualize the solutions obtained. This framework includes



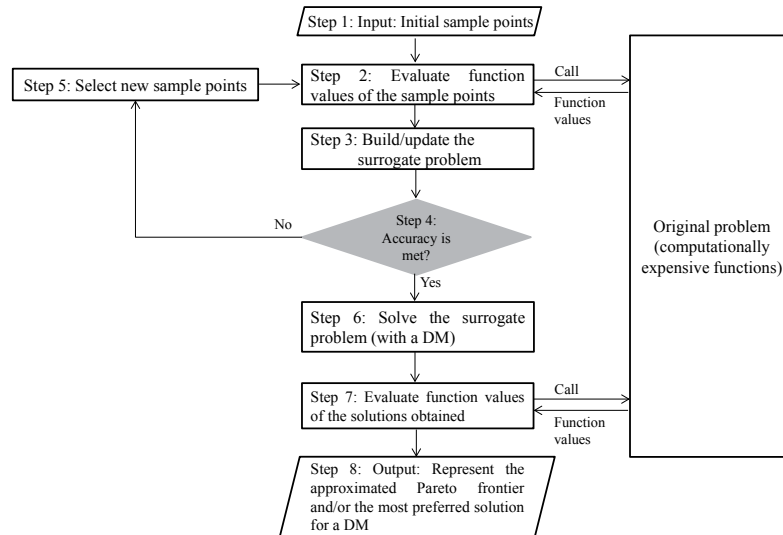


FIGURE 2 Sequential framework.

a special case termed *one-stage sampling*. In such a case, points are sampled only once to build a surrogate problem, i.e., Step 5 is not conducted. Then the surrogate problem is built and solved as described.

In this framework, the closeness of the set of approximated Pareto optimal solutions to the set of Pareto optimal solutions of the original problem depends on the accuracy of the surrogate problem. Moreover, the surrogate functions of the original functions are typically accurate over the entire decision (or objective) space.

### 3.2 Adaptive Framework: Types 1 and 2

Another framework introduced in [PI] is the adaptive framework as shown in Figures 3 and 4. In this framework, an initial surrogate problem is built. The approximated Pareto frontier obtained may not be able to represent the Pareto frontier of the original problem. This necessitates updating the surrogate problem by sampling new points and solving the updated surrogate problem iteratively. In this way, the approximated Pareto frontier is intended to coincide with the Pareto frontier of the original problem.

Depending on when new points are sampled to improve the accuracy of the surrogate problem, we consider two different types in this framework. In type 1, the sample points generated before assessing a stopping criterion (Step 4 highlighted in Figure 3) are utilized to update the surrogate problem. In type 2,

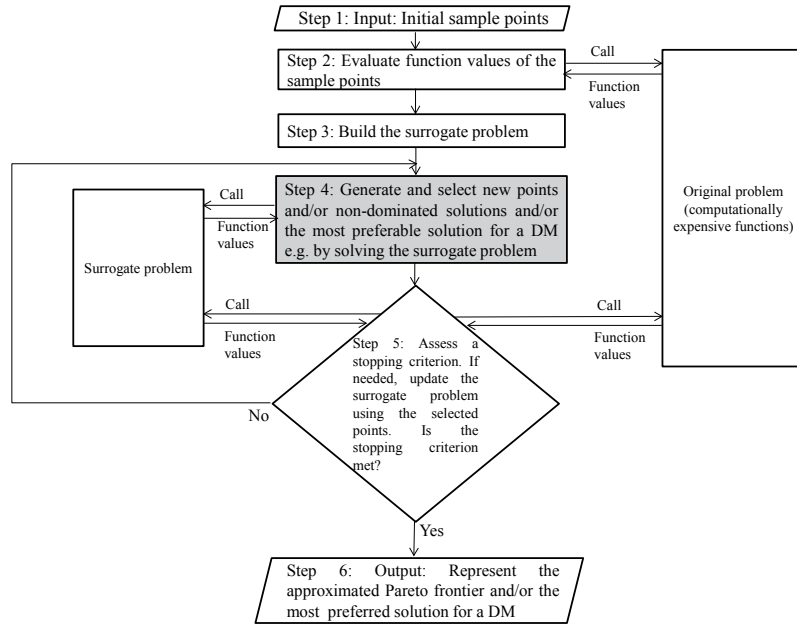


FIGURE 3 Adaptive framework: type 1.

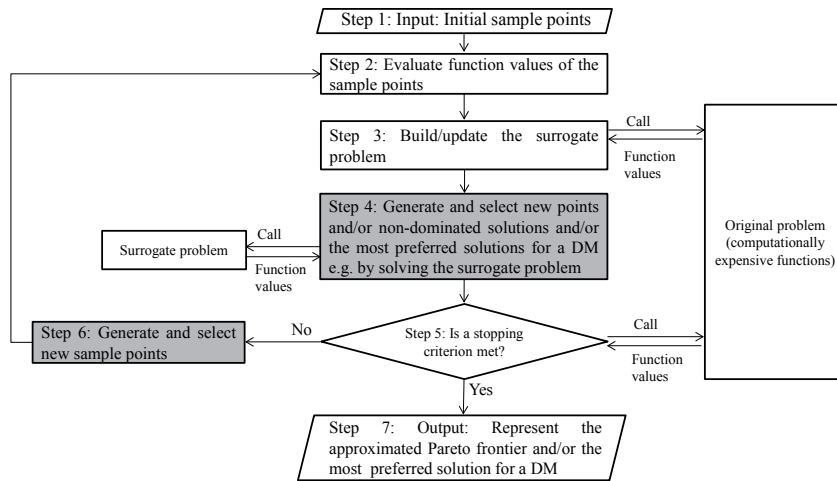


FIGURE 4 Adaptive framework: type 2.

not only are the sample points generated before assessing the stopping criterion (Step 4 highlighted in Figure 4) considered, but also new sample points are generated and selected after the stopping criterion is assessed (Step 6 highlighted in Figure 4) to update the surrogate problem.

As can be seen in the flowcharts in Figures 3 and 4, Steps 1-5 are similar in both types. In Step 1, initial sample points are selected. Then, they are evaluated with the original functions in Step 2. Using the evaluated points, the initial surrogate problem is built. New sample points are selected in Step 4 to improve the accuracy of the surrogate problem in regions corresponding to the preferences of the DM, if available, or the Pareto frontier. These points can be generated by solving the surrogate problem or sampling unexplored areas in the decision and/or objective space based on a sampling strategy. A subset of sample points from the generated samples is chosen by relying on a method-dependent criterion. In Step 5, a stopping criterion is assessed. We observe that different stopping criteria have been developed. For instance, in some methods, the selected sample points are evaluated with the original functions, and then a stopping criterion is checked. In some other methods, the selected sample points are evaluated with the original functions and the surrogate problem is updated. Then, a stopping criterion is assessed.

In type 1, if a stopping criterion is satisfied, in Step 6, the approximated Pareto frontier of the original problem or the most preferred solution for a DM is the set of non-dominated (or Pareto optimal) solutions or the most preferred solution of the last updated surrogate problem (these solutions may have been evaluated with the original functions). Depending on the user's needs, the obtained solutions can also be visualized by applying an appropriate visualization approach. Otherwise, i.e., when the stopping criterion is not yet met, Steps 4-5 are repeated. One should note that if the surrogate problem has not already been updated in Step 6, we first update the surrogate problem with the evaluated sample points and then Steps 4-5 are repeated.

In type 2, if the stopping criterion is not yet met in Step 5, a set of new sample points from unexplored regions of the decision and/or objective space is selected in Step 6. These new points as well as the (evaluated) points selected in Step 5 are considered, and Steps 2-5 are then repeated. Otherwise, similar to type 1, the solutions obtained by solving the last updated surrogate problem are considered for the original problem.

### 3.3 Review of Non-nature-inspired Methods

In [PI], we surveyed the methods developed in both sequential and adaptive frameworks. Then, we compared the methods surveyed from four points of view:

1. Are the methods capable of solving problems including black-box functions when there is a lack of information about mathematical properties, such as convexity or continuity?

2. Are the methods capable of capturing different types of Pareto frontiers, such as nonconvex and/or disconnected ones?
3. What kinds of problems can be solved by the methods in terms of number of objective and constraint functions as well as decision variables?
4. How is the interaction with a DM conducted during the solution process?

This comparison led us to identify future research directions for solving computationally expensive MOPs. In what follows, we summarize the comparison by taking into account the above questions. Throughout this comparison, we also discuss future research directions.

Regarding the first question, we observe that all methods surveyed except [41] can solve problems including black-box functions. In [41], the problem to be solved is supposed to have convex objective functions. This assumption typically does not hold when dealing with black-box functions. Noisy black-box functions are only discussed in [42]. However, the performance of the method developed in [42] is evaluated on noise-free benchmark problems. We also observe that approximating non-differentiable functions has not been discussed deeply. As can be understood, there is a need to conduct more research on applying metamodeling techniques to model noisy and/or non-differentiable black-box functions.

As far as the second question is concerned, after reviewing the papers surveyed, we have realized that to capture nonconvex and disconnected Pareto frontiers, selecting sample points in potentially relevant regions plays an essential and substantial role. Methods surveyed devise various strategies to consider this role. We observe that in [35, 42, 43, 44, 45], metamodeling techniques are applied not only to build the surrogate problem but also to construct some functions for selecting sample points. For instance, in [43, 45] these functions are built to do sampling toward extreme solutions as well as the Pareto frontier. In [35], a sampling function is built to generate evenly distributed sample points. The authors in [35] discuss that selecting evenly distributed sample points to improve the accuracy of the surrogate problem may assist to capture disconnected parts of the Pareto frontier.

The variances of the objective function values of the sample points predicted by the Kriging model are considered as a criterion of selecting samples in [46]. In [44], to select new samples, the authors address the Lagrangian multiplier values calculated by the metamodeling technique employed. A trust region strategy is discussed in [34, 47] to search the explored and unexplored regions in the decision and/or objective space. Convex hulls are considered in [48] as a method for capturing a nonconvex Pareto frontier. The PAINT method developed in [49] is capable of capturing a nonconvex Pareto frontier. In [38, 50], the idea of the normal constraint method [51, 52] is employed to generate sample points near the Pareto frontier of the original problem. In [38], a hybrid surrogate method is developed in which the sequential framework and type 1 of the adaptive framework are considered. In this method, first a surrogate problem is built by type 1 of the adaptive framework to generate initial sample points. Then, these sample points are employed to construct a surrogate problem of a computationally

expensive MOP by the sequential framework.

We observe that the metamodeling technique applied in [35, 38, 42, 43, 45] is RBFs. The Pareto frontier of the benchmark or application problems considered in [38] is nonconvex, whereas in [35, 42, 43, 45] it is nonconvex and disconnected. One may ask whether the type of a metamodeling technique has any role in solving problems with nonconvex and disconnected Pareto frontiers.

Overall, the methods surveyed have not discussed deeply solving computationally expensive MOPs, including nonconvex and disconnected Pareto frontiers. Such observations motivate us to consider developing surrogate-based methods to tackle the issues of nonconvexity and discontinuity in the Pareto frontier as a research direction.

Regarding the third question relating to solving computationally expensive MOPs in terms of dimensionality, we observe that  $k = 5$  is the largest number of objective functions in [41, 46, 50, 53]. In other papers surveyed, benchmark and/or application problems with at most three objective functions are considered. Only in [54] was a problem solved that had 13 constraint functions and 31 decision variables. As mentioned in [43, 45], the methods developed can be applied for problems with two or three objective functions and less than seven decision variables. In addition, the authors in [42] note that the method is suitable for problems with less than three objective functions and six decision variables. In [32], techniques for high-dimensional problems are reviewed. This reference may inspire some idea to develop new methods to solve such problems. Moreover, no paper mainly focuses on developing a strategy to handle constraint functions. This indeed can be considered as a future research direction.

As far as the fourth question relating to interaction with a DM is concerned, the methods developed in [35, 41, 44, 49, 53, 55] are capable of providing the most preferred solution for a DM. Among these methods, the methods developed in [35, 44] are a priori types and others are interactive ones. In fact, in [35, 44], a DM can provide his/her preferences in the form of weights and a reference point, respectively. Then these methods find a solution corresponding to the preferences of the DM. However, if the DM changes his/her preferences, all of the methods should be run again to provide a solution for the DM. In [53], the role of the DM is to compare the solutions obtained. In the method developed in [41], a DM assigns a goal value for one of the objective functions and upper bounds for the other ones. In [55], a DM expresses his/her preferences by classifying the objective functions or as a reference point. We observe that in terms of considering the role of a DM during the solution process, the PAINT method is a distinguished one. In this method, first a mixed integer linear multiobjective surrogate problem of the original problem is constructed without involving preferences. As a result, the surrogate problem can be solved with any appropriate interactive method. The focus of the surrogate-based methods developed in [35, 41, 49, 53, 55] is to find the most preferred solution for a DM rather than to approximate the entire Pareto frontier. As can be seen, a few methods have considered the role of the DM during the solution process. Developing interactive methods to solve computationally expensive MOPs can be considered as a future research direction.

Since the year 2013, surrogate-based methods have been developed in [56, 57, 58, 59, 60, 61]. The method developed in [56] applies Kriging to approximate each objective function. A utility function is introduced based on which new points are selected for updating surrogates. In [57, 58], by using the Gaussian process, new criteria based on the Vorob'ev expectation and uncertainty reduction principles, respectively, have been developed to select new sample points and update surrogates. In [59, 60], solving MOPs with computationally expensive constraint functions is considered by devising new criteria to assess the feasibility of new sample points for updating surrogates. In [61], the idea of efficient global optimization (EGO) for computationally expensive single objective optimization problems is extended for computationally expensive MOPs.

To summarize, based on our observations, we conclude that the following challenges require research efforts:

1. solving problems that include nonconvex and disconnected Pareto frontiers,
2. developing interactive surrogate-based methods to provide the most preferred solution for a DM,
3. solving high-dimensional problems in terms of number of objective functions and decision variables,
4. solving problems that include non-differentiable black-box functions, and
5. solving problems that include black-box constraint functions.

In this thesis, in order to fill the gap in the above enumerated shortcomings (except the last one), we first focused on the first and second challenges. In this regard, we developed the SURROGATE-ASF method [PII] in which the DM provides his/her preferences in the form of a reference point. SURROGATE-ASF is the first interactive surrogate-based method which can provide the most preferred solution for a DM in both the decision and objective spaces during the solution process. Numerical results showed that SURROGATE-ASF performs well on problems including nonconvex and/or disconnected Pareto frontier, at most 5 objective functions and 12 decision variables.

When interacting with the DM to solve problems with a large number of objective functions, (s)he should provide a reference point including a large number of aspiration levels. As we discuss in [PIII], this could impose a cognitive load on the DM. This led us to develop ANOVA-MOP to solve problems with high-dimensional decision and objective spaces that include the preferences of the DM. This is also the first method in the literature that can solve MOPs with any number of objective functions. The metamodeling technique employed in ANOVA-MOP is capable of approximating differentiable black-box functions and conducting sensitivity analysis on such functions. This has motivated us to investigate how to approximate non-differentiable black-box functions and elicit sensitivity analysis information. To do so, in [PIV], we review the potential of applying the T-splines technique to be incorporated in the ANOVA-MOP method to solve computationally expensive MOPs. The mathematical properties of T-splines are limited to functions with two and three decision variables. In this thesis, we

do not discuss any actual method on function approximation through T-splines. To summarize, by developing the above-mentioned methods, we deal with the above-mentioned challenges.

## 4 THE SURROGATE-ASF METHOD

Some of the challenges identified in [PI] are to develop interactive surrogate-based methods as well as capturing a nonconvex and disconnected Pareto frontier. In addition to these challenges, when solving computationally expensive problems by taking into account the preferences of the DM, (s)he may wish to explore the entire Pareto frontier, different regions or a particular region. This means that an interactive surrogate-based method should not tie the hands of the DM during the solution process. In addition, the DM may wish to have solutions corresponding to his/her preferences in both the decision and the objective spaces simultaneously. Moreover, during the solution process, the DM should not wait for a long time to see solutions corresponding to his/her preferences. The SURROGATE-ASF method [PII] is the first method developed in the literature to handle such challenges. SURROGATE-ASF incorporates the preferred ranges of the objective functions provided by the DM. These preferred ranges for each objective function are given as sub-intervals. The lower and upper bounds of a sub-interval show the best and worst values of each objective function that the DM is interested in. Using two vectors consisting of the worst and the best values in the sub-intervals, we form vectors to be used instead of the utopian and nadir vectors corresponding to the region of interest for the DM in the objective space. If the DM does not provide any preferred range, one may incorporate an estimation of the utopian and nadir vectors.

### 4.1 Basis Elements

SURROGATE-ASF is a two-phase method involving initialization and decision-making phases. In the initialization phase, a set of non-dominated solutions in the decision and objective spaces are obtained within the preferred ranges given by the DM. These solutions are then utilized to form a finite number of hyper-boxes in the decision space. Within each individual hyper-box, the ASF (2) is approximated and a surrogate function is built. When approximating the ASF,



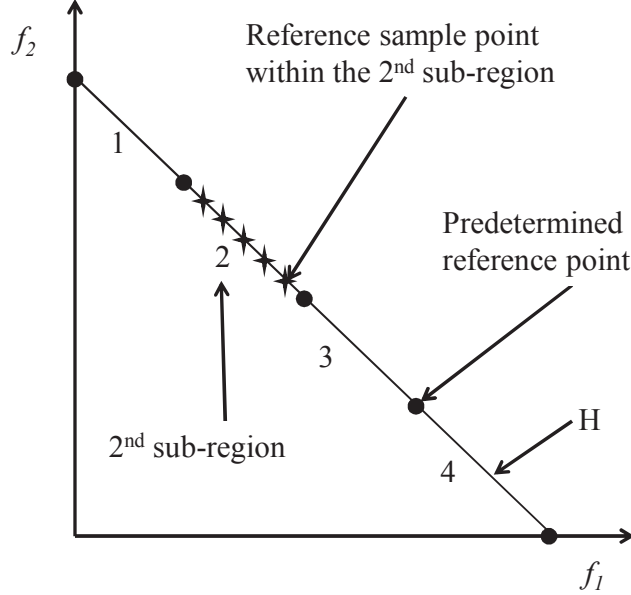


FIGURE 5 A convex hull with 5 predetermined reference points for a bi-objective problem. (Color online)

the decision variables of problem (1) and the aspiration levels appearing in the ASF are treated as variables of the surrogate function. In the decision-making phase, the single-objective surrogate functions built in the initialization phase are employed to form computationally inexpensive single-objective optimization problems to interact with the DM.

After preferred ranges are given by the DM, the extreme points corresponding to this region in the objective space are calculated. We then find the solutions in the decision space corresponding to these extreme points by solving problem (3) in which the extreme points are considered as reference points. To approximate the ASF, we require sample points for the aspiration levels and the decision variables. Before expressing steps involved in SURROGATE-ASF, however, we first discuss an interesting property regarding problem (3). This property leads us to find regions in the decision and objective spaces to select sample points.

Suppose  $H$  denotes the convex hull of all individual extreme points. Figures 5 and 6 depict a convex hull in  $\mathbb{R}^2$  and  $\mathbb{R}^3$ , respectively. The convex hull is formed by a convex combination of all extreme points in the objective space. Let  $\bar{z}^*$  be a reference point given by a DM, and  $\mathbf{x}^*$  be the corresponding optimal solution of problem (3). Then we have,  $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x} \in S} [\max_{i=1, \dots, k} (w_i (f_i(\mathbf{x}) - \bar{z}_i^*))]$ . On the other hand,  $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x} \in S} [\max_{i=1, \dots, k} (w_i (f_i(\mathbf{x}) - \bar{z}_i^*)) + t]$ , for all  $t \in \mathbb{R}$ . Thus

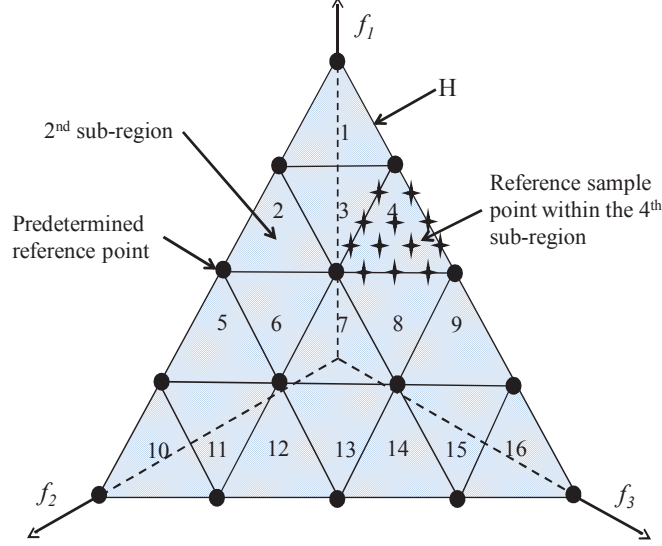


FIGURE 6 A convex hull with 15 predetermined reference points for a three-objective problem. (Color online)

$$\begin{aligned}
 \mathbf{x}^* &\in \operatorname{argmin}_{\mathbf{x} \in S} \left[ \max_{i=1, \dots, k} (w_i (f_i(\mathbf{x}) - \bar{z}_i^*)) + t \right] = \operatorname{argmin}_{\mathbf{x} \in S} \left[ \max_{i=1, \dots, k} (w_i (f_i(\mathbf{x}) - \bar{z}_i^*) + t) \right] \\
 &= \operatorname{argmin}_{\mathbf{x} \in S} \left[ \max_{i=1, \dots, k} (w_i (f_i(\mathbf{x}) - \bar{z}_i^* + \frac{t}{w_i})) \right] = \operatorname{argmin}_{\mathbf{x} \in S} \left[ \max_{i=1, \dots, k} (w_i (f_i(\mathbf{x}) - (\bar{z}_i^* - \frac{t}{w_i}))) \right] \\
 &= \operatorname{argmin}_{\mathbf{x} \in S} \left[ \max_{i=1, \dots, k} (w_i (f_i(\mathbf{x}) - \bar{z}_i^{**})) \right],
 \end{aligned}$$

where  $\bar{z}^{**}$  is considered as an arbitrary reference point on the ray passing through  $\bar{z}^*$  in parallel with  $\mathbf{w} = (w_1, \dots, w_k)^T$ . This means that the optimal solution of problem (3) for a reference point  $\bar{z}^*$  given by a DM is also the optimal solution of problem (3) for any reference point on the ray passing through  $\bar{z}^*$  in parallel with  $\mathbf{w}$  (see Figure 7). The projected reference point  $\bar{z}^{**}$  on  $H$  is obtained as follows:

$$\bar{z}^{**} \in \operatorname{argmin}_{\mathbf{h} \in H} \left\| \left( \frac{h_1 - \bar{z}_1^*}{w_1}, \dots, \frac{h_k - \bar{z}_k^*}{w_k} \right) \right\|, \quad (4)$$

where  $\|\cdot\|$  is the Euclidean norm and  $\mathbf{h} = (h_1, \dots, h_k)$ .

The above discussion shows that any reference point given by the DM within the preferred ranges of objective functions can be projected onto the convex hull  $H$ . By projection we mean finding the closest point on  $H$  along the direction  $\mathbf{w}$ . Thus, we select sample points for the aspiration levels as reference sample points on the convex hull  $H$ . To select reference sample points, we first generate a set of evenly distributed predetermined reference points including the extreme points on the convex hull  $H$ . We denote this set by  $\bar{Z}^P$ .

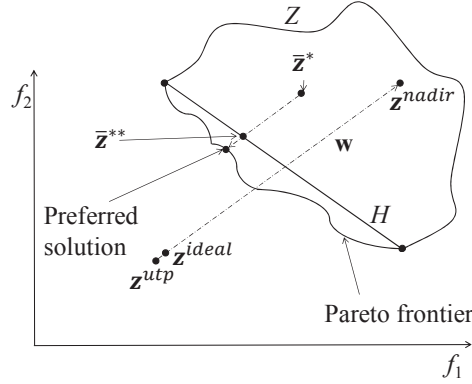


FIGURE 7 Projection of a given reference point onto the convex hull.

For each individual predetermined reference point in  $\bar{Z}^P$ , we find the corresponding reference solution (non-dominated solution) in the decision space by solving problem (3). Problem (3) is a computationally expensive single-objective optimization problem. This problem is solved by applying a surrogate-based single-objective optimization method. The choice of this method affects the performance of SURROGATE-ASF. In fact, this choice allows one to apply an appropriate state-of-the-art single-objective surrogate-based optimization method which (s)he believes to be the best. The reference solutions obtained are then evaluated with the original functions.

We denote the sets of reference solutions in the decision and the objective spaces by  $X^P$  and  $F^P$ , respectively. The points in  $\bar{Z}^P$  and  $X^P$  decompose the convex hull  $H$  and the decision space into a finite number of sub-regions and hyper-boxes, respectively. Figures 5 and 6 depict two convex hulls for bi- and three-objective problems, sub-regions on the convex hulls and predetermined reference points. Figure 8 shows hyper-boxes corresponding to the convex hull in Figure 5. In what follows, we denote each hyper-box by  $S^a$ , for  $a = 1, \dots, r$ , where  $r$  is the number of hyper-boxes (sub-regions). Once hyper-boxes and sub-regions are formed, sample points for the decision variables and the aspiration levels are selected within the hyper-boxes and the sub-region. By using these selected samples, the ASF in (2) is approximated. We denote the resulting single-objective surrogate function corresponding to each hyper-box by  $\widetilde{ASF}^a$ , for  $a = 1, \dots, r$ .

In the decision-making phase, surrogate functions built in the initialization phase are employed to interact with the DM. To do so, first, we show the DM the ideal and the nadir vectors, if (s)he did not provide any ranges. We also assume that reference points given by the DM are within the ranges. In any case, the DM specifies a reference point  $\bar{z}^*$ . In SURROGATE-ASF, the interaction with the DM corresponds to that of the reference point method developed in [8]. The projection of the reference point  $\bar{z}^*$  onto the convex hull  $H$  denoted by  $\bar{z}^{**}$  is obtained by solving problem (4). One should note that in practice, problem (4) is not solved. Instead, we can generate a large number of evenly distributed points on  $H$ . We then choose the point that minimizes the objective function of problem (4). Let us

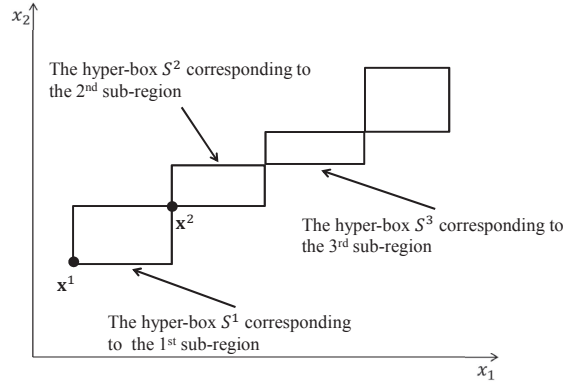


FIGURE 8 Hyper-boxes in the decision space corresponding to the sub-regions on the convex hull in Figure 5. For instance, solutions  $\mathbf{x}^1$  and  $\mathbf{x}^2$  form  $S^1$ .

denote the number of the sub-region and the hyper-box corresponding to  $\bar{\mathbf{z}}^{**}$  by  $\tilde{a}$ . The following surrogate problem is then formulated and solved by employing any appropriate single-objective optimization method:

$$\underset{\mathbf{x} \in S^{\tilde{a}}}{\text{minimize}} \quad \widetilde{\text{ASF}}^{\tilde{a}}(\mathbf{x}, \bar{\mathbf{z}}^{**}), \quad (5)$$

where  $S^{\tilde{a}}$  is the  $\tilde{a}^{\text{th}}$  hyper-box. The optimal solution of problem (5) is an approximation of the preferred solution in the decision space corresponding to the reference point  $\bar{\mathbf{z}}^*$  given by the DM. This solution is evaluated with the original functions and shown to the decision maker. The process of asking for a reference point from the DM, projecting it onto the convex hull and solving the corresponding surrogate problem is repeated until the most preferred solution satisfying the DM is obtained.

## 4.2 Discussion on the Performance of SURROGATE-ASF

In [PII], the performance of SURROGATE-ASF is evaluated by solving a bi-objective shape optimization problem of designing an airfoil [62] as well as ZDT [63] and DTLZ [64] academic benchmark problems. Since the objective functions in the shape optimization problem have practical meaning, we discussed the decision-making process thoroughly for this problem. The academic benchmark problems considered consist of different attributes, such as nonconvex Pareto frontiers and connected or disconnected sets of Pareto optimal solutions. For instance, ZDT4, ZDT6, DTLZ2, DTLZ3, DTLZ4 and DTLZ6 include many locally non-dominated solutions.

In the shape optimization problem and all benchmark problems, we considered the number of decision variables as  $n = 12$  and  $n = 7$ , respectively. In ZDT and DTLZ problems, we considered the number of objective functions as  $k = 2$

and  $k = 3$ , respectively. Moreover, we considered DTLZ2 with 4 and 5 objective functions.

According to the findings in [PI], we selected the method developed in [44] (henceforth called the Yun method) and the PAINT method [49] to compare the performance of SURROGATE-ASF. Since the Yun method and PAINT employ the same type of preference information in the form of a reference point, these three methods are comparable.

The Yun method is an a priori method. This means that, before solving a computationally expensive MOP, the DM has to provide a reference point. Then, the method builds a surrogate problem and finds the corresponding solution. In this way, surrogates built in the Yun method are accurate in the neighborhood of the solution obtained. If the DM provides a new reference point, the surrogates may not be accurate in the neighborhood of the solution corresponding to the new reference point. To be able to interact with the DM and to compare this method with SURROGATE-ASF, we have to run the entire method for each reference point given by the DM. To do so, the DM provided the first reference point. We then ran the Yun method and found the solution corresponding to this reference point. The DM then provided a new reference point. Using the surrogates built for the first reference point, we ran the Yun method such that by selecting new samples and updating the surrogates, they became accurate in the neighborhood of the solution corresponding to this new reference point. We repeated the Yun method as described above for all reference points given by the DM.

In the Yun method,  $\mu - \nu$ -SVR [44] is applied as the metamodeling technique. The idea of this method is to approximate each individual objective function and to formulate a surrogate multiobjective optimization problem. This surrogate problem is turned into a single-objective computationally inexpensive problem using the achievement scalarizing function (2) to be solved. To improve the accuracy of the surrogate problem, some sample points based on the Lagrangian coefficients in  $\mu - \nu$ -SVR as well as the solution of the single-objective optimization problem are selected. To train  $\mu - \nu$ -SVR, we applied the interior point algorithm implemented in MATLAB.

In the Yun method, as can be understood, a surrogate multiobjective optimization problem is formed. In fact, in this method, first each objective function is approximated and then the achievement scalarizing function is formulated, whereas in the SURROGATE-ASF method the order is opposite and only the achievement scalarizing function is approximated. The comparison between the performance of the Yun method and SURROGATE-ASF demonstrates the merits of the idea of approximating the ASF (2) as a computationally expensive objective function and then turning it into a computationally inexpensive one over the idea of first approximating each expensive objective function individually and then formulating the ASF.

The PAINT method similar to SURROGATE-ASF requires a set of non-dominated solutions to be available to construct a surrogate problem. The idea in PAINT is to build a linear mixed integer multiobjective optimization problem as a surrogate problem. During the decision-making process, the DM provides a

reference point. Then, this surrogate problem is converted into a linear mixed integer single-objective optimization using the ASF that includes the reference point given. This single-objective optimization problem is solved, and an approximated solution in the objective space is obtained. During the decision-making process, PAINT loses the connection with the decision space, i.e., the decision variable values of the approximated solution are not provided. The approximated solution is shown to the DM. If this solution does not satisfy the DM, (s)he specifies another new reference point. In PAINT, we interact with the DM until the DM accepts an approximated solution obtained.

We then follow an approach to find the corresponding solution in the decision space. To do so, the acceptable approximated solution is treated as a reference point for problem (3). This problem is solved including the original functions by using an appropriate (surrogate-based) single-objective optimization method. The optimal solution is the most preferred (approximated) solution and is provided to the DM. One should note that, although PAINT cannot provide solutions in the decision space during the intermediate interactions, one can find these solutions by repeating the described approach within each iteration. However, this can be time-consuming. In [PII], we applied the steps given in [49] and did not calculate approximated solutions in the decision space until the last iteration.

At the end of the solution process by an interactive method, the most preferred solution (which is usually a single solution) is obtained. To the best of our knowledge, there is no established quality measure in the literature that is suitable for a reference-point-based interactive method. Thus, by using the normalized Euclidean distance discussed in [PII], we compare the closeness of solutions in the objective space obtained by SURROGATE-ASF, the Yun method and PAINT with their corresponding exact solutions obtained by solving problem (3), in which the original functions (of the benchmark problems) are used.

As far as the shape optimization problem is concerned, in terms of computational burden, the DM was able to find the most preferred solution by SURROGATE-ASF, the Yun method and PAINT after 8 seconds, 32760 seconds and 346000 seconds, respectively. Moreover, SURROGATE-ASF outperformed other methods clearly in terms of solutions obtained. Regarding the benchmark problems with different characteristics discussed earlier, SURROGATE-ASF also surpassed the other methods.

As discussed in [PII], developing SURROGATE-ASF for high-dimensional problems in the objective space is a future research direction. Besides that, in such problems, the DM has to provide a reference point that includes a large number of aspiration levels. This could impose a cognitive load on the DM. In the next chapter we propose a method called ANOVA-MOP to solve such problems.

## 5 THE ANOVA-MOP METHOD

### 5.1 Motivation

In [PI], we concluded that developing a surrogate-based method to solve computationally expensive MOPs with high-dimensional objective and decision spaces taking into account the DM's preferences is a future research direction. By increasing the number of decision variables, high-dimensional problems become computationally more demanding. This is imposed by an issue known as the *curse of dimensionality* [32]. Although the computational cost can be reduced by using surrogates to replace computationally expensive functions, the issue of curse of dimensionality still remains. Due to this issue, when one is looking for a solution in a high-dimensional decision space, an exhaustive search is discouraged even in single-objective optimization problems [65, 66]. Thus an exact solution can hardly be obtained. This motivated us to develop the ANOVA-MOP method in [PIII] to solve high-dimensional MOPs. In this method, a high-dimensional MOP (under some conditions to be discussed later) is decomposed into a finite number of sub-problems with a lower dimension in the decision and objective spaces. The sub-problems are solved, and the solutions obtained are composed to form solutions for the original problem.

In high-dimensional problems, to find the most preferred solution for the DM, the cognitive task of decision-making includes analyzing and comparing solutions with a large number of components. In the decision-making phase, the DM should remember the preferences (s)he already explored. The authors in [67] discuss that cognitive tasks demand adequate ability to remember preferences as long as they are processed. This leads to the question of whether, from a cognitive point of view, the DM would be comfortable providing his/her preferences and analyzing solutions to such problems. To the best of our knowledge, decision-making through non-nature-inspired surrogate-based methods to solve computationally expensive MOPs with high-dimensional objective space has not been addressed in the literature.

The curse of dimensionality is tied with the sparsity-of-effects principle [68].

This means that typically a subset of decision variables, called *effective variables*, contributes most significantly to objective function values, while other decision variables, called *ineffective variables*, have a minor effect. According to [69], the effective variables in one region of the decision space may not be the same as those in another, and thus, global sensitivity analysis [70] has been developed. Similarly, decision variables may cooperate to provide interaction effects, in which the effectiveness of a subset of decision variables may not lead to the sum of individual decision variable effectivenesses. Sobol' and total sensitivity indices to be described in what follows have been developed to quantify decision variable effectiveness, clarifying these issues.

Global sensitivity analysis can be done through e.g., the functional analysis of variance decomposition (functional ANOVA decomposition) [71]. Suppose an arbitrary function  $g : S \rightarrow \mathbb{R}$  is square integrable, i.e.,  $\int_S g(\mathbf{x})^2 d\mathbf{x} < \infty$ . The *functional ANOVA decomposition* of  $g$  is

$$g(\mathbf{x}) = g_0 + \sum_i g_{(i)}(x_i) + \sum_{i < j} g_{(i,j)}(x_i, x_j) + \dots + g_{(1,\dots,n)}(\mathbf{x}), \quad (6)$$

where  $g_0 = \int_S g(\mathbf{x}) d\mathbf{x}$ , if

$$\int_{S_{i_u}} g_{(i_1,\dots,i_e)}(x_{i_1}, \dots, x_{i_e}) dx_{i_u} = 0, \quad (7)$$

for all  $u = 1, \dots, e$ , ( $e \geq 1$ ), and  $1 \leq i_1 < \dots < i_e \leq n$ . We call  $g_{(i)}$  a *main effect* and  $g_{(i_1,\dots,i_e)}$ , an *interaction* of order  $e$ . This results in decomposing the variance of  $g$ , i.e.,  $\text{var}(g) = D_0 = \int_S g^2(\mathbf{x}) d\mathbf{x} - g_0^2$ , into  $2^n - 1$  *variance components*

$$D_0 = \sum_i D_{(i)} + \sum_{i < j} D_{(i,j)} + \dots + D_{(1,\dots,n)}, \quad (8)$$

where  $D_{(i_1,\dots,i_e)} = \text{var}[g_{(i_1,\dots,i_e)}(x_{i_1}, \dots, x_{i_e})] = \int_{\prod_{u=1}^e S_{i_u}} g_{(i_1,\dots,i_e)}^2(x_{i_1}, \dots, x_{i_e}) dx_{i_1} \dots dx_{i_e}$ . The *Sobol' indices* [72] are calculated as

$$l_{(i_1,\dots,i_e)} = D_{(i_1,\dots,i_e)} / D_0, \quad (9)$$

for all  $(i_1, \dots, i_e) \in \Delta = \{(r_1, \dots, r_u) : 1 \leq r_1 < \dots < r_u \leq d, u \geq 1\}$ ,  $e = 1, \dots, n$ . The *total sensitivity indices*, which are measures of the contribution of each decision variable  $x_j$ , for  $j = 1, \dots, n$ , in the value of  $g$ , are given by

$$l_j^T = \sum_{(i_1,\dots,i_e) \in \Delta: j \in \{i_1,\dots,i_e\}} D_{(i_1,\dots,i_e)} / D_0, \quad j = 1, \dots, n. \quad (10)$$

Sobol' indices and total sensitivity indices are typically approximated in practice. In this thesis, they are called estimated Sobol' indices and estimated total sensitivity indices, respectively.

As can be seen in Equation (6), by the functional ANOVA decomposition, an objective function is decomposed hierarchically into a sum of terms depending on subsets of decision variables, based on their contribution to the total variance of the objective function. As pointed out in [68], higher interaction terms have a minor effect on function values such that typically 80% of the effects are due to 20% of the decision variables.



## 5.2 Basic Elements

In the ANOVA-MOP method, we reduce the dimension of the search domain in the decision space through the approach of applying the functional ANOVA decomposition and calculating the total sensitivity indices. By this approach we rank decision variables according to their effect on objective function values. Based on the sparsity-of-effects principle, one can typically observe that only a limited number of decision variables has a reasonable effect on objective function values. Thus, the remaining variables can be ignored without too significant losses in the solution quality. In this way, we deal with simpler and tractable problems. Before ANOVA-MOP, such an approach had only been employed in single-objective optimization problems. In fact, in MOPs, effective variables for one objective function may be ineffective for some other objectives.

Suppose  $O = \{1, \dots, k\}$  and  $D = \{1, \dots, n\}$  are the sets of indices of the objective functions and decision variables in problem (1), respectively. A general sketch of steps of ANOVA-MOP is as follows: first, for each individual objective function in problem (1), we build a surrogate function that is sufficiently accurate everywhere over the entire decision/objective space, as discussed in [73]. We then replace all functions in problem (1) with these surrogate functions. In this way, we formulate the following surrogate MOP:

$$\underset{\mathbf{x} \in S}{\text{minimize}} \{ \tilde{f}_1(\mathbf{x}), \dots, \tilde{f}_k(\mathbf{x}) \}, \quad (11)$$

where  $\tilde{f}_i : S \rightarrow \mathbb{R}$ , for  $i \in O$ , is a surrogate function corresponding to  $f_i$  in problem (1). The dimension of the decision and objective spaces in problem (11) is the same as those in problem (1). In what follows, by an objective function, we mean a surrogate function in problem (11).

Corresponding to each objective function, we estimate the total sensitivity indices of each decision variable. Suppose  $l_{i,j}^T$  denotes the estimated total sensitivity index of the  $j^{\text{th}}$  decision variable in the  $i^{\text{th}}$  objective function. We then consider an effective total sensitivity indices matrix (SM) where

$$SM = [l_{i,j}^T], \quad i \in O, \quad j \in D. \quad (12)$$

Figure 9 shows an example of SM for a problem with  $k = 8$  and  $n = 10$ . Decision variables with zero-value total sensitivity indices are treated as *ineffective* variables and others are treated as *effective* variables. We then form an incident matrix  $M = [m_{i,j}]$ ,  $i = 1, \dots, k$ ,  $j = 1, \dots, n$ , as:

$$m_{i,j} = \begin{cases} 1, & \text{if decision variable } x_j \text{ is effective for objective function } \tilde{f}_i, \\ 0, & \text{if decision variable } x_j \text{ is ineffective for objective function } \tilde{f}_i. \end{cases} \quad (13)$$

Figure 10 (a) depicts an incident matrix  $M$  corresponding to the effective total sensitivity indices matrix (SM) in Figure 9. In Figure 10 (a), for example, the decision variables  $x_2$  and  $x_5$  are effective variables for objective functions  $\tilde{f}_3$  and  $\tilde{f}_6$  and

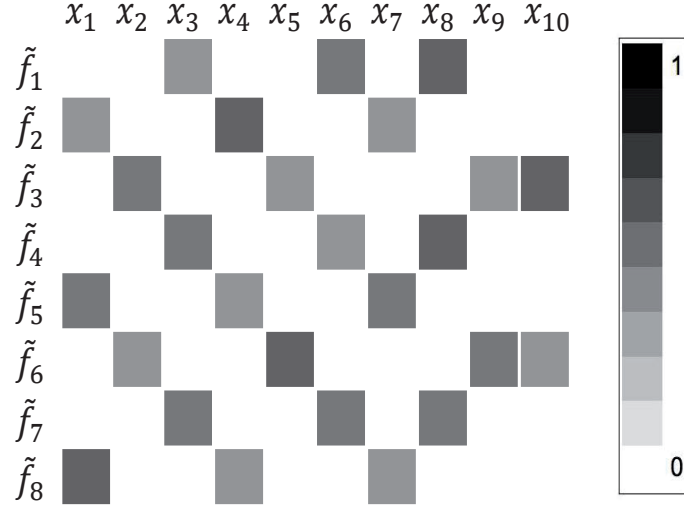


FIGURE 9 An example of an effective total sensitivity indices matrix (SM). The bar shows the range of the estimated total sensitivity indices.

ineffective for other objective functions. After effective and ineffective variables are identified, objective functions with the same effective variables are grouped together. This leads to forming a block-diagonal matrix with independent blocks (i.e., each decision variable and objective function belongs to only one block). Figure 10 (b) shows a block-diagonal matrix with three independent blocks corresponding to the incident matrix  $M$  in Figure 10 (a).

Sub-problems are formed based on objective functions and decision variables whose indices are grouped in the same block. In fact, we say that problem (11) is decomposable if and only if

1. There exist subsets  $O_a$  and  $D_a$  for  $a = 1, \dots, d$ , such that  $O = \bigcup_{a=1}^d O_a$ ,  $\bigcap_{a=1}^d O_a = \emptyset$ ,  $D = \bigcup_{a=1}^d D_a$  and  $\bigcap_{a=1}^d D_a = \emptyset$ ,
2.  $m_{i,j} = 1$  implies that there exist an index  $\bar{a}$  such that  $i \in O_{\bar{a}}$  and  $j \in D_{\bar{a}}$ .

In [PIII], we show that if problem (11) is decomposable, in the simplest case, matrix  $M$  has at least  $k + n - 2$  zero-value components. As the result of decomposition, in each sub-problem, objective functions have the same effective variables. In this way, problem (11) is decomposed into a limited number of sub-problems which have a lower dimension in the decision and objective spaces. Sub-problems are then solved independently from each other by any appropriate multiobjective optimization method. The solutions obtained are composed to form solutions for problem (11). In [PIII], we discuss the relation between the solutions composed and the solutions of problem (11).

The ANOVA-MOP method can be applied as an interactive or a non-interactive method. As a non-interactive method, each sub-problem can be solved by

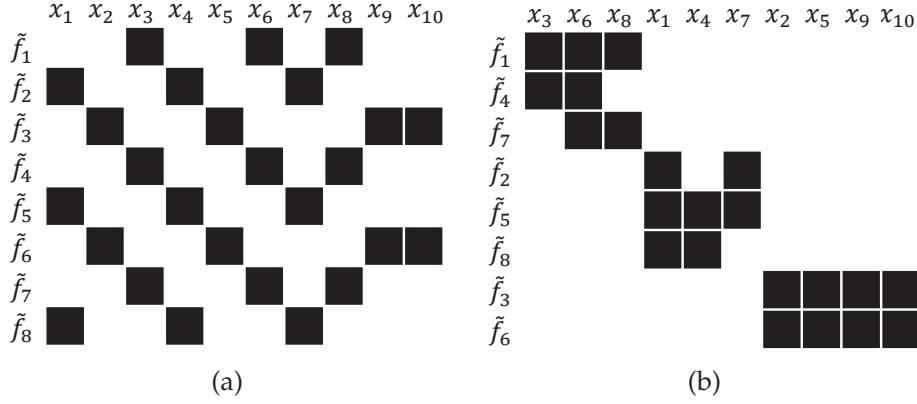


FIGURE 10 (a) An incidence matrix  $M$ , (b) a decomposition corresponding to the matrix  $M$  in (a).

using multiobjective optimization methods, and a set of Pareto optimal solutions is obtained. Without loss of generality, suppose problem (11) is decomposed into  $d$  sub-problems. Let  $\bar{X}^a$  ( $\bar{F}^a$ ),  $a = 1, \dots, d$ , denote the set of Pareto optimal solutions in the decision space (objective space) of the  $a^{\text{th}}$  sub-problem. In [PIII], we show that  $\bar{X} = \bar{X}^1 \times \bar{X}^2 \times \dots \times \bar{X}^d$  ( $\bar{F} = \bar{F}^1 \times \bar{F}^2 \times \dots \times \bar{F}^d$ ) is the set of Pareto optimal solutions of problem (11) in the decision (objective) space in which  $\times$  stands for the Cartesian set product.

When using ANOVA-MOP as an interactive method, any appropriate interactive method can be applied to solve sub-problems and find solutions corresponding to the DM's preferences. As mentioned in [74], providing preferences by the DM in a form of a reference point is a valid way of expressing preferences and is meaningful for the DM. Thus, in [PIII] we study the interactive aspect of ANOVA-MOP as a reference-point-based method in which the DM provides his/her preference information in the form of a reference point. Considering the  $a^{\text{th}}$  sub-problem, for  $a = 1, \dots, d$ , the indices of the objective functions in this sub-problem are shown to the DM. Then, (s)he specifies a reference point including aspiration levels corresponding to these objective functions. Suppose  $\bar{z}^{a*}$  is a reference point corresponding to the  $a^{\text{th}}$  sub-problem given by the DM. We then formulate the following computationally inexpensive single-objective optimization problem:

$$\underset{\mathbf{x}^a \in S^a}{\text{minimize}} \quad \text{ASF}(\mathbf{x}^a, \bar{z}^{a*}), \quad (14)$$

where  $\mathbf{x}^a$  is the vector of effective decision variables of the  $a^{\text{th}}$  sub-problem and  $S^a$  is the feasible decision set of the  $a^{\text{th}}$  sub-problem formed based on the effective variables in this sub-problem. The ASF is the achievement scalarizing function (2) formed based on the objective functions considered in the  $a^{\text{th}}$  sub-problem.

After specifying a reference point and formulating problem (14), this problem is solved by a single-objective optimization method which is appropriate for the characteristics of the problem in question. One should note that this single-objective optimization problem has a lower dimension in the decision space in

comparison with problem (11). The solution obtained by solving problem (14) is shown to the DM. We ask the DM to provide his/her preferences corresponding to the objective functions in the  $a^{th}$  sub-problem and solve problem (14) until the most preferred solution for the DM is obtained as in reference-point-based methods [1].

Let us denote the optimal solution of problem (14) corresponding to the  $a^{th}$  sub-problem by  $\bar{\mathbf{x}}^{a*}$ , for  $a = 1, \dots, d$ . In [PIII] we show that

$$\mathbf{x}^* = (\mathbf{x}^{1*}, \dots, \mathbf{x}^{d*}) \in \underset{\mathbf{x} \in S}{\operatorname{argmin}} \operatorname{ASF}(\mathbf{x}, \bar{\mathbf{z}}^*), \quad (15)$$

where  $\bar{\mathbf{z}}^* = (\bar{\mathbf{z}}^{1*}, \dots, \bar{\mathbf{z}}^{d*})$ . This means that  $\mathbf{x}^*$  is the most preferred solution for the DM corresponding to  $\bar{\mathbf{z}}^*$ , which is the reference point containing aspiration levels for all objective functions simultaneously.

When surrogate functions in problem (11) emulate computationally expensive objective functions in problem (1) precisely,  $\bar{X}$  and  $\mathbf{x}^*$  are also the set of Pareto optimal solutions and the most preferred solution for the DM in problem (1), respectively. Otherwise, they are an approximation of the set of Pareto optimal solutions and the most preferred solution, respectively.

The decomposability of problem (11) is connected with the sparsity of M, and matrix M is formed based on SM. In some problems, M may not be sparse enough to be decomposed. For example,  $m_{i,j} \neq 0$  for all  $i \in O$  and  $j \in D$ . In this case, all variables are effective for all objective functions. However, based on the sparsity-of-effects principle, we expect that for some  $i \in O$  and  $j \in D$ ,  $l_{i,j}^T \ll 1$  in SM. In [PIII], we say that the objective function  $\tilde{f}_i$  is  $\epsilon$ -insensitive (for some  $\epsilon > 0$ ) with respect to the decision variable  $x_j$ . We provide an approach by which decision variables with  $l_{i,j}^T \ll 1$  are treated as ineffective variables. As a result, a sparse enough reduced incident matrix (RM) is obtained which can be decomposed. Based on RM, we decompose problem (11) into a finite number of sub-problems. Sub-problems are then solved in the same manner as if problem (11) were decomposable. In this case, we say that problem (11) is  $\epsilon$ -decomposable. As mentioned in [PIII], solutions obtained by ANOVA-MOP for an  $\epsilon$ -decomposable problem are approximated solutions for problems (1) and (11).

Before further discussion, we need to define some concepts. Following [75], for  $\epsilon > 0$ , we say that a solution  $\mathbf{x}^1 = (x_1^1, \dots, x_n^1) \in S$   $\epsilon$ -dominates a solution  $\mathbf{x}^2 = (x_1^2, \dots, x_n^2) \in S$  if and only if for all  $i \in O$ ,  $\tilde{f}_i(\mathbf{x}^1) \leq \tilde{f}_i(\mathbf{x}^2) + \epsilon$ , and there exists an index  $j \in O$  such that  $\tilde{f}_j(\mathbf{x}^1) < \tilde{f}_j(\mathbf{x}^2) + \epsilon$ . Suppose  $X_{po} \subset S$  is the set of Pareto optimal solutions in the decision space of problem (11). In [PIII], for  $\epsilon > 0$ , we say that  $X_s \subset S$  is a *synthetic Pareto optimal set* if and only if for all  $\mathbf{x}_{po} \in X_{po}$ , there exist  $\mathbf{x}_s \in X_s$  such that  $\mathbf{x}_s$   $\epsilon$ -dominates  $\mathbf{x}_{po}$ . The image of  $X_s$  in the objective space of problem (11), i.e.,

$$\bar{F}_s := \operatorname{Im}_{\tilde{\mathbf{f}}}(X_s) = \left\{ \tilde{\mathbf{f}}(\mathbf{x}_s) \in \mathbb{R}^k \mid \mathbf{x}_s \in X_s \right\}. \quad (16)$$

is called a *synthetic Pareto frontier*.

Suppose problem (11) is  $\epsilon$ -decomposable and is decomposed into  $d$  sub-problems  $G_1, G_2, \dots, G_d$ , in which the dimension of the decision space is  $n_1, \dots, n_d$ , respectively, and the dimension of the objective space is  $k_1, \dots, k_d$ , respectively. Let us denote the set of Pareto optimal solutions in the decision space of  $G_1, \dots, G_d$  by  $\bar{X}_1, \dots, \bar{X}_d$ , respectively, and the Cartesian product of  $\bar{X}_1, \dots, \bar{X}_d$  by

$$\bar{X} := \left\{ (\mathbf{x}_1, \dots, \mathbf{x}_d) \in \mathbb{R}^{n_1} \times \dots \times \mathbb{R}^{n_d} = \mathbb{R}^n \mid \mathbf{x}_1 \in \bar{X}_1, \dots, \mathbf{x}_d \in \bar{X}_d \right\}. \quad (17)$$

In [PIII] we show that  $\bar{X}$  and  $\text{Im}_{\bar{\epsilon}}(\bar{X})$  are the synthetic Pareto optimal set and the synthetic Pareto frontier (i.e., solutions in  $\bar{X}$   $\bar{\epsilon}$  dominate solutions in  $X_{po}$  where  $\bar{\epsilon} = 2\epsilon$ ) for problem (11). While for MOPs with  $k \geq 3$ , the synthetic Pareto frontier includes one or more than one solutions, for  $k = 2$ , the synthetic Pareto frontier has only one solution. In fact, if the sets  $\bar{X}_1$  and  $\bar{X}_2$  have more than one solution, we choose one solution arbitrarily from each set and form  $\bar{X}$  (5.2). As mentioned earlier, the synthetic Pareto frontier is an approximation of the Pareto frontier of problems (1) and (11).

### 5.3 Discussion and an Example

As mentioned in [PIII], generally, the conditions of  $\epsilon$ -insensitivity and  $\epsilon$ -decomposability can be checked only when the functions given have explicit mathematical formulations and these formulations are amenable to analysis. This may not be possible when dealing with black-box functions. However, the functional ANOVA decomposition assists to identify effective variables for black-box objective functions and quantify their effectiveness.

In the literature, a Pareto optimal solution is said to be a *properly Pareto optimal solution* if impairing some objective function value is bounded by the result of improvement in some other objective function value [1]. Since solutions in the synthetic Pareto frontier are not necessarily Pareto optimal solutions for problem (11), they cannot be considered as properly Pareto optimal solutions.

In the evolutionary computation literature, there is a concept known as a *knee point* [76]. In a knee point in the Pareto frontier of problem (11), a reduction in one of the objective function value will result in a severe increment in value of at least another objective function.

In what follows, we provide a bi-objective optimization problem. This problem is not a high-dimensional one. However, to be able to visualize how ANOVA-MOP works, we consider it. Through the following example, we first show how the synthetic Pareto frontier is obtained. We then give a counter example to show that the concepts of the synthetic Pareto frontier and the knee point defined in [76] are not generally equal.

Let us consider the bi-objective problem:

$$\underset{\mathbf{x} \in [0.1, 4\pi]^2}{\text{minimize}} \{f_1(\mathbf{x}), f_2(\mathbf{x})\}$$

where

$$f_1(\mathbf{x}) = \sin(3x_1) + \gamma \cos(2x_1) \left( \frac{x_2}{x_1} - \frac{\pi}{2} \right), \quad (18)$$

$$f_2(\mathbf{x}) = \gamma \sin(2x_2) \left( \frac{x_1}{x_2} - \frac{3\pi}{5} \right) + \cos(3x_2),$$

$$\gamma = 10^{-3}.$$

By building surrogate functions  $\tilde{f}_1$  and  $\tilde{f}_2$  by e.g., the method developed in [73] (henceforth called BPC method), we form the bi-objective surrogate problem

$$\underset{\mathbf{x} \in [0.1, 4\pi]^2}{\text{minimize}} \{\tilde{f}_1(\mathbf{x}), \tilde{f}_2(\mathbf{x})\}. \quad (19)$$

We then calculate total sensitivity indices by the BPC method. As a result, we have

$$SM = \begin{bmatrix} 9.999 \times 10^{-1} & 1.000 \times 10^{-3} \\ 1.000 \times 10^{-3} & 9.999 \times 10^{-1} \end{bmatrix}, \quad M = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \quad (20)$$

Matrix M is not decomposable. However, as can be seen in Matrix SM,  $l_{1,2}^T \ll 1$  and  $l_{2,1}^T \ll 1$ . By applying the approach mentioned in [PIII], we consider  $x_2$  and  $x_1$  as ineffective variables for  $f_1$  and  $f_2$ , respectively. As a result, the following reduced incident matrix RM is obtained:

$$RM = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (21)$$

This RM is decomposable. Thus, we formulate the following two single-objective optimization sub-problems:

$$\underset{\mathbf{x} \in [0.1, 4\pi]^2}{\text{minimize}} \tilde{f}_1(\mathbf{x}) \quad (22)$$

$$\text{where } x_2 = 2\pi,$$

$$\underset{\mathbf{x} \in [0.1, 4\pi]^2}{\text{minimize}} \tilde{f}_2(\mathbf{x}) \quad (23)$$

$$\text{where } x_1 = 2\pi.$$

As mentioned earlier, the ineffective variables have no effect on the objective function values, therefore, in each sub-problem, we assign them a fixed arbitrary value within the variable range (in this example, we choose  $x_2 = 2\pi$  and  $x_1 = 2\pi$  for sub-problems (22) and (23), respectively). By solving these two sub-problems, we have:

$$\bar{X}_1 = \left\{ \frac{\pi}{2}, \frac{7\pi}{6} \right\}, \quad \bar{X}_2 = \left\{ \frac{\pi}{3}, \pi \right\}. \quad (24)$$

In this example,  $\bar{X}_1$  and  $\bar{X}_2$  have more than one solution. Let us consider the Cartesian product of these sets in the decision space of problem (19) and its image in the objective space by:

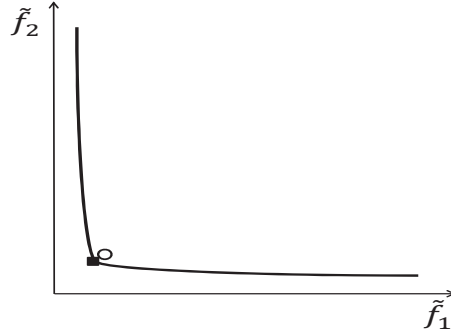


FIGURE 11 The curve, circle and square represent the Pareto frontier, synthetic Pareto frontier and the knee point in problem (19), respectively.

$$\mathfrak{X} = \left\{ \left( \frac{\pi}{2}, \frac{\pi}{3} \right), \left( \frac{\pi}{2}, \pi \right), \left( \frac{7\pi}{6}, \frac{\pi}{3} \right), \left( \frac{7\pi}{6}, \pi \right) \right\}, \quad (25)$$

$$\mathfrak{F} = \{ (-0.9991, -1.0003), (-1.0004, -1.0000), \\ (-1.0006, -0.9986), (-1.0004, -1.0000) \}. \quad (26)$$

As can be seen in (25), solutions in  $\mathfrak{X}$  are different whereas the image of two solutions is the same in  $\mathfrak{F}$  (26). Following [PIII], we have four candidate from which to choose a synthetic Pareto frontier. Suppose we choose  $x_1 = \frac{\pi}{2}$  and  $x_2 = \frac{\pi}{3}$  from  $\bar{X}_1$  and  $\bar{X}_2$ , respectively. Then  $\bar{X} = \{(\frac{\pi}{2}, \frac{\pi}{3})\}$  and the synthetic Pareto frontier is  $\{(-0.9991, -1.0003)\}$ . Figure 11 depicts the synthetic Pareto frontier and the Pareto frontier of problem (19). In this example, the BPC method managed to approximate objective functions in problem (18) accurately. Thus, the synthetic Pareto frontier and the Pareto frontier of problem (19) can also be counted for problem (18). A more general discussion on the relation between the synthetic Pareto frontier and the Pareto frontier of problems (1) and (11) is given in [PIII].

According to the definition of a knee point given in [76], we see that problem (18) has only one knee point  $(-1.0031, -1.0013)$  in the Pareto frontier of problem (19) (see Figure 11). However, by applying ANOVA-MOP, we have four candidates (i.e., solutions in  $\mathfrak{F}$  (26)) from which to choose a synthetic Pareto frontier. In general, the synthetic Pareto frontier and a knee point are not necessarily equal.

Due to the decomposition, in each phase of decision-making, a limited number of objective functions is considered. Thanks to this, the DM can focus on a subset of objective functions at a time. Therefore, it is more convenient to observe the changes in the solutions obtained after specifying a reference point. Moreover, the solution process is not cognitively as demanding as when dealing with a large number of objective functions simultaneously.

To put it briefly, we can summarize the strengths of ANOVA-MOP as:

- it can be applied on MOPs with a large number of objective functions,
- it can be employed as a non-interactive or an interactive method,

- sub-problems to be solved have a lower dimension in the decision and objective spaces in comparison to those in the original problem,
- when finding solutions corresponding to the preferences of the DM, (s)he can concentrate on a low number of objective functions at a time and can provide preferences for them rather than a large number of objective functions,
- since all surrogates are built before interaction with the DM, problem (14) is computationally inexpensive. Thus the DM can obtain solutions corresponding to his/her preferences quickly,
- not only can it be applied as a surrogate-based method, but it can also be employed to solve computationally inexpensive MOPs.

If problem (1) is not computationally expensive, there is no need to build surrogate functions, but one can apply the functional ANOVA decomposition directly on the original objective functions to identify effective and ineffective variables. Then the Pareto frontier or the synthetic Pareto frontier for problem (1) can be obtained as discussed. To the best of our knowledge, ANOVA-MOP is the first method in the literature with the above enumerated strengths.

Solving high-dimensional computationally inexpensive MOPs is an ongoing research topic in the literature (see e.g., [75, 77, 78]). These methods do not typically perform well on high-dimensional computationally expensive MOPs with a limited number of function evaluations. In ANOVA-MOP, when problem (11) is formed, one can apply such a method (i.e., methods developed for solving high-dimensional computationally inexpensive MOPs) to solve it. In this situation, typically, the set of Pareto optimal solutions (or the set of non-dominated solutions) in the objective space contains many solutions that have a large number of components (i.e.,  $k$  components). Even when this set is obtained with a satisfactory accuracy, its exploration and analysis by the DM when  $k > 3$  is challenging. Solving sub-problems (with a lower dimension in the decision and objective space) separately can facilitate analyzing and visualizing solutions in an objective space with a lower dimension. Because the focus in this thesis is on solving computationally expensive problems, we do not evaluate the performance of ANOVA-MOP on computationally inexpensive MOPs.

When problem (11) is ( $\epsilon$ -)decomposable, the matrix (R)M can be reordered such that it becomes block-diagonal with at least two independent blocks, i.e., the index of each objective function and each effective variable belongs to only one block. In this way, there is no effective variable belonging to two or more sub-problems. However, there are problems where the matrix (R)M cannot be reordered to form independent blocks, i.e., the index of at least one effective decision variable belongs to at least two blocks. As a result, at least two sub-problems have at least one common effective variable. This can be expected to be the case in most real-world problems. Although ANOVA-MOP in its current format may not be applicable for such problems, it can be directly applied for a real-world problem inheriting the decomposability condition. ANOVA-MOP has been a fundamental effort toward developing the next version of ANOVA-MOP



to solve real-world problems as well as to solve problems with computationally expensive constraint functions. These are future research directions.

To build surrogates and conduct sensitivity analysis in ANOVA-MOP, one can apply any other suitable method by taking into account the characteristics of the given problem rather than the BPC method employed in this chapter and [PIII]. In the next chapter, we discuss a new technique termed *T-splines* [79] as a potential metamodeling technique to employ in ANOVA-MOP.

## 6 POTENTIAL FUTURE DEVELOPMENT FOR ANOVA-MOP

### 6.1 Motivation

In Chapter 5, we have discussed the ANOVA-MOP method [PIII] for solving high-dimensional computationally expensive or inexpensive MOPs. In ANOVA-MOP, sensitivity analysis is a key element. In this chapter, we discuss how ANOVA-MOP can benefit from sensitivity analysis methods developed in the literature. We then discuss the potential of a new metamodeling technique called *T-splines*, presented in [PIV], to be incorporated into ANOVA-MOP for surrogate building and sensitivity analysis.

The mathematical properties of T-splines have been studied for functions with two and three decision variables [80]. Computationally expensive MOPs typically contain more than three decision variables. Thus, before applying T-splines for such problems, we may first develop T-splines for functions with more than three decision variables. This is an initial step for studying T-splines for surrogate building and sensitivity analysis in computationally expensive problems. We do not yet provide any actual method on function approximation through T-splines.

In [73], methods developed to conduct sensitivity analysis are divided into two categories, i.e., methods that do not employ surrogates and methods that apply surrogates. Methods developed in [70, 72, 81, 82] are examples of the former category, and methods developed in [73, 69, 83, 84, 85, 86, 87, 88] are examples of the latter one. A review of sensitivity analysis methods is given in, e.g., [89].

As mentioned in [73], methods in the first category typically require more function evaluation in comparison with those in the second category. Moreover, as discussed in [73], methods in the first category generally rely on minimal assumptions about a given function, whereas methods in the latter one depend on some assumptions (e.g., differentiability). As a result, methods in the first category can be applied for computationally inexpensive functions which can be e.g., discontinuous or non-differentiable, and methods in the second category are more

suitable for computationally expensive smooth functions.

When ANOVA-MOP is employed to solve a computationally inexpensive MOP, one can apply a method belonging to the first category to do sensitivity analysis. In this way, even if the problem includes non-differentiable functions, information obtained from the sensitivity analysis can be used to decompose the problem and form sub-problems as discussed in Chapter 5 and [PIII]. To apply ANOVA-MOP on problems with computationally expensive differentiable functions, methods in the second category are more suitable to be employed for sensitivity analysis.

MOPs may contain computationally expensive non-differentiable functions. In [PI], approximating such functions was identified as a future research direction. In this situation, methods in the first category that rely on, e.g., differentiability may not provide accurate enough information from the sensitivity analysis for computationally expensive non-differentiable functions. As a result, in ANOVA-MOP, sub-problems may not be formed accurately and solutions obtained may not be reliable. In this chapter, we discuss T-splines which are built based on spline functions as a potential technique to incorporate into ANOVA-MOP. In this way, we expect that sensitivity analysis on computationally expensive non-differentiable functions can provide accurate enough information and that computationally expensive MOPs including these kinds of functions can be solved efficiently.

In the literature, spline functions have been applied as one way to approximate computationally inexpensive non-differentiable functions [90]. To build a spline function, a set of linearly independent piecewise-polynomial functions of an arbitrary degree is considered. These piecewise-polynomial functions form basis functions and are called *B-splines*. Each linear combination of these B-splines constructs a spline function. B-splines are built based on a set of knots. Thanks to the knot set, the spline function becomes flexible enough to approximate non-differentiable functions. High-degree polynomial functions can also be approximated by low-degree B-splines [91].

Bivariate functions are approximated through bivariate B-splines [90]. In bivariate B-splines, a set of basis functions is constructed corresponding to each decision variable. By using these sets of basis functions, tensor spline bases are formed and bivariate B-splines are obtained. In bivariate B-splines, a refinement of the set of basis functions for one of the decision variables means adding some new basis functions into this set. This assists to improve the accuracy of the approximation. Nevertheless, refining a set of basis functions corresponding to one of the decision variables leads to an exponential increase in the number of tensor spline bases in bivariate B-splines. This is a drawback in bivariate B-splines, which do not provide local refinement [79, 91, 92].

To overcome the drawback of bivariate B-splines, T-splines have been developed in computer aided design (CAD) [79]. T-splines follow the same flexibility as bivariate B-splines, but the refinement in T-splines is local. To the best of our knowledge, so far in the literature, T-splines have not been used to approximate computationally expensive non-differentiable functions when solving multiob-

jective optimization problems. In [PIV], we provide a review on T-splines as a potential technique to approximate these kinds of functions and to estimate total sensitivity indices.

## 6.2 Function Approximation Through Spline Functions

Suppose  $U = \{u_1, u_2, u_3, \dots, u_{e+p+1}\}$  is a set  $e + p + 1$  non-decreasing number; then  $U$  is called a *global knot vector*,  $u_i$  for  $i = 1, e + p + 1$  are called *knots*,  $p$  is the degree of B-splines functions and  $e$  is the *number of B-splines* needed to build a spline function. A knot  $u_i$  can be repeated in  $U$  at most  $p + 1$  times. If  $u_1 = u_2 = \dots = u_{p+1}$  and  $u_{e+1} = u_{e+2}, \dots = u_{e+p+1}$ , then  $U$  is called an *open global knot vector*.

The *B-splines*  $N_{i,p}(x)$  (here  $x$  means only one decision variable) of degree  $p$  for a given global knot vector  $U$  are defined by the Cox-de Boor recursion formula [90] as follows:

$$N_{i,p}(x) = \begin{cases} 1 & x \in [u_i, u_{i+1}), \\ 0 & \text{otherwise,} \end{cases} \quad (27)$$

$$N_{i,p}(x) = \frac{x - u_i}{u_{i+2} - u_i} N_{i,p-1}(x) + \frac{u_{i+p+1} - x}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(x). \quad (28)$$

In the above equations, a term with a zero-denominator is considered to be zero. If  $p = 3$ , we call  $N_{i,3}$  a cubic-basis function. Let us suppose that the multiplicity of a knot  $u_j$  in  $U$  is  $\kappa_j$ . According to [90],  $N_{i,p}$  is  $p - \kappa_j$  times continuously differentiable at  $u_j$ . This means that if  $\kappa_j = p$ , then the spline function is continuous but not differentiable at  $u_j$ . This is an important property of B-splines by which e.g. non-differentiable functions can be approximated.

Now, suppose that  $f : [a, b] \rightarrow \mathbb{R}$  is a uni-variate black-box function and  $\hat{f} : [a, b] \rightarrow \mathbb{R}$  is a surrogate function to be built by an spline function. For a given degree  $p$  and a global knot vector  $U$  including  $e + p + 1$  knots,

$$\hat{f}(x) = \sum_{i=1}^e \alpha_i N_{i,p}(x), \quad (29)$$

where the coefficients  $\alpha_i$ ,  $i = 1, \dots, e$ , must be estimated for a given set of sample points  $(x^i, y^i)$ ,  $i = 1, \dots, n$ , ( $n \geq e$ ) and  $y^i = f(x^i)$ . The coefficients  $\alpha_i$  are estimated by solving the following least square problem

$$\underset{(\alpha_1, \dots, \alpha_e)^T \in \mathbb{R}^e}{\text{minimize}} \sum_{i=1}^n (\hat{f}(x^i) - y^i)^2. \quad (30)$$

When a set of sample points and a degree  $p$  are given, the task in applying spline functions to approximate a black-box function is specifying an appropriate global knot vector. See, e.g., [93, 94] for more detail.

To approximate a bivariate function corresponding to each decision variable, a set of knots is defined. Then, the set of B-splines corresponding to this knot set is built. We then construct tensor spline bases relying on these sets of B-splines, and bivariate spline functions are formed.

Let us suppose that  $f : [a, b]^2 \rightarrow \mathbb{R}$  is a bivariate black-box function and  $\hat{f} : [a, b]^2 \rightarrow \mathbb{R}$  is the corresponding surrogate function to be constructed. Then

$$\hat{f}(x_1, x_2) = \sum_{i=1}^{e_1} \sum_{j=1}^{e_2} \alpha_{i,j} N_{i,j}^{p,q}(x_1, x_2), \quad (31)$$

in which  $N_{i,j}^{p,q}(x_1, x_2), i = 1, \dots, e_1, j = 1, \dots, e_2$ , are called *tensor spline bases* and formed as  $N_{i,j}^{p,q}(x_1, x_2) = N_{i,p}(x_1)N_{j,q}(x_2)$  where  $N_{i,p}, i = 1, \dots, e_1$ , are B-splines of degree  $p$  corresponding to a global knot vector  $U = \{u_1, u_2, \dots, u_{e_1+p+1}\}$ ,  $N_{j,q}, j = 1, \dots, e_2$ , are B-splines of degree  $q$  corresponding to a global knot vector  $V = \{v_1, v_2, \dots, v_{e_2+q+1}\}$  and  $\alpha_{i,j}, i = 1, \dots, e_1, j = 1, \dots, e_2$ , are coefficients to be calculated. When  $U$  and  $V$  and a set of sample points  $(x_1^i, x_2^i, y^i), y^i = f(x_1^i, x_2^i), i = 1, \dots, \Delta, \Delta \geq e_1 \cdot e_2$ , are given, the coefficients  $\alpha_{i,j}$  are estimated by adopting problem (30) for a bivariate function. As can be seen, a bivariate B-spline function has a tensor product structure.

To improve the accuracy of a surrogate function built by spline functions, a refinement is necessary. A refinement means adding new B-splines into the set of B-splines. However, adding a new B-spline into the set of B-splines corresponding to one of the decision variables increases the number of tensor spline bases exponentially in bivariate B-spline functions.

### 6.3 T-splines Review

In order to reduce the number of bases in bivariate B-spline functions, researchers in CAD [79] have developed T-splines. While T-splines inherit the same flexibility as bivariate B-spline functions, they can be refined locally. Before introducing T-splines, we need to define some basic concepts.

Let  $U = \{u_1, u_2, \dots, u_{e_1+p+1}\}$  and  $V = \{v_1, v_2, \dots, v_{e_2+q+1}\}$  be global knot vectors corresponding to  $x_1$  and  $x_2$  coordinates, respectively, and  $I_1 = \{1, \dots, e_1 + p + 1\}$  and  $I_2 = \{1, \dots, e_2 + q + 1\}$  be a set of knot indices. We call the Cartesian product of  $I_1$  and  $I_2$ , i.e.,  $I = I_1 \times I_2$  the *knot-index-coordinate system*. Figure 12 shows a part of a knot-index-coordinate system. The net grid in Figure 12 is called a *T-mesh* in the knot-index-coordinate system. The elements in the knot-index-coordinate system are called *nodes*. Nodes are indicated by red circles in Figure 12.

T-splines were first introduced in [79]. Following [79, 92, 95], we assume that  $p = q = 3$ . T-splines are defined based on global knot vectors  $U = \{u_1, \dots, u_{e_1+p+1}\}$  and  $V = \{v_1, \dots, v_{e_2+q+1}\}$  and a T-mesh. We define a T-mesh as mentioned in [95]. In a T-mesh, nodes occupy some, but not all  $(i, j) \in I_1 \times I_2$  as

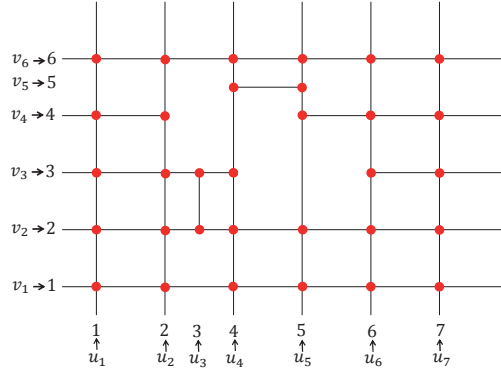


FIGURE 12 A knot-index-coordinate system. Nodes are shown in red circles. (Color online)

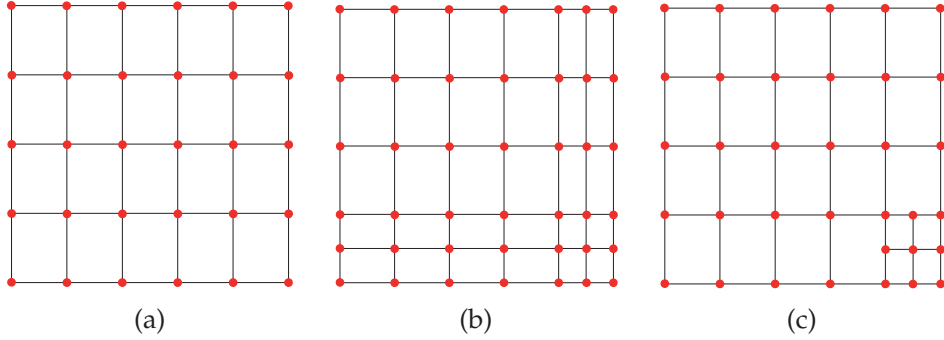


FIGURE 13 (a) An initial mesh, (b) bivariate B-spline refinement, (c) T-spline refinement. (Color online)

shown in Figure 12. Each pair of neighboring nodes on the same horizontal or vertical line must be connected by an *edge*. Two different edges can intersect only at a node. In a T-mesh, nodes shared by two vertical edges and one horizontal edge or one vertical edge and two horizontal edges are called *T-junctions*. For example, the node corresponding to (4, 5) in Figure 12 is a T-junction. One should note that a T-junction cannot be located on the boundary of the T-mesh.

A T-spline representation is given by

$$T(x_1, x_2) = \sum_{i=1}^{e_T} \alpha_i B_i(x_1, x_2), \quad (32)$$

in which  $\alpha_i \in \mathbb{R}$  are coefficients to be calculated,  $e_T$  is the number of nodes in the T-mesh and  $e_T \leq e_1 \cdot e_2$ . For  $i = 1, \dots, e_T$ ,  $B_i(x_1, x_2)$  is called a *blending function* and defined as

$$B_i(x_1, x_2) = N_i[\mathbf{u}_i](x_1) N_i[\mathbf{v}_i](x_2), \quad (33)$$

where  $\mathbf{u}_i = [u_{\delta_1^i}, u_{\delta_2^i}, u_{\delta_3^i}, u_{\delta_4^i}, u_{\delta_5^i}]$  and  $\mathbf{v}_i = [v_{\beta_1^i}, v_{\beta_2^i}, v_{\beta_3^i}, v_{\beta_4^i}, v_{\beta_5^i}]$  are sub-sequences of  $U$  and  $V$ , respectively,  $N_i[\mathbf{u}_i]$  and  $N_i[\mathbf{v}_i]$  are cubic-basis functions defined on  $\mathbf{u}_i$

and  $\mathbf{v}_i$ , respectively. We call  $\mathbf{u}_i$  and  $\mathbf{v}_i$  local knot vectors of  $N_i$  in  $x_1$  and  $x_2$  coordinates, respectively. As discussed in [95], blending functions (33) in T-splines are not necessarily linearly independent. Therefore, they may not form base functions. We discuss this matter in more detail in [PIV]. One should note that the coefficients  $\alpha_i, i = 1, \dots, e_T$ , for a given set of sample points  $(x_1^i, x_2^i, y^i), y^i = f(x_1^i, x_2^i), i = 1, \dots, \Delta, \Delta \geq e_T$ , and a given set of blending functions with their corresponding local knot vectors are calculated through minimizing a least square error by adopting problem (30) for a bivariate case. If the T-mesh in Figure 12 consists of all  $(i, j) \in I_1 \times I_2$ , then T-spline representation (32) is equal to bivariate B-spline functions (31).

In T-splines, there is a one-to-one correspondence between nodes in a T-mesh and the blending function in (32). As mentioned earlier, bivariate B-spline functions have a tensor product structure, whereas T-splines do not follow a tensor product structure. Therefore, a T-mesh can form an incomplete net grid, as shown in Figure 12. An initial mesh is shown in Figure 13 (a). If this mesh is refined by bivariate B-spline functions, the mesh shown in Figure 13 (b) is the result. However, a refinement by T-splines results in the T-mesh shown in Figure 13 (c). As can be seen in this figure, the refinement has a local effect.

Suppose  $\mathbf{u}_i$  and  $\mathbf{v}_i, i = 1, \dots, e_T$ , are given and  $\tilde{x}_{1,j} = x_1 - u_{\delta_j^i}$  and  $\tilde{u}_{j,k} = u_{\delta_j^i} - u_{\delta_k^i}$ . From (27) and (28) with  $p = 3$ , we have

$$N_i[\mathbf{u}_i](x_1) = \begin{cases} 0 \\ \frac{(\tilde{x}_{1,1})^3}{\tilde{u}_{4,1}\tilde{u}_{3,1}\tilde{u}_{2,1}} \\ \frac{-(\tilde{x}_{1,1})^2\tilde{x}_{1,3}}{\tilde{u}_{4,1}\tilde{u}_{3,1}\tilde{u}_{3,2}} + \frac{-\tilde{x}_{1,4}\tilde{x}_{1,2}\tilde{x}_{1,1}}{\tilde{u}_{3,2}\tilde{u}_{4,2}\tilde{u}_{4,1}} + \frac{-\tilde{x}_{1,5}(\tilde{x}_{1,2})^2}{\tilde{u}_{5,2}\tilde{u}_{4,2}\tilde{u}_{3,2}} \\ \frac{\tilde{x}_{1,1}(\tilde{x}_{1,4})^2}{\tilde{u}_{4,1}\tilde{u}_{4,2}\tilde{u}_{4,3}} + \frac{\tilde{x}_{1,2}\tilde{x}_{1,4}\tilde{x}_{1,5}}{\tilde{u}_{4,3}\tilde{u}_{4,2}\tilde{u}_{5,2}} + \frac{(\tilde{x}_{1,5})^2\tilde{x}_{1,3}}{\tilde{u}_{5,2}\tilde{u}_{5,3}\tilde{u}_{4,3}} \\ \frac{-(\tilde{x}_{1,5})^3}{\tilde{u}_{5,2}\tilde{u}_{5,3}\tilde{u}_{5,4}} \\ 1 \\ 0 \end{cases} \quad \begin{cases} x_1 = u_{\delta_1^i} = u_{\delta_2^i} = u_{\delta_3^i} = u_{\delta_4^i}, \\ u_{\delta_1^i} \leq x_1 < u_{\delta_2^i}, \\ u_{\delta_2^i} \leq x_1 < u_{\delta_3^i}, \\ u_{\delta_3^i} \leq x_1 < u_{\delta_4^i}, \\ u_{\delta_4^i} \leq x_1 < u_{\delta_5^i}, \\ x_1 = u_{\delta_2^i} = u_{\delta_3^i} = u_{\delta_4^i} = u_{\delta_5^i}, \\ \text{otherwise.} \end{cases} \quad (34)$$

Similarly,  $N_i[\mathbf{v}_i](x_2)$  is defined.

As mentioned earlier, for each node in a T-mesh, there is a blending function and vice versa. Each node has a coordinate in the knot-index-coordinate system. Without loss of generality, suppose the node  $(\delta_3^i, \beta_3^i)$  belongs to  $I$ . For this node there exists a blending function  $B_i$ . We identify  $\mathbf{u}_i = \{u_{\delta_1^i}, u_{\delta_2^i}, u_{\delta_3^i}, u_{\delta_4^i}, u_{\delta_5^i}\}$  and  $\mathbf{v}_i = \{v_{\beta_1^i}, v_{\beta_2^i}, v_{\beta_3^i}, v_{\beta_4^i}, v_{\beta_5^i}\}$  for this blending function. Let  $u$ -point be any  $(l, m) \in \mathbb{N}^2$  such that  $u_l \in U$  and  $v_m \in V$ , or a vertical edge passes through  $(l, m)$  in the knot-index-coordinate system, or  $l \notin \{3, \dots, e_1\}$ . Then,  $\delta_2^i$  is the largest integer that is smaller than  $\delta_3^i$  such that for some  $m \in I_2$ ,  $(\delta_2^i, m)$  is a  $u$ -point and,  $\delta_1^i$  is the largest integer that is smaller than  $\delta_2^i$  such that for some  $m \in I_2$ ,  $(\delta_1^i, m)$  is a  $u$ -point. In addition,  $\delta_4^i$  is the smallest integer greater than  $\delta_3^i$  such that for some  $m \in I_2$ ,  $(\delta_4^i, m)$  is a  $u$ -point and  $\delta_5^i$  is the smallest integer greater than  $\delta_4^i$  such that

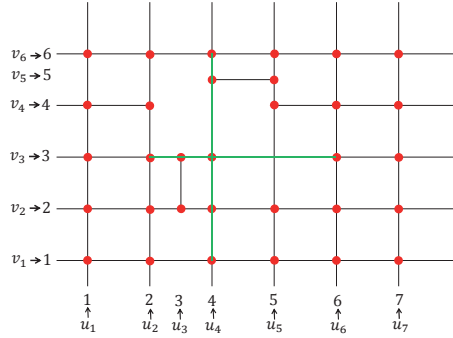


FIGURE 14 The indices in the local knot vectors corresponding to the node  $(4, 3)$  are shown in green. (Color online)

for some  $m \in I_2$ ,  $(\delta_5^i, m)$  is a  $u$ -point. Let a  $v$ -point be any  $(l, m) \in \mathbb{N}^2$  such that  $u_l \in U$  and  $v_m \in V$  or a horizontal edge passes through  $(l, m)$  in the knot-index-coordinate system or  $m \notin \{3, \dots, e_2\}$ . Then,  $\beta_2^i$  is the largest integer that is smaller than  $\beta_3^i$  such that for some  $l \in I_1$ ,  $(l, \beta_2^i)$  is a  $v$ -node and so on. For instance, in the blending function corresponding to the node  $(4, 3)$  in the T-mesh shown in Figure 14, we have  $\mathbf{u} = \{u_2, u_3, u_4, u_5, u_6\}$  and  $\mathbf{v} = \{v_1, v_2, v_3, v_5, v_6\}$ .

So far we have introduced T-splines. As mentioned in [PIII] and Chapter 5, in the ANOVA-MOP method, a surrogate function is built for each individual computationally expensive function. We expect that for computationally expensive MOPs including non-differentiable functions, T-splines can be considered a potential metamodeling technique to incorporate into ANOVA-MOP for sensitivity analysis of such functions. Consequently, sub-problems are formed accurately by ANOVA-MOP and solutions obtained can be reliable. The advantage of T-splines is that it has a local refinement property. Thus, computationally expensive functions can be approximated with fewer base functions in comparison with bivariate B-splines.

As mentioned earlier, blending functions in T-splines may not be linearly independent. Mathematical properties of T-splines such as linearly independent blending functions have been studied for functions with two and three decision variables [80]. We discuss them in [PIV]. To the best of our knowledge, T-splines have not been applied to conduct sensitivity analysis in computationally expensive MOPs. Typically, computationally expensive MOPs include functions with more than three decision variables. Thus, we should put more research effort toward applying T-splines for functions with more than three decision variables. The idea of applying T-splines for computationally expensive MOPs is new and demands extensive and profound investigation.



## 7 AUTHOR'S CONTRIBUTION

The research topic of solving computationally expensive multiobjective optimization problems was suggested by the author's supervisors. At the beginning, he conducted a literature review on this topic. He realized that there is no unified description of the main steps involved in surrogate-based methods for solving such problems. Thus, the author developed the sequential and adaptive frameworks. Based on these frameworks, he surveyed and compared surrogate-based methods. This led him to writing Paper [PI] by taking into account comments and feedback given by his supervisors as co-authors.

Initially, the idea in Paper [PII] was to develop an interactive, surrogate-based method for problems with a large number of objective functions. Prof. Kaisa Miettinen suggested that he first investigates this idea for problems with a low number of objective functions. Prof. Miettinen, Dr. Markus Hartikainen, Dr. Karthik Sindhya and Dr. Jussi Hakanen as the his supervisors and as co-authors emphasized that the DM should not have to wait for a long time to see solutions corresponding to his/her preferences. As the result, the author developed the steps involved in the SURROGATE-ASF method.

The author then considered developing SURROGATE-ASF for problems with a large number of objective functions. In SURROGATE-ASF the DM provides his/her preferences in the form of a reference point. This means that for problems with a large number of objective functions, the DM must specify an aspiration for each objective function.

In the 23<sup>rd</sup> International Conference on Multiple Criteria Decision-Making, in August 2015, the author presented the SURROGATE-ASF method. In this conference, the author had the privilege of meeting and discussing with Dr. Alberto Lovison. This fruitful discussion formed the basic idea of ANOVA-MOP presented in [PIII]. The author started to investigate how to apply sensitivity analysis for computationally expensive functions. This led the author to the method developed by Dr. Matthias Tan in [73]. Through an email exchange, Dr. Tan kindly provided the MATLAB code of his method. The author then had the honor to collaborate with Dr. Tan to develop ANOVA-MOP. Discussions, comments and feedback given by Dr. Lovison, Dr. Tan and Dr. Hartikainen and Prof. Miett-

nen as co-authors guided the author of this thesis to develop the ANOVA-MOP method including theoretical analysis.

The author conducted a research visit under the supervision of Dr. Tan. In this research visit, Dr. Tan suggested applying T-splines to conduct sensitivity analysis on computationally expensive non-differentiable functions. The author started doing a literature review to identify how T-splines could be incorporated into the ANOVA-MOP method as presented in [PIV].

In all papers included in this thesis, the author was the primary source of new ideas. Feedback and comments received from co-authors assisted the author to consider different aspects in solving computationally expensive MOPs when developing the methods proposed in this thesis.

## 8 CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this thesis, we have discussed challenges of solving computationally expensive multiobjective optimization problems and the importance of supporting a DM when solving such problems. We have introduced two frameworks, i.e., a sequential and an adaptive framework, to develop surrogate-based methods. We then have identified shortcomings of methods proposed in the literature, such as

1. capturing nonconvexity and/or discontinuity in the Pareto frontier when solving MOPs,
2. taking into account the preferences of the DM to develop interactive surrogate-based methods,
3. solving high-dimensional problems, and
4. solving problems involving non-differentiable functions.

Based on these shortcomings we have developed the SURROGATE-ASF and the ANOVA-MOP methods. In addition, we have discussed the potential of T-splines to be incorporated into ANOVA-MOP for sensitivity analysis.

The SURROGATE-ASF method [PII] consists of two phases, i.e., the initialization and decision-making phases. In the initialization phase, the decision space is decomposed into a limited number of hyper-boxes. Corresponding to each individual hyper-box, a surrogate is built. In the decision-making phase, the DM provides his/her preferences in the form of a reference point. Then a single-objective surrogate problem is formed and solved. The solution obtained is shown to the DM. The interaction with the DM is repeated until the solution obtained satisfies him/her. We have evaluated the SURROGATE-ASF performance on benchmark and application problems with at most 12 decision variables and 5 objective functions. When applying SURROGATE-ASF, the DM does not need to wait for a long time to find solutions corresponding to his/her preferences. Numerical results have shown that SURROGATE-ASF is capable of providing the most preferred solution for the DM even when the Pareto frontier is nonconvex and/or disconnected.

In the ANOVA-MOP method [PIII], using information obtained from sensitivity analysis, computationally expensive multiobjective optimization problems with a large number of decision variables and objective functions are decomposed into a finite number of sub-problems with a lower number of decision variables and objective functions. These sub-problems are solved, and the solutions obtained are composed to form solutions for the original problem.

ANOVA-MOP can be applied as a non-interactive or an interactive method. When applying it as an interactive method, the DM can only focus on a low number of objective functions at a time and can provide preferences for them rather than a large number of them. In this way, the DM can track easily how objective function values are changed according to his/her preferences. Moreover, the DM does not need to wait for a long time to see solutions corresponding to his/her preferences. Beside computationally expensive MOPs, ANOVA-MOP can be applied for solving computationally inexpensive MOPs.

We have discussed issues of sensitivity analysis on computationally expensive non-differentiable functions in [PIV]. We have reviewed T-splines and discussed their importance as a potential technique for sensitivity analysis on these kinds of functions.

Developing SURROGATE-ASF for problems with more than 12 decision variables and more than 5 objective functions is a future research direction. Combining the ideas in SURROGATE-ASF and ANOVA-MOP can be seen as another future research direction. To do so, one might study sensitivity analysis on decision variables and aspiration levels to identify their effect on the ASF values. This might lead to decomposing high-dimensional problems into a finite number of low-dimensional single-objective optimization problems.

In ANOVA-MOP, problems may arise in which the sub-problems obtained have some common variables. Developing ANOVA-MOP to solve such problems is a future research direction. Incorporating T-splines into ANOVA-MOP for solving problems involving non-differentiable functions is a research topic. Last but not least, developing surrogate-based methods to solve problems including computationally expensive constraints deserves research efforts.

The research effort devoted to this thesis has resulted in developing surrogate-based methods to rectify some issues in solving computationally expensive multiobjective optimization problems. By these methods, problems with different attributes, such as nonconvexity and/or discontinuity of the Pareto frontier or high-dimensional decision and objective spaces, can be solved effectively. In terms of decision-making, the DM can see solutions corresponding to his/her preferences quickly. Further methods are required to tackle issues that have not been considered in this thesis.

## YHTEENVETO (FINNISH SUMMARY)

### Lähestymistapoja laskennallisesti kalliiden monitavoiteoptimointitehtävien ratkaisemiseen

Tässä väitöskirjassa tarkastellaan laskennallisesti kalliiden monitavoiteoptimointitehtävien ratkaisemista niin, että päätöksentekijän preferenssitieto otetaan huomioon. Tavoite on tukea päätöksentekijää löytämään häntä parhaiten tyydyttävä ratkaisu tehtäville, joissa on useita ristiriitaisia tavoitteita ja joissa ratkaisukandidaatteja vastaavien tavoitteiden arvojen laskeminen vie aikaa. Aikaavievien tavoitteiden sijaan käytetään yleensä sijaisia. Tehdyn kirjallisuuskatsauksen pohjalta esitellään kaksi kehikkoa: peräkkäinen ja mukautuva, joihin olemassa olevat sijaispohjaiset menetelmät luokitellaan ja niitä vertaillaan. Kirjallisuuskatsauksen nojalla on tunnistettu lisätutkimusta vaativia haasteita, joihin väitöskirjassa paneudutaan.

Tässä tutkimuksessa on kehitetty kaksi sijaispohjaista menetelmää: SURROGATE-ASF ja ANOVA-MOP. SURROGATE-ASF on vuorovaikutteinen menetelmä ja siten sisältää kaksi vaihetta: aloitus- ja päätöksentekovaihe. Aloitusvaiheessa muuttuja-avaruus hajotetaan äärelliseen määrään hyperlaatikoita. Kullekin hyperlaatikolle muodostetaan yksitavoitteinen sijaistehtävä. Lopulta päätöksentekovaiheessa päätöksentekijän preferenssejä vastaava ratkaisu löydetään ratkaisemalla sopiva sijaistehtävä. Numeeriset testit tukevat menetelmän soveltuvuutta tehtäville joissa on korkeintaan 12 muuttujaa, 5 tavoitetta sekä epäkonvekksi ja/tai epäyhtenäinen Pareto-optimaalisten ratkaisujen joukko.

Jotta useampia muuttujia ja tavoitteita voitaisiin käsitellä, työssä on kehitetty ANOVA-MOP -menetelmä. Siinä tehtävä hajotetaan herkkyyksianalyysin avulla aputehtäviin joissa on kussakin vähemmän muuttujia ja tavoitteita. Nämä aputehtävät ratkaistaan ja ratkaisusta koostetaan approksimaatoratkaisuja alkupe- räiselle tehtävälle. ANOVA-MOP -menetelmää voidaan käyttää joko ilman päätöksentekijän vuorovaikutusta tai vuorovaikutteisena.

Työn lopussa tarkastellaan uuden T-spline -nimisen sijaismallinnuksen menetelmän mahdollisuuksia yhdessä ANOVA-MOP:n kanssa. Yhdistelmällä voidaan käsitellä tehtäviä, joissa on epäsiteitä tavoitteita. Väitöskirjassa kehitettyjen menetelmien avulla laajennetaan vuorovaikutteisten menetelmien soveltuvuutta laskennallisesti kalliille tehtäville.

Avainsanat: monitavoiteoptimointi, päätöksenteko, laskennallinen vaativuus, sijaismalli, hajotelma

## REFERENCES

- [1] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, 1999.
- [2] Y. Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.
- [3] T. Hildebrandt and J. Branke. On using surrogates with genetic programming. *Evolutionary Computation*, 23(3):343–367, 2014.
- [4] V. Chankong and Y. Y. Haimes. Optimization-based methods for multiobjective decision-making: An overview. *Large Scale Systems*, 5(1):1–33, 1983.
- [5] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.
- [6] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of Multiobjective Optimization*. Academic Press, 1985.
- [7] I. Kaliszewski. *Soft Computing for Complex Multiple Criteria Decision Making*. Springer US, 2006.
- [8] A. P. Wierzbicki. A mathematical basis for satisficing decision making. *Mathematical Modelling*, 3(5):391–405, 1982.
- [9] A. P. Wierzbicki. On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR Spectrum*, 8(2):73–87, 1986.
- [10] T. Aittokoski and K. Miettinen. Cost effective simulation-based multiobjective optimization in the performance of an internal combustion engine. *Engineering Optimization*, 40(7):593–612, 2008.
- [11] J. Hakanen, K. Miettinen, and K. Sahlstedt. Wastewater treatment: New insight provided by interactive multiobjective optimization. *Decision Support Systems*, 51(2):328–337, 2011.
- [12] Y. I. Tenne and C. K. Goh. *Computational Intelligence in Expensive Optimization Problems*. Springer, Berlin, 2010.
- [13] A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.
- [14] J. I. Madsen, W. Shyy, and R. T. Haftka. Response surface techniques for diffuser shape optimization. *AIAA Journal*, 38(9):1512–1518, 2000.

- [15] J. P. C. Kleijnen. Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, 192(3):707–716, 2009.
- [16] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, New York, 2003.
- [17] J. H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.
- [18] M. T. Hagan, H. B. Demuth, and M. Beale. *Neural Network Design*. PWS Publishing Co., Boston, 1996.
- [19] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.
- [20] J. C. Helton, J. D. Johnson, C. J. Sallaberry, and C. B. Storlie. Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering & System Safety*, 91(10-11):1175–1209, 2006.
- [21] T. W. Simpson, J. D. Poplinski, P. N. Koch, and J. K. Allen. Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with Computers*, 17(2):129–150, 2001.
- [22] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and K. P. Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1–28, 2005.
- [23] G. G. Wang and S. Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, 129(4):370–380, 2006.
- [24] A. I. J. Forrester, A. Sóbester, and A. J. Keane. *Engineering design via surrogate modelling: a practical guide*. Wiley, Chichester, 2008.
- [25] R. Jin, W. Chen, and T. W. Simpson. Comparative studies of metamodeling techniques under multiple modelling criteria. *Structural and Multidisciplinary Optimization*, 23(1):1–13, 2001.
- [26] S. M. Clarke, J. H. Griebisch, and T. W. Simpson. Analysis of support vector regression for approximation of complex engineering analyses. *Journal of Mechanical Design*, 127(6):1077–1087, 2004.
- [27] A. I. J. Forrester and A. J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45:50–79, 2009.
- [28] H. Nakayama, Y. Yun, and M. Yoon. *Sequential Approximate Multiobjective Optimization Using Computational Intelligence*. Springer, Berlin, 2009.
- [29] Y. F. Li, S. H. Ng, M. Xie, and T. N. Goh. A systematic comparison of metamodeling techniques for simulation optimization in decision support systems. *Applied Soft Computing*, 10(4):1257–1273, 2010.

- [30] A. Giunta, L. T. Watson, and J. Koehler. A comparison of approximation modeling techniques: Polynomial versus interpolating models. In *Proceedings of 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, volume 1, pages 392–404, St. Louis, MO, 1998.
- [31] M. H. Kutner, C. J. Nachtsheim, J. Neter, and W. Li. *Applied Linear Statistical Models*. McGraw Hill, Boston, 2005.
- [32] S. Shan and G. G. Wang. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization*, 41(2):219–241, 2010.
- [33] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:183–197, 2002.
- [34] G. P. Liu, X. Han, and C. Jiang. A novel multi-objective optimization method based on an approximation model management technique. *Computer Methods in Applied Mechanics and Engineering*, 197:2719–2731, 2008.
- [35] S. Kitayama, J. Srirat, M. Arakawa, and K. Yamazaki. Sequential approximate multi-objective optimization using radial basis function network. *Structural and Multidisciplinary Optimization*, 48(3):501–515, 2013.
- [36] T. Shao and S. Krishnamurty. A clustering-based surrogate model updating approach to simulation-based engineering design. *Journal of Mechanical Design*, 130(4):041101–041101, 2008.
- [37] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.
- [38] A. Messac and A. A. Mullur. A computationally efficient metamodeling approach for expensive multiobjective optimization. *Optimization and Engineering*, 9(1):37–67, 2008.
- [39] B. Wilson, D. Cappelleri, T. W. Simpson, and M. Frecker. Efficient Pareto frontier exploration using surrogate approximations. *Optimization and Engineering*, 2(1):31–50, 2001.
- [40] A. B. Ruiz, K. Sindhya, K. Miettinen, F. Ruiz, and M. Luque. E-NAUTILUS: A decision support system for complex multiobjective optimization problems based on the nautilus method. *European Journal of Operational Research*, 246(1):218–231, 2015.
- [41] M. Monz, K. H. Küfer, T. R. Bortfeld, and C. Thieke. Pareto navigation—algorithmic foundation of interactive multi-criteria IMRT planning. *Physics in Medicine and Biology*, 53(4):985–998, 2008.



- [42] S. Jakobsson, M. Patriksson, J. Rudholm, and A. Wojciechowski. A method for simulation based optimization using radial basis functions. *Optimization and Engineering*, 11(4):501–532, 2010.
- [43] S. Shan and G. G. Wang. An efficient Pareto set identification approach for multiobjective optimization on black-box functions. *Journal of Mechanical Design*, 127(5):866–874, 2004.
- [44] Y. Yun, M. Yoon, and H. Nakayama. Multi-objective optimization based on meta-modeling by using support vector regression. *Optimization and Engineering*, 10(2):167–181, 2009.
- [45] Z. O. Khokhar, H. Vahabzadeh, A. Ziai, G. G. Wang, and C. Menon. On the performance of the PSP method for mixed-variable multi-objective design optimization. *Journal of Mechanical Design*, 132(7):1–11, 2010.
- [46] J. Zhou and L. S. Turng. Adaptive multiobjective optimization of process conditions for injection molding using a Gaussian process approach. *Advances in Polymer Technology*, 26(2):71–85, 2007.
- [47] G. Chen, X. Han, G. Liu, C. Jiang, and Z. Zhao. An efficient multi-objective optimization method for black-box functions using sequential approximate technique. *Applied Soft Computing*, 12(1):14–27, 2012.
- [48] T. Goel, R. Vaidyanathan, R. T. Haftka, W. Shyy, N. V. Queipo, and K. Tucker. Response surface approximation of Pareto optimal front in multi-objective optimization. *Computer Methods in Applied Mechanics and Engineering*, 196:879–893, 2007.
- [49] M. Hartikainen, K. Miettinen, and M. M. Wiecek. PAINT: Pareto front interpolation for nonlinear multiobjective optimization. *Computational Optimization and Applications*, 52(3):845–867, 2012.
- [50] M. Gobbi, P. Guarneri, L. Scala, and L. Scotti. A local approximation based multi-objective optimization algorithm with applications. *Optimization and Engineering*, pages 1–23, 2013.
- [51] A. Messac, A. Ismail-Yahaya, and C. A. Mattson. The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization*, 25(2):86–98, 2003.
- [52] A. Messac and C. A. Mattson. Normal constraint method with guarantee of even representation of complete Pareto frontier. *AIAA Journal*, 42(10):2101–2111, 2004.
- [53] A. V. Lotov, V. A. Bushenkov, and G. K. Kamenev. *Feasible Goals Method. Search for Smart Decisions*. Computing Center of RAS, Moscow, 2001.

- [54] R. Su, L. Gui, and Z. Fan. Multi-objective optimization for bus body with strength and rollover safety constraints based on surrogate models. *Structural and Multidisciplinary Optimization*, 44(3):431–441, 2011.
- [55] P. Eskelinen, K. Miettinen, K. Klamroth, and J. Hakanen. Pareto navigator for interactive nonlinear multiobjective optimization. *OR Spectrum*, 32(1):211–227, 2010.
- [56] A. Zilinskas. A statistical model-based algorithm for ‘black-box’ multi-objective optimisation. *International Journal of Systems Science*, 45(1):82–93, 2014.
- [57] M. Binois, D. Ginsbourger, and O. Roustant. Quantifying uncertainty on Pareto fronts with gaussian process conditional simulations. *European Journal of Operational Research*, 243(2):386–394, 2015.
- [58] V. Picheny. Multiobjective optimization using gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, 25(6):1265–1280, 2015.
- [59] J. Martínez-Frutos and D. Herrero-Pérez. Kriging-based infill sampling criterion for constraint handling in multi-objective optimization. *Journal of Global Optimization*, 64(1):97–115, 2016.
- [60] R. Regis. Multi-objective constrained black-box optimization using radial basis function surrogates. *Journal of Computational Science*, 16:140 – 155, 2016.
- [61] J. Svenson and T. Santner. Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models. *Computational Statistics and Data Analysis*, 94:250–264, 2016.
- [62] A. Arias-Montano, C. A. Coello Coello, and E. Mezura-Montes. Multi-objective airfoil shape optimization using a multiple-surrogate approach. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2012.
- [63] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.
- [64] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Proceedings of Congress on Evolutionary Computation*, volume 1, pages 825–830, Honolulu, USA, 2002. IEEE.
- [65] A. Lovison. Global search perspectives for multiobjective optimization. *Journal of Global Optimization*, 57(2):385–398, 2013.
- [66] Y. D. Sergeyev and D. E. K. Kvasov. Global search based on efficient diagonal partitions and a set of lipschitz constants. *SIAM Journal on Optimization*, 16(3):910–937, 2006.

- [67] N. Cowan. The magical mystery four: How is working memory capacity limited, and why? *Current directions in psychological science*, 19(1):51–57, 2010.
- [68] C. F. J. Wu and M. S. Hamada. *Experiments: planning, analysis, and parameter design optimization*. Wiley, New York, 2 edition, 2009.
- [69] A. Owen. Variance components and generalized Sobol’ indices. *SIAM/ASA Journal on Uncertainty Quantification*, 1(1):19–41, 2013.
- [70] A. Saltelli, S. Tarantola, and K. P. S. Chan. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics*, 41(1):39–56, 1999.
- [71] A. Saltelli, K. Chan, and M. Scott, editors. *Sensitivity Analysis*. Wiley, New York, 2000.
- [72] I. M. Sobol. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55:271–280, 2001.
- [73] M. Tan. Sequential Bayesian polynomial chaos model selection for estimation of sensitivity indices. *SIAM/ASA Journal on Uncertainty Quantification*, 3:146–168, 2015.
- [74] O. I. Larichev. Cognitive validity in design of decision-aiding techniques. *Journal of Multi-Criteria Decision Analysis*, 1(3):127–138, 1992.
- [75] D. Hadka and P. Reed. Borg: An auto-adaptive many-objective evolutionary computing framework. *Evolutionary Computation*, 21(2):231–259, 2013.
- [76] K. Deb and S. Gupta. Understanding knee points in bicriteria problems and their implications as preferred solution principles. *Engineering Optimization*, 43(11):1175–1204, 2011.
- [77] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1, 2013.
- [78] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1, 2016.
- [79] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCs. *ACM Trans. Graph.*, 22(3):477–484, 2003.
- [80] P. Morgenstern. Globally structured three-dimensional analysis-suitable T-splines: Definition, linear independence and  $m$ -graded local refinement. *SIAM Journal on Numerical Analysis*, pages 2163–2186, 2016.

- [81] M. J. W. Jansen. Analysis of variance designs for model output. *Computer Physics Communications*, 117:35–43, 1999.
- [82] M. D. Morris, L. M. Moore, and M. D. McKay. Using orthogonal arrays in the sensitivity analysis of computer models. *Technometrics*, 50(2):205–215, 2008.
- [83] J. E. Oakley and A. O’Hagan. Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(3):751–769, 2004.
- [84] L. Le Gratiet, C. Cannamela, and B. Iooss. A Bayesian approach for global sensitivity analysis of (multifidelity) computer codes. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):336–363, 2014.
- [85] S. Touzani and D. Busby. Smoothing spline analysis of variance approach for global sensitivity analysis of computer codes. *Reliability Engineering & System Safety*, 112:67–81, 2013.
- [86] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering and System Safety*, 93(7):964–979, 2008.
- [87] S. Da Veiga, F. Wahl, and F. Gamboa. Local polynomial estimation for sensitivity analysis on models with correlated inputs. *Technometrics*, 51(4):452–463, 2009.
- [88] B. J. Reich, C. B. Storlie, and H. D. Bondell. Variable selection in bayesian smoothing spline ANOVA models: Application to deterministic computer codes. *Technometrics*, 51(2):110–120, 2009.
- [89] E. Borgonovo and E. Plischke. Sensitivity analysis: A review of recent advances. *European Journal of Operational Research*, 248(3):869 – 887, 2016.
- [90] L. Piegl and W. Tiller. *The NURBS Book (2nd Ed.)*. Springer-Verlag New York, Inc., New York, NY, USA, 1997.
- [91] K. Höllig and J. Hrner. *Approximation and Modeling with B-Splines*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2013.
- [92] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. In *ACM SIGGRAPH 2004 Papers, SIGGRAPH ’04*, pages 276–283, New York, NY, USA, 2004. ACM.
- [93] J. Z. Huang, C. O. Wu, and L. Zhou. Polynomial spline estimation and inference for varying coefficient models with longitudinal data. *Statistica Sinica*, 14(3):763–788, 2004.
- [94] M. Landajo, J. De Andrés, and P. Lorca. Measuring firm performance by using linear and non-parametric quantile regressions. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 57(2):227–250, 2008.

- [95] X. Li, J. Zheng, T. W. Sederberg, T. J. R. Hughes, and M. A. Scott. On linear independence of T-spline blending functions. *Computer Aided Geometric Design*, 29(1):63–76, 2012.

## **ORIGINAL PAPERS**

**PI**

### **A SURVEY ON HANDLING COMPUTATIONALLY EXPENSIVE MULTIOBJECTIVE OPTIMIZATION PROBLEMS USING SURROGATES: NON-NATURE INSPIRED METHODS**

by

M. Tabatabaei, J. Hakanen, M. Hartikainen, K. Miettinen, and K. Sindhya

Structural and Multidisciplinary Optimization, 2015, 52, 1-25

# A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods

Mohammad Tabatabaei · Jussi Hakanen ·  
Markus Hartikainen · Kaisa Miettinen ·  
Karthik Sindhya

Received: 6 June 2014 / Revised: 19 December 2014 / Accepted: 2 January 2015 / Published online: 10 March 2015  
© Springer-Verlag Berlin Heidelberg 2015

**Abstract** Computationally expensive multiobjective optimization problems arise, e.g. in many engineering applications, where several conflicting objectives are to be optimized simultaneously while satisfying constraints. In many cases, the lack of explicit mathematical formulas of the objectives and constraints may necessitate conducting computationally expensive and time-consuming experiments and/or simulations. As another challenge, these problems may have either convex or nonconvex or even disconnected Pareto frontier consisting of Pareto optimal solutions. Because of the existence of many such solutions, typically, a decision maker is required to select the most preferred one. In order to deal with the high computational cost, surrogate-based methods are commonly used in the literature. This paper surveys surrogate-based methods proposed in the literature, where the methods are independent of the underlying optimization algorithm and mitigate the computational burden to capture different types of Pareto frontiers. The methods considered are classified, discussed and then compared. These methods are divided into two

frameworks: the sequential and the adaptive frameworks. Based on the comparison, we recommend the adaptive framework to tackle the aforementioned challenges.

**Keywords** Multicriteria Decision Making (MCDM) · Black-box function · Computational cost · Metamodeling technique · Sampling technique · Pareto frontier

## 1 Introduction

Many practical engineering problems often involve optimizing (either minimizing or maximizing) multiple, possibly incommensurable objective functions subject to a feasible set determined by constraint functions. In such problems known as *multiobjective optimization problems (MOPs)*, the best solution for an objective function may be the worst solution for some other objective functions. As a matter of fact, in a solution of an MOP, improvement in the value of one objective is only possible by allowing impairment in the values of at least one of the other objectives which is known as the concept of *Pareto optimality* (Miettinen 1999). A solution of an MOP satisfying this concept is called a *Pareto optimal solution*, and the set of all such solutions is referred to as a *Pareto frontier* (often also known as a *Pareto optimal set*). Mathematically, without any additional information, all Pareto optimal solutions are equally acceptable solutions of an MOP. It is, however, generally desirable to obtain one solution to be implemented. Therefore, when solving an MOP, we need a *decision maker (DM)* to compare several different solutions or to provide preference information in some other way and to select the most preferred one. In this survey, we define solving an MOP in two ways, i.e., finding a representation of the entire Pareto frontier to a DM or obtaining the most preferred solution based on the preferences of a DM.

---

M. Tabatabaei (✉) · J. Hakanen · M. Hartikainen ·  
K. Miettinen · K. Sindhya  
University of Jyväskylä, Department of Mathematical  
Information Technology, P.O. Box 35 (Agora),  
FI-40014 University of Jyväskylä, Finland  
e-mail: mohammad.tabatabaei@jyu.fi

J. Hakanen  
e-mail: jussi.hakanen@jyu.fi

M. Hartikainen  
e-mail: markus.hartikainen@jyu.fi

K. Miettinen  
e-mail: kaisa.miettinen@jyu.fi

K. Sindhya  
e-mail: karthik.sindhya@jyu.fi

In real-world MOPs, the mathematical formulas of objective and constraint functions could be either computationally expensive to evaluate and/or of a black-box type. For black-box functions, all that is known about them is the output for a given input. When dealing with these functions, mathematical properties such as convexity or continuity are not available. In some problems, objective and constraint functions are evaluated using real and/or computational experiments such as thermodynamic analysis, structural analysis, computational fluid dynamics (CFD) or reservoir simulation which involve differential equations to be solved. Numerical techniques such as finite element (FE) and finite difference methods may be applied to solve these equations. These experiments are time consuming and such problems are known as *computationally expensive (intensive, costly) MOPs*. For example, each objective function evaluation in reservoir simulation problems may take several days even after applying various techniques to improve the computational speed (Rezaveisi et al. 2014). The Pareto frontiers of such problems may be convex, nonconvex or disconnected. How to most efficiently solve these computationally expensive problems is an open research question in the literature.

In this paper, we present a survey of methods to handle computationally expensive MOPs. The focus of this survey is on general methods which are independent of the type of the optimization algorithms used in them. The basic idea in such methods is to introduce a computationally less expensive replacement problem known as a *surrogate problem*. Besides methods considered here, methods have been developed in the literature where mechanisms of nature-inspired methods such as evolutionary and particle swarm algorithms are essential elements of building the surrogates of the methods. See Tenne and Goh (2010), Jin (2011), Zhou et al. (2011) for reviews of such methods. As said, these methods are not considered here. Surveys on methods to solve computationally expensive single objective optimization problems utilizing surrogate problems are given in Simpson et al. (2001, 2004, 2008), Shan and Wang (2010), Koziel et al. 2011. In such methods, only one objective function is concerned, while in multiobjective optimization methods, at least two objectives are considered. In single objective optimization, comparing two solutions based on the concept of optimality is possible: the smaller (or larger) the objective function value, the better the solution. In multiobjective optimization problems, however, the concept of Pareto optimality is needed. In the literature, there are scalarization-based methods (Steuer 1986; Miettinen et al. 2008) that transform an MOP into a single objective optimization problem with respect to the preferences of a DM. Then, the optimal solution of the single objective optimization problem is considered as a preferred solution for the decision maker. In this survey, wherever such methods are

employed to deal with computationally expensive MOPs, we discuss them.

As far as we know, this is the first survey which fully concentrates on handling computationally expensive MOPs by general methods as defined above. This survey covers 20 selected papers written in English and published in scientific journals before 2013. Besides as an overview of the methods available, one can use this survey to find a method applicable to one's own problems.

A method where a computationally expensive MOP is handled using a surrogate problem is here termed a *surrogate-based multiobjective optimization method*. In what follows, for the sake of simplicity, such methods are called surrogate-based methods. Handling a computationally expensive MOP relying on a surrogate problem involves selecting sample points, building, updating and solving the surrogate problem. Based on when the surrogate problem is updated, surrogate-based methods are here classified into two frameworks: the adaptive and the sequential framework. In accordance with when the sample points are selected to update the surrogate problem, the adaptive framework is divided into types 1 and 2.

The motivation of this survey is to focus on the characteristics of the surrogate-based methods to solve a computationally expensive MOP and compare these methods in four aspects: (1) Can the methods deal with general black-box functions where information regarding mathematical properties of the functions such as convexity or continuity is not available? (2) Can the methods capture different types of Pareto frontiers? (3) How many objective and constraint functions as well as decision variables can be handled by the methods? (4) What is the role of a DM during the solution process? For this comparison, we rely on the results given by the authors of the papers considered on the employed benchmark and real-world problems.

In surrogate-based methods, the quality of the Pareto frontier of the surrogate problem depends on the accuracy of the surrogate problem and the performance of the optimization algorithm employed to solve this surrogate problem. In the literature (Wu and Azarm 2000; Okabe et al. 2003; Zitzler et al. 2003; Zitzler et al. 2008), several performance indices for measuring the quality of the Pareto frontier have been proposed. According to Okabe et al. (2003), this quality can be assessed based on the number of solutions in the Pareto frontier of the surrogate measured by e.g., overall non-dominated vector generation, the distribution and spread of these solutions measured by e.g.,  $\Delta$  index and closeness of the surrogate's Pareto frontier to the Pareto frontier of the computationally expensive MOP measured by e.g., generational distance, inverted generational distance and hyper-volume. If such information is given in the papers considered, we also mention it.



The rest of this paper is organized as follows. In Section 2, the basic concepts used in this survey, and a brief discussion on how a surrogate problem can be built, are addressed. In Section 3, the classification of surrogate-based methods into the sequential and the adaptive frameworks is discussed. Details of the sequential framework and the related methods are discussed in Section 4. Types 1 and 2 of the adaptive framework, and methods belonging to them are discussed in Sections 5 and 6, respectively. There is a method in which the sequential framework and type 1 of the adaptive framework are hybridized to handle computationally expensive MOPs. Therefore, it is discussed in Section 7. In Section 8, the surrogate-based methods considered in this survey are compared. Future research directions are also discussed. Finally, conclusions are drawn in Section 9.

## 2 Concepts

### 2.1 Definitions and notations

In the following, basic definitions and notations used in this survey are given (mostly from Miettinen (1999)). Some of these are illustrated in Fig. 1. In this paper, we deal with multiobjective optimization problems of the form

$$\underset{x \in S}{\text{minimize}} \{f_1(x), \dots, f_k(x)\}, \tag{1}$$

where the set  $S$  is called the *feasible decision region* (set) (often also called the *feasible design space*) which is a subset of the *decision space*  $\mathbb{R}^n$ . We have  $k(\geq 2)$  objective functions  $f_i : S \rightarrow \mathbb{R}$ . We denote the vector of objective functions by  $f(x) = (f_1(x), \dots, f_k(x))^T$ . An example of the feasible decision region is  $S = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \dots, m, h_j(x) = 0, j = 1, \dots, p\}$ , where  $g_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, m$ , and  $h_j : \mathbb{R}^n \rightarrow \mathbb{R}, j = 1, \dots, p$ , are called *constraint functions*. A solution  $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$  is called a *decision (variable) vector*, where  $x_i, i = 1, \dots, n$ , are *decision variables*. A decision vector  $x \in S$  satisfying all the constraint functions is called a *feasible decision vector*. The image of the feasible decision region in the *objective space*  $\mathbb{R}^k$  is called the *feasible objective region* (set) (often also called *performance space*) denoted by  $Z (= f(S))$ . The elements of  $Z$  are called *feasible objective vectors* denoted by  $f(x)$  or  $z = (z_1, \dots, z_k)^T$ , where  $z_i = f_i(x), i = 1, \dots, k$ , are *objective (function) values*. For the sake of simplicity, we use the term *feasible solution* which refers to either a feasible decision vector or a feasible objective vector. If needed, we clarify whether a feasible solution belongs to either  $S$  or  $Z$ .

A feasible solution  $x^* \in S$  and the corresponding  $f(x^*) \in Z$  are said to be *weakly Pareto optimal* for the problem (1), if there does not exist another feasible solution

$x \in S$  such that  $f_i(x) < f_i(x^*)$  for all  $i = 1, \dots, k$ . Correspondingly, they are *Pareto optimal* for the problem (1), if there does not exist another feasible solution  $x \in S$  such that  $f_i(x) \leq f_i(x^*)$  for all  $i = 1, \dots, k$ , and  $f_j(x) < f_j(x^*)$  for at least one index  $j \in \{1, \dots, k\}$ . Obviously, a Pareto optimal solution is a weakly Pareto optimal solution. The set of all Pareto optimal solutions in the objective space is called a *Pareto frontier* (often referred to as a *Pareto optimal set*). We also define a feasible solution  $x_e^i = \operatorname{argmin}_{x \in S} \{f_i(x)\}$  for  $i = 1, \dots, k$ . The  $i$ -th *extreme solution* (also called the *anchor point* (Messac and Mullur 2008)) for  $i = 1, \dots, k$ , is defined as  $z_e^i = f(x_e^i)$  (see Fig. 1). A hyperplane passing through all extreme solutions is called a *utopia hyperplane* (Messac and Mullur 2008).

As mentioned earlier, in the process of solving an MOP, a DM may be involved, whose role is to give preference information, e.g., by comparing the obtained solutions. Based on the literature (Miettinen 1999; Luque et al. 2011; Ruiz et al. 2012), (s)he can provide his/her preferences, e.g., in the form of a *reference point*  $\bar{z} = (\bar{z}_1, \dots, \bar{z}_k)^T$  where  $\bar{z}_i$  is an *aspiration level* consisting of a desirable value for the  $i$ -th objective function. Another approach to elicit information about a DM's preference during the solution process is classification, i.e., the DM classifies the objectives into classes in which the objective function values should be improved, can impair or are satisfactory. See Miettinen (1999) for further information of the roles of a DM during the solution process.

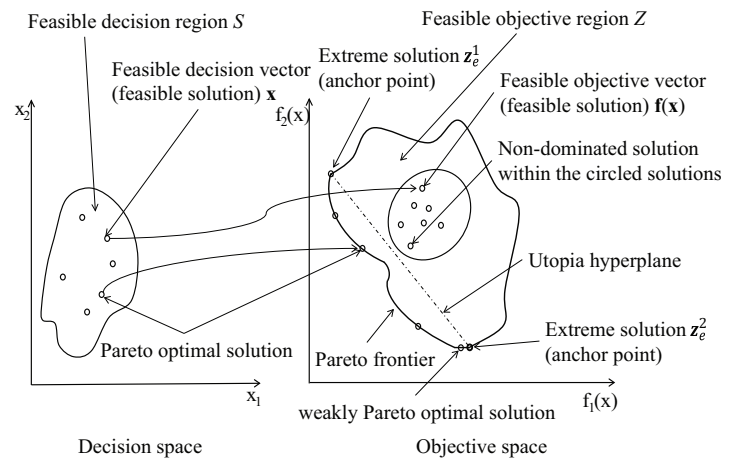
In the literature, scalarizing an MOP means formulating a single objective optimization problem such that an optimal solution for the single objective optimization problem is a (weakly) Pareto optimal solution for the MOP. The following is an example of scalarization. It involves a so-called *achievement scalarizing function* (Wierzbicki 1986):

$$\underset{x \in S}{\text{minimize}} \max_{i=1, \dots, k} [w_i(f_i(x) - \bar{z}_i)] \tag{2}$$

where  $w_i \geq 0$  for all  $i = 1, \dots, k$ , are nonnegative weights, and  $\bar{z}_i$  for all  $i = 1, \dots, k$ , the aspiration level for the  $i$ -th objective function provided by a DM. Different (weakly) Pareto optimal solutions can be obtained by changing the reference point. Pareto optimal solutions can be obtained by adding a so-called augmentation term to the objective function of the problem (2) (see, e.g. Miettinen (1999)).

Let us assume that the set  $X = \{x^1, \dots, x^q\}$  is an arbitrary subset of feasible solutions in  $S$ , and  $F = \{f(x^1), \dots, f(x^q)\}$  the corresponding objective vectors in  $Z$ . A solution  $x^i$  (or  $f(x^i)$ ) for  $i = 1, \dots, q$ , that satisfies the definition of Pareto optimality with respect to all solutions in  $X$  (or  $F$ ), is called a *non-dominated solution* in  $X$  (or  $F$ ) (see Fig. 1). A Pareto optimal solution is a non-dominated solution, but a non-dominated solution is not necessarily a Pareto optimal solution. If  $X = S$  (or  $F = Z$ ),

**Fig. 1** Some definitions in MOPs



then every non-dominated solution is a Pareto optimal solution and vice versa. Non-dominated solutions for the given set  $X$  can be identified by e.g., the *Pareto fitness function* (Schaumann et al. 1998) defined as

$$G_i = 1 - \max_{j \in \{1, \dots, q\} \setminus \{i\}} \left[ \min \left\{ \bar{f}_1(x^i) - \bar{f}_1(x^j), \dots, \bar{f}_k(x^i) - \bar{f}_k(x^j) \right\} \right], \quad (3)$$

$$i = 1, \dots, q,$$

where  $G_i$  denotes the Pareto fitness of the  $i$ -th feasible solution  $x^i$ , and  $\bar{f}_l(x^i)$  for  $l = 1, \dots, k$ , is the  $l$ -th normalized objective function value of the  $i$ -th feasible solution, given by:

$$\bar{f}_l(x^i) = \frac{f_l(x^i) - f_{l,\min}(x)}{f_{l,\max}(x) - f_{l,\min}(x)}, \quad (4)$$

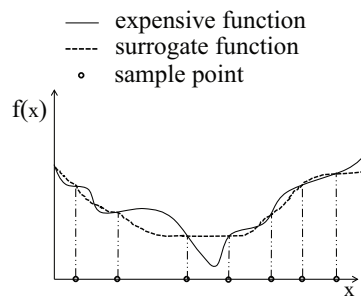
where  $f_{l,\max}(x)$  and  $f_{l,\min}(x)$  represent the maximum and minimum values of the  $l$ -th objective function among all feasible solutions in  $X$ , respectively. When the Pareto fitness  $G_i$  is greater than 1, the corresponding feasible solution is a non-dominated solution (Shan and Wang 2004).

## 2.2 How to build a surrogate problem

As mentioned earlier, the basic idea in a surrogate-based method is to replace a computationally expensive MOP with a computationally less expensive surrogate problem. One approach to building a surrogate problem is to approximate each computationally expensive function using metamodeling techniques such as polynomial functions (Madsen et al. 2000), Kriging models (Kleijnen 2009), radial basis functions (RBFs) (Buhmann 2003), multivariate

adaptive regression splines (MARS) (Friedman 1991), neural networks (Hagan et al. 1996) and support vector regression (SVR) (Smola and Schölkopf 2004). To approximate a computationally expensive function, sample points are required. These sample points which are solutions in the decision space can be selected by sampling techniques such as Latin hypercube sampling (LHS) (Helton et al. 2006), central composite design (CCD) (Simpson et al. 2001), orthogonal array sampling (OAS) (Simpson et al. 2001) and full factorial sampling (FFS) (Simpson et al. 2001). See Simpson (2001), Queipo et al. (2005), Wang and Shan (2006), Forrester et al. (2008) and Simpson (2001), Queipo et al. (2005), Wang and Shan (2006), Forrester et al. (2008), Jin et al. (2001), Clarke et al. (2004), Forrester and Keane (2009), Nakayama et al. (2009), Li et al. (2010) for surveys of the sampling and metamodeling techniques and their characteristics, respectively.

Once a set of points is sampled, these points are evaluated with the computationally expensive function for which the metamodel is to be fitted. Sample points can also be generated by other methods such as a posteriori methods for multiobjective optimization where a subset of (weakly) Pareto optimal solutions representing the Pareto frontier is obtained. These methods are surveyed in Miettinen (1999) and Marler and Arora (2004). The obtained (weakly) Pareto optimal solutions can be considered as a set of sample points. When a set of points evaluated with the computationally expensive function is available, a metamodeling technique is employed to fit a computationally inexpensive function to the sample points known as a *surrogate function* (see Fig. 2). Once the surrogate functions of all computationally expensive functions are constructed, a computationally inexpensive MOP known as a *surrogate problem* is formulated.



**Fig. 2** A surrogate function

According to Forrester and Keane (2009), in the context of surrogate-based single objective optimization, we want to build a surrogate problem to be most accurate in the region of the optimum. In surrogate-based multiobjective optimization, the accuracy of the surrogate problem in the region of the Pareto optimal solutions is desirable. Since each individual surrogate function may not be accurate in such a region, the Pareto frontier of the surrogate problem may not coincide with the Pareto frontier of the original, computationally expensive MOP. The accuracy of the surrogate functions can be evaluated with statistical measurements such as root mean square error (RMSE) (Giunta et al. 1998), predicted error sum of squares (PRESS) (Kutner et al. 2005), cross-validation method (Wang and Shan 2006) and  $R^2$  (Jin et al. 2001). See Shan and Wang (2010), Wang and Shan (2006), Jin et al. (2001) and Li et al. (2010) for surveys of commonly used criteria to evaluate the accuracy. In this survey, we study how the Pareto frontier of the surrogate problem represents the Pareto frontier of the computationally expensive MOP by the methods proposed in the papers considered.

Beside approximating each computationally expensive function, other approaches can be used to build a surrogate of a computationally expensive MOP, e.g., by approximating directly the Pareto frontier. These other approaches apply particular techniques which are discussed in the subsequent sections.

In this survey, when summarizing the methods considered we sometimes call a computationally expensive MOP as an *original problem*. The objective and/or constraint functions in the original and the surrogate problems are referred to as *computationally expensive* and *surrogate functions*, respectively. These functions involve models based on which functions are formed. For the sake of simplicity, we assume that all functions involved in the original problem are computationally expensive. A set of non-dominated (or Pareto optimal) solutions of the surrogate problem which may be evaluated with the computationally expensive

functions is considered as the *approximated Pareto frontier* of the original problem. We also use the terms *benchmark* and *application problems*. The first one refers to the available test problems in the literature, e.g. ZDT problems (Deb et al. 2002), used to compare the performance of different methods. Application problems refer to the problems arising from industries which are dealt with in the papers considered in this survey. If the information regarding the characteristics of the Pareto frontiers of the benchmark and application problems like convexity, nonconvexity or discontinuity as well as performance indices to assess the quality of the surrogate's Pareto frontier is provided by the authors of the surveyed papers, we also mention it. There are methods that have been developed to solve some particular application problems. While discussing such methods, first these problems are mentioned and then the summary of the methods is discussed. One should notice that the benchmark problems are not computationally expensive and, thus, their validity for properly testing the methods surveyed is questionable.

### 3 Classification of surrogate-based methods

After reviewing the journal papers considered in this survey, we have found out that two frameworks are employed to handle computationally expensive MOPs utilizing surrogates. Nevertheless, there is no unified description of the main steps involved in surrogate-based methods (see, e.g. Queipo et al. (2005), Wang and Shan (2006), Forrester and Keane (2009), Liu et al. (2008) and Kitayama et al. (2013)). Therefore, we consider two general frameworks, i.e., sequential and adaptive frameworks to classify surrogate-based methods, inspired by the classification in Wang and Shan (2006) and Liu et al. (2008), and based on when to update the surrogate problem. The key point in the sequential framework is to build an accurate surrogate problem and then to solve it. In this framework, the approximated Pareto frontier (i.e., a set of non-dominated (or Pareto optimal) solutions of the surrogate problem) is supposed to be as close as possible to the Pareto frontier of the original problem. Details of the sequential framework and methods belonging to this framework are discussed in Section 4. In the adaptive framework, however, the key point is first to construct an initial surrogate problem. As mentioned earlier, since the initial surrogate problem may not be accurate over the region of the Pareto optimal solutions of the original problem, the approximated Pareto frontier (obtained by solving the initial surrogate problem) may not represent the exact Pareto frontier of the original problem. Hence, by updating and solving the surrogate problem iteratively, the approximated Pareto frontier is supposed to coincide with the Pareto frontier of the original problem.

In order to update the surrogate problem, new sample points (which are also termed infill points in Forrester et al. (2008)) are required. These sample points can be selected from either a set of non-dominated (or Pareto optimal) solutions of the surrogate problem or unexplored regions in the decision and/or objective space. Based on when the sample points are selected to update the surrogate problem, inspired by Liu et al. (2008), we divide the adaptive framework into types 1 and 2. Details of the adaptive framework and methods in this framework are discussed in Sections 5 and 6, respectively. In addition, there is a method in which the sequential framework and type 1 of the adaptive framework are hybridized to handle computationally expensive MOPs. This method is discussed in Section 7 after describing both frameworks. All the methods considered in this survey are then compared in Section 8.

To solve a surrogate problem in both frameworks, two types of methods can be employed, i.e., *sampling-based* and *optimization-based* ones. In a sampling-based method, a surrogate problem is solved with an emphasis on a sampling process and without using any optimization algorithms. In contrast, in an optimization-based method, a surrogate problem is solved utilizing an optimization algorithm. If the aim of solving an MOP is to find the most preferred solution for a DM, depending on the way of giving the preference information, one can employ an interactive method. An overview of such methods has been presented in Miettinen (1999) and Miettinen et al. (2008). Several different solutions generated during the solution process can also be visualized for the DM to compare them. See Miettinen (2014) for a review of visualization techniques.

## 4 Sequential framework

### 4.1 General flowchart

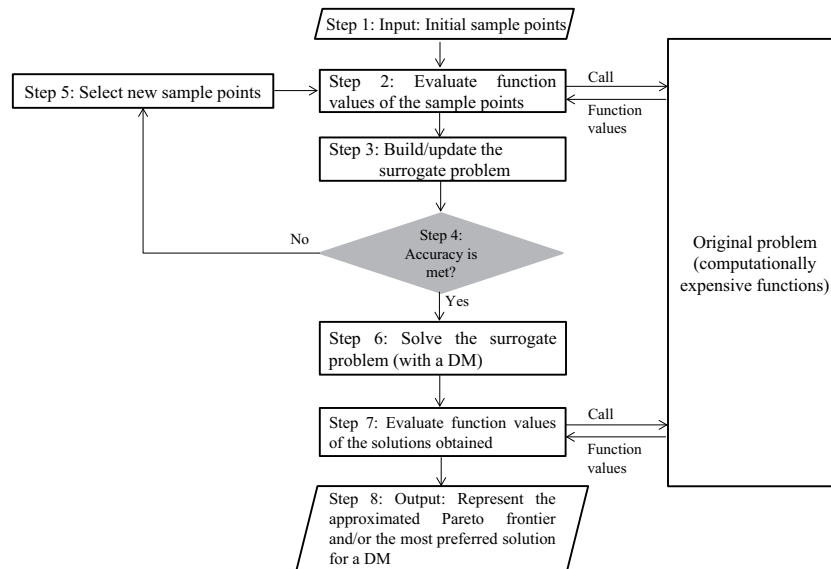
Constructing an accurate surrogate problem is the key point in the sequential framework. The flowchart in Fig. 3 presents the main steps of methods belonging to this framework. As can be seen in this figure, in Step 1 of a method in this framework, initial points are sampled, and then evaluated with the computationally expensive functions in Step 2. After this, function values at the sample points are available. A surrogate problem is then constructed in Step 3. Evaluating the accuracy of the surrogate problem in Step 4 (highlighted in Fig. 3) is critical in this framework. As mentioned earlier, the accuracy can be evaluated by statistical measurements such as the cross-validation method, root mean square error (RMSE), predicted error sum of squares (PRESS) and  $R^2$ . If the surrogate problem is not sufficiently accurate, it is updated by selecting new sample points in Step 5, and Steps 2–4 are then repeated. If

the sample points are selected only once to build a surrogate problem, this is considered as a special case termed as one-stage sampling. In this case, Step 5 is not conducted. After constructing the accurate enough surrogate problem, it is solved in Step 6 with a DM, if available. As mentioned earlier, the aim of solving the surrogate problem can be to represent non-dominated (or Pareto optimal) solutions to a DM or to provide the most preferred solution based on the preferences of a DM. The solution(s) obtained by solving the surrogate problem are typically evaluated with the computationally expensive functions in Step 7. As a result, the outcome of Step 7 is considered as the approximated Pareto frontier of the original problem and/or the most preferred solution for a DM in Step 8. This outcome can also be visualized using an appropriate visualization technique. In this framework, depending on the accuracy of the surrogate problem, the approximated Pareto frontier is as close as possible to the Pareto frontier of the original problem. In Section 4.2, we summarize the methods belonging to the sequential framework. These methods are then compared in Section 8.

### 4.2 Summary of methods in the sequential framework

In Goel et al. (2007), an optimization-based method using polynomial functions is introduced. In Step 1 of this method, initial points are sampled with OAS and then evaluated with the computationally expensive functions in Step 2. Each computationally expensive function is approximated by a quadratic or a cubic polynomial function in Step 3, and the surrogate problem is built. The accuracy of the surrogate problem is evaluated with a cross-validation method in Step 4. If needed, new points are sampled in Step 5 to improve the accuracy, and Steps 2–4 are repeated. After constructing the accurate enough surrogate problem, highly correlated objectives among the surrogate functions may be dropped or a representative objective can be used for all the correlated objectives (applying principal component analysis) (Goel et al. 2004). In Step 6, the surrogate problem is then solved using a population-based evolutionary multiobjective optimization method called NSGA-II (Deb et al. 2002). The non-dominated solutions obtained are then locally improved by the  $\epsilon$ -constraint method (Zanakis et al. 1998). In this paper, the authors do not discuss evaluating the non-dominated solutions obtained in Step 6 with the computationally expensive functions in Step 7. In Step 8, for an MOP with less than three objective functions, the approximated Pareto frontier is visualized by fitting a polynomial function to the non-dominated solutions in the objective space. This function is accurate in a limited region of the objective space which is identified by convex hulls.

In Goel et al. (2007), besides solving a computationally expensive MOP, the authors discuss the problem of Pareto



**Fig. 3** Flowchart of the sequential framework

drift (losing some non-dominated solutions in each generation of the dominance based Multiobjective Evolutionary Algorithms (MOEAs)). They propose an implementation of an archiving strategy to preserve all good solutions. Regarding the issue of capturing a nonconvex Pareto frontier, the authors mention that the number of convex hulls is important. In this method, a metamodeling technique is applied for two reasons, i.e., to build the accurate enough surrogate problem, and to visualize the approximated Pareto frontier. Finding the correlated functions along with a representation of an objective as a function of the remaining objectives to visualize the approximated Pareto frontier can be a barrier of using this method for MOPs with more than three objective functions. The efficiency of this method was evaluated on an MOP of liquid-rocket single element injector with four black-box objective functions and four decision variables requiring CFD simulations. In this problem, the number of objective functions was reduced to three by applying principal component analysis. The quality of the approximated Pareto frontier was assessed based on the average and maximum distances between the non-dominated solutions in two successive iterations of the optimization algorithm employed. This application problem was also studied in Queipo et al. (2005), where a discussion on the global sensitivity analysis of the objective functions and the decision variables was considered.

In the sequential framework, there are two sampling-based methods which we discuss in the following paragraphs. In Wilson et al. (2001), a sampling-based method called efficient Pareto frontier exploration is introduced in

which metamodeling techniques are employed to approximate individual computationally expensive objective and/or constraint functions. In efficient Pareto frontier exploration, first the DM provides desirable ranges for the decision variables. Then, making use of two sampling techniques (i.e., CCD and LHS), two sets of initial points are sampled within these ranges in the decision space in Step 1 and evaluated with the computationally expensive functions in Step 2. For each set, the computationally expensive functions are approximated with a second-order polynomial function and a Kriging model in Step 3 to compare the results, and the surrogate problems are constructed. In Step 4, using a cross-validation method, the accuracies of the surrogates are evaluated. If needed, new sample points are selected in Step 5, and Steps 2-4 are repeated. In order to solve the surrogate problems and to represent the corresponding non-dominated solutions in Step 6, a considerable number of points is sampled and evaluated with the surrogate functions. Then non-dominated solutions of the surrogate problems among the evaluated points are identified using the Pareto fitness function (3). These non-dominated solutions are then evaluated in Step 7 with the computationally expensive functions. The sets of evaluated non-dominated solutions are considered as the approximated Pareto frontier of the original problem in Step 8.

The efficient Pareto frontier exploration method was applied in designing a piezoelectric bimorph actuator with two black-box objective functions and five decision variables. The function evaluations involved applying the FE method using Abaqus (2013). The performance of the

efficient Pareto frontier exploration method was also evaluated on two benchmark problems with convex, nonconvex and disconnected Pareto frontiers. The authors show that the surrogate problem constructed based on initial sample points of LHS and the Kriging model is more accurate than the other surrogates for these problems. The results demonstrated that, while this method can capture a nonconvex and disconnected Pareto frontier, it performs better for an MOP with a convex Pareto frontier. Since in this method, a large number of sample points is required to solve the surrogate problem, it cannot be applied to handle high-dimensional MOPs in the decision and objective spaces. The role of a DM is also to provide desirable ranges for the decision variables which does not match with the definition of a DM's role in the literature as defined in Section 2.1.

In most of the methods discussed so far, surrogates of the computationally expensive functions are built using metamodeling techniques to approximate computationally expensive functions. In contrast, (Lotov et al. 2001) proposes a sampling-based method called Feasible Goals Method (FGM) to approximate the set  $Z$  (the feasible objective region) by means of a collection of boxes in the objective space and without using any metamodeling technique. In order to approximate the set  $Z$  with a finite number of boxes, it is assumed to be bounded. In Step 1 of this method, evenly distributed initial points are sampled randomly, and then evaluated with the computationally expensive functions in Step 2. In Step 3, a box in the neighborhood of each point in the objective space is formed. The collection of the boxes is considered as the surrogate of the feasible objective region. Utilizing the Chebyshev metric, the authors introduce a probability function to evaluate the accuracy of the surrogate in Step 4. To do this, a set of new evenly distributed points is sampled randomly in Step 5, and evaluated with the computationally expensive functions in Step 2. If the probability function value is less than a pre-determined threshold, the farthest point is selected. Then, a new box in the neighborhood of the selected point is formed to update the surrogate in Step 3, and Steps 2-4 are repeated. Otherwise, that is, the surrogate is sufficiently accurate, the surrogate problem is considered in Step 6 to select the most preferred solution in the objective space by the DM. In this step, the collection of boxes is visually shown to a DM. Then, the DM identifies a preferred solution. The center of the related box is computed along with the associated decision vector value in Step 7, and displayed to the DM in Step 8. FGM was applied to a set of application problems such as pollution monitoring station problem with five nonlinear objective functions and two decision variables. The feasible objective region of this problem was nonconvex. This method can handle black-box functions, since in practice, the boundedness of  $Z$  can be assumed by considering boundaries for the decision variables. Developments

of FGM to approximate both convex and nonconvex Pareto frontiers are discussed in Lotov et al. (2004).

So far, we have discussed methods involving all steps in the sequential framework. In the following paragraphs, we discuss methods involving one-stage sampling. As mentioned earlier, in one-stage sampling, sample points are selected only once. In Liao et al. (2008), an optimization-based method is developed which is applied to solve an MOP of crash safety design of vehicles with three black-box objective functions and five decision variables. LS-DYNA (Livermore Software Technology Corporation (LSTC) 2013) is employed as a simulation software. In this method, initial points are sampled with an extension of LHS called optimal Latin hypercube sampling (OLHS) in Step 1 and then evaluated with the computationally expensive functions involving the FE method in Step 2. After that, stepwise regression and a quadratic polynomial function are applied to approximate each computationally expensive function and, then the surrogate problem is built in Step 3. The accuracy of this problem is evaluated based on  $R^2$  in Step 4. Non-dominated solutions of the surrogate problem are obtained using NSGA-II in Step 6. This set of non-dominated solutions is considered as the approximated Pareto frontier of the original problem in Step 8. This set is not evaluated with the computationally expensive functions in Step 7.

In Su et al. (2011), an optimization-based method is developed, and is applied to solve a biobjective optimization problem of designing a bus body with 13 constraints and 31 decision variables. The objective and constraint functions are evaluated with the FE method using simulation software called MSC Nastran (MSC Nastran-Multidisciplinary structural analysis 2013) and LS-DYNA. In Step 1 of this method, initial points are sampled with OLHS and evaluated with the computationally expensive functions in Step 2. Then, stepwise regression is applied to approximate each objective and constraint function making use of a hybrid of a linear polynomial function and a Gaussian RBF, and the surrogate problem is constructed in Step 3. The accuracy of the surrogate problem is evaluated based on PRESS in Step 4. The surrogate problem is then solved in Step 6. The non-dominated solutions obtained are evaluated with the computationally expensive functions in Step 7. The set of evaluated solutions is considered as the approximated Pareto frontier of the original problem in Step 8. In this paper, two evolutionary algorithms, i.e., NSGA-II and AMISS-MOP (Su et al. 2011) were employed to solve the surrogate problem. The authors concluded that the spread of non-dominated solutions obtained by NSGA-II was wider than AMIS-MOP, but the convergence of AMISS-MOP was better than NSGA-II.

In Bornatico et al. (2013) steps similar to those of Liao et al. (2008) and Su et al. (2011) are followed to solve a

design optimization problem of a solar thermal building system with two black-box objective functions and two decision variables involving the Polysun simulation software (Polysun 2013). In this method, the sample points are selected only once with the Poisson disk node distribution (Gamito and Maddock 2009). Each computationally expensive function is approximated with a cubic RBF. The biobjective surrogate problem is solved with NSGA-II. In this paper, the accuracies of the surrogate problems constructed with four sampling techniques, i.e., Cartesian distribution (Bornatico et al. 2011), Hexagonal distribution (Steinhaus 2011), uniform distributions (Hickernell et al. 2005) and Poisson disk node distribution were also compared. To assess the quality of the approximated Pareto front, the original problem was also solved. Then, the average Euclidean distance between the Pareto frontiers of the original problem and the surrogates was considered. The authors concluded that the surrogate problem constructed with the Poisson disk node distribution sample points and a cubic RBF outperformed the others in terms of the accuracy.

In the optimization-based methods discussed so far, a surrogate problem is constructed using metamodeling techniques to approximate each computationally expensive function. In Hartikainen et al. (2012), however, an optimization-based method called PAINT is proposed to approximate the Pareto frontier of a computationally expensive MOP directly without utilizing any metamodeling techniques. In this method, a linear mixed integer multiobjective optimization problem is introduced as a surrogate of the original problem. In PAINT, initial sample points are generated by an a posteriori method in Steps 1 and 2. These sample points are Pareto optimal solutions of the original problem. Then, a PAINT interpolation between the sample points is created in Step 3 and the surrogate problem is introduced. The accuracy of the surrogate problem is high, if a large number of well-distributed Pareto optimal solutions is used as sample points. This problem can be solved with any interactive multiobjective optimization method, and a preferred approximated solution for a DM is obtained in Step 6. After finding this solution, it is then projected to the Pareto frontier of the original problem with the achievement scalarizing function (2) in Steps 7 and 8.

In PAINT, only the objective space is considered. Although the authors claim that PAINT can represent a nonconvex Pareto frontier, it cannot capture a disconnected Pareto frontier. In addition, the approximation of the Pareto frontier loses the connection to the decision space when solving the surrogate problem. However, after projecting the preferred approximated solution to the Pareto frontier, the closest Pareto optimal solution both in the decision and objective spaces is obtained. The efficiency of this method was evaluated on a benchmark problem with a convex Pareto frontier. This method was also applied in

approximating the Pareto frontier of an MOP of wastewater treatment planning with three black-box objective functions described in Hakanen et al. (2011). The number of constraint functions and decision variables was not mentioned.

All the described methods are compared in Section 8. Table 1 summarizes the characteristics of the methods in the sequential framework with respect to sampling techniques, metamodeling techniques, number of objective and constraint functions as well as decision variables in the considered benchmark and application problems, whether the methods involve one-stage sampling and whether they are optimization- or sampling-based. In this table, for every method, the most challenging MOP that was considered as a benchmark or an application problem is mentioned. Since the number of equality constraints in all problems are zero ( $p = 0$ ), it is not mentioned in the table. As can be seen, most of the problems considered are limited to two or three objective functions except (Lotov et al. 2001) with five objective functions. As mentioned earlier, since the method in Messac and Mullur (2008) hybridizes both the sequential and the adaptive frameworks, we discuss it in Section 7.

## 5 Adaptive framework: type 1

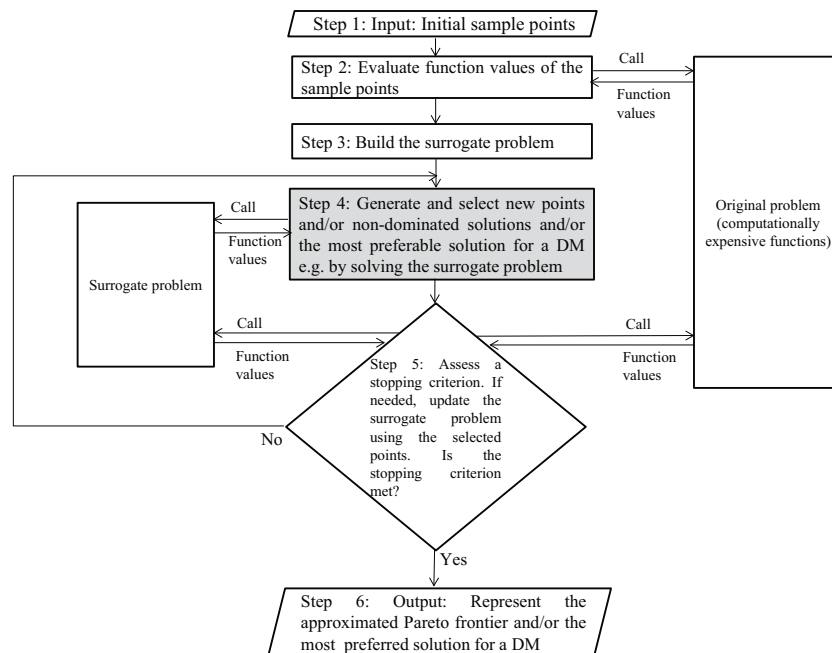
### 5.1 General flowchart

As mentioned in Section 3, another class of surrogate-based methods is the adaptive framework. In this framework, as can be seen in the flowcharts in Figs. 4 and 6, in comparison with the sequential framework (Fig. 3), after sampling, first an initial surrogate problem is constructed. As mentioned earlier, the approximated Pareto frontier (obtained by solving the initial surrogate problem) may not represent the exact Pareto frontier of the original problem. Therefore, the surrogate problem is iteratively solved and updated by selecting new sample points such that, the approximated Pareto frontier is supposed to coincide with the Pareto frontier of the original problem. As described in Section 3, the new sample points can be selected from either a set of non-dominated (or Pareto optimal) solutions of the surrogate problem or unexplored regions in the decision and/or objective space. Based on when new sample points are employed to update the surrogate problem, we divide this framework into types 1 and 2. In type 1, the sample points generated before assessing a stopping criterion (Step 4 highlighted in Fig. 4) are utilized to update the surrogate problem. In type 2, not only the sample points generated before assessing the stopping criterion (Step 4 highlighted in Fig. 6) are considered, but also new sample points are generated and selected after assessing the stopping criterion (Step 6 highlighted in Fig. 6) to update the surrogate problem. In this section, we discuss

**Table 1** Characteristics of methods in the sequential framework ( $k, m, n$  are the number of objective functions, inequality constraint functions and decision variables, respectively)

Characteristics Ref.	Sampling technique	Metamodeling technique	Features of the benchmark problem	Features of the application problem	Type of the method
(Goel et al. 2007)	OAS	Quadratic and cubic polynomial function	No benchmark problem	$k = 4, m = 0, n = 4$ nonconvex Pareto frontier	Optimization-based
(Wilson et al. 2001)	LHS, CCD	Quadratic polynomial function, Kriging model	$k = 2, m = 2, n = 2$ , nonconvex and disconnected Pareto frontier	$k = 2, m = 0, n = 5$	Sampling-based
(Lotov et al. 2001)	Random sampling	No metamodeling technique	No benchmark problem	$k = 5, m = 0, n = 2$	Sampling-based
(Messac and Mullur 2008)	LHS, FFS	Multiquadric RBF, E-RBF	$k = 2, m = 0, n = 5$ , nonconvex Pareto frontier	$k = 2, m = 2, n = 3$	Optimization-based
(Liao et al. 2008)	OLHS	Quadratic polynomial function	No benchmark problem	$k = 3, m = 0, n = 5$	Optimization-based (one-stage sampling)
(Su et al. 2011)	OLHS	Hybrid of a linear polynomial function and Gaussian RBF	No benchmark problem	$k = 2, m = 13, n = 31$ ,	Optimization-based (one-stage sampling)
(Bornatico et al. 2013)	Poisson disk node distribution	Cubic RBF	No benchmark problem	$k = 2, m = 0, n = 2$ ,	Optimization-based (one-stage sampling)
(Hartikainen et al. 2012)	A posteriori method	No metamodeling technique	$k = 4, m = 0, n =$ not mentioned, nonconvex Pareto frontier	$k = 3, m = 0, n =$ not mentioned	Optimization-based (one-stage sampling)





**Fig. 4** Flowchart of the adaptive framework: type 1

type 1 of the adaptive framework. Type 2 is considered in Section 6.

The flowchart in Fig. 4 shows the main steps of methods belonging to type 1 of the adaptive framework. As can be seen, in the first step of type 1, initial points are sampled, and then evaluated with the computationally expensive functions in Step 2. After this, function values of the sample points are available. In Step 3, the initial surrogate problem is constructed. In order to capture the Pareto frontier or to provide the most preferred solution based on the preferences of a DM, if available, new sample points are generated in Step 4. These points can be obtained by solving the surrogate problem and/or sampling unexplored regions in the decision and/or objective space depending on a sampling process. In accordance with a method-dependent criterion, a subset of points among the generated points is selected. In Step 5, assessing a stopping criterion may require to evaluate the selected sample points with the computationally expensive functions and/or to update the surrogate problem. There are different stopping criteria which are discussed in more detail in the following subsection. If a stopping criterion is met, the set of non-dominated (or Pareto optimal) solutions or the most preferred solution of the last surrogate problem which may have been evaluated with the computationally expensive functions are considered as the approximated Pareto frontier of the

original problem or the most preferred solution for a DM in Step 6. These solutions can also be visualized using an appropriate visualization technique. Otherwise, that is, when a stopping criterion is not met, if the surrogate problem has already been updated, Steps 4–5 are repeated. If not, first the surrogate problem is updated with the evaluated sample points and then Steps 4–5 are repeated.

In Section 5.2, we describe how the above steps are conducted by methods in type 1 of the adaptive framework. We compare these methods in Section 8.

## 5.2 Summary of methods in the adaptive framework: type 1

In Yang et al. (2002), first an optimization-based method called adaptive approximation in single objective optimization (AASO) is developed, and then it is extended to adaptive approximation in multiobjective optimization (AAMO). AAMO is the oldest method in the literature developed based on the adaptive framework. In Step 1 of AAMO, initial points are sampled with any sampling technique (e.g., LHS) and evaluated with the computationally expensive functions in Step 2. Then in Step 3, each computationally expensive function is approximated with a Kriging model, and an initial surrogate problem is built. In Step 4, a set of non-dominated solutions of the surrogate problem is obtained by Multiobjective Genetic Algorithm

(MOGA) (Goldberg 1989) and denoted by  $P_0$ . Among these points, using the maximin distance design criterion (Johnson et al. 1990), a number of isolated points is selected. In Step 5, these points are evaluated with the computationally expensive functions, and utilized to update the surrogate problem. The solutions in the set  $P_0$  obtained in Step 4 are again evaluated with the updated surrogate problem. A set of non-dominated solutions among these evaluated points is identified and denoted by  $P_{new}$ . If the difference between the number of the non-dominated solutions in  $P_0$  and in  $P_{new}$  is less than a predetermined threshold, the set  $P_{new}$  is considered as the approximated Pareto frontier in Step 6. Otherwise, the set  $P_{new}$  is inserted into an initial population of MOGA and the method continues with Step 4.

Similar to Goel et al. (2007), the authors in Yang et al. (2002) also discuss the difficulties of GA-based algorithms in identifying extreme solutions in the Pareto frontier. To overcome these difficulties, they propose a method called combined AASO-AAMO in which the AASO and AAMO methods are combined. In combined AASO-AAMO, after obtaining a set of non-dominated solutions of the surrogate problem in Step 4, all objective functions in the surrogate problem are individually minimized. The obtained optimal solutions along with the set  $P_{new}$  are inserted into an initial population of MOGA. The combined AASO-AAMO method was tested on solving an I-beam design problem with two nonlinear objective functions, one nonlinear constraint function and four decision variables. The quality of the approximated Pareto frontier was assessed based on closeness of it to the Pareto frontier of the application problem. While this method can handle black-box functions, the application of combined AASO-AAMO to a practical optimization problem involving actual computationally expensive or noisy simulations is mentioned as a future research direction.

In Zhou and Turng (2007), an optimization-based method is developed which is applied to solve an MOP of injection-molding process with five black-box objective functions and three decision variables. To do this, initial points are sampled with LHS in Step 1 and evaluated with the computationally expensive functions in Step 2 involving the Moldex3D simulation software (Moldex3d: Plastic injection molding simulation software 2013). Each computationally expensive function is approximated in Step 3 with a Kriging model, and the initial surrogate problem is built. In Step 4, new sample points are selected randomly. The objective function values of these points are evaluated with the surrogate functions. The variances of the objective function values of these points are also calculated with the Kriging model. For each point, a vector consisting of these variances is considered. These vectors are sorted using the non-dominated sorting method (Deb et al. 2002) (variances

are maximized). Then, a subset of points with the largest variances (at the first front) is selected randomly. In Step 5, the selected points are evaluated with the computationally expensive functions and added to the sample points set. Then, the surrogate problem is updated. In other methods in type 1 of the adaptive framework, the surrogate problem is solved in Step 4. In this method, however, the surrogate problem is solved using NSGA-II in Step 5 to assess the stopping criterion.

The authors in Zhou and Turng (2007) employ the concept of user-defined indifference threshold (Wu and Azarm 2000) as the stopping criterion in Step 5. This threshold refers to the change in each objective function value within which the non-dominated solutions are indifferent to each other. Applying the user-defined indifference thresholds, the objective space is discretized into a collection of hyperboxes. Using the method of Wu and Azarm (2000), if the distances between the boundaries of the hyperboxes are less than the predetermined threshold, the method stops. The set of non-dominated solutions of the last surrogate problem is considered as the approximated Pareto frontier of the original problem in Step 6. Otherwise, the method repeats Steps 4-5. According to the authors, the quality of the approximated Pareto frontier was assessed based on the average error percentage between the non-dominated solutions obtained by solving the surrogate and the original problems.

In Yun et al. (2009), an optimization-based method is developed to represent an approximation of the Pareto frontier as well as to identify the most preferred solution for a DM with respect to a reference point consisting of desirable aspiration levels for all objectives. In this method, initial points are sampled randomly in Step 1, and evaluated with the computationally expensive functions in Step 2. Then, each computationally expensive function is approximated with an extension of SVR called  $\mu - \nu$ -SVR, and the surrogate problem is built in Step 3. This surrogate problem is then solved with an evolutionary multiobjective optimization method called SPEA2 (Zitzler et al. 2001) in Step 4. In order to approximate well a neighborhood of the closest Pareto optimal solution to the reference point provided by the DM, among the generated non-dominated solutions, the point with the minimum value of an order-approximating achievement function (Miettinen 1999) is selected. Moreover, to update the surrogate problem, another point is selected based on a specific parameter associated with the  $\mu - \nu$ -SVR. In Step 5, if the number of computationally expensive function evaluations is less than a predetermined threshold, the selected points are evaluated with the computationally expensive functions. The surrogate problem is then updated, and the method returns to Step 4. Otherwise, the set of non-dominated solutions of the last surrogate problem is considered as the approximated Pareto frontier of

the original problem. The point with the minimum value of the order-approximating achievement function is also considered as the most preferred solution for the DM. The efficiency of this method was evaluated on a benchmark problem with a nonconvex and disconnected Pareto frontier, and on a welded beam design problem with two nonlinear objective functions, four nonlinear constraint functions and four decision variables.

In Jakobsson et al. (2010), an optimization-based method is developed to handle noisy black-box single objective optimization problems. This method is then extended to handle computationally expensive MOPs. The authors call both methods qualSolve. They claim that it is the first method to handle a computationally expensive noisy MOP. However, the performance of the qualSolve method for MOPs is only evaluated with deterministic benchmark problems. Here, we focus on qualSolve for MOPs. In Step 1 of qualSolve, initial points are sampled with LHS and evaluated with the computationally expensive functions in Step 2. Each computationally expensive function is approximated with a thin plate spline RBF in Step 3, and a surrogate problem is built. In Step 4, the surrogate problem is solved by a multiobjective optimization method to represent its Pareto frontier. Using this Pareto frontier, the extended Pareto frontier, i.e., the set of weakly Pareto optimal solutions in  $Z_s + \mathbb{R}^k$  is represented where  $Z_s$  is the feasible objective region of the surrogate problem. Figure 5 which is a modification of a figure in Jakobsson et al. (2010) shows the extended Pareto frontier in which  $s_1$  and  $s_2$  are the approximated objective functions.

To update the surrogate problem, in every iteration of the method, a sample point is selected from either the Pareto frontier of the surrogate problem or in an area near the extended Pareto frontier of the surrogate problem. This point can be a Pareto optimal solution, or a feasible solution in the objective space, near the extended Pareto frontier

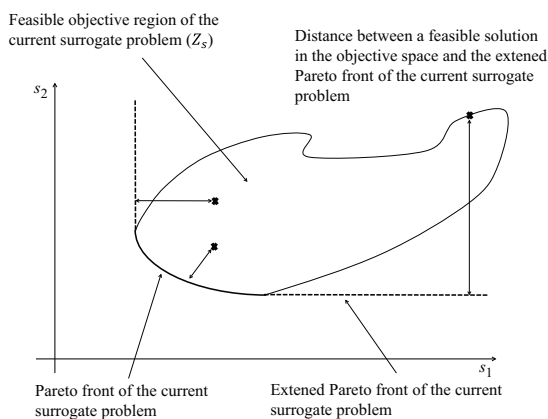


Fig. 5 Extended approximated Pareto front

of the current surrogate problem. To select the new sample point, the authors introduce a quality function based on two functions, i.e., a distance and a weight function. The distance function is defined as the smallest Euclidean distance between a feasible solution in the objective space and a solution on the extended Pareto frontier of the current surrogate problem. The weight function controls the procedure of selecting new sample points from either the Pareto frontier or an area near the extended Pareto frontier in the feasible objective region of the current surrogate problem. The quality function is formulated with respect to the weight and distance functions. The point that maximizes the quality function is used to update the surrogate problem. In Step 5, if the number of computationally expensive function evaluations is less than a predetermined threshold, the selected point is evaluated with the computationally expensive functions. The surrogate problem is then updated, and the method returns to Step 4. Otherwise, the Pareto frontier of the last surrogate problem is considered as the approximated Pareto frontier of the original problem in Step 6.

The performance of this method was evaluated on a set of benchmark problems such as Kursawe (1991) and OKA1 (2006). The authors concluded that qualSolve performed well on the Kursawe problem with a nonconvex and disconnected Pareto frontier. It, however, failed to represent the convex Pareto frontier of OKA1, since OKA1 has a very strong nonlinear behavior close to the minimum solution to one of the objective functions. They note that qualSolve was developed in a project on simulation-based multiobjective optimization of the Volvo D5 diesel engine with three black-box objective functions and five decision variables (Jakobsson et al. 2010) involving CFD simulation using Star-CD (Star-CD 2013). The authors consider the execution time, as a main downside of the qualSolve method for MOPs. Because the quality function includes an integral equation, the evaluation time of this function rises exponentially with the dimension of the decision space. They mention that qualSolve is suitable for problems with less than six decision variables. They also discuss that representing the extended Pareto frontier can be hard for three or more objective functions.

In Gobbi et al. (2013), an optimization-based method called approximate normal constraint (ANC) is introduced in which the idea of employing the normal constraint method (Messac et al. 2003; Messac and Mattson 2004) is followed. In the ANC method, a neural network with a single hidden layer is used to handle computationally expensive MOPs. To do this, first one of the computationally expensive objective functions is optimized to calculate the corresponding extreme solution. The authors, however, do not provide any guideline to choose this function. In a neighborhood of the extreme solution obtained in the decision

space, initial points are sampled randomly in Step 1 and evaluated with the computationally expensive functions in Step 2. These points are used to train a neural network to approximate each computationally expensive function, and to build a surrogate problem in Step 3. In Step 4, the other extreme solutions are calculated by minimizing the corresponding objective functions in the surrogate problem, and a utopia hyperplane is constructed. A set of evenly distributed points is then generated on the utopia hyperplane. For each point on the hyperplane, the surrogate problem is scalarized using the scalarizing technique introduced in the normal constraint method (Messac et al. 2003; Messac and Mattson 2004), and a single objective optimization problem is formulated.

The authors claim that by solving the single objective optimization problem, a non-dominated solution in an area near the Pareto frontier of the original problem can be obtained. Therefore, a set of non-dominated solutions can be generated near the Pareto frontier of the original problem by forming and solving the single objective optimization problem for each point on the hyperplane. In Step 5, these non-dominated solutions are evaluated with the computationally expensive functions. To keep the number of the training sample points constant, the points that are the farthest from the current utopia hyperplane are removed. The evaluated points are added to the training set, and the surrogate problem is updated. If the number of computationally expensive function evaluations is less than a threshold, the method returns to Step 4. Otherwise, the set of non-dominated solutions obtained near the Pareto frontier of the original problem is considered as the approximated Pareto frontier.

The efficiency of the ANC method was evaluated on benchmark problems with nonconvex Pareto frontiers. The efficiency was also assessed on an MOP of ground vehicle suspension design with five black-box objective functions and eleven decision variables. The authors compared the efficiency of the ANC method with MOGA and a method called Parameter Space Investigation (PSI) described in Miettinen (1999) and references therein. They provided remarks to show that the ANC outperformed the others. The authors claim that ANC can be used to refine the solutions obtained with other methods which approximate the Pareto frontier at once (i.e., genetic algorithms (Goldberg 1989) and quasi Monte Carlo method (Niederreiter 1992)).

In Kitayama et al. (2013), an optimization-based method is developed to handle computationally expensive black-box MOPs. In this method, the aim is to find the most preferred solution for a DM rather than to represent an approximation of the entire Pareto frontier. The authors introduce two sampling functions. They describe how to capture disconnected parts of the Pareto frontier by generating a set of sample points as evenly distributed as possible using the first

sampling function. The second sampling function provides the most preferred solution for a DM. In addition, to approximate computationally expensive functions with RBF, they utilize a new estimation proposed in Kitayama et al. (2011a) for the width parameter of a Gaussian kernel in RBF based on the number of and distances between sample points. In Kitayama et al. (2013), a set of initial points is sampled with LHS in Step 1 and evaluated with the computationally expensive functions in Step 2. The width parameter is also calculated based on the number of sample points. In Step 3, each computationally expensive function is approximated with the Gaussian RBF, and a surrogate problem is built. The multiobjective surrogate problem is scalarized with the weighted  $L_p$ -norm function known e.g., in Miettinen (1999) as the method of weighted metrics in which a DM provides his/her preferences by the weights. In Step 4, this scalarized problem is optimized with the differential evolution algorithm (Kitayama et al. 2011b). This optimal solution is added to the sample points set and the width parameter is recalculated.

Introducing the first sampling function based on the current sample points, a set of sample points as evenly distributed as possible is obtained. These points are added to the current sample points. Then, the second sampling function is introduced and optimized based on the updated sample points and a modification of the Pareto fitness function (3). Both functions are approximated using a Gaussian RBF. In Step 5, if a stopping criterion based on the selected points is not met, the optimal solution obtained with the second sampling function is added to the current sample point. The updated sample points are evaluated with the computationally expensive functions. The width parameter is recalculated, and the surrogate problem is updated. The method then returns to Step 4. Otherwise, the last optimal solution of the second sampling function is considered as the most preferred solution for the DM.

Regarding the new estimated width parameter, the authors mention that this estimation has been obtained with a heuristic approach. Therefore, the validity of the new parameter is not proved mathematically. They also propose that this parameter can be used for SVR. The performance of this method was evaluated on a set of benchmark problems with nonconvex and disconnected Pareto frontiers. The efficiency of the method was also evaluated on an MOP of variable blank holder force trajectory in deep drawing with two nonlinear objective functions and six decision variables. The numerical simulation was done by LS-DYNA.

In the optimization-based methods discussed so far in this subsection, a surrogate problem is constructed with metamodeling techniques to approximate each computationally expensive function. In Monz et al. (2008), however, an interactive optimization-based method called Pareto Navigation method is introduced for convex problems to

find the most preferred solution with respect to a DM's preference without using any metamodeling techniques, and is applied in intensity-modulated radiation therapy planning. Based on a figure in this paper, we conclude that the application problem has six objective functions. Moreover, there is no information regarding the number of constraint functions and the decision variables. In this method, a set of initial sample points is generated with an a posteriori method (Steps 1 and 2). Then, in Step 3, convex hulls of these points are constructed in both the decision and objective spaces. The set of sample points is shown to the DM who must select one of them. The DM provides his/her preference information by specifying an aspiration level called a goal for one of the objective functions and upper bounds for the other objective functions. With respect to the given preference information, a convex scalarization problem introduced in Pascoletti and Serafini (1984) (based on a reference point) is formulated as the surrogate problem in Step 3, which is optimized in Step 4. The optimal solution of this problem is a convex combination of the Pareto optimal solutions in the sample points set in the decision and objective spaces corresponding to the given preferences. In Step 5, if the DM desires, this point can be evaluated with the computationally expensive functions. If the DM is not satisfied with this point, it can be added to the sample points set, and the surrogate problem is updated. Then, the method returns to Step 4. Otherwise, the obtained point is considered as the most preferred solution for the DM in Step 7.

In Eskelinen et al. (2010), an interactive method called Pareto navigator is introduced. In this method, conducting Steps 1 and 2, i.e., constructing a convex hull, and showing the set of sample points to the DM are similar to the Pareto navigation method (Monz et al. 2008). In Step 3 of the Pareto navigator method, however, the DM provides the preference information in a form of a classification or a reference point consisting of desirable values for all objectives. The main difference between this method and the method in Monz et al. (2008) is that in the latter, only one solution with respect to the given preference is generated whereas, in this method, multiple solutions corresponding to the given preference are available when moving from the current solution towards the reference point specified.

To be more specific, a search direction with respect to the preference information is formed. Then, based on the convex hull, the single objective optimization problem (2), where the reference point is parametrically moved along the search direction, is formulated as the surrogate problem. If the DM desires, (s)he can control the speed of movement. Pareto optimal solutions to the surrogate problem are generated by solving the single objective optimization problem for different reference points until the DM wants to stop. If the DM wishes, the corresponding Pareto optimal solu-

tion of any point generated is obtained by projecting this point to the Pareto frontier of the original problem using the achievement scalarization function (2) in Step 5. If the DM is satisfied with the projected Pareto optimal solution, it is considered as the most preferred solution for the DM in Step 7. Otherwise, the DM can change the preference information, that is, the direction of movement and/or the starting solution. If desired, it is possible to add the projected Pareto optimal solution to the sample points set in order to improve the accuracy of the surrogate problem. Then, Steps 4-5 are repeated. The performance of this method was evaluated on a simple problem with three nonlinear objective functions, three linear constraint functions, two decision variables and a simple nonconvex Pareto frontier. In the Pareto navigation method (Monz et al. 2008), the objective functions and the Pareto frontier are assumed to be convex, whereas in Pareto navigator (Eskelinen et al. 2010), a simple nonconvex Pareto frontier can be captured by the convex hull. Capturing a complex nonconvex Pareto frontier is mentioned as a future research direction in Eskelinen et al. (2010). While the Pareto navigation method keeps the connection to the decision space, Pareto navigator loses the connection with the decision space when navigating in the objective space. In the Pareto navigation method (Monz et al. 2008), convexity of the objective functions is assumed, which is an obstacle to deal with black-box functions. However, the method in Eskelinen et al. (2010) can handle black-box functions.

In Shan and Wang (2004), a sampling-based method called Pareto Set Pursuing (PSP) is introduced, which is an extension of the Mode Pursuing Sampling (MPS) method proposed in Wang et al. (2004). In Step 1 of PSP, initial points are sampled randomly. These points are evaluated with the computationally expensive functions in Step 2, and are saved in an archive based on their Pareto fitness values. In Step 3 of the method, two metamodeling techniques (a quadratic polynomial function and a linear RBF) are used to approximate the computationally expensive functions. Using the two surrogate problems, the authors introduce two functions to guide the sampling process, and devise two criteria for selecting between them during the sampling in Step 4. The first function selects sample points towards the extremes of the Pareto frontier, and the second one towards the entire Pareto frontier of the original problem. These sample points are then combined with the sample points in the archive and a subset of these is selected with the help of the second function. The selected points are evaluated in Step 5 with the computationally expensive functions and added to the archive. The authors have two criteria to stop the sampling process. If the stopping criteria are not met, the surrogate problem is updated. Then, the method returns to Step 4. Otherwise, the non-dominated solutions in the archive are considered as the approximated

Pareto frontier. The authors claim that PSP can capture a nonconvex and disconnected Pareto frontier. Similar to Goel et al. (2007) and Yang et al. (2002), the authors also discuss the difficulties of GA-based algorithms in identifying extreme solutions in the Pareto frontier. They claim that the first function can overcome this difficulty. As far as the accuracy of the surrogate problem is concerned, the authors claim that the accuracy of their method is not critical when solving an MOP. The efficiency of PSP was evaluated on solving an MOP of fuel cell component design with one nonlinear objective function, one black-box objective function, one black-box constraint function and three decision variables.

In Khokhar et al. (2010), the PSP method (Shan and Wang 2004) is modified for mixed integer MOPs as the MV-PSP method. In the MV-PSP method, the basic idea of employing a surrogate problem is similar to the PSP method. To deal with discrete variables, however, random sampling in PSP would most likely generate infeasible discrete sample points (Khokhar et al. 2010). In order to rectify this issue, the authors use a method to sample feasible discrete points during the sampling process. The performance of the MV-PSP method was evaluated on solving an MOP of welded beam design with two nonlinear objective functions, four nonlinear constraint functions and four decision variables. In this paper, the performance of the MV-PSP method was also compared with the performance of six evolutionary multiobjective optimization algorithms, i.e., AbYSS (Nebro et al. 2008), CellDE (Durillo et al. 2008), FastPGA (Eskandari and Geiger 2008), NSGA-II (Deb et al. 2002), OMOPSO (Reyes and Coello 2005) and SPEA2 (Zitzler et al. 2002). These evolutionary algorithms, however, have not been developed to deal with computationally expensive MOPs. To compare these methods, a set of benchmark problems with convex, nonconvex and disconnected Pareto frontiers was considered. Since the basic idea of MV-PSP and PSP is the same, for MOPs with real-valued and discrete decision variables, PSP and MV-PSP were applied, respectively. In this comparison, spread, generational distance, inverted generational distance, hyper-volume, generalized spread and percentage of the Pareto optimal solutions were considered as the performance indices to assess the quality of the solutions obtained by these methods. Based on this comparison, the authors claim to be relatively safe to mention that with a limited number of computationally expensive function evaluations, the PSP and MV-PSP methods outperform the compared methods for MOPs with two to three objective functions and less than eight decision variables. They observe that when the dimensionality increases to ten decision variables, PSP does not offer any superiority over the others. Handling this weakness is mentioned as a future research direction in Khokhar et al. (2010).

To summarize, in this subsection, we have surveyed methods in type 1 of the adaptive framework. In the following section, methods in type 2 of this framework are considered. After surveying all methods in the adaptive framework, we summarize characteristics of them in Table 2 of Section 6.2.

## 6 Adaptive framework: type 2

### 6.1 General flowchart

The flowchart in Fig. 6 outlines the main steps of methods belonging to type 2 of the adaptive framework. As mentioned earlier, the initial approximated Pareto frontier may not represent the exact Pareto frontier of the original problem. Thus, by solving and updating the surrogate problem iteratively, the approximated Pareto frontier is supposed to coincide with the Pareto frontier of the original problem. To do this, sampling new points is required iteratively. In type 1, sample points selected before assessing a stopping criterion are utilized to update the surrogate problem (Step 4 highlighted in Fig. 4). In type 2, not only the sample points generated before assessing the stopping criterion are considered (Step 4 highlighted in Fig. 6), but also new points are sampled after assessing the stopping criterion in other regions of the decision and/or objective space to update the surrogate problem (Step 6 highlighted in Fig. 6).

As can be seen in the flowchart in Fig. 6, a set of initial points is sampled in Step 1, and then evaluated with the computationally expensive functions in Step 2. After this, the function values of the sample points are available. In Step 3, an initial surrogate problem is built. In Step 4, a set of new sample points is generated relying on the surrogate problem. Similar to type 1, these points can be generated and selected e.g., by solving the surrogate problem with a DM if available. In Step 5, a stopping criterion is checked, which may require to evaluate the selected sample points with the computationally expensive functions. If the criterion is not met, based on a method-dependent criterion, a subset of non-dominated solutions generated in Step 4, which may have been evaluated with the computationally expensive functions, is selected. Then, in Step 6, a set of new points in other regions of the decision and/or objective space is sampled. These points along with the sample points selected in Step 5 are considered, and Steps 2-5 are then repeated. Otherwise, the set of non-dominated (or Pareto optimal) solutions or the most preferred solution of the last surrogate problem, which may have been evaluated with the computationally expensive functions, is considered as either the approximated Pareto frontier of the original problem or the most preferred solution for a DM, respectively. These solutions can also be visualized by an appropriate

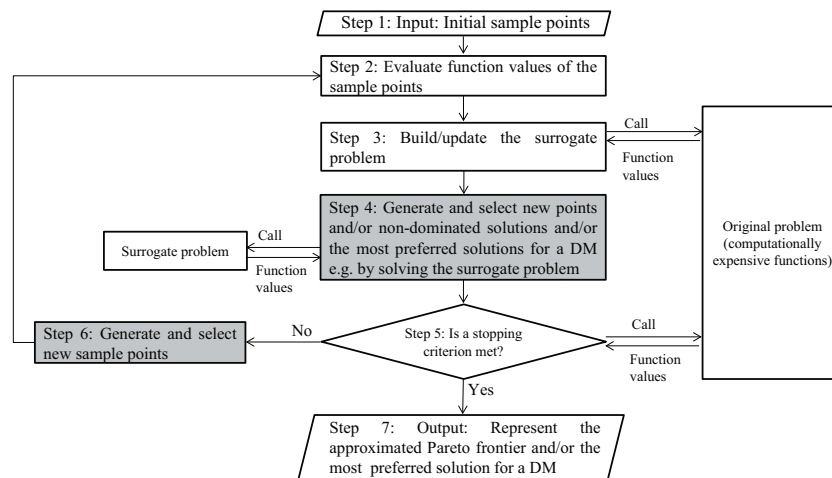


Fig. 6 Flowchart of the adaptive framework: type 2

visualization technique. In Section 6.2, we summarize the methods in type 2 of the adaptive framework. These methods are then compared with other methods in type 1 of the adaptive framework in Section 8.

### 6.2 Summary of methods in the adaptive framework: type 2

In Liu et al. (2008), an optimization-based method is introduced using a quadratic polynomial function. The authors claim that this method is highly efficient and is less dependent on the accuracy of the surrogate problem. From this point of view, they claim that the framework of their method is a new framework which covers surrogate-based methods, and consider it as a novel multiobjective optimization method based on an approximation model management technique. Type 2 of the adaptive framework has been inspired by this paper. In Chen et al. (2012) also an optimization-based method similar to the idea in Liu et al. (2008) is introduced. Without loss of generality, we summarize these methods together. In both methods, the sample points are selected within a trust region in the decision space. To do this, an initial trust region is considered in the decision space by initializing its parameters, i.e., trust region center, radius and bounds, threshold and control constants to update the trust region. The authors also define a set of equations to update the trust region in each iteration of the methods.

After initializing the trust region, in Liu et al. (2008) and Chen et al. (2012) initial points are sampled using OLHS and LHS, respectively, within the trust region in Step 1, and evaluated with the computationally expensive functions in Step 2. Each computationally expensive func-

tion is approximated in Step 3 with a quadratic polynomial function and a Gaussian RBF in Liu et al. (2008) and Chen et al. (2012), respectively, and the initial surrogate problem is constructed. In Step 4, non-dominated solutions of these surrogate problems are obtained by solving them using  $\mu$ -MOGA (Liu and Han 2006). Then, in both methods, the set  $P_a$  consisting of a subset of evenly distributed non-dominated solutions in the decision space from the obtained non-dominated solutions is considered. In Step 5, the points in  $P_a$  are evaluated with the computationally expensive functions. The non-dominated solutions of the evaluated points are stored in an archive  $P_e$ . In order to check a stopping criterion, the set  $P_t = P_a \cap P_e$  is considered. Then, a ratio between the number of points in  $P_t$  and  $P_a$  is calculated. Based on this ratio, the predefined equations and the constants, the trust region is updated by adjusting its parameters.

In Liu et al. (2008), if the radius of the updated trust region is less than a given threshold, or the number of iterations of solving and updating the surrogate problem is more than a given threshold, the method stops. In (Chen et al. 2012), other criteria based on the calculated ratio or the accuracy of the surrogate problem are also checked. If either of the stopping criteria is met, in both methods, the set  $P_e$  is considered as the approximated Pareto frontier of the original problem in Step 7. Otherwise, in both methods, a set of new points is sampled with LHS in Step 6. These points and those points from  $P_e$  that fall into the updated trust region are considered. In Liu et al. (2008), Steps 2–5 are then repeated. In Chen et al. (2012), a subset of those points from  $P_e$  that are in the updated trust region is first removed with respect to a distance coefficient. This action avoids

singularity in the matrix of updating the RBF. The remaining points along with the new sample points are selected, and Steps 2–5 are repeated.

These methods can capture nonconvex and disconnected Pareto frontiers. The authors in Liu et al. (2008) discuss that capturing such Pareto frontiers depends on the initial trust region radius. The efficiency of the method in Liu et al. (2008) was evaluated on a black-box biobjective optimization problem of variable binder force in a car sheet metal forming with three decision variables. LS-DYNA was employed as a simulator. As far as the method in Chen et al. (2012) is concerned, the distribution of the non-dominated solutions relies on the quality of  $\mu$ -MOGA. The distance coefficient affects the performance of this method. Based on the obtained results using the benchmark problems, the authors suggest the best value for the distance coefficient. The performance of this method was evaluated on a structure optimization problem of a vehicle door with two black-box objective functions, two black-box constraint functions and five decision variables involving the FE method. In Liu et al. (2008), closeness of the Pareto frontiers of the surrogate and benchmark problems was considered to assess the quality. In Chen et al. (2012), the quality was assessed based on the spread of non-dominated solutions obtained by the surrogate problem and closeness of them to the Pareto frontier of the benchmark problems.

All the methods described in Sections 5 and 6 are compared in Section 8. Table 2 summarizes the characteristics of the methods belonging to both types 1 and 2 in the adaptive framework with respect to sampling techniques, metamodeling techniques, number of objective and constraint functions as well as decision variables in the considered benchmark and application problems, and whether the methods are optimization- or sampling-based. In this table, for every method, the most challenging MOP that was considered as a benchmark or an application problem is mentioned. Since the number of equality constraints in all problems are zero ( $p = 0$ ), it is not mentioned in the table. As can be seen, most of the problems consist of two or three objective functions except (Zhou and Turng 2007; Gobbi et al. 2013) with five objective functions.

## 7 A hybrid method

In Messac and Mullur (2008), a two-stage optimization-based method called Pseudo Response Surface (PRS) which hybridizes both the sequential framework and type 1 of the adaptive framework is introduced. Since discussion on the frameworks utilized in Sections 4 and 5 are necessary to describe this method, we discuss it in this section. This method is compared with other methods in Section 8. PRS consists of two stages where handling

computationally expensive constraint functions is not addressed. In the first stage, a surrogate problem is built and updated iteratively in the same way in type 1 of the adaptive framework to generate a set of as evenly distributed feasible solutions as possible near the Pareto frontier of the original problem. This set is utilized to build another surrogate problem in stage two with the sequential framework. The second surrogate problem is accurate near the Pareto frontier of the original problem and intentionally inaccurate in other regions of the objective space. By formulating and solving the second surrogate problem, an approximation of the Pareto frontier is represented.

In stage one of PRS, extreme solutions are calculated. The extreme solutions, the utopia hyperplane and normal constraint method (Messac et al. 2003; Messac and Mattson 2004) are employed to generate feasible solutions near the Pareto frontier of the original problem. In Step 1 of type 1 of the adaptive framework utilized in this stage, a set of evenly distributed points including the extreme solutions is generated on the utopia hyperplane. Near every point on the hyperplane, a set of new points is sampled with FFS or LHS. These points are evaluated with the computationally expensive functions in Step 2. The individual computationally expensive functions are then approximated with a multi-quadratic RBF in Step 3, and an initial surrogate problem is built. In Step 4, the surrogate problem is scalarized using the normal constraint method introduced in Messac et al. (2003) and Messac and Mattson (2004). A feasible solution near the Pareto frontier is generated by solving this scalarized problem. In Step 5, this point is evaluated with the computationally expensive functions, and replaced with the oldest sample point in the sample points set. The surrogate problem is then updated. Steps 4 and 5 are repeated until all points on the hyperplane are used. As a result, in the first stage, a set of evenly distributed feasible solutions near the Pareto frontier is obtained in Step 6. These points are considered as the evaluated initial sample points for Steps 1 and 2 of the sequential framework used in stage two.

As mentioned earlier, the second stage of PRS concentrates on building an accurate surrogate problem near the Pareto frontier and is intentionally inaccurate in other regions of the objective space. To do this, based on the dimensionality of the decision space, a number of new sample points is selected far from the sample points set obtained from stage one. Instead of evaluating the selected points with the computationally expensive functions, every objective function value of these points is set such that each point has a higher objective value in comparison with the objective function value of the nearest point among the sample points evaluated with the computationally expensive functions. Then, each computationally expensive function is approximated with an extension of RBF called E-RBF (Mullur and Messac 2005) in Step 3 of the sequential



**Table 2** Characteristics of methods in the adaptive framework ( $k, m, n$  are the number of objective functions, inequality constraint functions and decision variables, respectively)

Characteristics Ref.	Sampling technique	Metamodeling technique	Features of the benchmark problem	Features of the application problem	Type of the method
(Yang et al. 2002)	LHS	Kriging model	No benchmark problem	$k = 2, m = 1, n = 4$	Optimization-based (type 1)
(Zhou and Turng 2007)	LHS	Kriging model	No benchmark problem	$k = 5, m = 0, n = 3$	Optimization-based (type 1)
(Yun et al. 2009)	Random sampling	$\mu - v - SVR$	$k = 2, m = 0, n = 2$ , nonconvex and disconnected Pareto frontier	$k = 2, m = 4, n = 4$	Optimization-based (type 1)
(Jakobsson et al. 2010)	LHS	Thin plate spline RBF	$k = 2, m = 0, n = 3$ , nonconvex and disconnected Pareto frontier	$k = 3, m = 0, n = 5$	Optimization-based (type 1)
(Gobbi et al. 2013)	Random sampling	Neural network	$k = 2, m = 0, n = 5$ , nonconvex Pareto frontier	$k = 5, m = 0, n = 11$	Optimization-based (type 1)
(Kitayama et al. 2013)	LHS	Gaussian RBF	$k = 2, m = 0, n = 2$ , nonconvex and disconnected Pareto frontier	$k = 2, m = 0, n = 6$	Optimization-based (type 1)
(Monz et al. 2008)	A posterior method	No metamodeling technique	No benchmark problem	$k = 6, m$ and $n$ not mentioned	Optimization-based (type 1)
(Eskelinen et al. 2010)	A posterior method	No metamodeling technique	$k = 3, m = 3, n = 2$ , nonconvex Pareto frontier	No application problem	Optimization-based (type 1)
(Shan and Wang 2004)	Random sampling	Polynomial function, Linear RBF	$k = 3, m = 1, n = 5$ , nonconvex and disconnected Pareto frontier	$k = 2, m = 0, n = 3$	Sampling-based (type 1)
(Khokhar et al. 2010)	Random sampling	Polynomial function, Linear RBF	$k = 3, m = 0, n = 7$ , nonconvex and disconnected Pareto front	$k = 2, m = 4, n = 4$	Sampling-based (type 1)
(Liu et al. 2008)	LHS	Polynomial function	$k = 2, m = 2, n = 2$ , nonconvex and disconnected Pareto front	$k = 2, m = 0, n = 3$	Optimization-based (type 2)
(Chen et al. 2012)	LHS	Gaussian RBF	$k = 2, m = 0, n = 3$ , nonconvex and disconnected Pareto frontier	$k = 2, m = 2, n = 5$	Optimization-based (type 2)

framework, and an initial surrogate problem is built. In Step 4, the accuracy of the surrogate problem is evaluated near the Pareto frontier of the original problem by RMSE. In Step 5, if the accuracy is not sufficient, new points near the Pareto frontier are obtained using the first stage, and the surrogate problem is updated in Step 3.

Having the accurate enough surrogate problem, the set of non-dominated solutions is obtained in Step 6 by solving the surrogate problem of the stage two. This set is considered as the approximated Pareto frontier of the original problem in Step 8. These solutions are not evaluated with the computationally expensive functions in Step 7. The performance of PRS was evaluated on benchmark problems with non-convex Pareto frontiers. The PRS method was also applied to a complex structural optimization problem requiring the FE method and analysis of rigidified inflatable structures. It had two black-box objective functions, three black-box constraints and three decision variables. The authors claim that the computational cost in simulation-based MOPs can be reduced considerably since the surrogate problem obtained with PRS is accurate near the Pareto frontier. While, the authors state that the approximated Pareto frontier is the Pareto frontier of the original problem, mathematical proof of this statement was mentioned as a future research direction. This method assumes that the ranges of the objective function values are available to compute the extreme solutions. This assumption can be a challenge. Although the authors expect that PRS can capture a disconnected Pareto frontier, they mention handling multimodalities and disconnected Pareto frontiers as a future research direction.

## 8 Comparison of surrogate-based multiobjective optimization methods

After giving an overview of surrogate-based methods proposed in the literature, we compare them in this section. As mentioned in Section 1, in this comparison, we concentrate on the attributes of the methods in four aspects: 1) Can the methods handle general black-box functions where information regarding mathematical properties of the functions such as convexity and continuity is not available? 2) Can the methods capture different types of Pareto frontiers? 3) How many objective and constraint functions as well as decision variables can be handled by the methods? 4) What is the role of a DM during the solution process? For this comparison, we consider the results given by the authors of individual papers on the benchmark and application problems that they have used to evaluate the performance of their methods. Throughout this comparison, we also discuss future research directions.

As far as black-box functions are concerned, all the described methods except (Monz et al. 2008) can handle

such functions. In Monz et al. (2008), the convexity of the objective functions is assumed which cannot be guaranteed when considering black-box functions. We observe that only in Jakobsson et al. (2010) dealing with noisy black-box functions is discussed. However, the authors evaluated the efficiency of their method on benchmark problems without noise. Based on these observations, more research to figure out how metamodeling techniques can be applied to handle noisy black-box functions is required.

As far as handling nonconvex and disconnected Pareto frontiers is concerned, we observe that capturing a potential region to select sample points for updating the surrogate problem has a vital and a critical role. In Shan and Wang (2004), Kitayama et al. (2013), Yun et al. (2009), Jakobsson et al. (2010) and Khokhar et al. (2010), metamodeling techniques not only are employed to build the surrogate problem, but also to introduce functions for sampling. In Shan and Wang (2004) and Khokhar et al. (2010) these functions choose sample points towards the extreme solutions and the Pareto frontier. In Jakobsson et al. (2010), the sampling function selects sample points from or close to the extended Pareto frontier of the surrogate problem. In Kitayama et al. (2013), the aim of employing the sampling function is to generate evenly distributed sample points. The authors argue that sampling evenly distributed points to build and update the surrogate problem may help to capture disconnected parts of the Pareto frontier.

In Zhou and Turng (2007), the sample points are selected based on the variances of the objective function values of the sample points predicted by the Kriging model. In addition, expected improvement is another criterion related to Kriging model used in nature-inspired methods (Wagner et al. 2010) to select a new point for updating a surrogate problem. Such a criterion can also be considered in non-nature inspired methods. In Yun et al. (2009), Lagrangian multiplier values given by the metamodeling technique play a role to select new sample points. In Liu et al. (2008) and Chen et al. (2012), the trust region strategy is employed to select sample points from the explored and unexplored regions in the decision and/or objective space. In both methods, the sample points are selected to update the surrogate problem before and after assessing a stopping criterion. In Wilson et al. (2001), a considerable number of sample points is selected to capture nonconvexity and discontinuity, which highly relies on the accuracy of the surrogate problem. In Goel et al. (2007), convex hulls are discussed to capture a nonconvex Pareto frontier. In this paper, a metamodeling technique is utilized to approximate a function which visualizes the approximated Pareto frontier. This function represents an objective as a function of other objectives.

The authors in Eskelinen et al. (2010) claim that a simple nonconvex Pareto frontier can be captured by their method, because a convex hull is applied. In the PAINT method

(Hartikainen et al. 2012), which can capture a nonconvex Pareto frontier, an interpolation between sample points is built. In Messac and Mullur (2008) and Gobbi et al. (2013), the idea of the normal constraint method (Messac et al. 2003; Messac and Mattson 2004) is employed to generate sample points near the Pareto frontier of the original problem. In Messac and Mullur (2008), the surrogate problem is constructed by type 1 of the adaptive framework to generate initial sample points. These points are used to form a surrogate problem of a computationally expensive MOP by the sequential framework. However, handling a disconnected Pareto frontier in Messac and Mullur (2008), Goel et al. (2007), Hartikainen et al. (2012) and Zhou and Turng (2007) is not considered. Based on the above observations, developing a method to handle the issues of non-convexity and discontinuity in the Pareto frontier deserves research efforts.

One can see that in Messac and Mullur (2008), Shan and Wang (2004), Kitayama et al. (2013), Jakobsson et al. (2010) and Khokhar et al. (2010), RBFs have been employed to handle computationally expensive MOPs. The benchmark or application problems in Messac and Mullur (2008) have nonconvex Pareto frontiers, while in Shan and Wang (2004), Kitayama et al. (2013), Jakobsson et al. (2010) and Khokhar et al. (2010), they have nonconvex and disconnected Pareto frontiers. Now, a question arises whether the type of a metamodel technique has any impact on capturing a nonconvex and disconnected Pareto frontier. In addition, we observe that many methods developed in the adaptive framework can capture a nonconvex and disconnected Pareto frontier. Based on this observation, we recommend employing the adaptive framework to handle the issue of nonconvexity and discontinuity in the Pareto frontier.

The methods of Hartikainen et al. (2012), Monz et al. (2008) and Eskelinen et al. (2010) introduce surrogate problems to approximate directly the Pareto frontier. Both methods in Hartikainen et al. (2012) and Eskelinen et al. (2010) lose the connection to the decision space when solving the surrogate problems. Nevertheless, the corresponding decision vector value is available after projecting the approximated preferred solution obtained by the surrogate problems to the Pareto frontier of the original problem (although, this can be time consuming). In contrast, the Pareto navigation method (Monz et al. 2008) keeps the connection to the decision space during the navigation in the objective space. Apart from an approximation of the Pareto frontier in the objective space, there is a possibility to approximate the set of Pareto optimal solutions in the decision space. In Bhardwaj et al. (2013), a method is proposed to approximate this set in the decision space rather than in the objective space. In this method, however, handling a computationally expensive MOP is not concerned.

Moreover, this method requires differentiability of the objective and constraint functions, and, thus, we do not discuss it further. However, one can consider the idea of approximating the set of Pareto optimal solutions in the decision space as an approach to handle computationally expensive MOPs. In addition, developing a method to approximate the Pareto frontier directly rather than each computationally expensive function in which the connection to the decision space is kept, can be considered for further research.

As far as the number of objective and constraint functions as well as decision variables is concerned, except in Lotov et al. (2001), Su et al. (2011), Zhou and Turng (2007), Gobbi et al. (2013), the authors in other considered papers employed benchmark and/or application problems limited to at most three objective functions, four constraint functions and seven decision variables. The highest number of objective functions ( $k = 5$ ) was considered in Lotov et al. (2001), Zhou and Turng (2007), Gobbi et al. (2013) and Monz et al. (2008). In Lotov et al. (2001), a surrogate problem is formed based on a sampling strategy, in Gobbi et al. (2013), based on a neural network, in Zhou and Turng (2007), based on a Kriging model and in Monz et al. (2008), based on the convex hulls. The sampling strategy, however, can be unstable to handle an MOP with a high number of decision variables. In addition, training a neural network is time consuming. In Lotov et al. (2001), Su et al. (2011), Zhou and Turng (2007) and Gobbi et al. (2013), it remains unclear how to capture a disconnected Pareto frontier. In (Monz et al. 2008), the convexity of the objective functions is required. An MOP with the highest number of constraint functions ( $m = 13$ ) and decision variables ( $n = 31$ ) was considered in Su et al. (2011) based on the sequential framework. In (Shan and Wang 2004; Khokhar et al. 2010), the authors note that their methods can be employed to solve a computationally expensive MOP with two or three objective functions and less than seven decision variables. Moreover, applying the method in Jakobsson et al. (2010) to handle an MOP with more than three objective functions and six decision variables is unsuitable. In other methods, the authors do not discuss upper bounds on the number of objective and constraint functions as well as decision variables. See Shan and Wang (2010) for a review on handling a high-dimensional design problem. The ideas may be extended to deal with an MOP with a high number of objective and constraint functions and decision variables.

Considering the role of a DM as defined in Section 2.1 in solving an MOP has been discussed in Kitayama et al. (2013), Lotov et al. (2001), Hartikainen et al. (2012), Yun et al. (2009), Monz et al. (2008) and Eskelinen et al. (2010). A DM provides his/her preferences in the form of weights in Kitayama et al. (2013) and Yun et al. (2009), while in Lotov et al. (2001), (s)he compares the generated solutions.

In Monz et al. (2008), a DM provides his/her preferences as a goal for one of the objective functions and upper bounds for the other ones. In Eskelinen et al. (2010), (s)he can express the preferences by means of a classification or a reference point consisting of aspiration levels for all objectives. In this aspect, PAINT has a distinguished feature, because a mixed integer linear multiobjective surrogate problem of the original problem is created without involving preferences. Therefore, it can be employed with any interactive method. The aim in methods of Kitayama et al. (2013), Lotov et al. (2001), Hartikainen et al. (2012), Monz et al. (2008) and Eskelinen et al. (2010) is to provide the most preferred solution for a DM rather than to approximate the entire Pareto frontier. Involving a DM when solving an MOP has not been considered in many papers and, thus, can be considered as another future research direction.

As mentioned earlier, the quality of the approximated Pareto frontier can be assessed based on Wu and Azarm (2000), Okabe et al. (2003) and Zitzler et al. (2003, 2008), the number of solutions in the approximated Pareto frontier, the distribution and spread of these solutions and closeness of the approximated Pareto frontier to the Pareto frontier of the original problem. Evaluating the performance of the surveyed methods based on these aspects with respect to a limited number of computationally expensive function evaluations requires implementing and testing them on a common set of benchmark problems. Nevertheless, the benchmark problems considered in the surveyed papers were not computationally expensive. Thus, to assess the performance of surrogate-based methods to handle computationally expensive problems, developing computationally expensive benchmark problems is a future research topic.

## 9 Conclusions

Many real-world (engineering) problems involve multiple conflicting objectives (and possibly constraints) that are computationally expensive to evaluate. In order to deal with the high computational cost, various surrogate-based methods which are independent of the type of the optimization algorithms used in them, have been proposed in the literature. So far, however, no survey has been available summarizing their characteristics, similarities and differences. This survey provided an overview of the methods available and with it, one can find out whether there exist a method applicable for one's own problems.

Based on when the surrogate problem is built and updated, we classified these methods into the sequential and the adaptive frameworks. A special case of the sequential framework termed as one-stage sampling was also considered. In this case, points were sampled only once to build the surrogate problem. In accordance with when sample points

are selected to update the surrogate problem, we divided the adaptive framework into types 1 and 2. The considered papers were then summarized and compared. Based on this comparison, we can identify the following challenges as future research directions:

- handling noisy black-box functions,
- capturing a nonconvex and disconnected Pareto frontier,
- handling a high number of objective and constraint functions as well as decision variables,
- providing the most preferred solution for a DM when solving computationally expensive MOPs,
- developing computationally expensive benchmark problems.

In order to capture a nonconvex and disconnected Pareto frontier, we recommend developing a method in the adaptive framework. We hope that this survey opens new horizons for the researchers interested in this topic.

**Acknowledgments** Mohammad Tabatabaei thanks the COMAS doctoral program in computing and mathematical sciences and Karthik Sindhya thanks the TEKES -the Finnish Funding Agency for Technology and Innovation (the SIMPRO project) for the financial assistance.

## References

- Abaqus (2013). <http://www.3ds.com/products-services/simulia/portfolio/abaqus/overview/>. Accessed Dec 2013
- Bhardwaj P, Dasgupta B, Deb K (2013) Modelling the Pareto-optimal set using b-spline basis functions for continuous multi-objective optimization problems. *Eng Optim*:1–27
- Bornatico R, Pfeiffer M, Witzig A (2011) Untersuchung ausgewählter solarsysteme durch abtasten grosser parameteräume. In: Proceedings of the 21th OTTI solar thermal technology symposium. Bad Staffelstein, Germany
- Bornatico R, Hüsey J, Witzig A, Guzzella L (2013) Surrogate modeling for the fast optimization of energy systems. *Energy* 57: 653–662
- Buhmann MD (2003) Radial basis functions: theory and implementations. Cambridge University Press, New York
- Clarke SM, Griebisch JH, Simpson TW (2004) Analysis of support vector regression for approximation of complex engineering analyses. *J Mech Des* 127(6):1077–1087
- Chen G, Han X, Liu G, Jiang C, Zhao Z (2012) An efficient multi-objective optimization method for black-box functions using sequential approximate technique. *Appl Soft Comput* 12(1):14–27
- Deb K, Thiele L, Laumanns M, Zitzler E (2002) Scalable multi-objective optimization test problems. In: Proceedings of congress on evolutionary computation, vol 1. IEEE, Honolulu, USA, pp 825–830
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6:183–197
- Durillo J, Nebro A, Luna F, Alba E (2008) Solving three-objective optimization problems using a new hybrid cellular genetic algorithm. In: Rudolph G, Jansen T, Lucas C, Poloni S, Beume N (eds) Parallel problem solving from nature – PPSN X, volume 5199 of Lecture notes in computer science. Springer Berlin Heidelberg, pp 661–670

- Eskandari H, Geiger CD (2008) A fast Pareto genetic algorithm approach for solving expensive multiobjective optimization problems. *J Heuristics* 14:203–241
- Eskelinen P, Miettinen K, Klamroth K, Hakanen J (2010) Pareto navigator for interactive nonlinear multiobjective optimization. *OR Spectrum* 32(1):211–227
- Friedman JH (1991) Multivariate adaptive regression splines. *Ann Stat* 19(1):1–67
- Forrester AIJ, Sobester A, Keane AJ (2008) *Engineering design via surrogate modelling: a practical guide*. Wiley, Chichester
- Forrester AIJ, Keane AJ (2009) Recent advances in surrogate-based optimization. *Prog Aerosp Sci* 45:50–79
- Gamito MN, Maddock SC (2009) Accurate multidimensional poisson-disk sampling. *ACM Trans Graph* 29(1):1–19
- Giunta A, Watson LT, Koehler J (1998) A comparison of approximation modeling techniques: polynomial versus interpolating models. In: *Proceedings of 7th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization*, vol 1. St. Louis, MO, pp 392–404. AIAA-98-4758
- Gobbi M, Guarneri P, Scala L, Scotti L (2013) A local approximation based multi-objective optimization algorithm with applications. *Optim Eng*:1–23
- Goel T, Vaidyanathan R, Haftka R, Shyy W, Queipo N, Tucker K (2004) Response surface approximation of Pareto optimal front in multi-objective optimization. In: *Proceedings of the 10th AIAA/ISSMO multidisciplinary analysis and optimization conferences*, Albany, NY, pp 2230–2245
- Goel T, Vaidyanathan R, Haftka RT, Shyy W, Queipo NV, Tucker K (2007) Response surface approximation of Pareto optimal front in multi-objective optimization. *Comput Methods Appl Mech Eng* 196:879–893
- Goldberg DE (1989) *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Longman Publishing Co., New York
- Hagan MT, Demuth HB, Beale M (1996) *Neural network design*. PWS, Boston
- Hakanen J, Miettinen K, Sahlstedt K (2011) Wastewater treatment: new insight provided by interactive multiobjective optimization. *Decis Support Syst* 51(2):328–337
- Hartikainen M, Miettinen K, Wiecek MM (2012) PAINT: Pareto front interpolation for nonlinear multiobjective optimization. *Comput Optim Appl* 52(3):845–867
- Helton JC, Johnson JD, Sallaberry CJ, Storlie CB (2006) Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliab Eng Syst Saf* 91(10–11):1175–1209
- Hickernell FJ, Lemieux C, Owen AB (2005) Control variates for quasi-monte carlo. *Stat Sci* 20(1):1–31
- Jakobsson S, Patriksson M, Rudholm J, Wojciechowski A (2010) A method for simulation based optimization using radial basis functions. *Optim Eng* 11(4):501–532
- Jakobsson S, Saif-Ul-Hasnain M, Rundqvist R, Edelvik F, Andersson B, Patriksson M, Ljungqvist M, Lortet D, Wallesten J (2010) Combustion engine optimization: a multiobjective approach. *Optim Eng* 11(4):533–554
- Jin R, Chen W, Simpson TW (2001) Comparative studies of metamodelling techniques under multiple modelling criteria. *Struct Multidiscip Optim* 23(1):1–13
- Jin Y (2011) Surrogate-assisted evolutionary computation: recent advances and future challenges. *Swarm Evol Comput* 1(2):61–70
- Johnson ME, Moore LM, Ylvisaker D (1990) Minimax and maximin distance designs. *J Stat Plan Inf* 26(2):131–148
- Kleijnen JPC (2009) Kriging metamodeling in simulation: a review. *Eur J Oper Res* 192(3):707–716
- Kitayama S, Arakawa M, Yamazaki K (2011) Sequential approximate optimization using radial basis function network for engineering optimization. *Optim Eng* 12(4):535–557
- Kitayama S, Arakawa M, Yamazaki K (2011) Differential evolution as the global optimization technique and its application to structural optimization. *Appl Soft Comput* 11(4):3792–3803
- Khokhar ZO, Vahabzadeh H, Ziai A, Wang GG, Menon C (2010) On the performance of the PSP method for mixed-variable multi-objective design optimization. *J Mech Des* 132(7):1–11
- Kitayama S, Srirat J, Arakawa M, Yamazaki K (2013) Sequential approximate multi-objective optimization using radial basis function network. *Struct Multidiscip Optim* 48(3):501–515
- 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
- Knowles J, Nakayama H (2008) Meta-modeling in multiobjective optimization. In: *Blaszczyński J, Jin Y, Shimoyama K, Slowinski R (eds) Multiobjective optimization: interactive and evolutionary approaches*. Springer-Verlag Berlin Heidelberg, pp 245–284
- Koziel S, Ciaurri DE, Leifsson L (2011) Surrogate-based methods. In: *Koziel S, Yang X (eds) Studies in computational intelligence*, vol 356. Springer Berlin Heidelberg, pp 33–59
- Kursawe F (1991) A variant of evolution strategies for vector optimization. In: *Schwefel HP, Männer R (eds) Parallel Problem Solving from Nature*, vol 496. Springer Berlin Heidelberg, pp 193–197
- Kutner MH, Nachtsheim CJ, Neter J, Li W (2005) *Applied linear statistical models*. McGraw Hill, Boston
- Li YF, Ng SH, Xie M, Goh TN (2010) A systematic comparison of metamodeling techniques for simulation optimization in decision support systems. *Appl Soft Comput* 10(4):1257–1273
- Liao X, Li Q, Yang X, Zhang W, Li W (2008) Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Struct Multidiscip Optim* 35(6):561–569
- Liu GP, Han X (2006) A micro multi-objective genetic algorithm for multi-objective optimizations. In: *The 4th China–Japan–Korea joint symposium on optimization of structural and mechanical systems*, Kunming, China
- Liu GP, Han X, Jiang C (2008) A novel multi-objective optimization method based on an approximation model management technique. *Comput Methods Appl Mech Eng* 197:2719–2731
- Livermore Software Technology Corporation (LSTC) (2013). <http://www.lstc.com/products/ls-dyna>. Accessed Dec 2013
- Lotov AV, Bushenkov VA, Kamenev GK (2001) Feasible goals method. Search for smart decisions. Computing Center of RAS, Moscow
- Lotov AV, Bushenkov VA, Kamenev GK (2004) *Interactive decision maps: approximation and visualization of Pareto frontier*. Kluwer Academic Publishers, Massachusetts
- Luque M, Ruiz F, Miettinen K (2011) Global formulation for interactive multiobjective optimization. *OR Spectrum* 33(1):27–48
- Madsen JJ, Shyy W, Haftka RT (2000) Response surface techniques for diffuser shape optimization. *AIAA J* 3(9):1512–1518
- Marler RT, Arora JS (2004) Survey of multi-objective optimization methods for engineering. *Struct Multidiscip Optim* 26(6):369–395
- Messac A, Ismail-Yahaya A, Mattson CA (2003) The normalized normal constraint method for generating the Pareto frontier. *Struct Multidiscip Optim* 25(2):86–98
- Messac A, Mattson CA (2004) Normal constraint method with guarantee of even representation of complete Pareto frontier. *AIAA J* 42(10):2101–2111
- Messac A, Mullur AA (2008) A computationally efficient metamodeling approach for expensive multiobjective optimization. *Optim Eng* 9(1):37–67
- Miettinen K (1999) *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, Norwell
- Miettinen K, Ruiz F, Wierzbicki AP (2008) Introduction to multi-objective optimization: interactive approaches. In: *Branke J, Deb K, Miettinen K, Slowinski R (eds) Multiobjective optimization:*

- interactive and evolutionary approaches. Springer-Verlag Berlin Heidelberg, pp 27–57
- Miettinen K (2014) Survey of methods to visualize alternatives in multiple criteria decision making problems. *OR Spectrum* 36(1): 3–37
- Moldex3d: Plastic injection molding simulation software (2013). <http://www.moldex3d.com/en/>. Accessed Dec 2013
- Monz M, Küfer KH, Bortfeld TR, Thieke C (2008) Pareto navigation—algorithmic foundation of interactive multi-criteria IMRT planning. *Phys Med Biol* 53(4):985–998
- MSC Nastran-Multidisciplinary structural analysis (2013). <http://www.mscsoftware.com/product/msc-nastran>, Accessed Dec 2013
- Mullur AA, Messac A (2005) Extended radial basis functions: more flexible and effective metamodeling. *AIAA J* 43(6):1306–1315
- Nakayama H, Yun Y, Yoon M (2009) Sequential approximate multi-objective optimization using computational intelligence, 1st edn. Springer Publishing Company, Incorporated, Berlin
- Nebro AJ, Luna F, Alba E, Dorronsoro B, Durillo JJ, Beham A (2008) AbYSS: adapting scatter search to multiobjective optimization. *IEEE Trans Evol Comput* 12:439–457
- Niederreiter H (1992) Random number generation and quasi-Monte Carlo methods. Philadelphia
- Okabe T, Jin Y, Sendhoff B (2003) A critical survey of performance indices for multi-objective optimisation. In: *The 2003 congress on Evolutionary computation, 2003. CEC '03. vol 2*, pp 878–885
- Pascoletti A, Serafini P (1984) Scalarizing vector optimization problems. *J Optim Theory Appl* 42(4):499–524
- Polysun (2013). <http://www.polysunonline.com/PsoPublic/app/home/access>. Accessed Dec 2013
- Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Tucker KP (2005) Surrogate-based analysis and optimization. *Prog Aerosp Sci* 41(1):1–28
- Reyes M, Coello CA (2005) Improving PSO-based multi-objective optimization using crowding, mutation and epsilon-dominance. In: Coello Coello C, Aguirre AH, Zitzler E (eds) *Proceedings of 3rd international conference on evolutionary multi-criterion optimization*, number 505–519 in *EMO'05*, Guanajuato, Mexico. Springer, Berlin
- Rezaveisi M, Sepehrnoori K, Johns RT (2014) Tie-simplex-based phase-behavior modeling in an IMPEC reservoir simulator. *Soc Pet Eng* 19(02):327–339
- Ruiz F, Luque M, Miettinen K (2012) Improving the computational efficiency in a global formulation (GLIDE) for interactive multi-objective optimization. *OR Spectrum* 197(1):47–70
- Schaumann E, Balling R, Day K (1998) Genetic algorithms with multiple objectives. In: *Proceedings of the 7th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization*, vol 3. Washington, DC, pp 2114–2123
- Shan S, Wang GG (2004) An efficient Pareto set identification approach for multiobjective optimization on black-box functions. *J Mech Des* 127(5):866–874
- Shan S, Wang GG (2010) Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Struct Multidiscip Optim* 41(2):219–241
- Simpson TW, Poplinski JD, Koch PN, Allen JK (2001) Metamodels for computer-based engineering design: survey and recommendations. *Engineering with Computers* 17(2):129–150
- Simpson TW, Booker AJ, Ghosh D, Giunta AA, Koch PN, Yang RJ (2004) Approximation methods in multidisciplinary analysis and optimization: a panel discussion. *Struct Multidiscip Optim* 27(5):302–313
- Simpson TW, Toropov V, Balabanov V, Viana F (2008) Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come - or not. In: *Proceedings of the 12th AIAA/ISSMO multidisciplinary analysis and optimization conference*, pages AIAA–2008–5802, British Columbia
- Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. *Stat Comput* 14(3):199–222
- Star-CD (2013). <http://www.cd-adapco.com/products/star-cd>. Accessed Dec 2013
- Steinhaus H (2011) *Mathematical snapshots*. Courier Dover Publications, New York
- Steuer RE (1986) *Multiple criteria optimization: theory, computation and application*. Wiley, New York
- Su R, Gui L, Fan Z (2011) Multi-objective optimization for bus body with strength and rollover safety constraints based on surrogate models. *Struct Multidiscip Optim* 44(3):431–441
- Su R, Wang X, Gui L, Fan Z (2011) Multi-objective topology and sizing optimization of truss structures based on adaptive multi-island search strategy. *Struct Multidiscip Optim* 43(2):275–286
- Tenne YI, Goh CK (2010) *Computational intelligence in expensive optimization problems*, 1st edition. Springer Publishing Company, Berlin
- Wang L, Shan S, Wang GG (2004) Mode-pursuing sampling method for global optimization on expensive black-box functions. *J Eng Optim* 36(4):419–438
- Wang GG, Shan S (2006) Review of metamodeling techniques in support of engineering design optimization. *J Mech Des* 129(4): 370–380
- Wierzbicki AP (1986) On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR Spectrum* 8(2):73–87
- Wilson B, Cappelleri D, Simpson TW, Frecker M (2001) Efficient Pareto frontier exploration using surrogate approximations. *Optim Eng* 2(1):31–50
- Wu J, Azarm S (2000) Metrics for quality assessment of a multiobjective design optimization solution set. *J Mech Des* 123(1):18–25
- Wagner T, Emmerich M, Deutz A, Ponweiser W (2010). In: Schaefer R, Cotta C, Kolodziej J, Rudolph G (eds) *On expected-improvement criteria for model-based multi-objective optimization*. Springer, Berlin
- Yang BS, Yeun YS, Ruy WS (2002) Managing approximation models in multiobjective optimization. *Struct Multidiscip Optim* 24(2):141–156
- Yun Y, Yoon M, Nakayama H (2009) Multi-objective optimization based on meta-modeling by using support vector regression. *Optim Eng* 10(2):167–181
- Zanakis SH, Solomon A, Wishart N, Dubish S (1998) Multi-attribute decision making: a simulation comparison of select methods. *Eur J Oper Res* 107(3):507–529
- Zitzler E, Laumanns M, Thiele L (2001) SPEA2: Improving the strength Pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Department of Electrical Engineering, Swiss Federal Institute of Technology (ETH)
- Zitzler E, Laumanns M, Thiele L (2002) SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: *Proceedings of the conference on evolutionary methods for design optimization and control, CIMNE, Barcelona, Spain*, pp 95–100

- Zitzler E, Thiele L, Laumanns M, Fonseca CM, da Fonseca VG (2003) Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans Evol Comput* 7(2):117–132
- Zitzler E, Knowles J, Thiele L (2008) Quality assessment of pareto set approximations, vol 5252. Springer, Berlin, pp 373–404
- Zhou J, Turng LS (2007) Adaptive multiobjective optimization of process conditions for injection molding using a Gaussian process approach. *Adv Polym Technol* 26(2):71–85
- Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm Evol Comput* 1(1):32–49

**PII**

**AN INTERACTIVE SURROGATE-BASED METHOD FOR  
COMPUTATIONALLY EXPENSIVE MULTIOBJECTIVE  
OPTIMIZATION**

by

M. Tabatabaei, M. Hartikainen, K. Sindhya, J. Hakanen, and K. Miettinen

Submitted



# An Interactive Surrogate-based Method for Computationally Expensive Multiobjective Optimization

Mohammad Tabatabaei\*, Markus Hartikainen†, Karthik Sindhya‡, Jussi Hakanen§,  
Kaisa Miettinen¶

University of Jyväskylä, Department of Mathematical Information Technology, P.O.  
Box 35 (Agora), FI-40014 University of Jyväskylä, Finland

## Abstract

Many disciplines involve computationally expensive multiobjective optimization problems. Surrogate-based methods are commonly used in the literature to alleviate the computational cost. In this paper, we develop an interactive surrogate-based method called SURROGATE-ASF to solve computationally expensive multiobjective optimization problems. This method employs preference information of a decision maker. Numerical results demonstrate that SURROGATE-ASF efficiently provides preferred solutions for a decision maker. It can handle different types of problems involving for example multimodal objective functions and nonconvex and/or disconnected Pareto frontiers.

**Keywords** Multiple Criteria Decision Making (MCDM), interactive method, computational cost, black-box functions, metamodeling techniques, multimodal functions, achievement scalarizing function

## Abbreviations and Notations

<b>MOP</b>	Multiobjective Optimization Problem
<b>DM</b>	Decision Maker
<b>ASF</b>	Achievement Scalarizing Function
<b>RBF</b>	Radial Basis Function
<b>SVR</b>	Support Vector Regression
<b>LOOCV</b>	The leave-one-out cross-validation
<b>MPF</b>	Modified Pareto Fitness
$k$	Number of (computationally expensive) objective functions
$n$	Number of decision variables

---

\*mohammad.tabatabaei@jyu.fi; corresponding author

†markus.hartikainen@jyu.fi

‡karthik.sindhya@jyu.fi

§jussi.hakanen@jyu.fi

¶kaisa.miettinen@jyu.fi

$f_i$	$i^{th}$ computationally expensive objective function
$S$	Feasible decision set
$\mathbf{x}$	Decision (variable) vector
$x_c$	$c^{th}$ decision variable
$x_c^l, x_c^u$	Lower and upper bounds of $x_c$
$\mathbb{R}^k$	Objective space
$\mathbf{f}(\mathbf{x})$	Objective vector
$Z = \mathbf{f}(S)$	Feasible objective set
$\mathbf{z}_e^i$	$i^{th}$ extreme point
$\mathbf{z}^{ideal}$	Ideal (objective) vector
$\mathbf{z}^{utp}$	Utopian (objective) vector
$\mathbf{z}^{nadir}$	Nadir (objective) vector
$\bar{\mathbf{z}}^*$	Reference point given by the decision maker
$H$	Convex hull of all individual extreme points
$\mathbf{z}^{**}$	Projected reference point on $H$ corresponding to $\mathbf{z}^*$
$\bar{Z}^P$	Set of predetermined reference points generated on $H$
$X^P$	Set of reference solutions in the decision space
$F^P$	Set of reference solutions in the objective space
$q$	Number of divisions along each objective coordinate axis
$r$	Number of sub-regions (or hyper-boxes)
$S^a$	$a^{th}$ hyper-box
$\bar{z}^{P,a,i}$	$i^{th}$ predetermined reference point forming the $a^{th}$ sub-region on $H$
$\bar{Z}^a$	Set of reference sample points corresponding to the $a^{th}$ sub-region
$X^{P,a}$	Set of non-dominated solutions in the decision space corresponding to the predetermined reference points in $\bar{Z}^P$
$F^{P,a}$	Set of non-dominated solutions in the objective space corresponding to the predetermined reference points in $\bar{Z}^P$
$\mathbf{x}^{P,a,i}$	$i^{th}$ of the non-dominated solutions defining $S^a$
$\bar{X}^a$	Set of sample points in the decision space corresponding to $S^a$
$\widetilde{\text{ASF}}^a$	Surrogate function corresponding to $S^a$
$T^a$	Cartesian product of sets $\bar{X}^a$ and $\bar{Z}^a$
$\bar{X}P^a$	Set of archived points in the decision space generated during the updating process
$\bar{F}P^a$	Set of archived points in the objective space generated during the updating process
$\bar{\mathbf{x}}^{cand}$	Minimizer of MPF
$\bar{\mathbf{x}}_{close}^{cand}$	Closest point in $\bar{X}^a$ to $\bar{\mathbf{x}}^{cand}$
$\bar{\mathbf{x}}_{close}^{close}$	Closest point in $\bar{X}^a \setminus \{\bar{\mathbf{x}}_{close}^{cand}\}$ to $\bar{\mathbf{x}}_{close}^{cand}$
$d_{close}^{cand}$	Euclidean distance between $\bar{\mathbf{x}}^{cand}$ and $\bar{\mathbf{x}}_{close}^{cand}$

$d_{close}^{close}$	Euclidean distance between $\bar{\mathbf{x}}_{close}^{cand}$ and $\bar{\mathbf{x}}_{close}^{close}$
$\bar{\mathbf{f}}_{close}^{cand}$	Objective vector corresponding to $\bar{\mathbf{x}}_{close}^{cand}$
$\bar{\mathbf{f}}_{artif}^{cand}$	Artificial objective vector

## 1 Introduction

Many multiobjective optimization problems (MOPs), arising e.g. in engineering applications, often involve multiple incommensurable, highly nonlinear, black-box and/or multimodal objective functions. Function evaluations in such problems may be conducted through time-consuming experiments and/or simulators. In the literature, these problems are called computationally expensive multiobjective optimization problems. MOPs typically have several (in many cases infinitely many) optimal solutions known as Pareto optimal solutions. The set of all Pareto optimal solutions in the objective space (called Pareto frontier) can be nonconvex and/or disconnected. From a mathematical point of view and without any preference consideration, Pareto optimal solutions are equally acceptable for an MOP. Therefore, when solving an MOP, a decision maker (DM) is required to provide preference information and to choose his/her preferred solutions.

Multiobjective optimization methods are often categorized according to a decision maker's role in the solution process, i.e., non-interactive and interactive methods [1]. According to [2], in non-interactive methods, the DM either is not involved or provides preference information before or after the actual solution process. In interactive methods, the DM plays an essential role and the intention is to support him/her in the search for the most preferred solution. In such methods, steps of an iterative solution algorithm are repeated and the DM progressively specifies preference information so that the most preferred solution can be found. Examples of types of specifying preference information are reference points and classification of objective functions. What is noticeable is that the DM can determine and alter his/her preferences between each iteration and in the meantime learn about the interdependencies in the problem as well as about one's own preferences. This is a significant advantage of interactive methods in light of the fact that becoming more acquainted with the problem, its possibilities and limitations is often very valuable for the DM. See [2, 3] for details of non-interactive and interactive methods.

Surrogate-based methods are commonly used in the literature to alleviate computational cost [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 15, 16]. The basic idea in such methods is to introduce a computationally less expensive problem called a surrogate problem and to replace the original problem with the surrogate one. In the literature, methods have been developed in which surrogate problems are built independently of the type of the optimization algorithms employed in them. In [17], a non-interactive surrogate-based method is developed. This surrogate-based method does not consider the role of a DM and cannot handle multimodal functions or disconnected Pareto frontiers. The surrogate-based methods proposed in [18, 19] consider the role of a DM in the solution process. However, they are not interactive methods. This means that if a DM wishes to provide new preferences, the entire methods should be run again. Therefore, the DM should wait for a long time to get new solutions corresponding to his/her preferences. In [20, 21, 22] methods have been developed by which the DM can find preferred solutions quickly. These methods require a set of approximated solutions a priori generated by some other surrogate-based methods.

In [23], we surveyed surrogate-based methods, and observed some shortcomings including inability to 1) involve a DM in the solution process, 2) deal with multimodal functions and 3) capture a nonconvex and disconnected Pareto frontier. To overcome these shortcomings, in this

paper, we develop an interactive surrogate-based method. In our method, a DM provides his/her preferences in the form of a reference point containing aspiration levels representing desirable values for objective functions. According to [24], the type of preference information in the form of a reference point has been regarded to be understandable for a DM.

The method proposed is called SURROGATE-ASF and involves two phases, i.e. initialization and decision making phases. In the initialization phase, a set of non-dominated solutions in the decision and objective spaces are generated. Using these solutions, hyper-boxes are formed in the decision space. Corresponding to each hyper-box, a single objective surrogate function is built by approximating an achievement scalarizing function (ASF) [25]. In the literature, this function is used to compute the (weakly) Pareto optimal solution for a given reference point. In the decision making phase, the single objective surrogate functions built in the initialization phase are employed in generating solutions reflecting the preferences of the DM. One should note that the DM is only involved in the latter phase. Based on the comparison in [26], a cubic radial basis function (RBF) with a linear tail is employed as the metamodeling technique in this paper, but SURROGATE-ASF is not limited to RBF.

SURROGATE-ASF has been developed for computationally expensive MOPs with box constraints. The novelty of SURROGATE-ASF can be summarized as follows: 1) the DM does not need to wait for a long time to obtain his/her solutions corresponding to his/her preferences, 2) SURROGATE-ASF utilizes reference points given by a DM as input, 3) the DM can explore different regions, a particular interesting region or the entire Pareto frontier, 4) It provides approximated solutions in both the decision and objective spaces simultaneously.

The rest of this paper is organized as follows. In Section 2, the basic concepts used in this paper are addressed. The SURROGATE-ASF method is presented in Section 3. Numerical results of evaluating the performance of SURROGATE-ASF on a practical shape optimization problem of designing an airfoil as well as some benchmark problems are presented in Section 4. Finally, in Section 5, we draw our conclusions and discuss paths for future research.

## 2 Basic Concepts

In this section, we introduce the concepts and notations used in this paper. We consider multi-objective optimization problems of the form:

$$\underset{\mathbf{x} \in S}{\text{minimize}} \{f_1(\mathbf{x}), \dots, f_k(\mathbf{x})\}, \quad (1)$$

where  $f_i : S \rightarrow \mathbb{R}$  are  $k$  ( $\geq 2$ ) conflicting, computationally expensive objective functions,  $S = \{\mathbf{x} \in \mathbb{R}^n : x_c^l \leq x_c \leq x_c^u, c = 1, \dots, n\}$  is a nonempty *feasible decision* set which is a subset of the *decision space*  $\mathbb{R}^n$ . A solution  $\mathbf{x} = (x_1, \dots, x_n)^T \in S$  is called a *feasible decision (variable) vector*, where  $x_c$ ,  $c = 1, \dots, n$ , are *decision variables* and,  $x_c^l$  and  $x_c^u$  are the lower and upper bounds of  $x_c$ , respectively. The image of  $\mathbf{x}$  in the *objective space*  $\mathbb{R}^k$  is called a *feasible objective vector* denoted by  $\mathbf{f}(\mathbf{x})$ . The image of  $S$  in the objective space is called the *feasible objective* set denoted by  $Z(= \mathbf{f}(S))$ .

A feasible solution  $\mathbf{x}^* \in S$  and the corresponding  $\mathbf{f}(\mathbf{x}^*) \in Z$  are termed *weakly Pareto optimal* for problem (1), if there does not exist another feasible solution  $\mathbf{x} \in S$  such that  $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$  for all  $i = 1, \dots, k$ . Correspondingly, they are *Pareto optimal* for problem (1), if there does not exist another feasible solution  $\mathbf{x} \in S$  such that  $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$  for all  $i = 1, \dots, k$ , and  $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$  for at least one index  $j \in \{1, \dots, k\}$ . The set of all Pareto optimal solutions in the objective space is called a *Pareto frontier*. Let the set  $X^d = \{\mathbf{x}^1, \dots, \mathbf{x}^d\}$  be an arbitrary subset of feasible

solutions in  $S$ , and  $F^d = \{\mathbf{f}(\mathbf{x}^1), \dots, \mathbf{f}(\mathbf{x}^d)\}$ , the corresponding objective vectors in  $Z$ . A solution  $\mathbf{x}^i$  (or  $\mathbf{f}(\mathbf{x}^i)$ ),  $i = 1, \dots, d$ , that satisfies the definition of Pareto optimality with respect to all solutions in  $X^d$  (or  $F^d$ ), is called a *non-dominated solution* in  $X^d$  (or  $F^d$ ). A solution  $\mathbf{x}'$  (or  $\mathbf{f}(\mathbf{x}')$ ) is called a locally non-dominated solution if there exist a non-empty set  $S' \subseteq S$  such that,  $\mathbf{x}'$  (or  $\mathbf{f}(\mathbf{x}')$ ) satisfies the definition of Pareto optimality with respect to all points in  $S'$  (or  $\mathbf{f}(S')$ ). A Pareto optimal solution is a non-dominated solution, but a non-dominated one is not necessarily Pareto optimal.

We also define a feasible solution  $\mathbf{x}_e^i \in \operatorname{argmin}_{\mathbf{x} \in S} \{f_i(\mathbf{x})\}$  for  $i = 1, \dots, k$ . The  $i^{\text{th}}$  *extreme point (solution)* for  $i = 1, \dots, k$ , is defined as  $\mathbf{z}_e^i = \mathbf{f}(\mathbf{x}_e^i)$ . These extreme points in the objective space are also extremes of the Pareto frontier. Based on the extreme points, a vector of the lower bound of the objective function values in the Pareto frontier in the objective space is defined as the *ideal (objective) vector* and denoted by  $\mathbf{z}^{ideal} = (z_1^{ideal}, \dots, z_k^{ideal})^T$  where  $z_i^{ideal} = f_i(\mathbf{x}_e^i)$  for  $i = 1, \dots, k$ . The *utopian (objective) vector*  $\mathbf{z}^{utp}$  is a vector in which its components are calculated by subtracting some small positive scalar (e.g.,  $10^{-6}$ ) from the components of  $\mathbf{z}^{ideal}$ . A vector of upper bounds of the objective function values in the Pareto frontier is defined as the *nadir (objective) vector* and denoted by  $\mathbf{z}^{nadir} = (z_1^{nadir}, \dots, z_k^{nadir})^T$ . The components of the nadir vector can be approximated by e.g., a pay-off table using the extreme points. More information of the ideal, utopian and nadir vectors is given e.g. in [1]. We define the difference operator  $\mathbf{a} - \mathbf{b} = (a_1 - b_1, \dots, a_k - b_k)$ , where  $\mathbf{a} = (a_1, \dots, a_k)$ ,  $\mathbf{b} = (b_1, \dots, b_k) \in \mathbb{R}^k$ .

In SURROGATE-ASF, the DM provides his/her preferences in the form of a *reference point*  $\bar{\mathbf{z}}^* = (\bar{z}_1^*, \dots, \bar{z}_k^*)^T$ , where  $\bar{z}_i^*$  is an *aspiration level* representing a desirable value for the objective function  $f_i$ . One can find a preferred solution for a reference point given by a DM by applying an appropriate scalarization. Scalarizing problem (1) means formulating a single objective optimization problem such that its (globally) optimal solution is a Pareto optimal solution for (1). In this paper, we consider the following widely used *achievement scalarizing function* (ASF) [25] as an important element of SURROGATE-ASF:

$$\begin{aligned} \text{ASF: } S \times \mathbb{R}^k &\rightarrow \mathbb{R} \\ (\mathbf{x}, \bar{\mathbf{z}}^*) &\mapsto \max_{i=1, \dots, k} (w_i(f_i(\mathbf{x}) - \bar{z}_i^*)), \end{aligned} \quad (2)$$

where  $w_i \geq 0$ , for  $i = 1, \dots, k$ , are non-negative fixed weights which actually set a direction where  $\bar{\mathbf{z}}^*$  is projected onto the Pareto frontier. In this paper we set  $w_i = \frac{1}{z_i^{nadir} - z_i^{utp}}$ , for  $i = 1, \dots, k$ , which are widely used [25]. A new (but still computationally expensive) single objective optimization problem is formulated as

$$\underset{\mathbf{x} \in S}{\text{minimize}} \quad \text{ASF}(\mathbf{x}, \bar{\mathbf{z}}^*). \quad (3)$$

The reference point in problem (3) can be feasible or infeasible, i.e., inside or outside of the feasible objective set. As proved in [25], theoretically by solving problem (3), a (weakly) Pareto optimal solution corresponding to the reference point is obtained, regardless of the feasibility or infeasibility of the reference point. This does not hold for all other scalarizing approaches [1]. Different (weakly) Pareto optimal solutions can be obtained by changing the reference point in problem (3). One can add an augmentation term to (2) to avoid weakly Pareto optimal solutions as discussed and proved in [1, 25]. Numerically, in cases with a limited number of function evaluations, the optimal solution of problem (3) is a locally non-dominated solution. In what follows, an optimal solution of problem (3) corresponding to a reference point given by a DM and a predetermined reference point is called a *preferred solution* and a *reference solution*, respectively. Moreover, problem (1) is referred to as the *original problem* and its computationally expensive functions as *original functions*.

### 3 The SURROGATE-ASF Method

In the SURROGATE-ASF method, a DM can provide preferred ranges of the objective functions. Then, they are considered as the utopian and the nadir vectors corresponding to the region which is interesting for the DM in the objective space. Alternatively, estimates of the utopian and nadir vectors can be incorporated into the method. SURROGATE-ASF involves two phases, i.e. initialization and decision making phases. In the initialization phase, a set of non-dominated solutions within the preferred ranges given by the DM is generated. Then, by calling Algorithm 1 to be discussed in the following subsection, a finite number of hyper-boxes is formed using the non-dominated solutions in the decision space. For each individual hyper-box, a single objective surrogate function is built by approximating the ASF (2) where the decision variables of problem (1) and the aspiration levels appearing in the ASF are treated as variables of the surrogate function. In the decision making phase to be discussed in Subsection 3.3, these computationally inexpensive surrogate functions are utilized to interact with the DM.

#### 3.1 Surrogate Building Module

Once the DM provides the preferred ranges, the extreme points of the corresponding region in the objective space are calculated. Then, the corresponding solutions in the decision space are found by solving problem (3) by considering the extreme points as reference points. The approximation of the ASF requires sample points for the aspiration levels and the decision variables. However, before discussing the procedure of selecting sample points, we first look at an interesting property regarding problem (3). This property assists to identify regions in the decision and objective spaces where to select sample points.

Suppose that  $H$  is the convex hull of all individual extreme points. This is constructed by a convex combination of all extreme points in the objective space. Moreover, let  $\bar{\mathbf{z}}^*$  be a reference point given by a DM, and  $\mathbf{x}^*$  be the corresponding optimal solution of problem (3). Thus,  $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x} \in S} [\max_{i=1, \dots, k} (w_i(f_i(\mathbf{x}) - \bar{z}_i^*))]$ . On the other hand,  $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x} \in S} [\max_{i=1, \dots, k} (w_i(f_i(\mathbf{x}) - \bar{z}_i^*)) + t]$ , for all  $t \in \mathbb{R}$ . Therefore, we have:

$$\begin{aligned} \mathbf{x}^* &\in \operatorname{argmin}_{\mathbf{x} \in S} \left[ \max_{i=1, \dots, k} (w_i(f_i(\mathbf{x}) - \bar{z}_i^*)) + t \right] = \operatorname{argmin}_{\mathbf{x} \in S} \left[ \max_{i=1, \dots, k} (w_i(f_i(\mathbf{x}) - \bar{z}_i^*) + t) \right] \\ &= \operatorname{argmin}_{\mathbf{x} \in S} \left[ \max_{i=1, \dots, k} (w_i(f_i(\mathbf{x}) - \bar{z}_i^* + \frac{t}{w_i})) \right] = \operatorname{argmin}_{\mathbf{x} \in S} \left[ \max_{i=1, \dots, k} (w_i(f_i(\mathbf{x}) - (\bar{z}_i^* - \frac{t}{w_i}))) \right] \\ &= \operatorname{argmin}_{\mathbf{x} \in S} \left[ \max_{i=1, \dots, k} (w_i(f_i(\mathbf{x}) - \bar{z}_i^{**})) \right], \end{aligned}$$

where  $\bar{z}_i^{**}$  is treated as an arbitrary reference point on the ray passing through  $\bar{z}_i^*$  in parallel with  $\mathbf{w} = (w_1, \dots, w_k)^T$ . It means that the optimal solution of problem (3) for a reference point  $\bar{\mathbf{z}}^*$  given by a DM is also the optimal solution of problem (3) for any reference point on the ray passing through  $\bar{\mathbf{z}}^*$  in parallel with  $\mathbf{w}$  (see Figure 1). The following problem gives the projected reference point  $\bar{\mathbf{z}}^{**}$  in  $H$ :

$$\bar{\mathbf{z}}^{**} \in \operatorname{argmin}_{\mathbf{h} \in H} \left\| \frac{\mathbf{h} - \bar{\mathbf{z}}^*}{\mathbf{w}} \right\| \quad (4)$$

where  $\|\cdot\|$  is the Euclidean norm. Problem (4) applies if and only if there exists a real number  $t \in \mathbb{R}$  such that  $\bar{\mathbf{z}}^* = \mathbf{h} + t\mathbf{w}$  for some  $\mathbf{h} \in H$ .

Since any reference point within the preferred ranges of objective functions given by the DM can be projected onto the convex hull  $H$  (i.e., finding the closest point on  $H$  along the direction

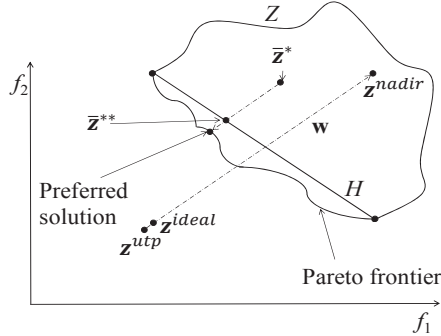


Figure 1: Projection of a given reference point onto the convex hull

$w$ ), sample points for the aspiration levels as reference sample points are selected on this convex hull. To do this, first, a set of evenly distributed predetermined reference points including the extreme points is generated on the convex hull  $H$  and denoted by  $\bar{Z}^P$ . The reference solutions (non-dominated solutions) in the decision space corresponding to these reference points are obtained by solving problem (3) for each individual predetermined reference point. This is conducted by employing a surrogate-based single objective optimization method. The choice of this method has an impact on the performance of SURROGATE-ASF. One can use an appropriate state-of-the-art method to solve problem (3) efficiently. These reference solutions are evaluated with the original functions.

In order to generate the predetermined reference points (without involving any DM), we use the method presented in [27] that places points on a convex hull of all individual extreme points (a  $(k - 1)$ - dimensional simplex). If  $h_z$  predetermined reference points (including the extreme points in the objective space) are considered, the  $q$  divisions along each objective coordinate axis in the objective space of a  $k$ -objective problem can be calculated using:

$$h_z = \binom{k + q - 1}{q}. \quad (5)$$

Figures 2 (a) and 2 (b) depict two convex hulls for bi- and three-objective problems with  $q = 4$  and  $h_z = \binom{2 + 4 - 1}{4} = 5$  and  $h_z = \binom{3 + 4 - 1}{4} = 15$  predetermined reference points (black circles), respectively.

The sets of reference solutions in the decision and the objective spaces are denoted by  $X^P$  and  $F^P$ , respectively. By making use of the set of predetermined reference points  $\bar{Z}^P$  and reference solutions  $X^P$ , the convex hull  $H$  and the decision space are decomposed into a finite number of sub-regions and hyper-boxes, respectively. In what follows,  $a = 1, \dots, r$ , represents the index corresponding to the  $a^{th}$  hyper-box or sub-region where  $r$  is the number of hyper-boxes (sub-regions).<sup>1</sup>

Algorithm 1 starts by constructing sub-regions on  $H$  using the predetermined reference points in  $\bar{Z}^P$ . Each sub-region is formed by selecting the  $k$  nearest neighbor points on  $H$  starting from one of the extreme points. Figures 2 (a) and 2 (b) represent one possible way of forming the sub-regions and numbering them where  $k = 2$  and  $k = 3$ , respectively. Then, reference sample points (the stars in Figures 2 (a) and 2 (b)) are selected within these sub-regions. To select reference sample points, the predetermined reference points corresponding to the  $a^{th}$  sub-region

<sup>1</sup>For  $k = 2$  and  $3$ ,  $r = q$  and  $q^2$ , respectively

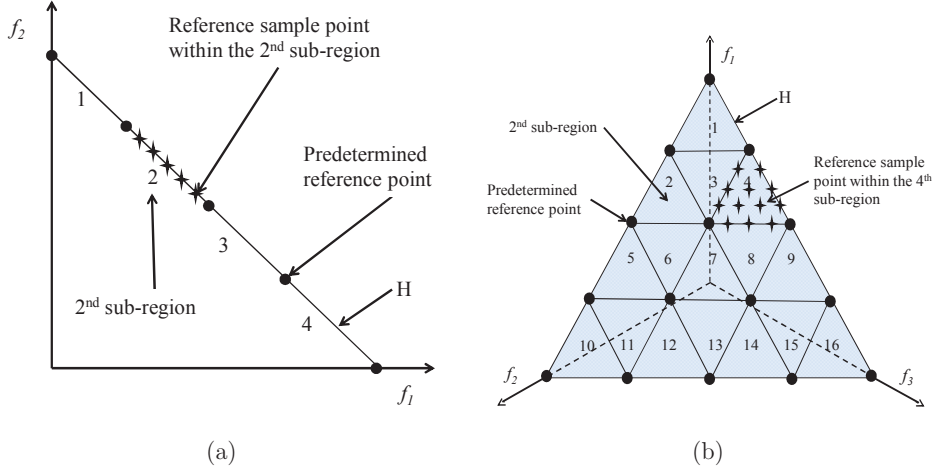


Figure 2: (a): A convex hull with 5 points for a bi-objective problem with  $q = 4$  divisions. (b): A convex hull with 15 points for a three-objective problem with  $q = 4$  divisions.

(i.e.,  $\bar{Z}^{P,a} = \{\bar{\mathbf{z}}^{P,a,1}, \dots, \bar{\mathbf{z}}^{P,a,k}\}$ ) are considered as the extreme points of this sub-region. Then, by using the method presented in [27], a set of evenly distributed reference sample points within this sub-region on the convex hull is generated. The larger the number of generated reference sample points, the higher is the accuracy of  $\widetilde{\text{ASF}}^a$ . These reference sample points along with the reference points in  $\bar{Z}^{P,a}$  are considered as sample points corresponding to this sub-region and denoted by  $\bar{Z}^a$ .

To decompose the decision space, hyper-boxes are built corresponding to sub-regions on the convex hull  $H$ . Considering the  $a^{\text{th}}$  sub-region on  $H$  and the set of predetermined reference points forming this sub-region, i.e.,  $\bar{Z}^{P,a} = \{\bar{\mathbf{z}}^{P,a,1}, \dots, \bar{\mathbf{z}}^{P,a,k}\}$ , the set of non-dominated solutions corresponding to these predetermined reference points in the decision and the objective spaces are denoted by  $X^{P,a} = \{\mathbf{x}^{P,a,1}, \dots, \mathbf{x}^{P,a,k}\}$  and  $F^{P,a} = \{\mathbf{f}(\mathbf{x}^{P,a,1}), \dots, \mathbf{f}(\mathbf{x}^{P,a,k})\}$ , respectively. Then, the lower and the upper bounds of the decision variable  $x_c$ , for  $c = 1, \dots, n$ , within the  $a^{\text{th}}$  hyper-box denoted by  $S^a$  corresponding to this sub-region are calculated as follows:

$$\min\{x_c^{P,a,1}, \dots, x_c^{P,a,k}\} \leq x_c \leq \max\{x_c^{P,a,1}, \dots, x_c^{P,a,k}\}, \forall c = 1, \dots, n. \quad (6)$$

Figure 3 shows a simple example of hyper-boxes for a problem with two decision variables. Once  $S^a$  is formed, a small number of initial sample points within this hyper-box is selected using some sampling technique such as Latin hypercube sampling (LHS) [28] and evaluated with the original functions. These evaluated points along with the non-dominated solutions in  $X^{P,a}$  are considered as the initial sample points in the decision space corresponding to  $S^a$  and denoted by  $\bar{X}^a$ .

To build the surrogate function corresponding to the hyper-box  $S^a$  denoted by  $\widetilde{\text{ASF}}^a$ , a cubic RBF with a linear tail is employed to approximate ASF (its description is given e.g. in [29]). To do this, the following Cartesian product of sets  $\bar{X}^a$  and  $\bar{Z}^a$  is formed as input data denoted by  $T^a$  for RBF:

$$T^a = \bar{X}^a \times \bar{Z}^a = \{(\bar{\mathbf{x}}, \bar{\mathbf{z}}) \mid \bar{\mathbf{x}} \in \bar{X}^a \text{ and } \bar{\mathbf{z}} \in \bar{Z}^a\}. \quad (7)$$



---

**Algorithm 1: Surrogate Building Module**

---

**Input:**  $X^P$ ,  $F^P$  and  $\bar{Z}^P$ **Output:** The hyper-boxes, the sub-regions and corresponding surrogate functions

- 1: Define sub-regions on the convex hull  $H$ .
  - 2: **For** the  $a^{th}$  sub-region ( $a = 1, \dots, r$ ), on the convex hull  $H$
  - 3: Select a set of reference sample points within the  $a^{th}$  sub-region on the convex hull  $H$  by the method given in [27]. Denote this set by  $\bar{Z}^a$ .
  - 4: Build the  $a^{th}$  hyper-box  $S^a$  in the decision space according to (6).
  - 5: Select the initial set of sample points within  $S^a$  and *evaluate them with the original functions*.
  - 6: Define  $\bar{X}^a$  and  $\bar{F}^a$  as the sets of sample points in the decision and the objective spaces, respectively.
  - 7: Form the set  $T^a$  according to (7).
  - 8: Calculate the ASF values of all elements in the set  $T^a$ .
  - 9: Using the elements in the set  $T^a$  and LOOCV (or a  $k$ -fold cross-validation method), build an initial  $\widetilde{\text{ASF}}^a$  by RBF.
  - 10: Check the prediction accuracy of  $\widetilde{\text{ASF}}^a$ .
  - 11: **if** the accuracy is acceptable
  - 12:     Consider  $\widetilde{\text{ASF}}^a$  as the surrogate function corresponding to  $S^a$ .
  - 13: **else**
  - 14:     Call Algorithm 2 with inputs  $\bar{X}^a, \bar{F}^a, \bar{Z}^P, \widetilde{\text{ASF}}^a$  and  $T^a$ .
  - 15: **end if**
  - 16: **end for**
  - 17: Return the hyper-boxes, the sub-regions and corresponding surrogate functions.
- 

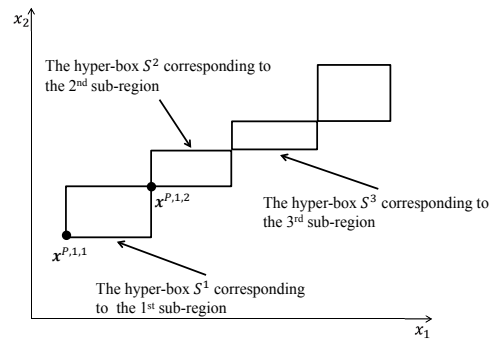


Figure 3: Hyper-boxes in the decision space corresponding to the sub-regions on the convex hull in Figure 2 (a). For example, reference solutions  $\mathbf{x}^{P,1,1}$  and  $\mathbf{x}^{P,1,2}$  form  $S^1$ .

The ASF values of all elements in  $T^a$  are calculated. One should note that the objective function values of all sample points in  $\bar{X}^a$  are available. Then,  $T^a$  is divided into validation and training sets using the leave-one-out cross-validation (LOOCV) or the  $k$ -fold cross-validation method discussed in [26] (note that it differs from the index  $k$  for the  $k^{th}$  objective function.). Let  $|T^a|$  be the cardinality of  $T^a$ . According to [26], if  $|T^a| \leq 50$ , we use LOOCV. If  $|T^a| > 50$ , then we use the  $k$ -fold cross-validation method where  $k = 10$ , for  $50 < |T^a| \leq 100$ ,  $k = 20$ , for  $100 < |T^a| \leq 150$  and  $k = 30$ , for  $150 < |T^a| \leq 200$ . Having the elements in the training set and their corresponding ASF values as input data, RBF is employed to build  $\widetilde{ASF}^a$ . When using either LOOCV or the  $k$ -fold cross-validation method, the prediction accuracy of  $\widetilde{ASF}^a$  is evaluated using standard error measures like Root Mean Squared Error (RMSE) [30] and/or  $R^2$  [31].

The steps of the decomposition procedure of the decision space into a finite number of hyper-boxes and building corresponding surrogate functions are shown in Algorithm 1. Note that the prediction accuracy of  $\widetilde{ASF}^a$  may not be satisfactory after steps 1-10 of Algorithm 1 and, in that case, it needs to be improved. In the following subsection, we discuss an update strategy called Algorithm 2 by which new sample points within the hyper-box  $S^a$  can be selected. The process of updating  $\widetilde{ASF}^a$  is repeated until a desired accuracy is achieved.

The non-dominated solutions used to form hyper-boxes correspond to the predetermined reference points on the convex hull. It may happen that at least two predetermined reference points have the same corresponding non-dominated solution. In such cases, to form sub-regions and hyper-boxes, out of all predetermined reference points with the same non-dominated solution (i.e., the same objective values but different decision variable values), only one reference point (selected arbitrary) and its corresponding non-dominated solution are considered along with other different predetermined reference points and non-dominated solutions. Then, sub-regions and hyper-boxes are formed. It may also occur that hyper-boxes overlap each other. In these cases, some sample points selected fall into more than one hyper-box. However, sub-regions and hyper-boxes are still formed as mentioned in Algorithm 1.

In practice, obtaining a desired accuracy may not be possible. In such a case, depending on the computational budget, one can consider a maximum number of function evaluations for each individual hyper-box. Then, either achieving a desired accuracy or reaching the maximum number of function evaluations can be considered as a stopping criterion.

### 3.2 Accuracy Improvement Module

To improve the accuracy prediction of  $\widetilde{ASF}^a$  corresponding to  $S^a$ , we discuss Algorithm 2. When updating  $\widetilde{ASF}^a$ , two matters are considered, i.e., sampling non-dominated solutions within  $S^a$  and having a good diversity among the sample points in  $\bar{X}^a$ .

In order to sample non-dominated solutions, we employ the sampling function discussed in [19] to generate new sample points within  $S^a$ . The authors of [19] modify the Pareto fitness function [32] and introduce a sampling function by using RBF such that by minimizing this function, the optimal solution can be a non-dominated solution. See [19] for details of building this sampling function. For the sake of simplicity, we refer to this sampling function as the modified Pareto fitness function (MPF).

By minimizing the MPF iteratively, non-dominated solutions within  $S^a$  can be generated. Nevertheless, the diversity of the points in  $\bar{X}^a$  to cover the hyper-box should be considered. Thus, in some iteration of updating  $\widetilde{ASF}^a$ , some solution obtained by minimizing the MPF is not evaluated with the original functions. This requires a criterion to assess whether the optimal solution of the MPF should be evaluated with the original functions and added into  $\bar{X}^a$  or not. We discuss this criterion here.

---

**Algorithm 2: Accuracy Improvement Module**


---

**Input:**  $\bar{X}^a, \bar{F}^a, \bar{Z}^P, \widetilde{\text{ASF}}^a, T^a$   
**Output:** An accurate enough  $\widetilde{\text{ASF}}^a$

- 1: Define  $\overline{XP}^a = \bar{X}^a$  and  $\overline{FP}^a = \bar{F}^a$
  - 2: **While**  $\widetilde{\text{ASF}}^a$  is not accurate
  - 3:   Calculate the MPF values of all points in  $\overline{XP}^a$  as discussed in [19].
  - 4:   Build the MPF by RBF.
  - 5:   Minimize the MPF to obtain the optimal solution  $\bar{x}^{cand}$ .
  - 6:   Find  $\bar{x}_{close}^{cand} \in \bar{X}^a$  and  $\bar{f}_{close}^{cand} \in \bar{F}^a$ .
  - 7:   Denote the Euclidean distance between  $\bar{x}_{close}^{cand}$  and  $\bar{x}^{cand}$  by  $d_{close}^{cand}$ .
  - 8:   Find  $\bar{x}_{close}^{close} \in \bar{X}^a \setminus \{\bar{x}_{close}^{cand}\}$ .
  - 9:   Denote the Euclidean distance between  $\bar{x}_{close}^{close}$  and  $\bar{x}_{close}^{cand}$  by  $d_{close}^{close}$ .
  - 10:   **if**  $\frac{d_{close}^{close}}{2} < d_{close}^{cand}$
  - 11:     *Evaluate  $\bar{x}^{cand}$  with the original functions and calculate  $\bar{f}^{cand}$ .*
  - 12:     Form the test set by using  $\bar{x}^{cand}$  and all points in  $\bar{Z}^P$ .
  - 13:     Evaluate the prediction accuracy of all elements in the test set with  $\widetilde{\text{ASF}}^a$  and ASF.
  - 14:     **If** the prediction accuracy is acceptable
  - 15:       Set  $\bar{X}^a = \bar{X}^a \cup \{\bar{x}^{cand}\}$  and  $\bar{F}^a = \bar{F}^a \cup \{\bar{f}^{cand}\}$ .
  - 16:       Update  $T^a$ .
  - 17:       Update  $\widetilde{\text{ASF}}^a$ .
  - 18:       Stop the updating process.
  - 19:     **else**
  - 20:       Set  $\overline{XP}^a = \overline{XP}^a \cup \{\bar{x}^{cand}\}$ ,  $\bar{X}^a = \bar{X}^a \cup \{\bar{x}^{cand}\}$ ,  $\overline{FP}^a = \overline{FP}^a \cup \{\bar{f}^{cand}\}$   
       and  $\bar{F}^a = \bar{F}^a \cup \{\bar{f}^{cand}\}$ .
  - 21:     **end if**
  - 22:     **else**
  - 23:       Consider the artificial objective vector  $\bar{f}_{artif}^{cand}$  by adding a random  
       vector with positive components to  $\bar{f}_{close}^{cand}$ .
  - 24:        $\overline{XP}^a = \overline{XP}^a \cup \{\bar{x}^{cand}\}$  and  $\overline{FP}^a = \overline{FP}^a \cup \{\bar{f}_{artif}^{cand}\}$ .
  - 25:     **end if**
  - 26: **end while**
-

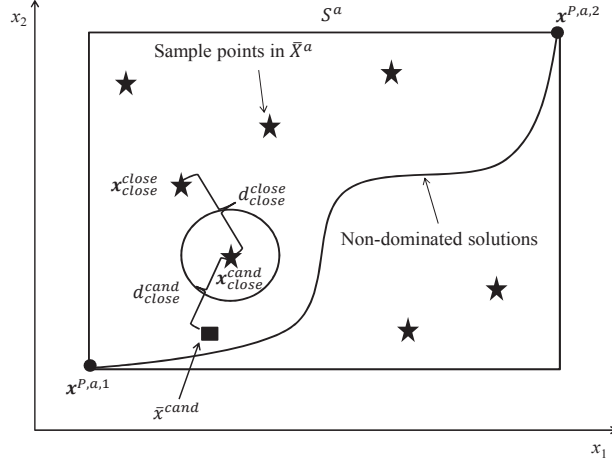


Figure 4: Selecting sample points within  $S^a$  to update  $\widetilde{ASF}^a$

Since some points are not evaluated with the original functions, we introduce  $\overline{XP}^a$  and  $\overline{FP}^a$  as sets of archived points generated during the updating process in the decision and the objective spaces, respectively. These sets include all points in  $\overline{X}^a$  and  $\overline{F}^a$  (the points evaluated with the original functions) and sample points which are not evaluated with the original functions. In what follows, a procedure is discussed to assign objective function values to the points not evaluated with the original functions.

Building the MPF is based on the points in  $\overline{XP}^a$  and  $\overline{FP}^a$ . To do this, all points in  $\overline{XP}^a$  and their corresponding MPF values are considered as input data to train a cubic RBF with a linear tail. Once the MPF is built, it is minimized and the optimal solution  $\overline{x}^{cand}$  within  $S^a$  is obtained. Then, as depicted in Figure 4, the closest point in  $\overline{X}^a$  in terms of the Euclidean distance to  $\overline{x}^{cand}$  is found. This point is denoted by  $\overline{x}_{close}^{cand}$ ; the corresponding objective vector by  $\overline{f}_{close}^{cand}$  and the Euclidean distance between  $\overline{x}^{cand}$  and  $\overline{x}_{close}^{cand}$  by  $d_{close}^{cand}$ . The closest point in  $\overline{X}^a \setminus \{\overline{x}_{close}^{cand}\}$  to  $\overline{x}_{close}^{cand}$  is also found. We denote this point by  $\overline{x}_{close}^{close}$  and the Euclidean distance between  $\overline{x}_{close}^{close}$  and  $\overline{x}_{close}^{cand}$  by  $d_{close}^{close}$ .

To have diversity among sample points within  $\overline{X}^a$ , the criterion  $d_{close}^{cand} > \frac{d_{close}^{close}}{2}$  is checked. This means that if  $\overline{x}^{cand}$  is outside the circle centered on  $\overline{x}_{close}^{cand}$  with radius  $\frac{d_{close}^{close}}{2}$  (see Figure 4), then  $\overline{x}^{cand}$  is evaluated with the original functions and the objective vector  $\overline{f}^{cand}$  is calculated. Before adding  $\overline{x}^{cand}$  and  $\overline{f}^{cand}$  into  $\overline{X}^a$  and  $\overline{F}^a$ , respectively, the prediction accuracy of  $\widetilde{ASF}^a$  is assessed. To accomplish this, a test set as defined in  $T^a$  (7) is formed with  $\overline{x}^{cand}$  and all reference sample points in  $\overline{Z}^a$  are utilized to form the Cartesian product set. All elements in the test set are evaluated with  $\widetilde{ASF}^a$  and ASF, and the prediction accuracy of  $\widetilde{ASF}^a$  is checked. If the accuracy is not acceptable,  $\overline{x}^{cand}$  is added into  $\overline{X}^a$  and  $\overline{XP}^a$  and  $\overline{f}^{cand}$  into  $\overline{F}^a$  and  $\overline{FP}^a$ , respectively. Then, the points in  $\overline{X}^a$  are utilized to update  $\widetilde{ASF}^a$  and the points in  $\overline{XP}^a$  to update the MPF. The process of minimizing the MPF and checking the diversity criterion is repeated. If the prediction accuracy is acceptable, then the process of updating  $\widetilde{ASF}^a$  is terminated.

If  $d_{close}^{cand} \leq \frac{d_{close}^{close}}{2}$ ,  $\overline{x}^{cand}$  is inside the circle centered on  $\overline{x}_{close}^{cand}$  with radius  $\frac{d_{close}^{close}}{2}$ , then  $\overline{x}^{cand}$  is

not evaluated with the original functions and is not added into  $\overline{X}^a$ . However, to generate new sample points by the MPF, the objective vector corresponding to  $\overline{x}^{cand}$  is required. In this case, to avoid generating  $\overline{x}^{cand}$  again, the objective vector corresponding to this point is artificially treated as a dominated point in the objective space by the objective vector  $\overline{f}_{close}^{cand}$ . To do this, an artificial objective vector denoted by  $\overline{f}_{artif}^{cand}$  is considered for this point by adding a random vector with positive components to the objective vector  $\overline{f}_{close}^{cand}$ . This random vector is a fraction of  $\mathbf{z}^{nadir} - \overline{f}_{close}^{cand}$ , e.g.,  $\frac{\mathbf{z}^{nadir} - \overline{f}_{close}^{cand}}{10}$ . By this choice,  $\overline{f}_{close}^{cand}$  is non-dominated with respect to  $\overline{f}_{artif}^{cand}$ , and is inside the feasible objective space. Then,  $\overline{x}^{cand}$  and  $\overline{f}_{artif}^{cand}$  are added into  $\overline{XP}^a$  and  $\overline{FP}^a$ . Note that these points are not added into  $\overline{X}^a$  and  $\overline{F}^a$ . Then, the MPF values of all points in  $\overline{XP}^a$  are calculated and the MPF is updated. The updating process is repeated until the stopping criterion regarding the accuracy of  $\widehat{ASF}^a$  is met.

### 3.3 SURROGATE-ASF and Decision Making

In this subsection, we present both the decision making phase as well as the algorithm of SURROGATE-ASF. As discussed in Subsection 3.1, in the initialization phase, surrogate functions are built by calling Algorithm 1. These surrogate functions are employed to interact with the DM.

---

**Algorithm 3: Decision making using the surrogate problem (8)**

---

**Input:** The preferred ranges of the objectives or the utopian and nadir objective vectors.

**Output:** The most preferred solution in the decision and the objective spaces.

- 1: **Initialization phase**
  - 2: Form the set of predetermined reference points  $\overline{Z}^P$ .
  - 3: Employ a surrogate-based single objective optimization method to solve problem (3) with respect to all points in  $\overline{Z}^P$  and obtain the set of reference solutions  $X^P$  and  $F^P$  in the decision and the objective spaces, respectively.
  - 4: Form hyper-boxes and build corresponding surrogate functions by calling Algorithm 1 with inputs  $X^P$ ,  $F^P$  and  $\overline{Z}^P$ .
  - 5: **Decision making phase**
  - 6: **While** the decision maker is not satisfied
  - 7:     Ask the decision maker to provide a reference point  $\overline{\mathbf{z}}^*$ .
  - 8:     Determine  $\overline{\mathbf{z}}^{**}$  on the convex hull  $H$  according to (4).
  - 9:     Identify  $\tilde{a}$  as the sub-region number of  $\overline{\mathbf{z}}^{**}$ .
  - 10:     Identify the  $\tilde{a}^{th}$  hyper-box and denote it by  $S^{\tilde{a}}$ .
  - 11:     Utilize  $\overline{\mathbf{z}}^{**}$  as a reference point, formulate and solve the surrogate problem (8).
  - 12:     *Evaluate the optimal solution of the surrogate problem with the original functions.*
  - 13:     Show the evaluated solution to the decision maker.
  - 14: **end while**
  - 15: Return the most preferred solution for the decision maker.
-

In the decision making phase, if the DM did not provide any preferred ranges, the utopian and the nadir vectors are shown to him/her. The DM is supposed to specify reference points within the ranges. In any case, the DM provides a reference point  $\bar{\mathbf{z}}^*$ . The interaction with the DM in SURROGATE-ASF corresponds to that of the reference point method given in [33]. The reference point  $\bar{\mathbf{z}}^*$  is projected onto the convex hull  $H$  by solving problem (4) as discussed in Subsection 3.1. In practice, to find the projected reference point denoted by  $\bar{\mathbf{z}}^{**}$ , we do not solve problem (4). A simple approach is to generate a large number of uniformly distributed points on  $H$ . Then, the point that minimize the objective function of problem (4) is selected. Then, the sub-region and the hyper-box number corresponding to this projected reference point are identified and denoted by  $\tilde{a}$  and the following surrogate problem is solved by using any appropriate single objective optimization method:

$$\underset{\mathbf{x} \in S^{\tilde{a}}}{\text{minimize}} \quad \widetilde{\text{ASF}}^{\tilde{a}}(\mathbf{x}, \bar{\mathbf{z}}^{**}), \quad (8)$$

where  $S^{\tilde{a}}$  is the  $\tilde{a}^{th}$  hyper-box. The optimal solution obtained is an approximation of the preferred solution in the decision space corresponding to the reference point  $\bar{\mathbf{z}}^*$  given by the DM. This solution is evaluated with the original functions and shown to the decision maker. The process of asking for a reference point from the DM, projecting it onto the convex hull and solving the corresponding surrogate problem is repeated until the most preferred solution satisfying the DM is obtained. An overview of the algorithm of SURROGATE-ASF including the initialization and the decision making phases is given in Algorithm 3.

## 4 Numerical results

In this section, we demonstrate the performance of SURROGATE-ASF through solving a shape optimization problem of designing an airfoil. Since this problem contains objective functions with a practical meaning, it is easier to describe the decision making process. Therefore, we discuss the decision making phase comprehensively for this problem. In addition to this practical problem, we consider academic benchmark problems attributing different characteristics such as non-convex Pareto frontiers, connected or disconnected sets of Pareto optimal solutions and multimodal functions. Results involving the benchmark problems are given in Subsection 4.2. The decision making phase for these academic benchmark problems is similar to the one in the shape optimization problem.

We compare the performance of SURROGATE-ASF with the method [18] (henceforth called here the Yun method) and the PAINT method [20]. According to the findings presented in [23], the Yun method and PAINT are the most similar methods to SURROGATE-ASF because in all of them the DM provides preference information in the form of a reference point and, thus, these three methods are comparable.

The Yun method is non-interactive in which  $\mu - \nu$ -SVR [18] is applied as the metamodeling technique. Because it is not interactive by nature, for each individual reference point given by the DM, the entire method must be run for enabling interaction with the DM. In this method, first each objective function is approximated and a surrogate multiobjective optimization problem is formulated. With the objective functions in the surrogate problem, problem (3) as a single objective computationally inexpensive problem is formulated to be solved. The solution of this problem as well as some sample points, selected based on the Lagrangian coefficients in  $\mu - \nu$ -SVR, are utilized to improve the accuracy of the surrogate problem. By comparing the performance of the Yun method and SURROGATE-ASF, we show the advantages of the approach of approximating the achievement scalarizing function (2) as a computationally expensive objective function and then converting it to a computationally inexpensive one over the approach

of first approximating each expensive function individually and then forming the achievement scalarizing function.

Similar to SURROGATE-ASF, the PAIN method requires a set of non-dominated solutions being available to build a surrogate problem. In PAIN, a linear mixed integer multiobjective optimization problem is introduced as a surrogate of the original problem. Then, this problem is scalarized using the achievement scalarizing function including a reference point given by the DM. This scalarized problem is solved, and an approximated solution in the objective space is obtained. PAIN can only provide solutions in the objective space, i.e., decision variable values of the approximated solutions cannot be obtained. The approximated solution is shown to the DM. If the DM is not satisfied, (s)he provides another reference point. The interaction with the DM is conducted until the approximated solution obtained is acceptable for the DM. Then, this solution is considered as a reference point. Using this reference point and the original, computationally expensive functions, problem (3) is formulated as a computationally expensive single objective optimization problem. This problem is solved by a (surrogate-based) single objective optimization method. The optimal solution is the most preferred (approximated) solution and provided to the DM. One should note that, for any approximated solution generated in the intermediate iterations, it is possible to obtain the corresponding decision variable values by solving an expensive single objective optimization problem. In this paper, we follow the steps given in [20] and do not calculate approximated solutions in the decision space until at the last iteration.

At the end of the solution process by an interactive method, the most preferred solution (which is usually a single solution) is obtained. In this aspect, to the best of our knowledge, there is no established quality measure in the literature which is suitable for a reference point based interactive method. Therefore, we compare the closeness of approximated solutions in the objective space obtained by SURROGATE-ASF, the Yun method and PAIN with their corresponding solutions obtained by solving problem (3) in which the original functions are used. To do so, problem (3) (for the academic benchmark problems) is solved including reference points given by the DM. In what follows, we refer to these solutions as original solutions. The closeness of the approximated solution  $\tilde{\mathbf{f}}^{j*}$  and the original solution  $\mathbf{f}^{j*}$  for  $j = 1, \dots, l$ , (where  $l$  is the number of reference points given by the DM) in the objective space is assessed based on the Euclidean distance. To do this, first normalized approximated solutions  $\tilde{\mathbf{f}}_{norm}^{j*} = \frac{\tilde{\mathbf{f}}^{j*} - \mathbf{z}^{ideal}}{\mathbf{z}^{nadir} - \mathbf{z}^{ideal}}$  and normalized original solutions  $\mathbf{f}_{norm}^{j*} = \frac{\mathbf{f}^{j*} - \mathbf{z}^{ideal}}{\mathbf{z}^{nadir} - \mathbf{z}^{ideal}}$  are calculated for  $j = 1, \dots, l$ . Then, the Euclidean distance between the normalized approximated and original solutions in the objective space is calculated as follows:

$$E_j = \|\tilde{\mathbf{f}}_{norm}^{j*} - \mathbf{f}_{norm}^{j*}\| \quad (9)$$

where  $\|\cdot\|$  is the Euclidean norm.

#### 4.1 Solving a shape optimization problem

In this subsection, we demonstrate the performance of SURROGATE-ASF by solving a bi-objective airfoil shape optimization problem

$$\underset{\mathbf{x} \in S}{\text{minimize}} \left\{ \frac{C_d(\mathbf{x})}{C_l(\mathbf{x})}, C_m^2(\mathbf{x}) \right\}, \quad (10)$$

where  $C_d$ ,  $C_l$  and  $C_m$  are the drag, lift and pitching moment coefficients, respectively, and  $S = \{\mathbf{x} \in \mathbb{R}^{12} : 0.0085 \leq x_1 \leq 0.0126, 0.0020 \leq x_2 \leq 0.0040, 7.0000 \leq x_3 \leq 10.0000, 10.0000 \leq x_4 \leq 14.0000, -0.0060 \leq x_5 \leq -0.0030, 0.0025 \leq x_6 \leq 0.0050, 0.4100 \leq x_7 \leq 0.4600, 0.1100 \leq x_8 \leq 0.1300, -0.9000 \leq x_9 \leq -0.7000, 0.2000 \leq x_{10} \leq 0.2600, -0.0230 \leq x_{11} \leq -0.0150, 0.0500 \leq x_{12} \leq 0.2000\}$  is the feasible decision set. The angle of attack, Reynolds and Mach numbers were  $4.0^\circ$ ,  $2.0 \times 10^6$  and 0.1, respectively. The CFD solver adopted in this problem was XFOIL [34]. Problem (10) termed as *ASO – MOP2* has been investigated with a non-interactive method in [35] with a budget of 2000 function evaluations. In our paper, we also used the same budget to solve the problem. In [35], the description of the decision variables and the geometry of the corresponding airfoil (see Figure 3 in [35]) is provided. In this problem, the DM was interested in exploring the objective space for ranges  $0.0043 \leq f_1 \leq 0.0064$  and  $0.0076 \leq f_2 \leq 0.034$ .

As said, the surrogate-based single objective optimization method to solve problem (3) as a computationally expensive one has an impact on the performance of SURROGATE-ASF in terms of reducing the computational cost. In [26], a toolbox implemented in MATLAB called MATSuMoTo (can be downloaded from <https://courses.cit.cornell.edu/jmueler/>) has been developed to solve computationally expensive global optimization problems. We applied MATSuMoTo to solve problem (3).

In the initialization phase of SURROGATE-ASF, according to equation (5),  $h_z = 4$  ( $q = 3$ ) predetermined reference points for the bi-objective problem were considered. For each predetermined reference point, 350 function evaluations were used to find the corresponding reference solutions by solving problem (3). In Algorithm 1, the criteria to check the accuracy of  $\widehat{\text{ASF}}^a$  corresponding to the  $a^{\text{th}}$  hyper-box (sub-region) for  $a = 1, \dots, r$ , were  $R^2 = 0.95$  and  $RMSE = 0.005$  as they are commonly used in the literature. Since the accuracy of the initial surrogate functions was not satisfactory with these choices, we applied Algorithm 2 to update the surrogate functions with respect to a maximum number of 100 function evaluations within each hyper-box. Tables A.1 and A.2 in Appendix A summarize the parameter settings of SURROGATE-ASF and MATSuMoTo, respectively. In Table A.1, the number of reference sample points (i.e.,  $h_z$ ) within each sub-region was chosen heuristically according to (5). In the initialization phase, we set 1400 function evaluations for MATSuMoTo to obtain the reference solutions  $X^P$  and  $F^P$  corresponding to the points in  $\overline{Z}^P$ . The number of function evaluations needed by SURROGATE-ASF to build all surrogate functions was 284. Therefore, the total number of function evaluations in the initialization phase of SURROGATE-ASF using MATSuMoTo to build all surrogate functions was 1684.

In the decision making phase, reference points given by the DM were projected onto the convex hull  $H$  according to (4). Then, the corresponding surrogate problem (8) was solved including the projected reference points. In this problem, while the computationally inexpensive surrogate problem (8) can be solved by any appropriate method, we employed a single objective genetic algorithm with the parameter settings given in [36]. As shown in Table 1, in the first iteration, the DM provided the first reference point  $\bar{\mathbf{z}}^{*1}$  within his preferred ranges. The corresponding surrogate problem was solved and the preferred solution was obtained. This solution was evaluated with the original functions and the corresponding objective vector  $\mathbf{f}^1$  was shown to the DM. For  $\mathbf{f}^1$ , the DM was interested in improving the ratio of drag to lift coefficient while keeping the pitching moment coefficient the same. Then, the next reference point  $\bar{\mathbf{z}}^{*2}$  was given by the DM and the preferred solution  $\mathbf{f}^2$  was presented to him. Comparing the ratio of drag to lift coefficient in  $\bar{\mathbf{z}}^{*2}$ ,  $\mathbf{f}^1$  and  $\mathbf{f}^2$ , the DM realized that with this pitching moment coefficient, the improvement in the ratio of drag to lift coefficient is very small. Thus, the DM sacrificed the pitching moment coefficient to improve the ratio of drag to lift coefficient, and provided the reference point  $\bar{\mathbf{z}}^{*3}$ . The preferred solution  $\mathbf{f}^3$  was shown to the DM. The change in the pitching moment coefficient from  $\mathbf{f}^2$  to  $\mathbf{f}^3$  was acceptable for the DM. Based on the pitching moment coefficient in  $\mathbf{f}^3$ , the



DM provided  $\bar{\mathbf{z}}^{*4}$  to improve the ratio of drag to lift coefficient. The corresponding preferred solution  $\mathbf{f}^4$  was presented to the DM.

When comparing  $\bar{\mathbf{z}}^{*4}$  and  $\mathbf{f}^4$ , the DM figured out that the ratio of drag to lift coefficient is very close to his preferred value given in  $\bar{\mathbf{z}}^{*4}$ . Since the pitching moment coefficient in  $\mathbf{f}^4$  was acceptable for the DM,  $\mathbf{f}^4$  satisfied him and the solution process was terminated. Table 1 summarizes the reference points given and the corresponding preferred solutions in the objective space obtained by SURROGATE-ASF (SUR) with MATSuMoTo. Figure 5 depicts the geometric shape of the airfoils corresponding to the preferred solutions obtained in each iteration. In the decision making phase, only four computationally expensive function evaluations were conducted, that is, one for each reference point. Because of this, the DM was able to find the most preferred solution fast.

For the reference points given by the DM, we also obtained the corresponding preferred solutions by the Yun method and PAINT. Since the Yun method is a non-interactive method, we ran the method for each individual reference point. The number of function evaluations used for each reference point was set as  $\lfloor \frac{1688}{4} \rfloor = 422$ . Regarding PAINT, to build a surrogate problem, we employed the same reference solutions in  $F^P$  used in SURROGATE-ASF. During the decision making phase, we only obtained the approximated preferred solutions in the objective space. At the end of the decision making by PAINT, we solved problem (3) for the last reference point given by the DM. The maximum number of function evaluations in this step was 288 function evaluations, since 1400 evaluations were already used in obtaining the reference solutions  $F^P$ .

Since the Pareto optimal set of this problem is not known, we cannot use the Euclidean distance (9) and, instead, compare the solutions obtained in terms of dominance. As can be seen in Table 1, all solutions obtained by SURROGATE-ASF with MATSuMoTo dominate the corresponding solutions obtained by the Yun method and PAINT.

In terms of computational burden, SURROGATE-ASF with MATSuMoTo required 66 minutes to build surrogate functions in the initialization phase. In the decision making phase, the DM waited 2 seconds for each reference point to see the preferred solutions. In PAINT, the surrogate problem was built in 57 minutes. Then, for all reference point except the last once, the DM was able to see the preferred solutions after 2 seconds. However, to see the preferred solution corresponding to the last iteration, the DM had to wait for 9 minutes. In the Yun method, for each reference point, the DM had to wait for 1 day to see the corresponding solution. To summarize, the last preferred solution was obtained with SURROGATE-ASF, PAINT and the Yun method after 8 seconds, 546 seconds and 4 days, respectively. This comparison highlights the advantage of SURROGATE-ASF in which the DM was able to find the preferred solutions quickly. All this was done on a Dell Desktop with Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz processor and 8.00 GB RAM.

## 4.2 Solving benchmark problems

In this subsection, we evaluate the performance of SURROGATE-ASF through solving ZDT [37] and DTLZ [38] benchmark problems. The Pareto frontiers of ZDT1, ZDT4 and DTLZ1 are convex, of ZDT2, ZDT6, DTLZ2, DTLZ3, DTLZ4, DTLZ5 and DTLZ6 are nonconvex and of ZDT3 and DTLZ7 are nonconvex and disconnected. Moreover, ZDT4, ZDT6, DTLZ2, DTLZ3, DTLZ4 and DTLZ6 have many locally nondominated solutions. These benchmark problems also contain multimodal objective functions. For all benchmark problems, we considered  $n = 7$  decision variables. The number of objective functions in ZDT and DTLZ problems was  $K = 2$  and  $K = 3$ , respectively. Additionally, to evaluate the performance of SURROGATE-ASF on problems with more than 3 objective functions, we considered DTLZ2 (with multimodal objective functions) with 4 and 5 objective functions. Although objective functions involved in the

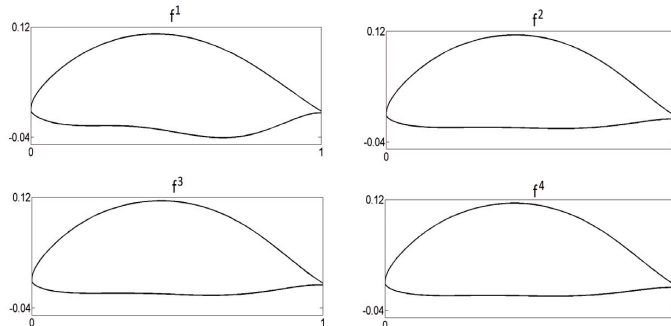


Figure 5: Geometric representation of the airfoils corresponding to each iteration

Table 1: Reference points given by the DM and corresponding preferred solutions in the objective space

Reference points	Preferred solutions (SUR.)	Solutions (PAINT.)	Solutions (Yun.)
1 $\bar{\mathbf{z}}^1 = (6.500 \times 10^{-3}, 8.000 \times 10^{-3})$	$\mathbf{f}^1 = (6.339 \times 10^{-3}, 7.586 \times 10^{-3})$	$\mathbf{f}^1 = (6.339 \times 10^{-3}, 7.838 \times 10^{-3})$	$\mathbf{f}^1 = (8.513 \times 10^{-3}, 8.382 \times 10^{-3})$
2 $\bar{\mathbf{z}}^2 = (5.500 \times 10^{-3}, 7.586 \times 10^{-3})$	$\mathbf{f}^2 = (6.215 \times 10^{-3}, 8.372 \times 10^{-3})$	$\mathbf{f}^2 = (7.337 \times 10^{-3}, 1.533 \times 10^{-2})$	$\mathbf{f}^2 = (9.218 \times 10^{-3}, 3.628 \times 10^{-2})$
3 $\bar{\mathbf{z}}^3 = (5.000 \times 10^{-3}, 1.500 \times 10^{-2})$	$\mathbf{f}^3 = (5.023 \times 10^{-3}, 1.662 \times 10^{-2})$	$\mathbf{f}^3 = (5.337 \times 10^{-3}, 1.718 \times 10^{-2})$	$\mathbf{f}^3 = (7.492 \times 10^{-3}, 2.636 \times 10^{-2})$
4 $\bar{\mathbf{z}}^4 = (5.000 \times 10^{-3}, 1.662 \times 10^{-2})$	$\mathbf{f}^4 = (5.006 \times 10^{-3}, 1.667 \times 10^{-2})$	$\mathbf{f}^4 = (9.164 \times 10^{-3}, 2.483 \times 10^{-2})$	$\mathbf{f}^4 = (7.317 \times 10^{-3}, 2.583 \times 10^{-2})$

benchmark problems are known, we treat them as black-box functions. In order to demonstrate the performance of SURROGATE-ASF in computationally expensive problems where solution times may be limited, we set the maximum number of function evaluations for each benchmark problem as 300.

For the problems considered here, along with MATSuMoTo discussed in Subsection 4.1, we applied the derivative-free DIRECT method [39] (can be downloaded from [http://www4.ncsu.edu/~ctk/Finkel\\_Direct/](http://www4.ncsu.edu/~ctk/Finkel_Direct/)) as the second method to solve problem (3) in the initialization phase. Although DIRECT has not been developed for computationally expensive problems, we show how these different types of methods affect the performance of SURROGATE-ASF. Table A.3 in Appendix A summarizes the parameter settings of DIRECT.

For each benchmark problem, we first applied MATSuMoTo and then DIRECT to solve problem (3) taking into account the preferences of the DM. In the initialization phase of SURROGATE-ASF, according to equation (5),  $h_z = 4$  ( $q = 3$ ) predetermined reference points were considered for biobjective problems. For three-, four- and five-objective problems we set  $h_z = 6$  ( $q = 2$ ),  $h_z = 10$  ( $q = 2$ ) and  $h_z = 15$  ( $q = 2$ ), respectively.

By increasing the number of objective functions, the number of predetermined reference points is also increased. On the other hand, the computational budget was fixed. Therefore, in this phase, for two-, three-, four- and five-objective problems, we considered 230, 230, 200 and 150 function evaluations, respectively, to find the reference solutions (non-dominated solutions) corresponding to the predetermined reference points by solving problem (3). In two-, three-, four- and five-objective problems, we considered  $l = 3, 4, 5$  and 5 reference points given by the DM. This means that in the decision making phase, only  $l$  function evaluations were required. The remaining number of function evaluations was used in the initialization phase to build a surrogate function for each sub-region. In the decision making phase, while the computationally inexpensive surrogate problem (8) can be solved by any appropriate global optimization method, we employed DIRECT.

Similar to Subsection 4.1, we compare the performance of SURROGATE-ASF (with both MATSuMoTo and DIRECT) taking into account the preferences of a DM to that of the Yun

method and PAINt. In this comparison, we find solutions corresponding to reference points given by the DM within the preferred ranges of the objective functions provided by the DM. These preferred ranges for all benchmark problems were set as  $[-50, 50]^k$ . This demonstrates that the method is not too sensitive on the choice of the ranges. As before, the Yun method was run for each individual reference point given by the DM. The number of function evaluations used for each reference point in all benchmark problems was set as  $\lfloor \frac{300}{l} \rfloor$  where  $l$  is the number of reference points given by the DM in the decision making phase. Regarding PAINt to build a surrogate problem, we employed the same non-dominated solutions used in SURROGATE-ASF. During the decision making phase, we only obtained the approximated preferred solutions in the objective space. At the end of the decision making by PAINt, we solved problem (3) for the last reference point given by the DM. The maximum number of function evaluations in this step was the remaining budget out of 300 function evaluations.

Tables 2.a and 2.b summarize the Euclidean distances  $E_j$ ,  $j = 1, \dots, l$ , between the approximated solutions obtained with SURROGATE-ASF (with MATSuMoTo), PAINt (with MATSuMoTo) and the Yun method and the original solutions for biobjective problems. Corresponding results for three-, four- and five-objective problems with MATSuMoTo and DIRECT are summarized in Tables 3.a, 3.b, 4.a and 4.b, respectively. Bold figures show the smallest distance between the approximated solutions obtained and the original solutions for each reference point. In all tables, SUR, MAT and DIR stand for SURROGATE-ASF, MATSuMoTo and DIRECT, respectively.

As far as applying MATSuMoTo is concerned to solve problem (3) to obtain reference solutions corresponding to the predetermined reference points, SURROGATE-ASF outperformed PAINt and the Yun method on ZDT1, ZDT2, ZDT3, ZDT6 (highlighted distances in Table 2.a), on DTLZ1, DTLZ4, DTLZ5, DTLZ7 (highlighted distances in Table 3.a) and on DTLZ2 with 4 and 5 objective functions (highlighted distances in Table 4.a). This is due to the fact that the reference solutions obtained by MATSuMoTo were closer to the Pareto frontiers of these benchmark problems in comparison to those by DIRECT.

Regarding problems with disconnected Pareto frontiers, SURROGATE-ASF with MATSuMoTo outperformed PAINt and Yun on ZDT3 and DTLZ7. As can be seen in Tables 2.a, 2.b, 3.a and 3.b, the Euclidean distances corresponding to solutions obtained by PAINt are larger than those by SURROGATE-ASF. This confirms the fact that PAINt does not perform well on problems with disconnected Pareto frontiers as mentioned in [20].

As can be seen in Table 3.a, on DTLZ2 with 3 objective functions, SURROGATE-ASF performed better than the other methods expect on the last reference point. Due to the limited function evaluation budget for each hyper-box (sub-problem), Algorithm 2 stopped updating the surrogate function corresponding to this reference point before reaching the desirable accuracy  $R^2 = 0.95$  and  $RMSE = 0.005$ . As a result, for this reference point, the approximated solution obtained by PAINt was slightly closer to the original one. On DTLZ6, the Yun method performed slightly better than SURROGATE-ASF with MATSuMoTo. For other reference points, though, approximated solutions obtained by SURROGATE-ASF were closer to the original ones.

On ZDT4 (in Table 2.a) and DTLZ3 (in Table 3.a), the reference solutions obtained by DIRECT were closer to the Pareto frontier. As a result, SURROGATE-ASF with DIRECT performed better than PAINt and the Yun method (as show in Tables 2.b and 3.b). In particular, SURROGATE-ASF with DIRECT surpassed the Yun method significantly (in Table 3.b). On the other hand, the Yun method outperformed SURROGATE-ASF with MATSuMoTo on these problem.

Regarding DTLZ4 (in Table 3.b), DIRECT found the same reference solutions for all pre-determined reference points. Therefore, SURROGATE-ASF and PAINt could not proceed to build surrogate problems (shown by a dash in the table). As can be seen in Tables 3.a and

Table 2.a: The Euclidean distance between approximated and original solutions in the objective space (biobjective problems, SUR. and PAINT with MAT.)

	$E_1$			$E_2$			$E_3$		
	SUR. (MAT.)	PAINT (MAT.)	Yun	SUR. (MAT.)	PAINT (MAT.)	Yun	SUR. (MAT.)	PAINT (MAT.)	Yun
ZDT1	<b>0.0006</b>	0.0473	0.0037	<b>0.0009</b>	0.0154	0.2682	<b>0.0007</b>	0.0010	0.7480
ZDT2	<b>0.0034</b>	0.0471	0.0056	<b>0.0064</b>	0.0153	0.1690	<b>0.0094</b>	0.0097	0.7491
ZDT3	<b>0.0008</b>	0.1298	0.0968	<b>0.0250</b>	0.2445	0.2737	<b>0.0010</b>	0.4811	0.6548
ZDT4	2.2863	2.931	<b>1.0034</b>	2.3712	2.8613	<b>0.8359</b>	2.2841	2.737	<b>1.0210</b>
ZDT6	<b>1.0806</b>	1.1191	1.3533	<b>1.3481</b>	1.4189	1.5894	<b>1.4123</b>	1.6133	2.0300

Table 2.b: The Euclidean distance between approximated and original solutions in the objective space (biobjective problems, SUR. and PAINT with DIR.)

	$E_1$			$E_2$			$E_3$		
	SUR. (DIR.)	PAINT (DIR.)	Yun	SUR. (DIR.)	PAINT (DIR.)	Yun	SUR. (DIR.)	PAINT (DIR.)	Yun
ZDT1	1.4974	1.9321	<b>0.0037</b>	1.7398	1.8968	<b>0.2682</b>	2.0128	1.9826	<b>0.7480</b>
ZDT2	2.4853	2.7492	<b>0.0056</b>	2.6397	2.8363	<b>0.1690</b>	2.5849	2.8959	<b>0.7491</b>
ZDT3	1.5763	2.3218	<b>0.0968</b>	1.4856	2.1968	<b>0.2737</b>	1.6003	2.8539	<b>0.6548</b>
ZDT4	<b>0.7493</b>	0.9910	1.0034	<b>0.5268</b>	1.0044	0.8359	<b>0.8862</b>	1.1073	1.0210
ZDT6	6.8529	6.9215	<b>1.3533</b>	7.2893	6.9851	<b>1.5894</b>	7.1002	7.2638	<b>2.0300</b>

Table 3.a: The Euclidean distance between approximated and original solutions in the objective space (3-objective problems, SUR. and PAINT with MAT.)

	$E_1$			$E_2$			$E_3$			$E_4$		
	SUR. (MAT.)	PAINT (MAT.)	Yun	SUR. (MAT.)	PAINT (MAT.)	Yun	SUR. (MAT.)	PAINT (MAT.)	Yun	SUR. (MAT.)	PAINT (MAT.)	Yun
DTLZ1	<b>0.1157</b>	0.9976	0.2237	<b>0.2417</b>	0.2731	0.3567	<b>0.2149</b>	0.2693	0.6598	<b>0.2719</b>	0.2958	0.3345
DTLZ2	<b>0.0554</b>	0.1019	37.2177	<b>0.0844</b>	0.1129	38.6381	<b>0.0288</b>	0.0821	7.6622	0.0854	<b>0.0803</b>	37.4879
DTLZ3	103.6793	104.2896	<b>86.0869</b>	106.8422	105.3853	<b>86.3371</b>	109.1386	110.3392	<b>86.1709</b>	104.3749	105.5237	<b>86.0868</b>
DTLZ4	<b>0.4887</b>	0.6264	0.7113	<b>0.5538</b>	0.5846	0.5657	<b>1.0033</b>	1.2785	1.9086	<b>0.7783</b>	1.1804	0.9340
DTLZ5	<b>0.1586</b>	1.0246	0.8646	<b>0.0118</b>	0.0545	0.2236	<b>0.0213</b>	0.0538	0.5872	<b>0.0081</b>	0.0470	0.9028
DTLZ6	<b>4.2795</b>	6.0290	5.1001	4.9227	5.0669	<b>4.8404</b>	<b>5.1369</b>	6.0671	6.3696	<b>5.7331</b>	6.0458	7.4418
DTLZ7	<b>0.0568</b>	0.9655	43.7997	<b>0.3099</b>	0.8358	41.4128	<b>0.1975</b>	0.9003	42.6465	<b>0.3172</b>	0.9182	42.2017

Table 3.b: The Euclidean distance between approximated and original solutions in the objective space (3-objective problems, SUR. and PAINT with DIR.)

	$E_1$			$E_2$			$E_3$			$E_4$		
	SUR. (DIR.)	PAINT (DIR.)	Yun	SUR. (DIR.)	PAINT (DIR.)	Yun	SUR. (DIR.)	PAINT (DIR.)	Yun	SUR. (DIR.)	PAINT (DIR.)	Yun
DTLZ1	<b>0.0233</b>	0.9076	0.2237	<b>0.0121</b>	0.1102	0.3567	<b>0.0072</b>	0.0107	0.6598	<b>0.0061</b>	0.0109	0.3345
DTLZ2	<b>0.1257</b>	0.1738	37.2177	<b>0.1439</b>	0.1857	38.6381	<b>0.1355</b>	0.1769	7.6622	<b>0.1578</b>	0.1698	37.4879
DTLZ3	<b>0.0554</b>	0.1239	86.0869	<b>0.0145</b>	0.1086	86.3371	<b>0.0944</b>	0.1028	86.1709	<b>0.0384</b>	0.0870	86.0868
DTLZ4	----	----	<b>0.7113</b>	----	----	<b>0.5657</b>	----	----	<b>1.9086</b>	----	----	<b>0.9340</b>
DTLZ5	<b>0.1871</b>	1.5392	0.8646	<b>0.1549</b>	0.1973	0.2236	<b>0.1052</b>	0.1215	0.5872	<b>0.1639</b>	0.1861	0.9028
DTLZ6	5.2739	5.4839	<b>5.1001</b>	5.7392	5.8254	<b>4.8404</b>	6.4821	6.6292	<b>6.3696</b>	7.6935	7.5284	<b>7.4418</b>
DTLZ7	49.2815	52.1276	<b>43.7997</b>	43.5382	51.2959	<b>41.4128</b>	44.2883	51.3695	<b>42.6465</b>	43.5293	54.1857	<b>42.2017</b>

Table 4.a: The Euclidean distance between approximated and original solutions in the objective space (DTLZ2 problem, SUR. and PAINT with MAT.)

	$E_1$			$E_2$			$E_3$			$E_4$			$E_5$		
	SUR. (MAT.)	PAINT (MAT.)	Yun	SUR. (MAT.)	PAINT (MAT.)	Yun	SUR. (MAT.)	PAINT (MAT.)	Yun	SUR. (MAT.)	PAINT (MAT.)	Yun	SUR. (MAT.)	PAINT (MAT.)	Yun
k=4	<b>0.1200</b>	0.1597	0.2102	<b>0.1010</b>	0.3723	0.1903	<b>0.1037</b>	0.3651	0.1683	<b>0.0792</b>	0.3409	0.2196	<b>0.1412</b>	0.5129	0.3159
k=5	<b>0.3427</b>	0.7312	0.4676	<b>0.3963</b>	0.9581	0.8426	<b>0.3234</b>	0.9502	0.7885	<b>0.3732</b>	0.8194	0.5601	<b>0.1854</b>	0.1937	0.6058

Table 4.b: The Euclidean distance between approximated and original solutions in the objective space (DTLZ2 problem, SUR. and PAINT with DIR.)

	$E_1$			$E_2$			$E_3$			$E_4$			$E_5$		
	SUR. (DIR.)	PAINT (DIR.)	Yun	SUR. (DIR.)	PAINT (DIR.)	Yun	SUR. (DIR.)	PAINT (DIR.)	Yun	SUR. (DIR.)	PAINT (DIR.)	Yun	SUR. (DIR.)	PAINT (DIR.)	Yun
k=4	<b>0.1538</b>	0.1729	0.2102	<b>0.1323</b>	0.4281	0.1903	<b>0.1273</b>	0.4134	0.1683	<b>0.1639</b>	0.4317	0.2196	<b>0.1381</b>	0.5126	0.2017
k=5	<b>0.4179</b>	0.8183	0.4676	<b>0.4261</b>	1.0111	0.8426	<b>0.5192</b>	1.0016	0.7885	<b>0.4673</b>	0.9713	0.5601	<b>0.3416</b>	0.5492	0.6058

3.b, MATSuMoTo and DIRECT solved problem (3) for DTLZ1 and DTLZ2. On DTLZ1, not only SURROGATE-ASF with DIRECT (Table 3.b) outperformed PAINTE and the Yun method, but also SURROGATE-ASF with MATSuMoTo. For DTLZ2, SURROGATE-ASF with MATSuMoTo (Table 3.a) performed better than SURROGATE-ASF with DIRECT, PAINTE and the Yun method.

Tables 4.a and 4.b summarize the results for the problems with four and five objective functions. By increasing the number of objective functions, the number of sub-problems is also increased. With a limited number of function evaluations for each hyper-box (sub-problem), Algorithm 2 terminated the updating process based on the computational budget allocated rather than reaching the desirable accuracy  $R^2 = 0.95$  and  $RMSE = 0.005$ . However, SURROGATE-ASF with MATSuMoTo and DIRECT surpassed both PAINTE and the Yun method. Moreover, SURROGATE-ASF with MATSuMoTo performed better than SURROGATE-ASF with DIRECT. Overall, SURROGATE-ASF performed very well in the numerical tests when compared to the two other methods.

## 5 Conclusions and future research directions

In this paper, we developed an interactive surrogate-based method called SURROGATE-ASF to solve computationally expensive MOPs. This method consists of two phases: initialization and decision making phases. In the initialization phase, the decision space is decomposed into a finite number of hyper-boxes. For each hyper-box, a single objective surrogate of the achievement scalarizing function is built by using e.g. a cubic RBF with a linear tail. In the decision making phase, iterations with a decision maker are conducted in an interactive fashion. In this phase, at each iteration a reference point is specified by the decision maker. Then, a surrogate single objective optimization problem is formulated and solved by any appropriate single objective optimization method. The optimal solution is an approximation of the preferred solution in the decision space. This solution is evaluated with the original functions and shown to the decision maker. The interaction with the decision maker is repeated until the most preferred solution for him/her is found.

In SURROGATE-ASF, the approach of building a computationally inexpensive surrogate of the achievement scalarizing function is applied instead of a typical approach of building a surrogate of each individual computationally expensive objective function and then forming the achievement scalarizing function as discussed in [18]. Therefore, when interacting with the DM, only a single objective optimization problem is solved. The surrogate assists the DM to find his/her preferred solutions in both the decision and the objective spaces quickly. Numerical results confirmed that SURROGATE-ASF performed very well, in particular, on problems with multimodal objective functions, non-convex and/or disconnected Pareto frontier and it fills a gap in the selection of methods available for computationally expensive problems. Solving a real-world computationally expensive airfoil optimization problem demonstrated that SURROGATE-ASF reduced the computational burden significantly in comparison with the typical approach by which the DM had to wait a couple of days to find the most preferred solution. The speed can be further improved by applying parallel computing when building single objective surrogate functions.

By increasing the number of objective functions, the number of sub-problems is also increased. Therefore, with a fixed budget of function evaluations, the number of function evaluations for each sub-problem is decreased. This may lead to a sacrificed accuracy of the sub-problems. A possible idea is to merge hyper-boxes and sub-regions to reduce the number of sub-problems. This is a future research direction when SURROGATE-ASF is to be applied for problems with a large number of objective functions. SURROGATE-ASF with the current setting along with an

appropriate single objective surrogate-based method can be employed for problem with a high-dimensional decision space. However, the accuracy of the surrogate functions for each hyper-box may not be satisfactory. A possible approach is to apply a dimensionality reduction method within each hyper-box to obtain a low-dimensional hyper-box. Then, SURROGATE-ASF with the current format can be applied for each new hyper-box. This is another future research direction.

## Acknowledgement

Mohammad Tabatabaei thanks the financial support from the COMAS Doctoral Program in Computing and Mathematical Sciences at the University of Jyväskylä, the Academy of Finland project No. 287496 and the Early Career Scheme (ECS) project No. 21201414 sponsored by the Research Grants Council of Hong Kong (Dr. Matthias Hwai Yong Tan). He also thanks Dr. Alfredo Arias Montano for providing the airfoil optimization problem code and Prof. Raimo Mäkinen for fruitful discussions on this problem.

## References

- [1] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, 1999.
- [2] K. Miettinen. Introduction to multiobjective optimization: Noninteractive approaches. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 1–26. Springer-Verlag Berlin Heidelberg, 2008.
- [3] K. Miettinen, F. Ruiz, and A. P. Wierzbicki. Introduction to multiobjective optimization: Interactive approaches. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 27–57. Springer-Verlag Berlin Heidelberg, 2008.
- [4] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.
- [5] R. G. Regis and C. A. Shoemaker. Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. *Engineering Optimization*, 45(5):529–555, 2012.
- [6] C. Audet. A survey on direct search methods for blackbox optimization and their applications. In P. M. Pardalos and T. M. Rassias, editors, *Mathematics Without Boundaries*, pages 31–56. Springer New York, 2014.
- [7] M. Tan. Sequential Bayesian polynomial chaos model selection for estimation of sensitivity indices. *SIAM/ASA Journal on Uncertainty Quantification*, 3:146–168, 2015.
- [8] B. Dengiz, C. Alabas-Uslu, and O. Dengiz. Optimization of manufacturing systems using a neural network metamodel with a new training approach. *Journal of the Operational Research Society*, 60(9):1191–1197, 2009.
- [9] M. Chih. A more accurate second-order polynomial metamodel using a pseudo-random number assignment strategy. *Journal of the Operational Research Society*, 64(2):198–207, 2013.

- [10] I. M. Reis dos Santos and M. P. Reis dos Santos. Construction and validation of distribution-based regression simulation metamodels. *Journal of the Operational Research Society*, 62(7):1376–1384, 2011.
- [11] D. R. Hurrion. A sequential method for the development of visual interactive meta-simulation models using neural networks. *Journal of the Operational Research Society*, 51(6):712–719, 2000.
- [12] C. J. P. Kleijnen and M. W. C. van Beers. Monotonicity-preserving bootstrapped kriging metamodels for expensive simulations. *Journal of the Operational Research Society*, 64(5):708–717, 2013.
- [13] M. W. C. van Beers and C. J. P. Kleijnen. Kriging for interpolation in random simulation. *Journal of the Operational Research Society*, 54(3):255–262, 2003.
- [14] T. Yang and L. Tseng. Solving a multi-objective simulation model using a hybrid response surface method and lexicographical goal programming approach—a case study on integrated circuit ink-marking machines. *Journal of the Operational Research Society*, 53(2):211–221, 2002.
- [15] E. Mehdad and C. J. P. Kleijnen. Classic kriging versus kriging with bootstrapping or conditional simulation: classic kriging’s robust confidence intervals and optimization. *Journal of the Operational Research Society*, 66(11):1804–1814, 2015.
- [16] Taejin Park, Bongjin Yum, Ying Hung, Young-Seon Jeong, and K. Myong Jeong. Robust kriging models in computer experiments. *Journal of the Operational Research Society*, 67(4):644–653, 2016.
- [17] A. Messac and A. A. Mullur. A computationally efficient metamodeling approach for expensive multiobjective optimization. *Optimization and Engineering*, 9(1):37–67, 2008.
- [18] Y. Yun, M. Yoon, and H. Nakayama. Multi-objective optimization based on meta-modeling by using support vector regression. *Optimization and Engineering*, 10(2):167–181, 2009.
- [19] S. Kitayama, J. Srirat, M. Arakawa, and K. Yamazaki. Sequential approximate multi-objective optimization using radial basis function network. *Structural and Multidisciplinary Optimization*, 48(3):501–515, 2013.
- [20] M. Hartikainen, K. Miettinen, and M. M. Wiecek. PAINT: Pareto front interpolation for nonlinear multiobjective optimization. *Computational Optimization and Applications*, 52(3):845–867, 2012.
- [21] M. Hartikainen and A. Lovison. Paint-SiCon: constructing consistent parametric representations of Pareto sets in nonconvex multiobjective optimization. *Journal of Global Optimization*, 62(2):243–261, 2015.
- [22] A. B. Ruiz, K. Sindhya, K. Miettinen, F. Ruiz, and M. Luque. E-NAUTILUS: A decision support system for complex multiobjective optimization problems based on the nautilus method. *European Journal of Operational Research*, 246(1):218–231, 2015.
- [23] M. Tabatabaei, J. Hakanen, M. Hartikainen, K. Miettinen, and K. Sindhya. A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods. *Structural and Multidisciplinary Optimization*, 52(1):1–25, 2015.
- [24] O. I. Larichev. Cognitive validity in design of decision-aiding techniques. *Journal of Multi-Criteria Decision Analysis*, 1(3):127–138, 1992.
- [25] A. P. Wierzbicki. On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR Spectrum*, 8(2):73–87, 1986.

- [26] J. Müller and C. A. Shoemaker. Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems. *Journal of Global Optimization*, 60(2):123–144, 2014.
- [27] I. Das and J. Dennis. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.
- [28] J. C. Helton, J. D. Johnson, C. J. Sallaberry, and C. B. Storlie. Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering & System Safety*, 91(10-11):1175–1209, 2006.
- [29] Juliane Muller, Christine A. Shoemaker, and Robert Piche. So-mi: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems. *Computers & Operations Research*, 40(5):1383–1400, 2013.
- [30] A. Giunta, L. T. Watson, and J. Koehler. A comparison of approximation modeling techniques: Polynomial versus interpolating models. In *Proceedings of 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, volume 1, pages 392–404, St. Louis, MO, 1998.
- [31] R. Jin, W. Chen, and T. W. Simpson. Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and Multidisciplinary Optimization*, 23(1):1–13, 2001.
- [32] E. Schaumann, R. Balling, and K. Day. Genetic algorithms with multiple objectives. In *Proceedings of 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, volume 3, pages 2114–2123, St. Louis, MO, 1998.
- [33] A. P. Wierzbicki. A mathematical basis for satisficing decision making. *Mathematical Modelling*, 3(5):391–405, 1982.
- [34] M. Drela. Xfoil: An analysis and design system for low Reynolds number aerodynamics. University Of Notre Dame, IN, 1989.
- [35] A. Arias-Montano, C. A. Coello Coello, and E. Mezura-Montes. Multi-objective airfoil shape optimization using a multiple-surrogate approach. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2012.
- [36] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, 1989.
- [37] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.
- [38] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Proceedings of Congress on Evolutionary Computation*, volume 1, pages 825–830, Honolulu, USA, 2002. IEEE.
- [39] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.



## Appendix A

Table A.1: Parameter settings for SURROGATE-ASF

Initial number of sample points within each hyperbox	n+1	
Number of reference sample points within each sub-region	k=2	30 (q = 29)
	k=3	66 (q = 10)
	k=4	84 (q = 6)
	k=5	125 (q = 5)

Table A.2: Parameter settings for MATSuMoTo

Metamodeling technique	cubic RBF
Sampling technique	Latin hypercube sampling
New candidate points to be evaluated with the computationally expensive functions	adding random perturbations to the best point found so far
Initial number of sample points	2(n+1)

Table A.3: Parameter settings for DIRECT

	To solve problem (3)	To solve problem (8)
Maximum number of function evaluations	differs for each reference point and benchmark problem	500
Maximum number of iterations	500	500
Maximum number of rectangle divisions	1000	1000

**PIII**

**ANOVA-MOP: ANOVA DECOMPOSITION FOR  
MULTIOBJECTIVE OPTIMIZATION**

by

M. Tabatabaei, A. Lovison, M. Hartikainen, M. Tan, and K. Miettinen

Submitted

# ANOVA-MOP: ANOVA Decomposition for Multiobjective Optimization

Mohammad Tabatabaei<sup>\*1</sup>, Alberto Lovison<sup>+2</sup>, Matthias Tan<sup>‡3</sup>,  
Markus Hartikainen<sup>§1</sup>, and Kaisa Miettinen<sup>¶1</sup>

<sup>1</sup>University of Jyväskylä, Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland

<sup>2</sup>Dipartimento di Matematica, Università degli Studi di Padova, Via Trieste, 63, 35121 Padova, Italy

<sup>3</sup>Department of Systems Engineering and Engineering Management, City University of Hong Kong, Kowloon Tong, Hong Kong

## Abstract

Real-world optimization problems may involve several high-dimensional computationally expensive functions. Because of the curse-of-dimensionality, gaining insight into problems' structure and solving them are challenging. Surrogate-based methods are typically employed to mitigate the computational costs of expensive functions, but this does not reduce the curse-of-dimensionality. This challenge, however, can be tackled by the approach of computation of Sobol' indices and the functional ANOVA decomposition to reduce the dimension of the search domain. This approach allows ranking decision variables according to their impact on objective values. Due to the Sparsity of Effects principle, a small number of decision variables sensibly affects an objective value. Therefore the remaining variables can be neglected, leading to form simpler and tractable problems. Although this is an effective strategy, so far it has only

---

\*mohammad.tabatabaei@jyu.fi; corresponding author

+lovison@math.unipd.it

‡mathtan@cityu.edu.hk

§markus.hartikainen@jyu.fi

¶kaisa.miettinen@jyu.fi

been applied to single-objective optimization problems. In this paper, we develop a novel surrogate-based method called ANOVA-MOP in which ANOVA decomposition is applied to decompose a complex multiobjective optimization problem into a number of independent sub-problems with few decision variables and objective functions. The solutions of the sub-problems are then combined to build approximated solutions for the original complex problem. The quality of the approximated solutions and issues related to decision making are discussed.

**Keywords** Multiobjective Optimization, Sensitivity analysis, Metamodeling, Decision making

## 1 Introduction

Multiobjective optimization problems (MOPs) arise in many application domains such as engineering, finance, and even biology. Their peculiarity is the coexistence of a number of conflicting objective functions. Real-world MOPs typically contain black-box functions requiring computationally expensive and/or time-consuming experiments and/or simulations [1]. Because of the multiplicity of the objective functions, MOPs do not contain a unique optimal solution but rather several (usually infinite) equivalently optimal solutions, named Pareto optimal solutions, in honor of the mathematician and economist Vilfredo Pareto [2]. Mathematically, there is no a priori preference between Pareto optimal solutions, each one defined by a different trade-off among objectives, and at this stage, a decision maker (DM) is required to identify the most preferred solution to be implemented.

As explained in [3, 4, 5], multiobjective optimization methods are classified according to the role of the DM into non-interactive and interactive methods. In the former ones, the preference information of the DM either is not available or it is given only before or after the solution process. On the other hand, in interactive methods the DM is asked his/her preference at each iteration, (s)he analyzes the solutions and (s)he also can revise his/her preference information on the basis of the updated situation. The interaction with the DM is continued until (s)he finds the most preferred solution. As mentioned in [5], the DM can provide his/her preferences as e.g., reference points involving aspiration levels representing desirable values for objective functions or classification of objective functions. As discussed in [6], providing preference information in the form of a reference point has been regarded as a doable procedure to be adopted by a DM with no special preparation in mathematical programming. In the literature, surrogate-based methods are employed to relieve computational costs [7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. The principle of such methods is to devise a suitable computationally inexpensive numerical model of the problem, typically by means of splines, Gaussian processes, radial basis functions, Fourier expansions, etc. The numerical model is called the surrogate problem. Some of the computations can then be performed on the surrogate problem, for instance for computing derivatives, or the surrogate problem can even substituted for the original problem, comparing the solutions found on the surrogate problem with the original problem only at the final stage.

In literature, surrogate-based optimization methods have been developed in which the process of building a surrogate problem is independent of the type of optimization

algorithm employed in them. We surveyed such surrogate-based methods in [1] and identified some shortcomings such as the involvement of the DM in the solution process and solving MOPs with the possible high dimension of the decision and objective spaces. As discussed in [17], in general, the DM should not wait for a long time to see solutions corresponding to his/her preferences. Indeed, as discussed in [18], cognitive tasks require sufficient ability to hold the relevant information as long as it is processed. In literature, decision making in computationally expensive MOPs with a large dimension seems not been reported at all. This raises a question that how would the DM be comfortable in terms of cognitive load to provide preference information and compare solutions in such complex situations.

Although the use of surrogate models would also sensibly reduce the cost of computing objective functions, the intrinsic complexity of large dimensionality is not affected at all, i.e., the dimensionality is still challenging when computationally expensive functions are replaced with inexpensive numerical models. Indeed, searching for optima in an  $d$ -dimensional space, where  $d \gg 1$ , is usually very demanding and the use of global strategies exhaustively searching the decision space is discouraged for  $d > 5$ , also for single objective optimization (see, e.g., [19, 20]). On the other hand, when attempting to detect and describe the set of the optima of a MOP with  $m \gg 1$  objective functions means in the typical case to have to represent an  $(m - 1)$ -dimensional geometrical object [21]. Even when such a set is detected with satisfactory accuracy, it is virtually impossible to represent it and to allow for its exploration and analysis for a DM when  $m > 4$ .

The curse-of-dimensionality is accompanied by the sparsity of effects principle, i.e., usually there is a restricted selection of decision variables that really are important for the computation of objective function values, while the remaining have little effect. Detecting such important variables is called global sensitivity analysis, and one of the most powerful methods is called *functional analysis of variance decomposition* (functional ANOVA decomposition) [22]. Functional ANOVA consists in decomposing hierarchically an objective function in a summation of terms depending only on subsets of variables, according to their contribution to the total variance of the function. Terms depending on individual variables are called *main effects*, while the contributions due to pairs of variables are called *interactions*. Typically terms of higher interaction have less and less importance and as reported firstly by Pareto, we typically notice that the 80% of the effects are due to the 20% of the variables [23].

In single objective optimization problems, this would allow ignoring a large number of decision variables, greatly simplifying the complexity of the problems at hand. This, however, seems not to have been applied yet to multiple objectives, apparently because that if some variables are not important for an objective, this does not mean that they are not important for some other objective. So when multiple objectives are involved, the generalization of applicability of functional ANOVA decomposition is not straightforward. The seemingly most natural idea is to try to couple objective functions based on the decision variables which are most effective for them, and then grouping objectives and decision variables in non-overlapping blocks acting as independent sub-problems.

In this paper, we have formalized and implemented this idea, which we call the ANOVA-MOP method. The method can be employed as a non-interactive or an interactive method. In the later one, the method takes into account DM's preferences

in the form of a reference point. In some more details, we proceed by building a surrogate function for each computationally expensive function and forming a surrogate MOP. Then, by applying sensitivity analysis we identify effective and ineffective variables for each objective. The surrogate MOP is then decomposed into a set of sub-problems by partitioning decision variables and objective functions into independent blocks. Each sub-problem will have a reduced dimension both in the decision and objective spaces and will become tractable with standard methods. Solutions of the sub-problems then will be suitably combined to form approximated solutions for the original high-dimensional problem.

To put it in a nutshell, the novelty and attributes of ANOVA-MOP are considered as follows: 1) computationally expensive MOPs with any number of objective functions can be solved, 2) it can be applied as a non-interactive or an interactive method, 3) during the solution process, the focus is only on a low-dimensional sub-problem in the decision and objective spaces, which enables users to employ optimization algorithm for low dimensional problems, 4) as an interactive method, in each step of interaction with a DM, (s)he only provides his/her preferences for a few number of objective functions rather than a large number of them, 5) the DM does not require to wait for a long time to find solutions corresponding to his/her preferences and 5) besides a surrogate-based method, ANOVA-MOP can also be employed to solve computationally inexpensive MOPs. To the best of our knowledge, ANOVA-MOP is the first method in literature with such attributes.

The rest of this paper is organized as follows. In Section 2, basic concepts used in this paper are addressed. Decomposition in the context of MOPs is discussed in Section 3. In Section 4, we discuss functional ANOVA decomposition. The ANOVA-MOP method is presented in Section 5. Numerical results are given in Section 6. Finally, we conclude and provide some future research directions in Section 7.

## 2 Concepts

In this section, we introduce the concepts and notations used in this paper. We consider multiobjective optimization problems of the form

$$\underset{\mathbf{x} \in S}{\text{minimize}} \{f_1(\mathbf{x}), \dots, f_k(\mathbf{x})\}, \quad (1)$$

where  $f_i : S \rightarrow \mathbb{R}$  are  $k$  ( $\geq 2$ ) conflicting objective functions, the set  $S$  is a nonempty *feasible decision* set which is a subset of the *decision space*  $\mathbb{R}^d$ . A solution  $\mathbf{x} = (x_1, \dots, x_d)^T \in S$  is called a *feasible decision (variable) vector*, where  $x_j$  for  $j = 1, \dots, d$ , are *decision variables* and  $x_j \in S_j := [x_j^l, x_j^u]$ , where  $x_j^l$  and  $x_j^u$  are the lower and upper bounds of  $x_j$ , respectively and  $S = \prod_{j=1}^d S_j$ . We denote the *vector of objective functions* by  $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$  which is the image of  $\mathbf{x}$  in the *objective space*  $\mathbb{R}^k$ . The image of  $S$  in the objective space is called the *feasible objective* set denoted by  $Z (= \mathbf{f}(S))$ . In this paper, without loss of generality, we assume that  $S_j = [l, u]$ , for  $j = 1, \dots, d$ , because this could be obtained by affinely rescaling the decision variables. We denote the sets of indices of the objective functions and decision variables by  $O = \{1, \dots, k\}$  and  $D = \{1, \dots, d\}$ , respectively.

We say a feasible solution  $\mathbf{x}^1 \in S$  (and corresponding  $\mathbf{f}(\mathbf{x}^1) \in Z$ ) dominates a feasible solution  $\mathbf{x}^2 \in S$  (and corresponding  $\mathbf{f}(\mathbf{x}^2) \in Z$ ) if for all  $i \in O$ ,  $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$  and some  $j \in O$ ,  $f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2)$ . A feasible solution  $\mathbf{x}^* \in S$  and the corresponding  $\mathbf{f}(\mathbf{x}^*) \in Z$  are called *weakly Pareto optimal* for problem (1), if there does not exist another feasible solution  $\mathbf{x} \in S$  such that  $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$  for all  $i \in O$ . Correspondingly, they are *Pareto optimal* for problem (1), if there does not exist another feasible solution  $\mathbf{x} \in S$  such that dominates  $\mathbf{x}^* \in S$ . The set of Pareto optimal solutions in the decision space is called Pareto optimal set and denoted by  $X$ . The set of Pareto optimal solutions in the objective space is called a *Pareto frontier* and denoted by  $F$ .

We also define a feasible solution for the individual scalar functions  $\mathbf{x}^{i,min} \in \operatorname{argmin}_{\mathbf{x} \in S} \{f_i(\mathbf{x})\}$  for  $i \in O$ . The vector of minima of the objective functions is defined as the *ideal (objective) vector* and denoted by  $\mathbf{z}^{ideal} = (z_1^{ideal}, \dots, z_k^{ideal})^T$  where  $z_i^{ideal} = f_i(\mathbf{x}^{i,min})$  for  $i \in O$ . The *utopian (objective) vector*  $\mathbf{z}^{utp}$  is a vector in which its components are calculated by subtracting some predefined small positive scalar number  $\xi \ll 1$  from the components of  $\mathbf{z}^{ideal}$ . We also define the *nadir (objective) vector* denoted by  $\mathbf{z}^{nadir} = (z_1^{nadir}, \dots, z_k^{nadir})^T$  to be the vector of maxima of the objective function values in the Pareto frontier. Existence of ideal and nadir points is guaranteed by a compactness argument (see [3]).

In ANOVA-MOP, the DM provides his/her preferences in the form of a *reference point*  $\bar{\mathbf{z}} = (\bar{z}_1, \dots, \bar{z}_k)^T$ , where  $\bar{z}_i$  is an *aspiration level* representing a desirable value for the objective function  $f_i$ . One can find a solution corresponding to a reference point given by a DM by applying an appropriate scalarization. Scalarizing problem (1) means formulating a single objective optimization problem such that its (globally) optimal solution is a weakly Pareto optimal or Pareto optimal solution for problem (1). In this paper, we consider the following widely used *achievement scalarizing function* (ASF) [24]:

$$\begin{aligned} \text{ASF: } S \times \mathbb{R}^k &\rightarrow \mathbb{R} \\ (\mathbf{x}, \bar{\mathbf{z}}) &\mapsto \max_{i \in O} (w_i (f_i(\mathbf{x}) - \bar{z}_i)), \end{aligned} \quad (2)$$

where  $w_i \geq 0$ , for  $i \in O$ , are non-negative fixed weights which actually set a direction where  $\bar{\mathbf{z}}$  is projected onto the Pareto frontier. In this paper we set  $w_i = \frac{1}{z_i^{nadir} - z_i^{utp}}$ , for  $i \in O$ . A new single objective optimization problem is formulated as

$$\operatorname{minimize}_{\mathbf{x} \in S} \text{ASF}(\mathbf{x}, \bar{\mathbf{z}}). \quad (3)$$

By solving problem (3), a (weakly) Pareto optimal solution corresponding to a given reference point is obtained, regardless of the feasibility or infeasibility of the reference point. All other scalarizing approaches do not follow such a property [3]. By changing the reference point in problem (3), different (weakly) Pareto optimal solutions can be achieved. To avoid weakly Pareto optimal solutions, one can add a suitable augmentation term to (2) as discussed in [3].

We use the ANOVA decomposition to measure the importance of the variables in the objective function values. Suppose an arbitrary function  $g : S \rightarrow \mathbb{R}$  is square integrable, i.e.,  $\int_S g(\mathbf{x})^2 d\mathbf{x} < \infty$ . The *functional ANOVA decomposition* of  $g$  is

$$g(\mathbf{x}) = g_0 + \sum_i g_{(i)}(x_i) + \sum_{i < j} g_{(i,j)}(x_i, x_j) + \dots + g_{(1, \dots, d)}(\mathbf{x}), \quad (4)$$

where  $g_0 = \int_S g(\mathbf{x}) d\mathbf{x}$ , if

$$\int_{S_{i_u}} g_{(i_1, \dots, i_e)}(x_{i_1}, \dots, x_{i_e}) dx_{i_u} = 0, \quad (5)$$

for all  $u = 1, \dots, e, (e \geq 1)$ , and  $1 \leq i_1 < \dots < i_e \leq d$ . Following [25], in this paper, we call  $g_{(i)}$  a *main effect* and  $g_{i_1, \dots, i_e}$ , an *interaction* of order  $e$ . This results in decomposing the variance of  $g$ , i.e.,  $var(g) = D_0 = \int_S g^2(\mathbf{x}) d\mathbf{x}$ , into  $2^d - 1$  *variance components*

$$D_0 = \sum_i D_{(i)} + \sum_{i < j} D_{(i,j)} + \dots + D_{(1, \dots, d)}, \quad (6)$$

where  $D_{(i_1, \dots, i_e)} = var[g_{(i_1, \dots, i_e)}(x_{i_1}, \dots, x_{i_e})] = \int_{\prod_{u=1}^e S_{i_u}} g_{(i_1, \dots, i_e)}^2(x_{i_1}, \dots, x_{i_e}) dx_{i_1} \dots dx_{i_e}$ . The *Sobol indices* [26] are calculated as

$$l_{(i_1, \dots, i_e)} = D_{(i_1, \dots, i_e)} / D_0, \quad (7)$$

for all  $(i_1, \dots, i_e) \in \Delta = \{(r_1, \dots, r_u) : 1 \leq r_1 < \dots < r_u \leq d, u \geq 1\}, e = 1, \dots, d$ . The *total sensitivity indices*, which are measures of the contribution of each decision variable  $x_j$ , for  $j = 1, \dots, d$ , in the value of  $g$ , are given by

$$l_j^T = \sum_{(i_1, \dots, i_e) \in \Delta: j \in \{i_1, \dots, i_e\}} D_{(i_1, \dots, i_e)} / D_0, j = 1, \dots, d. \quad (8)$$

Practically, Sobol' indices and total sensitivity indices are approximated. In this paper, we refer to them as estimated Sobol' indices and estimated total sensitivity indices, respectively.

## 3 Decomposition in Multiobjective Optimization

### 3.1 Decomposability

In this subsection, we discuss the purpose of decomposition in the context of multi-objective optimization. Let us consider problem (1) with  $d$  decision variables and  $k$  objective functions. Assume that not all the objective functions depend on all decision variables. Let the incidence matrix  $M = [m_{i,j}]$ , with  $i \in O$  and  $j \in D$ , be defined as:

$$\begin{cases} m_{i,j} = 1, & \text{if function } f_i \text{ does depend on variable } x_j, \\ m_{i,j} = 0, & \text{if function } f_i \text{ does NOT depend on variable } x_j. \end{cases} \quad (9)$$

Figure 1(a) depicts the matrix  $M$  for a hypothetical MOP with  $k = 8$  and  $d = 8$ . As can be seen, in this matrix, the rows correspond to the objective functions and the columns correspond to the decision variables. The black and white squares correspond to matrix entries with values 1 and 0, respectively.

We apply decomposition to problem (1) to decrease the complexity of the problem in order to enhance the solution process towards a set of solutions. Such a decomposition procedure aims at decomposing problem (1) into a number of sub-problems with lower numbers of objective functions.



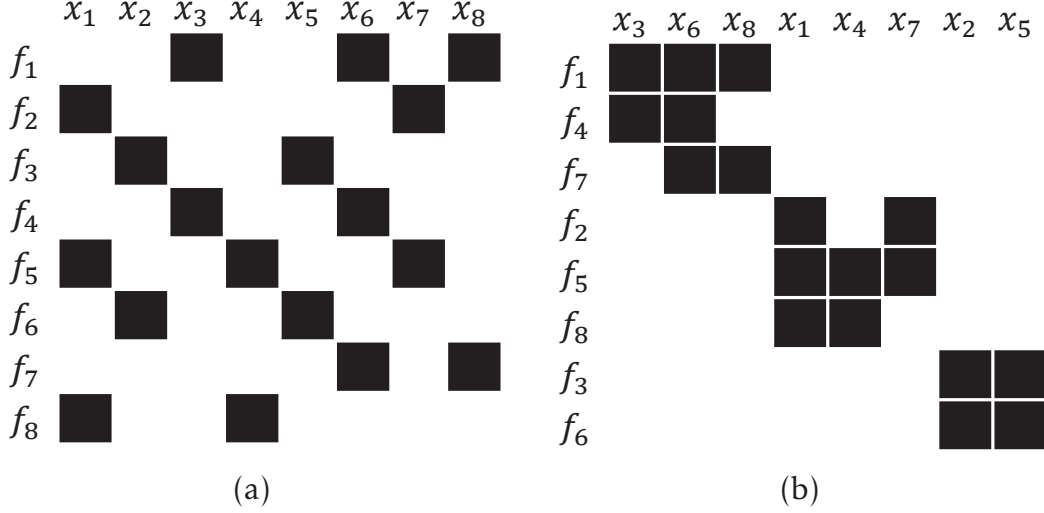


Figure 1: (a) the incidence matrix  $M$ , (b) A decomposition corresponding to the matrix  $M$  in (a).

**Definition 3.1.** We say that problem (1) (or equivalently matrix  $M$ ) is decomposable if the objective functions and the decision variables can be reordered such that the incidence matrix  $M$  becomes block-diagonal (with at least two non-degenerate blocks).

As shown in Figure 1 (b), for example, the first sub-problem includes the objective functions  $f_1, f_4$  and  $f_7$  and the decision variables  $x_3, x_6$  and  $x_8$ . The following remark defines when problem (1) is decomposable.

**Remark 3.2.** problem (1) is decomposable if and only if

1. There exist subsets  $O_a$  and  $D_a$  for  $a = 1, \dots, n$ , such that  $O = \bigcup_{a=1}^n O_a, \bigcap_{a=1}^n O_a = \emptyset,$   
 $D = \bigcup_{a=1}^n D_a$  and  $\bigcap_{a=1}^n D_a = \emptyset,$
2.  $m_{i,j} = 1$  implies that there exist an index  $\bar{a}$  such that  $i \in O_{\bar{a}}$  and  $j \in D_{\bar{a}}$ .

**Remark 3.3.** If problem (1) is decomposable, in the simplest case there are at least  $k + d - 2$  entries with 0 value in  $M$ .

In this paper, by complexity, we mean the complexity of solving problem (1) to obtain a set of solutions, as defined in [27] and called  $CI$ . The complexity of the decomposed problem is defined as the sum of the complexities of the sub-problems.

Figure 2 depicts a simple bi-objective decomposable problem. As can be seen, the Pareto frontier consists of a single point (the red point), rather than being a one-dimensional geometrical object, as it is in the typical case. This degeneracy and simplification of the solution set is due to the decomposability of the problem: the first objective function  $f_1$  depends only on a subset of the decision variables that are not involved in the definition of the second objective function  $f_2$  and vice versa. After

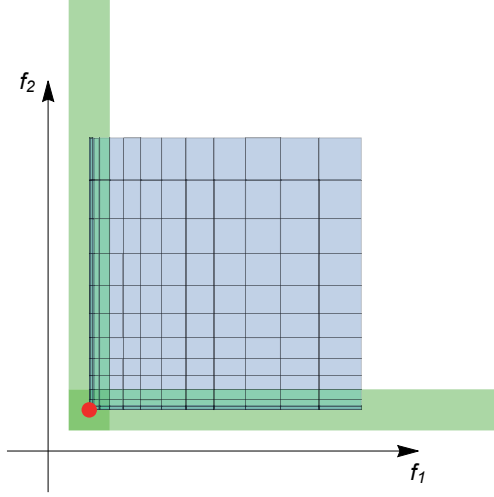


Figure 2: A bi-objective decomposable problem: the red point is the only Pareto optimal solution. (Color online)

performing a decomposition a question arises: how to build the solution set of the complete problem by solving the sub-problems?

### 3.2 Composing solutions of sub-problems

In this subsection, we discuss how to guide the solution process of problem (1) towards a set of solutions by solving sub-problems obtained through decomposition of problem (1). For the sake of simplicity, referring to an objective function, we mean either the original function (if it is inherently a computationally inexpensive one) or a surrogate function (if the original function is a computationally expensive one). We also assume that there are  $n$  sub-problems. We call the decision variables of the  $a^{th}$  sub-problem for  $a = 1, \dots, n$ , as effective variables and denote by  $\mathbf{x}^a \in S^a$ , in which  $S^a = \prod_{j \in D_a} S_j$  is the feasible decision set of the  $a^{th}$  sub-problem.

In each sub-problem, decision variables neglected are termed ineffective variables denoted by  $\underline{\mathbf{x}}^a \in \underline{S}^a$ . For a solution  $\mathbf{x} \in S$ ,  $(\mathbf{x}^a, \underline{\mathbf{x}}^a)$  is a representation of  $\mathbf{x}$  in terms of effective and ineffective variables of the  $a^{th}$  sub-problem. As mentioned earlier, the ineffective variables have no influence on the objective function values, therefore, during the solution process of each sub-problem, we assign them a conventional fixed value, e.g., the midpoint of the variable range (in our case 0). By decomposability, the solutions of the sub-problems are independent of each other, so sub-problems can be solved in any order. We first discuss the matter of decision making and then finding the Pareto frontier for problem (1).

### 3.2.1 Decision making

In ANOVA-MOP, a DM provides his/her preferences as a reference point containing aspiration levels representing desirable values for objective functions. A straightforward approach to finding the corresponding preferred solution is to optimize problem (3) including all objective functions and the reference point. As discussed in [18], to accomplish cognitive tasks, one requires sufficient ability to hold information as it is processed. For problems with a large number of objective functions (e.g.,  $k = 15$ ), the DM should provide a reference point with a large number of aspiration level (i.e., 15). This can impose a cognitive load to the DM. In what follows, we discuss how sub-problems obtained by ANOVA-MOP are utilized to interact with the DM.

Considering the  $a^{\text{th}}$  sub-problem, for  $a = 1, \dots, n$ , the ASF is formulated including objective functions in the sub-problem. To do this, the indices of the objective functions in the  $a^{\text{th}}$  sub-problem are shown to the DM. Then, (s)he provides  $\bar{\mathbf{z}}^{a*}$  as the reference point including aspiration levels for the objective functions considered. Then, the following  $a^{\text{th}}$  sub-problem is formulated:

$$\underset{\mathbf{x}^a \in S^a}{\text{minimize}} \quad \text{ASF}(\mathbf{x}^a, \bar{\mathbf{z}}^{a*}). \quad (10)$$

problem (10) is then solved. The solution obtained in the objective space is shown to the DM. The interaction with the DM is conducted until the most preferred solution corresponding to the objective functions in the  $a^{\text{th}}$  sub-problem is obtained as in the reference point methods [3]. By solving problem (3) for decision variables and objective functions grouped in each sub-problem, the corresponding preferred values for the decision variables and objective functions are obtained. The following theorem shows that the solutions obtained by solving the sub-problems form a preferred solution for the DM corresponding to problem (1).

**Theorem 3.4.** *Suppose the incidence matrix  $M$  can be decomposed into  $n$  sub-problems and  $\bar{\mathbf{z}}^{a*}$ , for  $a = 1, \dots, n$ , is a reference point given by the DM for the  $a^{\text{th}}$  sub-problem. We denote the optimal solution of the  $a^{\text{th}}$  sub-problem (10) by  $\bar{\mathbf{x}}^{a*}$ , for  $a = 1, \dots, n$ . Then*

$$\mathbf{x}^* = (\mathbf{x}^{1*}, \dots, \mathbf{x}^{n*}) \in \underset{\mathbf{x} \in S}{\text{argmin}} \text{ASF}(\mathbf{x}, \bar{\mathbf{z}}^*), \quad (11)$$

where  $\bar{\mathbf{z}}^* = (\bar{\mathbf{z}}^{1*}, \dots, \bar{\mathbf{z}}^{n*})$ .

*Proof.* We show that

$$\forall \mathbf{x} \in S, \text{ASF}(\mathbf{x}^*, \bar{\mathbf{z}}^*) \leq \text{ASF}(\mathbf{x}, \bar{\mathbf{z}}^*). \quad (12)$$

Suppose  $\text{ASF}(\mathbf{x}^*, \bar{\mathbf{z}}^*) = (w_h(f_h(\mathbf{x}^*) - \bar{z}_h^*))$ . Then, there exists an index  $e$  such that  $h \in O_e$ . The optimal solution of the  $e^{\text{th}}$  sub-problem is  $\mathbf{x}^{e*}$ . This means

$$(w_h(f_h(\mathbf{x}^*) - \bar{z}_h^*)) = \max_{j \in O_e} (w_j(f_j(\mathbf{x}^{e*}, \underline{\mathbf{x}}^e) - \bar{z}_j^{e*})). \quad (13)$$

For all  $\mathbf{x}^e \in S^e$ , we have

$$\max_{j \in O_e} (w_j(f_j(\mathbf{x}^{e*}, \underline{\mathbf{x}}^e) - \bar{z}_j^{e*})) \leq \max_{j \in O_e} (w_j(f_j(\mathbf{x}^e, \underline{\mathbf{x}}^e) - \bar{z}_j^{e*})). \quad (14)$$

On the other hand

$$\forall \mathbf{x}^e \in S^e, \max_{j \in O_e} (w_j (f_j(\mathbf{x}^e, \underline{\mathbf{x}}^e) - \bar{z}_j^{e*})) \leq \max_{a=1, \dots, n} (\max_{b \in O_a} (w_b (f_b(\mathbf{x}^a, \underline{\mathbf{x}}^a) - \bar{z}_b^a))). \quad (15)$$

Since,  $\max_{a=1, \dots, n} (\max_{b \in O_a} (w_b (f_b(\mathbf{x}^a, \underline{\mathbf{x}}^a) - \bar{z}_b^a))) = \text{ASF}(\mathbf{x}, \bar{\mathbf{z}}^*)$ , inequality (12) is resulted.  $\square$

The advantage of this approach is that, for each sub-problem, a low dimensional single objective optimization problem is solved. Moreover, the DM is only required to provide preferences for few objective functions in each interaction. As mentioned earlier, when applying ANOVA-MOP to solve computationally expensive problems, we use surrogate functions. Therefore, the DM does not need to wait for a long time to see the solutions corresponding to his/her preferences.

### 3.2.2 Representation of the Pareto frontier

In what follows, we discuss how to find the set of Pareto optimal solutions of problem (1) by solving sub-problems. In ANOVA-MOP, Pareto optimal solutions obtained by solving the sub-problems are composed to form the set of Pareto optimal solutions of problem (1). To do this, each individual sub-problem is solved including the effective variables and objective functions in the sub-problem. The following theorem shows how the set of Pareto optimal solutions of problem (1) is formed.

**Theorem 3.5.** *Let denote the set of Pareto optimal solutions in the decision and objective spaces of the  $a^{\text{th}}$  sub-problem, for  $a = 1, \dots, n$ , by  $\bar{X}^a$  and  $\bar{F}^a$ , respectively. Then,*

$$\bar{X} = \{\bar{\mathbf{x}} \in \mathbb{R}^d : \bar{\mathbf{x}} \in \bar{X}^1 \times \dots \times \bar{X}^n\}, \quad (16a)$$

$$\bar{F} = \{\bar{\mathbf{f}} \in \mathbb{R}^k : \bar{\mathbf{f}} \in \bar{F}^1 \times \dots \times \bar{F}^n\}, \quad (16b)$$

are the sets of Pareto optimal solutions of problem (1) in the decision and objective spaces, respectively, where  $\times$  stands for the Cartesian product.

*Proof.* Suppose  $\bar{\mathbf{x}}^* \in \bar{X}$  ( $\mathbf{f}(\bar{\mathbf{x}}^*) \in \bar{F}$ ) is not a Pareto optimal solution of problem (1). Then, there exists a solution  $\mathbf{x} \in S$  such that for all  $i \in O$ ,  $f_i(\mathbf{x}) \leq f_i(\bar{\mathbf{x}}^*)$ , and for some  $j \in O$ ,  $f_j(\mathbf{x}) < f_j(\bar{\mathbf{x}}^*)$ . Let consider an arbitrary index  $i \in O$ . Since problem (1) is decomposable, there is an index  $a \in 1, \dots, n$ , such that  $i \in O_a$ . Let  $\mathbf{x} = (\mathbf{x}^a, \underline{\mathbf{x}}^a)$  and  $\bar{\mathbf{x}}^* = (\bar{\mathbf{x}}^{*a}, \bar{\underline{\mathbf{x}}}^{*a})$  be representation of  $\mathbf{x}$  and  $\bar{\mathbf{x}}^*$  in terms of effective and ineffective variables of the  $a^{\text{th}}$  sub-problem. We have,  $\bar{f}_i(\mathbf{x}^a) := f_i(\mathbf{x}^a, \underline{\mathbf{x}}^a)$  and  $\bar{f}_i(\bar{\mathbf{x}}^{*a}) := f_i(\bar{\mathbf{x}}^{*a}, \bar{\underline{\mathbf{x}}}^{*a})$ . Thus,  $\bar{f}_i(\mathbf{x}) \leq \bar{f}_i(\bar{\mathbf{x}}^*)$ . This is a contradiction with Pareto optimality of  $\bar{\mathbf{x}}^{*a}$  for the  $a^{\text{th}}$  sub-problem.  $\square$

One should note that  $\text{card}(\bar{F}) = \prod_{a=1}^n \text{card}(\bar{F}^a)$  ( $\text{card}(\bar{X}) = \prod_{a=1}^n \text{card}(\bar{X}^a)$ ) where  $\text{card}(\cdot)$  is the cardinality. Cases may arise that one may wish to have more solutions in some particular regions in the objective space. Solving sub-problems separately brings an advantage to generate more solutions in those regions. If surrogates emulate the original computationally expensive functions precisely, the set  $\bar{X}$  ( $\bar{F}$ ) can be considered as the Pareto optimal set (the Pareto frontier) for the original, computationally expensive MOP. Otherwise, they represent an approximation of the Pareto optimal set (the Pareto frontier) for the original problem.

## 4 Approximately Decomposable Problems

In Section 3, we considered cases in which problem (1) or equivalently the incidence matrix  $M$  is decomposable. However, in a typical case, there is no reason for which a particular entry of the matrix  $M$  should be zero. Indeed, usually the matrix  $M$  is fully non-zero, or in some fortunate case, the matrix could have a small number of zero entries. In other words, problem (1) cannot be decomposed as it is, i.e., the functional dependence between any objective function and decision variable may not completely absent. Nevertheless, as mentioned earlier, 80% of the effects on an objective function value are due to 20% of the decision variables. We propose that by identifying ineffective variables and ignoring them, it should be possible in many cases to obtain a sparse enough approximated matrix  $M$ . Consequently, we are led to defining an auxiliary decomposable MOP, with corresponding simplified solutions, acting as approximations for the solutions of problem (1). We will call the set of these solutions in the decision and objective space as the *synthetic Pareto optimal set* and *synthetic Pareto frontier*, respectively. More precisely,

**Definition 4.1.** Following [28], for a tolerance  $\epsilon > 0$ , we say that a solution  $\mathbf{x}^1 = (x_1^1, \dots, x_d^1) \in S$   $\epsilon$ -dominates a solution  $\mathbf{x}^2 = (x_1^2, \dots, x_d^2) \in S$  if and only if for all  $i \in O$ ,  $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2) + \epsilon$ , and there exists an index  $j \in O$  such that  $f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2) + \epsilon$ .

**Definition 4.2.** Suppose  $X \subset S$  is the Pareto optimal set in the decision space of problem (1). For  $\epsilon > 0$ , we say that  $X_s \subset S$  is a synthetic Pareto optimal set if and only if for all  $\mathbf{x} \in X$ , there exist  $\mathbf{x}_s \in X_s$  such that  $\mathbf{x}_s$   $\epsilon$ -dominates  $\mathbf{x}$ . The image of  $X_s$  in the objective space of problem (1), i.e.,

$$F_s := \text{Im}_f(X_s) = \left\{ \mathbf{f}(\mathbf{x}_s) \in \mathbb{R}^k \mid \mathbf{x}_s \in X_s \right\}. \quad (17)$$

is called a synthetic Pareto frontier.

For instance, in bi-objective problems, the synthetic Pareto frontier is a single solution, while the Pareto frontier of the original problem is a full curve and not a single point.

This is well illustrated in Figure 3 (a). The Pareto frontier, in this case, is the orange line and not a single point. We have highlighted in red a special point  $\bar{P}$  of the Pareto frontier which is a candidate to become a synthetic Pareto frontier. We notice that all other different points in the Pareto frontier could outperform the value of  $\bar{P}$  for at least one of the objectives, but if a point  $P$  has a better value for the  $i$ -th objective in comparison with  $\bar{P}$ , the corresponding value of  $\bar{P}$  is not far from the value of  $P$  for more than a prescribed tolerance  $2\epsilon$ . Indeed, the Pareto frontier is wholly contained in the green region, which is a pair of half-strips of width  $2\epsilon$  and with mid-line passing through  $\bar{P}$ . These half strips also have a corner in  $P - \epsilon$ .

When the functional dependence is not enough weak, the Pareto frontier can trespass the bounds of the pair of half strips as opposed to the preceding example. Therefore,  $\bar{P}$  is not a valid approximation for the whole Pareto frontier. Figure 3 (b) illustrates this situation. The approximation could become valid if  $\epsilon$  is enlarged. In the next subsection, we define in some more details these questions of approximate decomposability for larger numbers of objective functions.

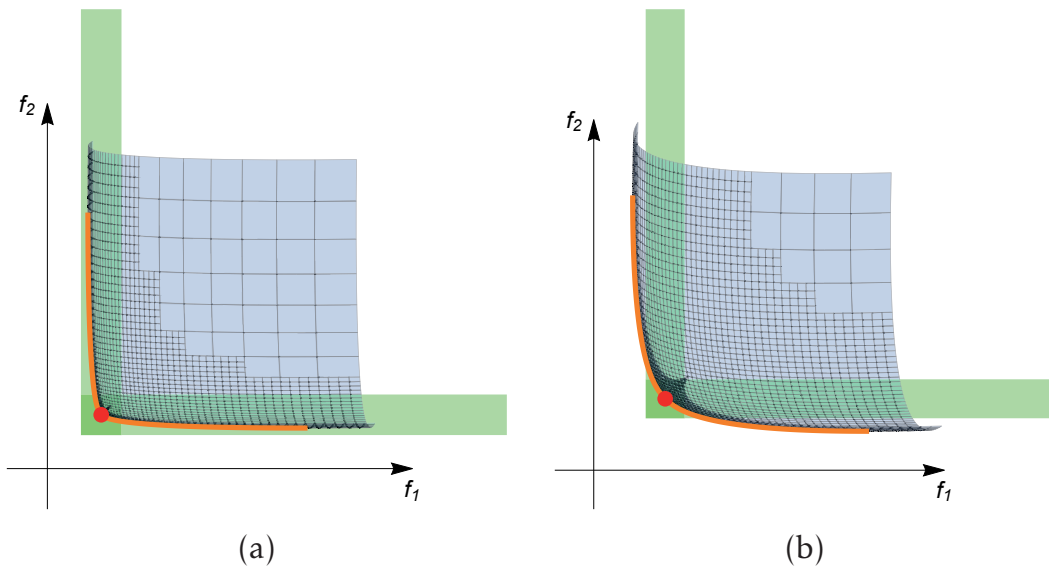


Figure 3: (a) the Pareto frontier (the orange line) and a synthetic Pareto frontier (the red point) for a hypothetical bi-objective optimization problem. The green strips represent the tolerance zone of width  $2\epsilon$  where the Pareto frontier should be located for eliciting the synthetic Pareto frontier to be an acceptable approximation. (b) A hypothetical bi-objective optimization problem where the functional dependence is not enough weak to be neglected and the Pareto front (orange line) violates the tolerance zone (green strips). Pareto sets are traced by singular continuation [19, 21]. (Color online)

## 4.1 Approximation quality

Suppose problem (1) is not decomposable. In this problem we have  $S = [l, u]^d$  and

$$\mathbf{f} : S \rightarrow \mathbb{R}^k. \quad (18)$$

Let  $1 \leq d_1, d_2 < d$  and  $1 \leq k_1, k_2 < k$ , with  $d_1 + d_2 = d$  and  $k_1 + k_2 = k$ , and fix  $\underline{\mathbf{x}}^{1\star} \in S^1 := [l, u]^{d_1}$  and  $\underline{\mathbf{x}}^{2\star} \in S^2 := [l, u]^{d_2}$  arbitrarily. Let us call  $\pi_1 : \mathbb{R}^k \rightarrow \mathbb{R}^{k_1}$  and  $\pi_2 : \mathbb{R}^k \rightarrow \mathbb{R}^{k_2}$  the two canonical projections, i.e.,  $\forall \mathbf{y} \in \mathbb{R}^k, \mathbf{y} = (\pi_1(\mathbf{y}), \pi_2(\mathbf{y}))$ .

We define two auxiliary sub-problems  $G_1$  and  $G_2$  including the following vector of objective functions, respectively:

$$\begin{aligned} \mathbf{g}_1^{\underline{\mathbf{x}}^{2\star}} : S^1 &\rightarrow \mathbb{R}^{k_1}, \\ \mathbf{x}_1 &\mapsto \pi_1(\mathbf{f}(\mathbf{x}_1, \underline{\mathbf{x}}^{2\star})), \end{aligned} \quad (19a)$$

$$\begin{aligned} \mathbf{g}_2^{\underline{\mathbf{x}}^{1\star}} : S^2 &\rightarrow \mathbb{R}^{k_2}, \\ \mathbf{x}_2 &\mapsto \pi_2(\mathbf{f}(\underline{\mathbf{x}}^{1\star}, \mathbf{x}_2)). \end{aligned} \quad (19b)$$

By composing sub-problems  $G_1$  and  $G_2$  as a new problem  $G$ , we have:

$$\begin{aligned} \mathbf{g} : S = S^1 \times S^2 &\rightarrow \mathbb{R}^k, \\ (\mathbf{x}_1, \mathbf{x}_2) &\mapsto (\mathbf{g}_1^{\underline{\mathbf{x}}^{2\star}}(\mathbf{x}_1), \mathbf{g}_2^{\underline{\mathbf{x}}^{1\star}}(\mathbf{x}_2)). \end{aligned} \quad (20)$$

Trivially one sees that  $\mathbf{g}(\underline{\mathbf{x}}^{1\star}, \underline{\mathbf{x}}^{2\star}) = \mathbf{f}(\underline{\mathbf{x}}^{1\star}, \underline{\mathbf{x}}^{2\star})$ , however it is interesting to investigate the situation in which  $\mathbf{g}(\mathbf{x}_1, \mathbf{x}_2) \simeq \mathbf{f}(\mathbf{x}_1, \mathbf{x}_2)$ , i.e., when  $\mathbf{g}$  is a good approximation of problem (1). To be more precise we say that:

**Definition 4.3.** *problem (1) defined by  $\mathbf{f} : S \rightarrow \mathbb{R}^k$  is called  $\epsilon$ -decomposable if there exist two auxiliary sub-problems  $G_1$  and  $G_2$  defined as above such that*

$$\|\mathbf{f}(\mathbf{x}_1, \mathbf{x}_2) - \mathbf{g}(\mathbf{x}_1, \mathbf{x}_2)\| < \epsilon, \quad \forall \mathbf{x}_1 \in S^1, \mathbf{x}_2 \in S^2. \quad (21)$$

**Remark 4.4.** *The  $\epsilon$ -decomposability can be rewritten only in terms of  $\mathbf{f}$  and of the partitioned domain and co-domain:*

$$\left\| \pi_1(\mathbf{f}(\mathbf{x}_1, \mathbf{x}_2)) - \pi_1(\mathbf{f}(\mathbf{x}_1, \bar{\mathbf{x}}_2)) \right\| < \epsilon, \quad \forall \mathbf{x}_1 \in S^1, \mathbf{x}_2, \bar{\mathbf{x}}_2 \in S^2, \quad (22a)$$

$$\left\| \pi_2(\mathbf{f}(\mathbf{x}_1, \mathbf{x}_2)) - \pi_2(\mathbf{f}(\bar{\mathbf{x}}_1, \mathbf{x}_2)) \right\| < \epsilon, \quad \forall \mathbf{x}_1, \bar{\mathbf{x}}_1 \in S^1, \mathbf{x}_2 \in S^2, \quad (22b)$$

*i.e., we can say that  $\pi_1 \circ \mathbf{f}$  and  $\pi_2 \circ \mathbf{f}$  are  $\epsilon$ -insensitive on  $\mathbf{x}_2$  and  $\mathbf{x}_1$ , respectively, or that the functional dependence is  $\epsilon$ -weak.*

In what follows, without loss of generality, we assume that  $\pi_1(\mathbf{f}(\mathbf{x})) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_{k_1}(\mathbf{x}))$  and  $\pi_2(\mathbf{f}(\mathbf{x})) = (f_{k_1+1}(\mathbf{x}), f_{k_1+2}(\mathbf{x}), \dots, f_k(\mathbf{x}))$ . Thus, the sets of indices of objective functions in  $G_1$  and  $G_2$  are  $O_1 = \{1, \dots, k_1\}$  and  $O_2 = \{k_1 + 1, \dots, k\}$ . The  $i^{\text{th}}$  objective function in  $G_1$  and  $G_2$  is denoted by  $g_{i,1}^{\underline{\mathbf{x}}^{2\star}}$  and  $g_{i,2}^{\underline{\mathbf{x}}^{1\star}}$ , respectively.

**Theorem 4.5.** Let problem (1),  $G_1$  and  $G_2$  as above admitting a Pareto frontier. Let denoting the Pareto optimal sets of  $G_1$  and  $G_2$  by  $\bar{X}_1$  and  $\bar{X}_2$ , respectively and the Cartesian product of  $\bar{X}_1$  and  $\bar{X}_2$  by

$$\bar{X} := \left\{ (\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} = \mathbb{R}^d \mid \mathbf{x}_1 \in \bar{X}_1, \mathbf{x}_2 \in \bar{X}_2 \right\}, \quad (23)$$

and the image of  $\bar{X}$  by

$$\bar{F} := \text{Im}_{\mathbf{f}}(\bar{X}) = \left\{ \mathbf{f}(\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{R}^k \mid \mathbf{x}_1 \in \bar{X}_1, \mathbf{x}_2 \in \bar{X}_2 \right\}. \quad (24)$$

Then, a solution in  $\bar{X}$  ( $\bar{F}$ )  $\bar{\epsilon}$ -dominates a solution in  $X$  ( $F$ ) where  $\bar{\epsilon} = 2\epsilon$ , i.e., the sets  $\bar{X}$  and  $\bar{F}$  are the synthetic Pareto optimal set and the synthetic Pareto frontier of problem (1) with tolerance  $\bar{\epsilon}$ , respectively. Equivalently, the Pareto frontier of problem (1) is completely contained in the union of two “tubular strips” defined as follows: let  $\epsilon_1 = (2\epsilon, \dots, 2\epsilon) \in \mathbb{R}^{k_1}$  and  $\epsilon_2 = (2\epsilon, \dots, 2\epsilon) \in \mathbb{R}^{k_2}$ , then

$$T_1 := \left\{ (\mathbf{y}^1, \mathbf{y}^2) \in \mathbb{R}^k \mid \exists (\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^2) \in \bar{X}, \quad \pi_1(\mathbf{f}(\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^2)) - \epsilon_1 < \mathbf{y}^1, \quad \|\mathbf{y}^2 - \pi_2(\mathbf{f}(\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^2))\| < 2\epsilon \right\}, \quad (25a)$$

$$T_2 := \left\{ (\mathbf{y}^1, \mathbf{y}^2) \in \mathbb{R}^k \mid \exists (\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^2) \in \bar{X} \quad \|\mathbf{y}^1 - \pi_1(\mathbf{f}(\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^2))\| < 2\epsilon, \quad \pi_2(\mathbf{f}(\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^2)) - \epsilon_2 < \mathbf{y}^2 \right\}. \quad (25b)$$

In other words, if a vector value in the Pareto frontier of problem (1) outperforms all the values in the synthetic Pareto frontier, the performance is not better than  $2\epsilon$ .

*Proof.* Suppose  $\mathbf{x}^* \in X$  is an arbitrary Pareto optimal solution in the decision space of problem (1),  $\mathbf{f}(\mathbf{x}^*) \in F$  is the corresponding Pareto optimal solution in the objective space and  $\mathbf{x}^* = (\mathbf{x}^{*1}, \mathbf{x}^{*2})$  is representation of  $\mathbf{x}^*$  in terms of effective variables in sub-problems  $G_1$  and  $G_2$ . We show that  $\mathbf{f}(\mathbf{x}^*) \in T_1$  or  $\mathbf{f}(\mathbf{x}^*) \in T_2$  through the following steps.

1. According to equations (19a) and (19b), we have  $\pi_1(\mathbf{f}(\mathbf{x}^{*1}, \underline{\mathbf{x}}^{2*})) = \mathbf{g}_1^{\underline{\mathbf{x}}^{2*}}(\mathbf{x}^{*1})$  and  $\pi_2(\mathbf{f}(\underline{\mathbf{x}}^{1*}, \mathbf{x}^{*2})) = \mathbf{g}_2^{\underline{\mathbf{x}}^{1*}}(\mathbf{x}^{*2})$ . Since  $\mathbf{g}_1^{\underline{\mathbf{x}}^{2*}}(\mathbf{x}^{*1}) \in \mathbb{R}^{k_1}$ , there exists a Pareto optimal solution  $\tilde{\mathbf{x}}_1 \in \bar{X}_1$  such that  $\mathbf{g}_1^{\underline{\mathbf{x}}^{2*}}(\tilde{\mathbf{x}}_1)$  dominates  $\mathbf{g}_1^{\underline{\mathbf{x}}^{2*}}(\mathbf{x}^{*1})$ . Analogously, there exists a Pareto optimal solution  $\tilde{\mathbf{x}}_2 \in \bar{X}_2$  such that  $\mathbf{g}_2^{\underline{\mathbf{x}}^{1*}}(\tilde{\mathbf{x}}_2)$  dominates  $\mathbf{g}_2^{\underline{\mathbf{x}}^{1*}}(\mathbf{x}^{*2})$ .
2. Without loss of generality, suppose there exist  $I_{\leq} \subset O_1$  and  $I_{>} \subset O_1$  such that

$$\forall i \in I_{\leq} \quad f_i(\mathbf{x}^{*1}, \mathbf{x}^{*2}) \leq f_i(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2), \quad (26a)$$

$$\forall i \in I_{>} \quad f_i(\mathbf{x}^{*1}, \mathbf{x}^{*2}) > f_i(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2). \quad (26b)$$

We then have

$$\begin{aligned} \forall i \in I_{\leq}, \quad f_i(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2) - f_i(\mathbf{x}^{*1}, \mathbf{x}^{*2}) &= f_i(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2) - f_i(\tilde{\mathbf{x}}_1, \underline{\mathbf{x}}^{2*}) \\ &\quad + f_i(\tilde{\mathbf{x}}_1, \underline{\mathbf{x}}^{2*}) - f_i(\mathbf{x}^{*1}, \underline{\mathbf{x}}^{2*}) \\ &\quad + f_i(\mathbf{x}^{*1}, \underline{\mathbf{x}}^{2*}) - f_i(\mathbf{x}^{*1}, \mathbf{x}^{*2}). \end{aligned} \quad (27)$$



Since  $\mathbf{f}$  is  $\epsilon$ -decomposable and  $f_i(\widetilde{\mathbf{x}}_1, \underline{\mathbf{x}}^{2*}) \leq f_i(\mathbf{x}^{*1}, \underline{\mathbf{x}}^{2*})$ , from (27), we have

$$\forall i \in I_{\leq}, \quad f_i(\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2) - f_i(\mathbf{x}^{*1}, \mathbf{x}^{*2}) < 2\epsilon. \quad (28)$$

From equations (26b) and (28), we have

$$\forall i \in O_1, \quad f_i(\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2) - 2\epsilon < f_i(\mathbf{x}^{*1}, \mathbf{x}^{*2}), \quad (29a)$$

$$\pi_1(\mathbf{f}(\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2)) - \epsilon_1 < \pi_1(\mathbf{f}(\mathbf{x}^{*1}, \mathbf{x}^{*2})). \quad (29b)$$

3. We then show that  $\|\pi_2(\mathbf{f}(\mathbf{x}^{*1}, \mathbf{x}^{*2})) - \pi_2(\mathbf{f}(\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2))\| < 2\epsilon$ . Since  $\mathbf{f}$  is  $\epsilon$ -decomposable and  $\pi_2(\mathbf{f}(\underline{\mathbf{x}}^{1*}, \widetilde{\mathbf{x}}_2)) \leq \pi_2(\mathbf{f}(\underline{\mathbf{x}}^{1*}, \mathbf{x}^{*2}))$ , we have

$$\begin{aligned} & \|\pi_2(\mathbf{f}(\mathbf{x}^{*1}, \mathbf{x}^{*2})) - \pi_2(\mathbf{f}(\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2))\| = \\ & \|\pi_2(\mathbf{f}(\mathbf{x}^{*1}, \mathbf{x}^{*2})) - \pi_2(\mathbf{f}(\underline{\mathbf{x}}^{1*}, \mathbf{x}^{*2})) + \pi_2(\mathbf{f}(\underline{\mathbf{x}}^{1*}, \mathbf{x}^{*2})) - \pi_2(\mathbf{f}(\underline{\mathbf{x}}^{1*}, \widetilde{\mathbf{x}}_2)) \\ & + \pi_2(\mathbf{f}(\underline{\mathbf{x}}^{1*}, \widetilde{\mathbf{x}}_2)) - \pi_2(\mathbf{f}(\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2))\| < \\ & \|\pi_2(\mathbf{f}(\mathbf{x}^{*1}, \mathbf{x}^{*2})) - \pi_2(\mathbf{f}(\underline{\mathbf{x}}^{1*}, \mathbf{x}^{*2}))\| + \|\pi_2(\mathbf{f}(\underline{\mathbf{x}}^{1*}, \widetilde{\mathbf{x}}_2)) - \pi_2(\mathbf{f}(\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2))\| < 2\epsilon. \end{aligned} \quad (30)$$

From equations (29b) and (30), we result that  $\mathbf{f}(\mathbf{x}^*) \in T_1$ . Analogously, if one assumes that there exist  $I_{\leq} \subset O_2$  and  $I_{>} \subset O_2$  such that

$$\forall i \in I_{\leq} \quad f_i(\mathbf{x}^{*1}, \mathbf{x}^{*2}) \leq f_i(\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2), \quad (31a)$$

$$\forall i \in I_{>} \quad f_i(\mathbf{x}^{*1}, \mathbf{x}^{*2}) > f_i(\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2). \quad (31b)$$

Then, one can show that  $\mathbf{f}(\mathbf{x}^*) \in T_2$ .  $\square$

**Theorem 4.6.** Let  $\mathbf{z}^{G,ideal} = (z_1^{G,ideal}, \dots, z_k^{G,ideal})$  and  $\mathbf{z}^{G,nadir} = (z_1^{G,nadir}, \dots, z_k^{G,nadir})$  be the ideal and nadir vectors of problem  $G$ , respectively. Then,

$$z_i^{ideal} \leq z_i^{G,ideal}, \quad \forall i \in O, \quad (32a)$$

$$z_i^{G,nadir} \leq z_i^{nadir}, \quad \forall i \in O. \quad (32b)$$

*Proof.* For an arbitrary function  $f_i$ , for  $i \in O$ , suppose  $\widetilde{\mathbf{x}}^{i,min} \in \operatorname{argmin}_{\mathbf{x} \in S} \{f_i(\mathbf{x})\}$ . There is a sub-problem  $G_a$  such that  $i \in O_a$ . Without loss of generality suppose  $a = 1$ . Let  $\widetilde{\mathbf{x}}^{i,min} = (\widetilde{\mathbf{x}}^{1,i,min}, \underline{\widetilde{\mathbf{x}}}^{1,i,min})$ . We have

$$f_i(\widetilde{\mathbf{x}}^{1,i,min}, \underline{\widetilde{\mathbf{x}}}^{1,i,min}) \leq f_i(\mathbf{x}^1, \underline{\mathbf{x}}^1), \quad \forall \mathbf{x}^1 \in S^1, \forall \underline{\mathbf{x}}^1 \in \underline{S}^1. \quad (33)$$

Suppose  $\mathbf{x}^{1,i,min} \in \operatorname{argmin}_{\mathbf{x}^1 \in S^1} \{g_{1,i}^{\mathbf{x}^{2*}}(\mathbf{x}^1)\}$ . Then,

$$g_{1,i}^{\mathbf{x}^{2*}}(\mathbf{x}^{1,i,min}) \leq g_{1,i}^{\mathbf{x}^{2*}}(\mathbf{x}^1) \quad \forall \mathbf{x}^1 \in S^1. \quad (34)$$

Since  $g_{1,i}^{\mathbf{x}^{2*}}(\mathbf{x}^{1,i,min}) = f_i(\mathbf{x}^{1,i,min}, \underline{\mathbf{x}}^{2*})$ , from inequality (33) we have

$$f_i(\widetilde{\mathbf{x}}^{i,min}) \leq f_i(\mathbf{x}^{1,i,min}, \underline{\mathbf{x}}^{2*}), \quad (35)$$

and inequality (32a) is resulted. Analogously, inequality (32b) can be proved.  $\square$

**Corollary 4.7.** Let  $k = 2$  and problem (1) be  $\epsilon$ -decomposable, with  $S = S^1 \times S^2$ . Then, by fixing  $\underline{x}^{2*} \in S^2$  and  $\underline{x}^{1*} \in S^1$  and setting  $\bar{x}_1 \in \operatorname{argmin}_{x_1 \in S^1} \{f(x_1, \underline{x}^{2*})\}$  and  $\bar{x}_2 \in \operatorname{argmin}_{x_2 \in S^2} \{f(\underline{x}^{1*}, x_2)\}$ , we have that  $\bar{x} := (\bar{x}_1, \bar{x}_2)$  is a synthetic Pareto set for problem (1) with tolerance  $2\epsilon$ .

**Remark 4.8.** The solutions  $\bar{x}_1$  and  $\bar{x}_2$  may not be uniquely defined. When there are multiple solutions realizing the same absolute optimum, it is not restrictive to choose any of these solutions.

One should note that the synthetic Pareto frontier is an approximation of the Pareto frontier of the original, computationally expensive MOP. Clearly, the condition of  $\epsilon$ -decomposability can be known only in cases when the functions at hand have an explicit formulation and this formulation is amenable to analysis. This is not possible clearly when the functions at hand are black boxes. However, a method called *functional ANOVA decomposition*, where ANOVA stands for analysis of variance, is designed specifically for estimating the extent of functional dependence among function values and variables. In this paper, we apply the functional ANOVA decomposition to detect effective and ineffective variables. Then, by ignoring ineffective variables, sub-problems  $G_1$  and  $G_2$  are formulated by which problem (1) is decomposed approximately.

## 4.2 Functional ANOVA decomposition in MOPs

In this subsection, we discuss how to introduce problem  $G$  based on problem (1). To do so, we introduce a reduced incidence matrix  $RM$  based on  $M$ . The reduced incidence matrix  $RM$  is decomposable and based on  $RM$ , we formulate problem  $G$  which is decomposable. We then decompose problem  $G$  into a limited number of sub-problems in lieu of problem (1). Once problem  $G$  is decomposed, the solution process in terms of decision making or representing the Pareto frontier is similar to those discussed in Subsection 3.2. One should note that in this case, the solutions of problem  $G$  are approximated solutions for problem (1). In this way, we say that problem (1) is approximated by problem  $G$ .

For each objective function in problem (1), it is important to realize which of the decision variables have more effect on the objective function value and then to quantify that effectiveness. According to [29], the effective variables in one region of the decision space may not be the same as those in another, and thus, global sensitivity analysis has been developed. Similarly, decision variables may cooperate to provide interaction effects, in which the effectiveness of a subset of decision variables may not lead to the sum of individual decision variable effectiveness. Sobol and total sensitivity indices, described in Section 2 have been developed to quantify decision variable effectiveness, clarifying these issues.

The functional ANOVA and variance decompositions [22] is the most popular approach to conducting global sensitivity analysis and calculating Sobol and total sensitivity indices [25]. According to [25], generally, two classes of methods have been developed to conduct sensitivity analysis, i.e., methods that do not utilize surrogates and methods that rely on surrogates. Examples of the first class are Monte Carlo methods [26, 30, 31], and Extended FAST [32]. The second class includes methods based

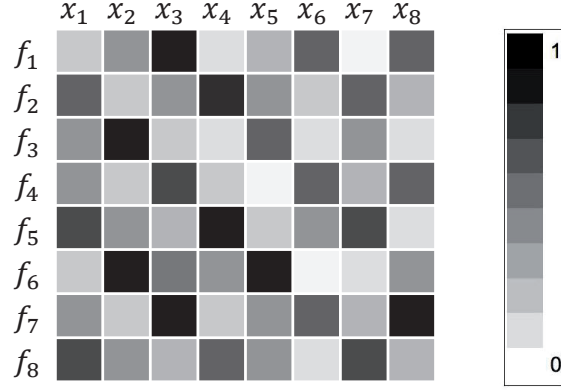


Figure 4: An example of an effective total sensitivity indices matrix (SM). The bar shows the range of the total sensitivity indices.

on Gaussian process [29, 33, 34], smoothing spline [35], polynomial chaos expansion [25, 36], and local polynomial regression [37].

As mentioned in [25], methods in the first class typically require more function evaluations than those in the other class. However, methods developed in the first class generally depend on minimal assumptions about a given function, while methods in the second class rely more on limiting assumptions (e.g., continuity and differentiability). Therefore, the first class methods are more desirable for computationally inexpensive, discontinuous, and non-smooth functions, whereas the second class methods can be applied for computationally expensive and/or smooth functions. Obviously, global sensitivity analysis can also be conducted by a hybrid method, in which a surrogate is first built and then a non-surrogate-based method is employed to estimate Sobol and total sensitivity indices based on evaluations of the surrogate.

The incidence matrix  $M$  captures the dependencies between objective functions and decision variables in problem (1). However, by applying sensitivity analysis and identifying effective and ineffective variables, we can form a reduced incidence matrix  $RM$  which is decomposable and we discuss it in what follows.

As far as computationally expensive functions are considered, in ANOVA-MOP, first, for each individual objective function, a surrogate function by an appropriate method is built where the surrogate function is globally accurate over the entire decision space. Then, by applying a suitable method, the total sensitivity indices are estimated relying on the surrogate function values. For computationally inexpensive functions, an appropriate method can be applied directly to the functions to estimate the total sensitivity indices. We denote the estimated total sensitivity index for the  $j^{th}$  decision variable in the  $i^{th}$  objective function by  $l_{ij}^T$ . Then, we define the effective total sensitivity indices matrix (SM) as

$$SM = [l_{ij}^T], i \in O, j \in D. \quad (36)$$

Figure 4 shows  $SM$  corresponding to the incidence matrix  $M$  in Figure 1. Next, we discuss how the estimated total sensitivity indices in  $SM$  can assist to form a reduced incidence matrix  $RM$  and to build problem  $G$  based on  $RM$ .

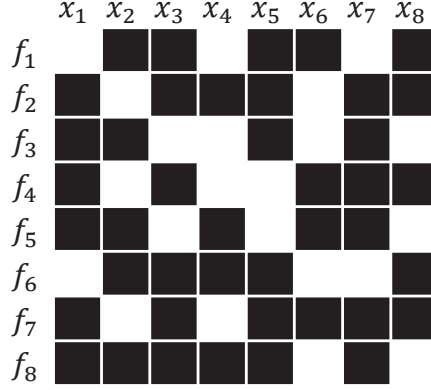


Figure 5: A reduced incidence matrix  $RM$  corresponding to a threshold considered.

### 4.3 Sparsity of a reduced incidence matrix $RM$

In this subsection, we discuss the application of estimated total sensitivity indices in approximating problem (1) and introducing problem  $G$ . The idea in ANOVA-MOP is to form a reduced incidence matrix  $RM$  which is sparse enough and decomposable. The sparsity of a reduced incidence matrix is directly governed by considering a threshold for the estimated total sensitivity indices. To do so, using a subset of entries in  $SM$ , an array of thresholds denoted by  $T$  is considered which has at most  $k \times d$  unique components. We define

$$ut = \min_{i \in O} \max_{j \in D} (l_{ij}^T), \quad (37)$$

as the upper bound for the thresholds in  $T$ . Then, all unique  $l_{ij}^T \leq ut, l_{ij}^T \neq 0$ , for  $i \in O$ , and  $j \in D$ , sorted from the smallest to the largest values are considered as components of  $T$ .

Considering the smallest threshold in  $T$ , we apply the following procedure to form a reduced incidence matrix  $RM$  which is decomposable. The decision variables for which their total sensitivity indices are greater than or equal to the threshold are treated as effective variables and the remaining decision variables as ineffective ones. Then, a reduced incidence matrix  $RM$  is formed in which in each row corresponding to each objective function, effective variables are denoted by 1's and ineffective ones by 0's. Once the reduced incidence matrix  $RM$  is formed, using the method in [38], the reduced incidence matrix  $RM$  is decomposed into a number of blocks.

One should note that any threshold greater than  $ut$  makes all decision variables in at least one of the objective functions ineffective, and as a result, all entries in the corresponding row in the reduced incidence matrix  $RM$  become zero. Such a situation is out of the scope of this paper. Figure 5 illustrates a reduced incidence matrix  $RM$ . In this case, for example for  $f_3$ ,  $l_{3j}$ , for  $j = 3, 4, 6, 8$ , is less than a threshold considered. Therefore, in the reduced incidence matrix  $RM$ , the decision variables  $x_j$ , for  $j = 3, 4, 6, 8$ , are treated as ineffective.

In the method developed in [38], for a given reduced incidence matrix  $RM$  and a fixed number of sub-problems, several decomposed problems are obtained. We con-

sider only those with the smallest complexity  $CI$ . Among them, we pick a decomposed problem in which the total number of decision variables which are common between the sub-problems is the smallest. This number of common decision variables may not be zero. In such a case, the next threshold in  $T$  is selected and the described above procedure is repeated until a reduced incidence matrix  $RM$  is obtained which is decomposable and there is no common variable between any block as shown in Figure 1(b).

The reduced incidence matrix  $RM$  shows effective and ineffective decision variables for each individual objective function. For each objective function, we neglect ineffective variables and formulate the decomposable multiobjective optimization problem  $G$ . It is notable that the larger the threshold, the more ineffective variables there are. However, cases may occur that a reduced incidence matrix  $RM$  cannot be decomposed for all threshold in  $T$  according to the criteria considered for sub-problems. Such cases are a future research direction.

As mentioned in [27], the total number of common variables is increased by the increment in the number of sub-problems and vice versa. Nevertheless, the less sub-problems is formed, the more objective functions in each sub-problem exist. In ANOVA-MOP, we fix the number of sub-problems. Then, by varying the threshold, we build problem  $G$ .

#### 4.4 $\epsilon$ -decomposability and sensitivity analysis

It is clear that if a function is decomposable, then some of the entries of the  $M$  matrix will be zero. Things are less immediate when the function is  $\epsilon$ -decomposable. Actually, in the typical case the functions at hand can be black box and therefore it will be impossible to give an estimate of the possible  $\epsilon$  ruling the decomposability of the function. The estimate of the Sobol' can be realized with a small campaign of experiments, and the computation of the indices could be a valid method for detecting ineffective variables. One has, however, to recall that the Sobol' indices are averages, therefore it is not possible to obtain an estimate of  $\epsilon$ , which is a global quantity. Nevertheless, the Sobol' indices are a good indicator of probable weak dependencies and a result of decomposability, so this justifies the fact that the procedure proposed is likely to produce good approximations with a limited computational cost of the black box functions involved in the problem at hand.

### 5 The ANOVA-MOP Method

In this section, we present the steps of the ANOVA-MOP method shown in Algorithm 1. In this method, first, by calling Algorithm 2 to be discussed in Subsection 5.1, decomposability of problem (1) is evaluated and sub-problems are obtained as the output of Algorithm 2. Then, the solution process as discussed in Subsection 3.2 is applied.

## 5.1 Decomposition Algorithm

By calling Algorithm 2, decomposability of problem (1) is assessed and sub-problems with minimum complexity are obtained. As can be seen in Algorithm 2, in the case of black-box functions, first, their surrogate functions are built. Total sensitivity indices are then calculated. In ANOVA-MOP, we fix the number of sub-problems as  $n = \lceil \frac{k}{3} \rceil$ . This means that we are interested in sub-problems with at most 3 objective functions.

---

### Algorithm 1

---

**Output:** A set of solutions or the most preferred solution for the DM

- 1: Assess decomposability of problem (1) and build sub-problems by calling Algorithm 2
  - 2: **If** the DM is involved
  - 3:     **For**  $a = 1, \dots, n$
  - 4:         **While** the DM is not satisfied
  - 5:             Ask the decision maker to provide a reference point  $\bar{z}^{a*}$
  - 6:             Solve problem (3) with the objective functions in the  $a^{th}$  sub-problem
  - 7:         **Endwhile**
  - 8:     **Endfor**
  - 9: **Else**
  - 10:     **For**  $a = 1, \dots, n$
  - 11:         Solve the  $a^{th}$  sub-problem
  - 12:     **Endfor**
  - 13:     Compose the solutions of the sub-problems as discussed in Sub-section 3.2
  - 14: **Endif**
  - 15: **If** the DM is involved, return the most preferred solution
  - 16: **Otherwise** return a set of solutions
-

---

**Algorithm 2: Decomposition**

---

**Input:** problem (1)**Output:** sub-problems

- 1: If the objective functions are black-box ones, build their corresponding surrogate functions.
  - 2: Calculate the total sensitivity indices for decision variables
  - 3: Define the number of sub-problems as  $n = \lceil \frac{k}{3} \rceil$
  - 4: Calculate  $ut$  from equation (37)
  - 5: Define the threshold array  $T$
  - 6: flag = True
  - 7: **While** flag
  - 8:     **For**  $t \in T$
  - 9:         Define decision variables with  $l_{ij}^T \geq t$  as effective ones and set others as ineffective ones.
  - 10:         Form a reduced incidence matrix  $RM$  based on the effective and ineffective variables
  - 11:         Decompose  $RM$
  - 12:         **If** there exist sub-problems with  $\widetilde{INT} = 0$
  - 13:             Return the sub-problems
  - 14:         flag = False
  - 15:     **Endif**
  - 16: **Endfor**
  - 17: **Endwhile**
-

The method developed in [38] may provide several decompositions with the same minimum complexity  $CI$  for a given reduced incidence matrix  $RM$ . In this method, we set the maximum number of common variables as  $d$ . As a result, the method can generate sub-problems with different numbers of common variables. Then, among all decompositions obtained, we search for sub-problems with the minimum number of common variables. To obtain the best decomposition, i.e., sub-problems with minimum complexity and without any common variables between them, the lowest threshold in  $T$  is considered, and the effective and ineffective variables are identified. The reduced incidence matrix  $RM$  is formed based on these variables. Then, the reduced incidence matrix  $RM$  is decomposed. Several decompositions may be formed. We denote the number of common variables in sub-problems by  $\widehat{INT}$ . Among these decompositions, if there exist sub-problems with  $\widehat{INT} = 0$ , these sub-problems are considered as the best decomposition. Otherwise, the next threshold in  $T$  is considered and the above procedure is repeated.

One should note that, if problem (1) is decomposable as discussed in Section 3, then  $0 \in T$ . This means that for the first threshold  $t = 0$ ,  $M=RM$  and the best decomposition is obtained immediately without increasing the threshold.

## 6 Numerical results

We demonstrate the performance of ANOVA-MOP both as an interactive and a non-interactive method. To do so, we first consider inherently decomposable problems and thereafter  $\epsilon$ -decomposable problems. We treat all objective functions as black-boxes to demonstrate the applicability of the method to solve computationally expensive problems. In Subsection 6.4 we consider the interactive aspects of ANOVA-MOP.

### 6.1 Experimental Setup

We consider objective functions as black-box functions and we employ the method developed in [25], henceforth called as Bayesian polynomial chaos (BPC) method to build surrogate inexpensive functions. The BPC method is based on the functional ANOVA decomposition (4) in which the interaction order  $e$  can be assigned up to  $d$ . The components in the functional ANOVA decomposition are approximated by using polynomial functions of an arbitrary degree  $p$ . Once the surrogate function is built, the method estimates the total sensitivity index for each individual decision variable along with a 95% confidence interval containing the estimated total sensitivity index. We assign  $e = d$  to capture all interactions between variables and  $p = 4$  to have an accurate enough approximation. In what follows, we assume that all necessary surrogate functions have been computed and have been substituted the original black box functions.

For decomposition we apply the method developed in [38] henceforth called the Chen method. For this method, the user can set the number of sub-problems (i.e.,  $n$ ), the maximum number of common variables between sub-problems and the minimum



number of effective variables within each sub-problem. We set these parameters to  $\lceil \frac{k}{3} \rceil$ ,  $d$  and 0, respectively.

We use NSGA-II [39] and DIRECT [40] for solving multiobjective and single objective problems respectively. Nevertheless, we recall that ANOVA-MOP can be coupled to any other optimization method, decomposition method or any other method to build surrogate functions.

## 6.2 Decomposable benchmark problems

We start by considering the following problem because it is possible to represent the synthetic Pareto optimal set and the synthetic Pareto frontier and to compare the solutions of the complete problem with the solution of the sub-problems. In what follows by  $k \times d = (k_1 \times d_1) \otimes \dots \otimes (k_n \times d_n)$ , we mean decomposing a problem with  $k$  objective functions and  $d$  decision variables to  $n$  sub-problems with  $k_i$  objective functions and  $d_i$  decision variables.

### 6.2.1 Decomposable problem $2 \times 2 = (1 \times 1) \otimes (1 \times 1)$

Let  $\ell = 2$ ,

$$\underset{\mathbf{x} \in [-\ell, \ell]^2}{\text{minimize}} \{f_1(\mathbf{x}), f_2(\mathbf{x})\} \quad \text{where} \quad \begin{cases} f_1(x_1, x_2) = \frac{1}{2}(x_1 - 1)^2, \\ f_2(x_1, x_2) := \frac{1}{2}(x_2 - 1)^2. \end{cases} \quad (38)$$

Being  $f_1$  independent from  $x_2$  and  $f_2$  independent from  $x_1$ , the effective total sensitivity indices matrix is

$$SM = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (39)$$

and the set of Pareto optimal solutions is  $\bar{X} = \{(1, 1)\}$  while the Pareto front is only  $\bar{F} = \{(0, 0)\}$ . The problem (38) can be decomposed into the following single objective optimization sub-problems

$$\underset{x_1}{\text{minimize}} \quad g_1(x_1) := f_1(x_1, 0), \quad (40) \qquad \underset{x_2}{\text{minimize}} \quad g_2(x_2) := f_2(0, x_2). \quad (41)$$

Trivially, the set of optimal solutions for problem (40) is  $\bar{X}^1 := \{1\}$  and for problem 41 is  $\bar{X}^2 := \{-1\}$ , so the Cartesian product of the solutions of the sub-problems is exactly  $\bar{X} = \bar{X}^1 \times \bar{X}^2 = \{(1, -1)\}$ , and at the same time  $\bar{F} = \bar{F}^1 \times \bar{F}^2 = \{(0, 0)\}$ , so the Theorem 3.5 is verified.

### 6.2.2 decomposable problem $10 \times 2 = (5 \times 1) \otimes (5 \times 1)$

The following problem has 10 decision variables and 2 objectives, but is only apparently more complicated than the previous. Indeed the two  $5 \times 1$  sub-problems have

only one minimum.

$$\underset{\mathbf{x} \in [-\ell, \ell]^{10}}{\text{minimize}} \{f_1(\mathbf{x}), f_2(\mathbf{x})\} \quad \text{where} \begin{cases} f_1(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^5 (x_i - 1)^2, \\ f_2(\mathbf{x}) = \frac{1}{2} \sum_{i=6}^{10} (x_i - 1)^2. \end{cases} \quad (42)$$

In problem (42), the sets of Pareto optimal solutions in the decision and objective spaces are  $\bar{X} = \{(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)\}$  and  $\bar{F} = \{(0, 0)\}$ , respectively. The effective total sensitivity indices matrix is

$$SM = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \end{bmatrix}, \quad (43)$$

which confirms that the decision variables  $x_i$  for  $i = 1, \dots, 5$ , are effective variables for  $f_1$  and  $x_i$  for  $i = 6, \dots, 10$  are effective variables for  $f_2$ . Decomposition reads as:

$$\underset{x_1, \dots, x_5 \in [-\ell, \ell]^5}{\text{minimize}} g_1(x_1, \dots, x_5) \quad (44) \qquad \underset{x_6, \dots, x_{10} \in [-\ell, \ell]^5}{\text{minimize}} g_2(x_6, \dots, x_{10}), \quad (45)$$

where  $g_1(x_1, \dots, x_5) := f_1(x_1, \dots, x_5, 0, \dots, 0)$  and  $g_2(x_6, \dots, x_{10}) := f_2(0, \dots, 0, x_6, \dots, x_{10})$ . The Pareto optimal solutions of the sub-problems combine as expected by Theorem 3.5 and the same holds for the Pareto fronts, i.e., we have  $\bar{X} = \bar{X}^1 \times \bar{X}^2$  and  $\bar{F} = \bar{F}^1 \times \bar{F}^2$ .

### 6.2.3 decomposable problem $5 \times 5 = (3 \times 3) \otimes (2 \times 2)$

Let  $\ell = 2$  and consider the following five decision variables and five objectives problem:

$$\underset{\mathbf{x} \in [-\ell, \ell]^5}{\text{minimize}} \{f_1(\mathbf{x}), \dots, f_5(\mathbf{x})\} \quad (46)$$

where

$$\begin{aligned} f_1(\mathbf{x}) &= ((x_1, x_2, x_3)^T - P_1^T)^2, & P_1 &= (1, 1, 1)^T, \\ f_2(\mathbf{x}) &= ((x_1, x_2, x_3)^T - P_2^T)^2, & P_2 &= (1, -1, -1)^T, \\ f_3(\mathbf{x}) &= ((x_1, x_2, x_3)^T - P_3^T)^2, & P_3 &= (1, 1, -1)^T, \\ f_4(\mathbf{x}) &= ((x_4, x_5)^T - P_4^T)^2, & P_4 &= (1, -1)^T, \\ f_5(\mathbf{x}) &= ((x_4, x_5)^T - P_5^T)^2, & P_5 &= (-1, 1)^T. \end{aligned}$$

The effective total sensitivity indices matrix is estimated as

$$SM = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}. \quad (47)$$

On the basis of this matrix, by applying ANOVA-MOP, we can decompose problem (46) into the following three- and bi-objective sub-problems

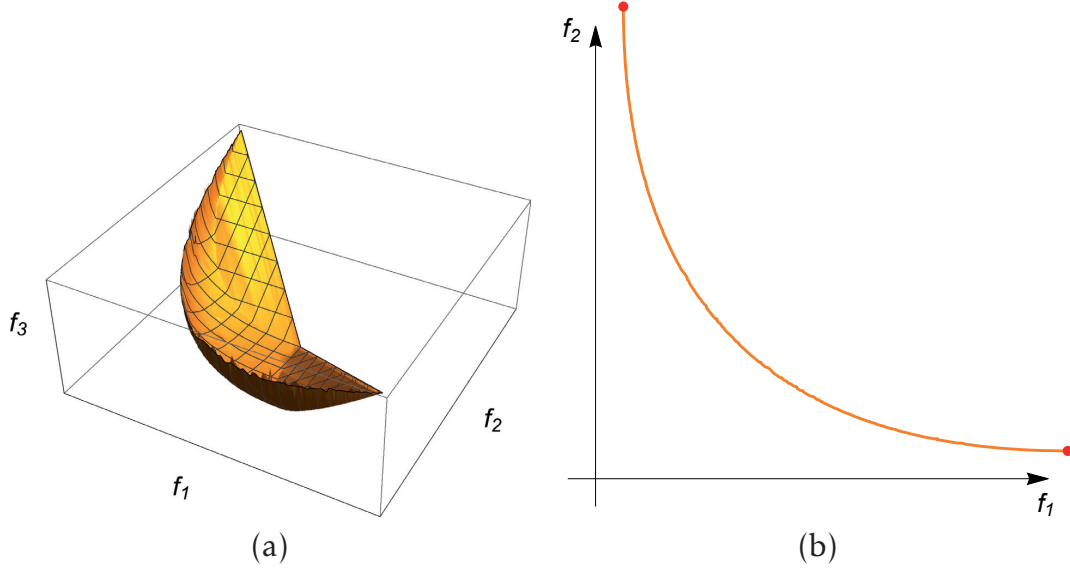


Figure 6: (a) Pareto frontier of sub-problem (48), (b) Pareto frontier of sub-problem (49). (Color online)

$$\underset{\mathbf{x}_a \in [-\ell, \ell]^3}{\text{minimize}} \{g_1(\mathbf{x}_a), g_2(\mathbf{x}_a), g_3(\mathbf{x}_a)\} \quad (48)$$

$$\underset{\mathbf{x}_b \in [-\ell, \ell]^2}{\text{minimize}} \{g_4(\mathbf{x}_b), g_5(\mathbf{x}_b)\} \quad (49)$$

where  $\mathbf{x}_a = (x_1, x_2, x_3)$  and  $\mathbf{x}_b = (x_3, x_4)$ , and  $g_1(x_1, x_2, x_3) := f_1(x_1, x_2, x_3, 0, 0, 0)$ . The remaining definitions, i.e.,  $g_2, \dots, g_5$ , are obvious. Figures 6 (a) and 6 (b) depict the Pareto frontiers of these sub-problems, respectively. According to Theorem 3.5, the set of Pareto optimal solutions of problem (46) is the Cartesian product of the sets of Pareto optimal solutions of sub-problems (48) and (49).

In other words, any of the Pareto optimal solution of problem (46) is obtained by joining a solution  $(x_1, x_2, x_3)$  of sub-problem (48) and a solution  $(x_4, x_5)$  of sub-problem (49).

### 6.3 Approximately decomposable benchmark problems

As discussed earlier, typical problems are not decomposable, but often are  $\epsilon$ -decomposable, with  $\epsilon$  small enough. The decomposition is determined via sensitivity analysis, and auxiliary sub-problems are defined, which will produce approximated solutions.

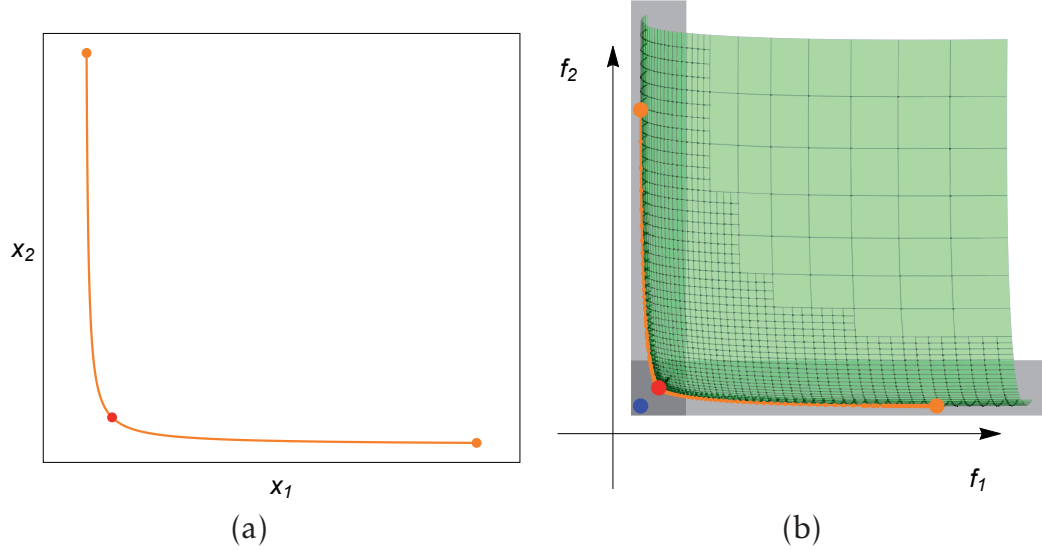


Figure 7: The orange curve in (a) and (b) shows the Pareto optimal set and the Pareto frontier of problem (50), respectively. (Color online)

### 6.3.1 $\epsilon$ -decomposable problem $2 \times 2 \simeq (1 \times 1) \otimes (1 \times 1)$

Let  $\ell = 2$ . The following parametric problem is a generalization of (38), where a weak dependence on  $x_1$  and  $x_2$  is added to  $f_2$  and  $f_1$  respectively.

$$\underset{\mathbf{x} \in [-\ell, \ell]^2}{\text{minimize}} \{f_1(\mathbf{x}), f_2(\mathbf{x})\} \quad \text{where} \quad \begin{cases} f_1(\mathbf{x}) = \frac{1}{2}((x_1 + 1)^2 + \gamma(x_2 - 1)^2), \\ f_2(\mathbf{x}) = \frac{1}{2}(\gamma(x_1 - 1)^2 + (x_2 + 1)^2), \end{cases} \quad (50)$$

with  $\gamma = 7.000 \times 10^{-3}$

As can be seen in Figure 7(a), by changing the  $\gamma$  value slightly, the set of solutions of problem (38) is turned from a single point into a full curve of solutions. The red circle in Figure 7(b) is the synthetic Pareto frontier obtained by ANOVA-MOP. In this figure, the blue circle and orange curve are solutions of problems (38) and (50), respectively. Next, we discuss how ANOVA-MOP generates the synthetic Pareto frontier for problem (50).

By means of sensitivity analysis we compute the effective total sensitivity indices matrix  $SM$  and the incidence matrix  $M$  for problem (50):

$$SM = \begin{bmatrix} 9.999 \times 10^{-1} & 1.000 \times 10^{-3} \\ 1.000 \times 10^{-3} & 9.999 \times 10^{-1} \end{bmatrix}, \quad M = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \quad (51)$$

Matrix  $M$  in (51) cannot be decomposed, nevertheless, as can be seen in matrix (51), the estimations of the total sensitivity indices of variables  $x_2$  and  $x_1$  are very small for objective functions  $f_1$  and  $f_2$ , respectively. We then form an array of thresholds denoted by  $T$ . According to equation (37),  $ut = 1.000 \times 10^{-3}$  is the upper bound for the thresholds in  $T$ . Therefore,  $T = \{1.000 \times 10^{-3}\}$ . Considering variables with estimated

total sensitivity index values less than the element in  $T$  as ineffective variables, we can form the following reduced incidence matrix:

$$RM = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (52)$$

According to Remark 3.2, matrix (52) is decomposable, and we form the following single objective optimization sub-problems:

$$\underset{\mathbf{x} \in [-\ell, \ell]^2}{\text{minimize}} \{f_1(\mathbf{x})\} \quad (53)$$

$$\text{where} \\ x_2 = 0,$$

$$\underset{\mathbf{x} \in [-\ell, \ell]^2}{\text{minimize}} \{f_2(\mathbf{x})\} \quad (54)$$

$$\text{where} \\ x_1 = 0.$$

One should note that, we only solve these sub-problems for their effective variables. In sub-problem (53), the set of optimal solutions in the decision space is,  $\bar{X}^1 := \{x_1 = -1\}$ , and for sub-problem (54),  $\bar{X}^2 := \{x_2 = -1\}$ . According to Theorem 3.5, the synthetic Pareto optimal set problem (50) in the decision space is obtained by the Cartesian product of the solutions of sub-problems (53) and (54), i.e.,  $\bar{X} = \bar{X}^1 \times \bar{X}^2$ . To obtain the synthetic Pareto frontier in the objective space, we evaluate the solutions in  $\bar{X}$  with objective functions of problem (50), i.e.,  $\bar{F} = \left\{ \left( \frac{\gamma}{2}, \frac{\gamma}{2} \right) \right\}$ .

### 6.3.2 $\epsilon$ -decomposable problem $10 \times 2 \simeq (5 \times 1) \otimes (5 \times 1)$

Now, let us consider the following parametrized bi-objective optimization problem with  $\ell = 2$

$$\underset{\mathbf{x} \in [-\ell, \ell]^{10}}{\text{minimize}} \{f_1(\mathbf{x}), f_2(\mathbf{x})\} \quad (55)$$

where

$$f_1(\mathbf{x}) = \frac{1}{8} \left( \sum_{i=1}^5 (x_i + (-1)^{i-1})^2 + \gamma \sum_{i=6}^{10} (x_i + (-1)^{i-1})^2 \right),$$

$$f_2(\mathbf{x}) = \frac{1}{8} \left( \gamma \sum_{i=1}^5 (x_i + (-1)^i)^2 + \sum_{i=6}^{10} (x_i + (-1)^i)^2 \right),$$

$$\gamma = 7.000 \times 10^{-3}.$$

This is problem (42) with the changed  $\gamma$  value. By applying ANOVA-MOP and conducting sensitivity analysis for problem (55), we have

$$SM = \begin{bmatrix} 1.999 \times 10^{-1} & 1.000 \times 10^{-3} \\ 1.999 \times 10^{-1} & 1.000 \times 10^{-3} \\ 1.999 \times 10^{-1} & 1.000 \times 10^{-3} \\ 1.999 \times 10^{-1} & 1.000 \times 10^{-3} \\ 1.999 \times 10^{-1} & 1.000 \times 10^{-3} \\ 1.000 \times 10^{-3} & 1.999 \times 10^{-1} \\ 1.000 \times 10^{-3} & 1.999 \times 10^{-1} \\ 1.000 \times 10^{-3} & 1.999 \times 10^{-1} \\ 1.000 \times 10^{-3} & 1.999 \times 10^{-1} \\ 1.000 \times 10^{-3} & 1.999 \times 10^{-1} \end{bmatrix}^T, \quad (56)$$

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (57)$$

According to Remark 3.2, matrix (57) is not decomposable. Matrix (56) shows that the estimation of the total sensitivity indices of variables  $x_i$  for  $i = 6, \dots, 10$ , for  $f_1$  and  $x_i$  for  $i = 1, \dots, 5$ , for  $f_2$  are very small. For this problem,  $ut = 1.000 \times 10^{-3}$  is the upper bound for thresholds in  $T$ , and  $T = \{1.000 \times 10^{-3}\}$ . We then form the following reduced incidence matrix

$$RM = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (58)$$

Since matrix (58) is decomposable (according to Remark 3.2), we form the following single objective optimization sub-problems

$$\text{minimize } \{f_1(\mathbf{x})\}_{\mathbf{x} \in [-\ell, \ell]^{10}} \quad (59)$$

where

$$x_6 = 0,$$

$$x_7 = 0,$$

$$x_8 = 0,$$

$$x_9 = 0,$$

$$x_{10} = 0,$$

$$\text{minimize } \{f_2(\mathbf{x})\}_{\mathbf{x} \in [-\ell, \ell]^3} \quad (60)$$

where

$$x_1 = 0,$$

$$x_2 = 0,$$

$$x_3 = 0,$$

$$x_4 = 0,$$

$$x_5 = 0.$$

In sub-problem (59), the set of optimal solutions in the decision space is  $\bar{X}^1 = \{(-1, 1, -1, 1, -1)\}$ , and for sub-problem (60)  $\bar{X}^2 = \{(-1, 1, -1, 1, -1)\}$ . The set of synthetic Pareto optimal solutions of problem (55) in the decision space is obtained by the Cartesian product of the solutions of sub-problems (59) and (60), i.e.,  $\bar{X} = \bar{X}^1 \times \bar{X}^2$ . By evaluating the solutions in  $\bar{X}$  with objective functions of problem (55) we have  $\bar{F} = \left\{ \left( \frac{5}{2}\gamma, \frac{5}{2}\gamma \right) \right\}$ .

### 6.3.3 $\epsilon$ -decomposable problem $5 \times 5 \simeq (3 \times 3) \otimes (2 \times 2)$

Next, we consider the following five-objective parametrized optimization problem with  $\ell = 1.3$ .

$$\underset{\mathbf{x} \in [-\ell, \ell]^5}{\text{minimize}} \{f_1(\mathbf{x}), \dots, f_5(\mathbf{x})\} \quad (61)$$

where

$$\begin{aligned} f_1(\mathbf{x}) &= g_1(\mathbf{x}) + \gamma g_4(\mathbf{x}), & g_1(\mathbf{x}) &= ((x_1, x_2, x_3)^T - P_1^T)^2, & P_1 &= (1, 1, 1)^T, \\ f_2(\mathbf{x}) &= g_2(\mathbf{x}) + \gamma g_5(\mathbf{x}), & g_2(\mathbf{x}) &= ((x_1, x_2, x_3)^T - P_2^T)^2, & P_2 &= (1, -1, -1)^T, \\ f_3(\mathbf{x}) &= g_3(\mathbf{x}) + \gamma(g_4(\mathbf{x}) + g_5(\mathbf{x})), & g_3(\mathbf{x}) &= ((x_1, x_2, x_3)^T - P_3^T)^2, & P_3 &= (1, 1, -1)^T, \\ f_4(\mathbf{x}) &= g_4(\mathbf{x}) + \gamma g_1(\mathbf{x}), & g_4(\mathbf{x}) &= ((x_4, x_5)^T - P_4^T)^2, & P_4 &= (1, -1)^T, \\ f_5(\mathbf{x}) &= g_5(\mathbf{x}) + \gamma(g_1(\mathbf{x}) + g_2(\mathbf{x})), & g_5(\mathbf{x}) &= ((x_4, x_5)^T - P_5^T)^2, & P_5 &= (-1, 1)^T, \end{aligned}$$

and  $\gamma = 7.000 \times 10^{-3}$ .

For this problem, the estimated total sensitivity indices matrix  $SM$  and the incidence matrix  $M$  are respectively:

$$SM = \begin{bmatrix} 3.330 \times 10^{-1} & 3.330 \times 10^{-1} & 3.330 \times 10^{-1} & 1.000 \times 10^{-3} & 1.000 \times 10^{-3} \\ 3.330 \times 10^{-1} & 3.330 \times 10^{-1} & 3.330 \times 10^{-1} & 1.000 \times 10^{-3} & 1.000 \times 10^{-3} \\ 3.330 \times 10^{-1} & 3.330 \times 10^{-1} & 3.330 \times 10^{-1} & 1.000 \times 10^{-3} & 1.000 \times 10^{-3} \\ 1.000 \times 10^{-3} & 1.000 \times 10^{-3} & 1.000 \times 10^{-3} & 3.330 \times 10^{-1} & 3.330 \times 10^{-1} \\ 1.000 \times 10^{-3} & 1.000 \times 10^{-3} & 1.000 \times 10^{-3} & 3.330 \times 10^{-1} & 3.330 \times 10^{-1} \end{bmatrix}, \quad (62)$$

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (63)$$

Then, we have  $ut = 1.000 \times 10^{-3}$  and  $T = \{1.000 \times 10^{-3}\}$  and

$$RM = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}. \quad (64)$$

According to matrix (64), problem (61) is decomposed into the following sub-problems

$$\underset{\mathbf{x} \in [-\ell, \ell]^5}{\text{minimize}} \{f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})\} \quad (65)$$

where

$$x_4 = 0,$$

$$x_5 = 0,$$

$$\underset{\mathbf{x} \in [-\ell, \ell]^5}{\text{minimize}} \{f_4(\mathbf{x}), f_5(\mathbf{x})\} \quad (66)$$

where

$$x_1 = 0,$$

$$x_2 = 0,$$

$$x_3 = 0.$$

These sub-problems are solved for their effective variables. Suppose  $\bar{X}^1$  and  $\bar{X}^2$  are the set of Pareto optimal solutions of sub-problems (65) and (66), respectively. Then  $\bar{X} = \bar{X}^1 \times \bar{X}^2$  is considered as the synthetic Pareto optimal set for problem (61). We denote the synthetic Pareto frontier of problem (61) by  $\bar{F}$ . To show the quality of the approximated solutions in  $\bar{F}$ , we employ the hyper-volume indicator [41]. To do so, we calculate hyper-volume for the solutions in  $\bar{F}$ . Then, we solve problem (61) and obtain the set of Pareto optimal solutions in the objective space denoted by  $\bar{F}^*$ . The hyper-volume is also calculated for the solutions in  $\bar{F}^*$ . This indicator for the solutions in  $\bar{F}$  and  $\bar{F}^*$  is 0.1220 and 0.4620, respectively. As said earlier, for a given problem which is not initially decomposable, the solution obtained by ANOVA-MOP is an approximation of the Pareto frontier of the given problem.

As can be understood, if a given problem is not decomposable at the first glance, by removing variables with negligible influence on objective function values, an approximated problem can be obtained which is decomposable. In the next subsection, we discuss the interactive aspect of ANOVA-MOP.

## 6.4 Decision making by ANOVA-MOP

When using ANOVA-MOP as an interactive method, the final solution obtained is the most preferred solution for the DM, as discussed in Subsection 3.2.1. In what follows, we discuss how the most preferred solution for the DM is obtained. As an example, we consider problem (46).

When interacting with the DM to solve the three objective functions in the first sub-problem, he provided the first reference point  $\bar{z}_1^{1*} = (4, 3, 7)^T$ . The ASF was formulated and problem (3) solved. The solution  $\bar{f}_1^{1*} = (2.5175, 1.5456, 1.9759)^T$  obtained was shown to him. Since he was not satisfied with this solution, he gave a new reference point  $\bar{z}_2^{1*} = (1, 3, 7)$ . The corresponding solution  $\bar{f}_2^{1*} = (1.1296, 3.1283, 2.3100)^T$  was obtained and shown to the DM. When comparing  $\bar{f}_1^{1*}$  and  $\bar{f}_2^{1*}$ , the DM was willing to let the value of the second objective function get worse. So, he provided a reference point  $\bar{z}_3^{1*} = (1, 5, 7)^T$ . The corresponding solution  $\bar{f}_3^{1*} = (0.4996, 4.5109, 2.6439)^T$  was obtained and shown to him. The DM was happy with the values of the first and the second objective functions, but he wished to improve the value of the third objective function. Therefore, he gave the reference point  $\bar{z}_4^{1*} = (0.4996, 4.5109, 1)^T$ . The corresponding solution  $\bar{f}_4^{1*} = (0.7528, 4.8276, 1.3168)^T$  was obtained and shown to him. This solution satisfied the DM and the interaction with him regarding the first sub-problem was terminated.

Regarding the two objective functions in the second sub-problem, the DM gave a reference point  $\bar{z}_1^{2*} = (6, 2)^T$ . The ASF including the objective functions in this sub-problem was formulated and problem (3) solved. The corresponding solution  $\bar{f}_1^{2*} = (4.4852, 0.5079)^T$  was obtained and shown to him. Then, he provided a new reference point  $\bar{z}_2^{2*} = (5, 3)^T$ , and the corresponding solution  $\bar{f}_2^{2*} = (3.1327, 1.1221)^T$  was obtained. The DM was happy with the value of the first objective function, but he wished to improve the second objective function. He then provided  $\bar{z}_3^{2*} = (3.2488, 4)^T$ .



The corresponding solution  $\bar{\mathbf{f}}_3^{2*} = (1.5893, 2.4579)^T$  was shown to him. This solution satisfied the DM and the interaction was stopped. Thanks to the decomposition, the DM could concentrate on a subset of objective functions at a time. Because he had to pay attention to three and two objective functions at a time, it was easier to follow the changes in the solutions obtained after having specified a reference point and the solution process was not cognitively as demanding as when dealing with five objective functions simultaneously.

## 7 Conclusion and future research directions

In this paper, we developed a novel surrogate-based method called ANOVA-MOP to solve computationally expensive multiobjective problems with high-dimensional decision and objective spaces. In this method, first a surrogate function for each individual objective function is built. Total sensitivity indices are then estimated. Using these indices, effective decision variables which have influence on the objective function values are identified and other variables are treated as ineffective ones. The objective functions with the same effective variables are grouped into a low dimensional sub-problem in the decision and objective spaces. This leads to decompose a complex problem into a limited number of simpler sub-problems with low dimension objective and decision spaces. The solutions of the sub-problems are composed to build a synthetic Pareto frontier for the main problem.

ANOVA-MOP can be applied as an interactive and a non-interactive method to solve computationally expensive MOPs with any number of objective functions. As an interactive method, due to the decomposition, in each step of interaction with a DM, (s)he focuses on a workable number of objective functions at a time rather than the whole objective functions. As a non-interactive method, sub-problems can be solved by parallel computing. This boosts the solution process considerably. In both versions of interactive and non-interactive ones, since ineffective variables are neglected in sub-problems, the dimension of the search domain of sub-problems in the decision space is reduced. This allows users to apply any appropriate optimization method for low-dimensional problems. Besides a surrogate-based method, ANOVA-MOP can also be applied to solve computationally inexpensive MOPs.

When decomposing problem (1), cases may arise that it is not possible to obtain sub-problems without any common variables. This means that, for any threshold considered, some effective variables are common between sub-problems. Such cases is a future research direction. Developing ANOVA-MOP for problems with constraint functions is also another future research direction.

## Acknowledgement

Mohammad Tabatabaei was supported by the financial support from the COMAS Doctoral Program in Computing and Mathematical Sciences at the University of Jyväskylä and the Academy of Finland project No. 287496. Matthias Tan and Mohammad Tabatabaei were supported by the Early Career Scheme (ECS) project No. 21201414 sponsored by the Research Grants Council of Hong Kong.

## References

- [1] M. Tabatabaei, J. Hakanen, M. Hartikainen, K. Miettinen, and K. Sindhya. A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods. *Structural and Multidisciplinary Optimization*, 52(1):1–25, 2015.
- [2] V. Pareto. *Cours d'économie politique/Professé à l'université de Lausanne*. Rouge, Lausanne,, 1896-1897.
- [3] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, 1999.
- [4] K. Miettinen. Introduction to multiobjective optimization: Noninteractive approaches. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 1–26. Springer-Verlag Berlin Heidelberg, 2008.
- [5] K. Miettinen, F. Ruiz, and A. P. Wierzbicki. Introduction to multiobjective optimization: Interactive approaches. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 27–57. Springer-Verlag Berlin Heidelberg, 2008.
- [6] O. I. Larichev. Cognitive validity in design of decision-aiding techniques. *Journal of Multi-Criteria Decision Analysis*, 1(3):127–138, 1992.
- [7] Q. Zhang, W. Liu, E. Tsang, and B. Virginas. Expensive multiobjective optimization by moea/d with gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3):456–474, 2010.
- [8] M. Hartikainen and A. Lovison. Paint-SiCon: constructing consistent parametric representations of Pareto sets in nonconvex multiobjective optimization. *Journal of Global Optimization*, 62(2):243–261, 2015.
- [9] R. Regis. Multi-objective constrained black-box optimization using radial basis function surrogates. *Journal of Computational Science*, 16:140 – 155, 2016.
- [10] H. Wang, Y. Jin, and J. O. Janson. Data-driven surrogate-assisted multi-objective evolutionary optimization of a trauma system. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1, 2016.
- [11] Y. Yue and K. Meerbergen. Accelerating optimization of parametric linear systems by model order reduction. *SIAM Journal on Optimization*, 23(2):1344–1370, 2013.
- [12] C. Audet and D. Orban. Finding optimal algorithmic parameters using derivative-free optimization. *SIAM Journal on Optimization*, 17(3):642–664, 2006.
- [13] C. Audet and J. Dennis. A pattern search filter method for nonlinear programming without derivatives. *SIAM Journal on Optimization*, 14(4):980–1010, 2004.
- [14] L. Hannah, W. Powell, and D. Dunson. Semiconvex regression for metamodeling-based optimization. *SIAM Journal on Optimization*, 24(2):573–597, 2014.
- [15] C. Audet, A. Ianni, S. Le Digabel, and C. Tribes. Reducing the number of function evaluations in mesh adaptive direct search algorithms. *SIAM Journal on Optimization*, 24(2):621–642, 2014.

- [16] S. Wild and C. Shoemaker. Global convergence of radial basis function trust region derivative-free algorithms. *SIAM Journal on Optimization*, 21(3):761–781, 2011.
- [17] A. B. Ruiz, K. Sindhya, K. Miettinen, F. Ruiz, and M. Luque. E-NAUTILUS: A decision support system for complex multiobjective optimization problems based on the nautilus method. *European Journal of Operational Research*, 246(1):218–231, 2015.
- [18] N. Cowan. The magical mystery four: How is working memory capacity limited, and why? *Current directions in psychological science*, 19(1):51–57, 2010.
- [19] A. Lovison. Global search perspectives for multiobjective optimization. *Journal of Global Optimization*, 57(2):385–39, 2013.
- [20] Y. D. Sergeyev and D. E. Kvasov. Global search based on efficient diagonal partitions and a set of lipschitz constants. *SIAM Journal on Optimization*, 16(3):910–937 (electronic), 2006.
- [21] A. Lovison. Singular continuation: Generating piecewise linear approximations to pareto sets via global analysis. *SIAM Journal on Optimization*, 21(2):463–490, 2011.
- [22] A. Saltelli, K. Chan, and M. Scott, editors. *Sensitivity Analysis*. Wiley, New York, 2000.
- [23] J. Wu and S. Azarm. Metrics for quality assessment of a multiobjective design optimization solution set. *Journal of Mechanical Design*, 123(1):18–25, 2000.
- [24] A. P. Wierzbicki. On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR Spectrum*, 8(2):73–87, 1986.
- [25] M. Tan. Sequential Bayesian polynomial chaos model selection for estimation of sensitivity indices. *SIAM/ASA Journal on Uncertainty Quantification*, 3:146–168, 2015.
- [26] I. M. Sobol. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55:271–280, 2001.
- [27] L. Chen and S. Li. Analysis of decomposability and complexity for design problems in the context of decomposition. *Journal of Mechanical Design*, 127(4):545–557, 2004.
- [28] D. Hadka and P. Reed. Borg: An auto-adaptive many-objective evolutionary computing framework. *Evolutionary Computation*, 21(2):231–259, 2013.
- [29] A. Owen. Variance components and generalized Sobol’ indices. *SIAM/ASA Journal on Uncertainty Quantification*, 1(1):19–41, 2013.
- [30] M. J. W. Jansen. Analysis of variance designs for model output. *Computer Physics Communications*, 117:35–43, 1999.
- [31] M. D. Morris, L. M. Moore, and M. D. McKay. Using orthogonal arrays in the sensitivity analysis of computer models. *Technometrics*, 50(2):205–215, 2008.

- [32] A. Saltelli, S. Tarantola, and K. P. S. Chan. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics*, 41(1):39–56, 1999.
- [33] J. E. Oakley and A. O’Hagan. Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(3):751–769, 2004.
- [34] L. Le Gratiet, C. Cannamela, and B. Iooss. A Bayesian approach for global sensitivity analysis of (multifidelity) computer codes. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):336–363, 2014.
- [35] S. Touzani and D. Busby. Smoothing spline analysis of variance approach for global sensitivity analysis of computer codes. *Reliability Engineering & System Safety*, 112:67–81, 2013.
- [36] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering and System Safety*, 93(7):964–979, 2008.
- [37] S. Da Veiga, F. Wahl, and F. Gamboa. Local polynomial estimation for sensitivity analysis on models with correlated inputs. *Technometrics*, 51(4):452–463, 2009.
- [38] L. Chen, Z. Ding, and S. Li. A formal two-phase method for decomposition of complex design problems. *Journal of Mechanical Design*, 127(2):184–195, 2005.
- [39] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:183–197, 2002.
- [40] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.
- [41] T. Okabe, Y. Jin, and B. Sendhoff. A critical survey of performance indices for multi-objective optimisation. In *The 2003 Congress on Evolutionary Computation, CEC 03*, volume 2, pages 878–885 Vol.2, 2003.

**PIV**

**PROSPECT OF T-SPLINES FOR THE ANOVA-MOP METHOD**

by

M. Tabatabaei and M. Tan

Reports of the Department of Mathematical Information Technology, Series B,  
Scientific Computing, No. 10, University of Jyväskylä, Jyväskylä, 2016

# Prospect of T-splines for the ANOVA-MOP Method

Mohammad Tabatabaei <sup>\*1</sup> and Matthias Tan <sup>†2</sup>

<sup>1</sup>University of Jyvaskyla,, Department of Mathematical Information Technology,, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland

<sup>2</sup>Department of Systems Engineering and Engineering Management, City University of Hong Kong, Kowloon Tong, Hong Kong

## Abstract

Our starting point is the surrogate-based method called ANOVA-MOP which has been developed to solve computationally expensive multiobjective optimization problems relying on sensitivity analysis. We discuss how ANOVA-MOP can take advantage from sensitivity analysis methods developed in the literature. We then discuss the potential of a new metamodeling technique termed T-splines for sensitivity analysis and how they could be employed in ANOVA-MOP.

**Keywords** Black-box function, computational cost, metamodeling technique, T-splines, non-differentiability

## 1 Introduction

In many application domains, computationally expensive functions in multiobjective optimization problems (MOPs) require conducting time-consuming experiments and/or simulations. These functions can have a black-box nature, i.e., for a given input we only know the output. Computationally expensive functions can be continuous, discontinuous, differentiable or non-differentiable. In the literature, *surrogate-based* methods are

---

<sup>\*</sup>mohammad.tabatabaei@jyu.fi; corresponding author

<sup>†</sup>mathtan@cityu.edu.hk

employed to solve such problems [1]. Typically, in surrogate-based methods, metamodeling techniques are applied to approximate computationally expensive functions. As a result, computationally inexpensive functions known as *surrogate functions* are built. Using surrogate functions, computationally inexpensive MOPs termed as *surrogate problems* are constructed. In [1], we surveyed how surrogate-based methods build surrogate problems and how they are solved to find solutions for the original, computationally expensive MOPs. We then concluded that solving computationally expensive MOPs with high-dimensional decision and objective spaces deserves further research efforts.

The curse-of-dimensionality is connected with the sparsity of effects principle. This means that typically a subset of decision variables called *effective variables* has an important effect on objective function values, while the remaining variables called *ineffective variables* have a little effect. In the literature, identifying effective variables is termed *global sensitivity analysis*, and *functional analysis of variance decomposition* (functional ANOVA decomposition) [2] is one way to do sensitivity analysis. By applying functional ANOVA decomposition, an objective function is hierarchically decomposed into a sum of terms depending only on subsets of decision variables, based on their contribution to the total variance of the function. Terms depending on individual variables are called *main effects*, while the contributions due to pairs of variables are called *interactions*. As mentioned in [3], typically 80% of the effects on objective function values are due to 20% of the variables.

Relying on sensitivity analysis, in [4], we have developed a surrogate-based method called *ANOVA-MOP* to solve high-dimensional, computationally expensive MOPs. In the ANOVA-MOP method, first, for each computationally expensive function, a surrogate function is built and a surrogate MOP is then formed. We then apply sensitivity analysis on each individual surrogate function and identify effective and ineffective variables for them. Based on sensitivity analysis information obtained, the surrogate MOP is decomposed into a finite number of sub-problems which have a lower dimension in the decision and objective spaces in comparison with those of the surrogate MOP. Sub-problems are solved and solutions obtained are composed to form solutions for the surrogate MOP. These solutions can be considered as approximated solutions for the original, computationally expensive MOP. As discussed in [4], besides computationally expensive MOPs, ANOVA-MOP can be used for solving computationally inexpensive MOPs as well.

In ANOVA-MOP, sensitivity analysis is an important element. As mentioned in [5], sensitivity analysis methods can be divided in two classes, i.e., methods that do not utilize surrogates and methods that utilize surrogates. Examples of methods in the first class are [6, 7, 8, 9] and those of the second class are [5, 10, 11, 12, 13, 14, 15, 16]. One can see e.g., [17] for a review on sensitivity analysis methods.

As indicated in [5], typically methods in the first class not involving surrogates demand more function evaluations in comparison to those belonging to the second class. Moreover, as discussed in [5], methods in the first class generally depend on minimal assumptions about functions, while the latter class relies on restrictive assumptions (e.g., differentiability). Thus, methods in the first class can be employed for computationally inexpensive functions which can be e.g., discontinuous and/or non-differentiable, and methods in the second category are more suitable for computationally expensive and smooth functions.

It is always desirable to elicit sensitivity analysis information accurately. ANOVA-

MOP can benefit from methods in the first class to solve computationally inexpensive MOPs. As a result, even if a problem contains non-differentiable functions, the information obtained from sensitivity analysis can be utilized to form sub-problems accurately as discussed in [4]. For solving computationally expensive MOPs including differentiable functions by ANOVA-MOP, methods in the second class are suitable.

A computationally expensive MOPs may have non-differentiable functions. In such problems, the second class of methods (which rely on e.g. differentiability) may not give sensitivity analysis information accurately. Therefore, sub-problems cannot be built accurately in ANOVA-MOP. In this report, we introduce T-splines which are built based on spline functions as a possible technique for incorporating into ANOVA-MOP. We expect that by applying T-splines, sensitivity analysis information for computationally expensive non-differentiable functions may be obtained accurately and computationally expensive MOPs including these kinds of functions can be solved efficiently.

In the literature, the mathematical properties of T-splines have been investigated for functions with two and three decision variables [18]. Typically, computationally expensive MOPs include more than three decision variables. Therefore, before applying T-splines for such problems, we must first develop T-splines for functions with more than three decision variables. This report is an initial step to study T-splines for sensitivity analysis in computationally expensive problems. We do not yet provide any actual method on function approximation through T-splines.

Spline functions have been used in curve fitting to model complicated curves, for instance, circles or ellipses [19]. They also have been applied to approximate uni-variate functions [20]. Spline functions are linear combinations of piecewise-polynomial functions of an arbitrary degree. These piecewise-polynomial functions are basis functions and are called B-splines. They are formed relying on a set of knots. This knots set provides a flexibility for spline functions to approximate non-differentiable functions [20]. Spline functions with low-degree B-splines can be employed to approximate higher degree polynomial functions.

To approximate functions with more than one decision variable through spline functions, bi-variate B-splines have been developed [19]. In bi-variate B-splines, corresponding to each decision variable, a set of basis functions is constructed. By using these sets of basis functions, tensor spline bases are formed and bi-variate B-splines are obtained. In bi-variate B-splines, a refinement of the set of basis functions for one of the decision variables means adding some new basis functions into this set. This assists to improve the accuracy of the approximation. Nevertheless, refining a set of basis functions corresponding to one of the decision variables leads to increasing the number of tensor spline bases in bi-variate B-splines, exponentially. This is a drawback in bi-variate B-splines which does not provide a local refinement [20, 21, 22].

To overcome the drawback of bi-variate B-splines, T-splines have been developed in computer-aided design (CAD) [21]. T-splines follow the same flexibility as bi-variate B-splines, but the refinement in T-splines is local. To the best of our knowledge, so far in the literature, T-splines have not been used to approximate computationally expensive functions in multiobjective optimization problems (MOPs). In this report, we discuss the potential of T-splines to approximate computationally expensive functions and to do sensitivity analysis when dealing with multiobjective optimization problems. This is the first step on investigating the idea of employing of T-splines in sensitivity analysis



for computationally expensive non-differentiable functions. Here, we do not yet provide any idea on function approximation through T-splines.

The rest of this report is organized as follows. In Section 2, we briefly discuss basic concepts in multiobjective optimization, functional ANOVA decomposition and give an overview of ANOVA-MOP. In Section 3, we discuss uni-variate function approximation by spline functions. Then, in Section 4, bi-variate function approximation is discussed. In Section 5, we discuss T-splines and the prospect of employing T-splines for ANOVA-MOP. Finally, conclusions are drawn in Section 6.

## 2 Background in Multiobjective Optimization

In this section, we briefly discuss some concepts used in this report. We then provide a brief description of the ANOVA-MOP method. We consider multiobjective optimization problems of the form:

$$\underset{\mathbf{x} \in S}{\text{minimize}} \{f_1(\mathbf{x}), \dots, f_k(\mathbf{x})\}, \quad (1)$$

where  $f_i : S \rightarrow \mathbb{R}$  are  $k$  ( $\geq 2$ ) conflicting, computationally expensive objective functions,  $S = \{\mathbf{x} \in \mathbb{R}^n : x_c^l \leq x_c \leq x_c^u, c = 1, \dots, n\}$  is a nonempty *feasible decision* set which is a subset of the *decision space*  $\mathbb{R}^n$ . A solution  $\mathbf{x} = (x_1, \dots, x_n)^T \in S$  is called a *feasible decision (variable) vector*, where  $x_c, c = 1, \dots, n$ , are *decision variables* and,  $x_c^l$  and  $x_c^u$  are the lower and upper bounds of  $x_c$ , respectively.

In the ANOVA-MOP method, we utilize the functional ANOVA decomposition and sensitivity analysis [2] to measure the importance of the variables in the objective function values. Suppose an arbitrary function  $g : S \rightarrow \mathbb{R}$  is square integrable, i.e.,  $\int_S g(\mathbf{x})^2 d\mathbf{x} < \infty$ . The *functional ANOVA decomposition* of  $g$  is

$$g(\mathbf{x}) = g_0 + \sum_i g_{(i)}(x_i) + \sum_{i < j} g_{(i,j)}(x_i, x_j) + \dots + g_{(1, \dots, n)}(\mathbf{x}), \quad (2)$$

where  $g_0 = \int_S g(\mathbf{x}) d\mathbf{x}$ , if

$$\int_{S_{i_u}} g_{(i_1, \dots, i_e)}(x_{i_1}, \dots, x_{i_e}) dx_{i_u} = 0, \quad (3)$$

for all  $u = 1, \dots, e$ , ( $e \geq 1$ ), and  $1 \leq i_1 < \dots < i_e \leq n$ . We call  $g_{(i)}$  a *main effect* and  $g_{(i_1, \dots, i_e)}$ , an *interaction* of order  $e$ . This results in decomposing the variance of  $g$ , i.e.,  $\text{var}(g) = D_0 = \int_S g^2(\mathbf{x}) d\mathbf{x} - g_0^2$ , into  $2^n - 1$  *variance components*

$$D_0 = \sum_i D_{(i)} + \sum_{i < j} D_{(i,j)} + \dots + D_{(1, \dots, n)}, \quad (4)$$

where  $D_{(i_1, \dots, i_e)} = \text{var}[g_{(i_1, \dots, i_e)}(x_{i_1}, \dots, x_{i_e})] = \int_{\prod_{u=1}^e S_{i_u}} g_{(i_1, \dots, i_e)}^2(x_{i_1}, \dots, x_{i_e}) dx_{i_1} \dots dx_{i_e}$ . The Sobol' indices [7] are calculated as

$$I_{(i_1, \dots, i_e)} = D_{(i_1, \dots, i_e)} / D_0, \quad (5)$$

for all  $(i_1, \dots, i_e) \in \Delta = \{(r_1, \dots, r_u) : 1 \leq r_1 < \dots < r_u \leq d, u \geq 1\}, e = 1, \dots, n$ . The total sensitivity indices, which are measures of the contribution of each decision variable  $x_j$ , for  $j = 1, \dots, n$ , in the value of  $g$ , are given by

$$I_j^T = \sum_{(i_1, \dots, i_e) \in \Delta: j \in \{i_1, \dots, i_e\}} D_{(i_1, \dots, i_e)} / D_0, \quad j = 1, \dots, n. \quad (6)$$

Sobol' indices and total sensitivity indices are typically approximated in practice. In this report, we call them estimated Sobol' indices and estimated total sensitivity indices, respectively.

In [4], we have developed the ANOVA-MOP method to solve high-dimensional, computationally expensive MOPs. A general sketch of steps of ANOVA-MOP is as follows: we first build a surrogate function for each individual objective function which is accurate everywhere in the entire decision/objective space and formulate a surrogate MOP. Corresponding each surrogate objective function, total sensitivity indices are estimated to identify effective and ineffective variables. Then, objective functions which have the same effective decision variables are considered as a sub-problem. In this way, the surrogate MOP is decomposed into a number of sub-problems with a lower dimension in the decision and objective spaces compared to those in the surrogate MOP. Sub-problems are then solved independently by any appropriate optimization method. Solutions obtained are composed to form solutions for the surrogate MOP. These solutions are treated as approximated solutions for the original, computationally expensive MOP.

Generally, it is important to estimate total sensitivity indices accurately. In Section 1, we discussed how ANOVA-MOP can benefit from sensitivity analysis methods developed in the literature. In Section 5, we introduce T-splines as a possible metamodeling technique to approximate terms in the functional ANOVA decomposition. In this way, non-differentiable functions can be approximated accurately and correspondingly total sensitivity indices are estimated accurately. To introduce T-splines, we need a discussion on function approximation by splines functions given in the next section.

### 3 Uni-variate B-splines

Basis-splines or in a short term, B-splines are the essence of both bi-variate B-splines and T-splines. In this section, we elaborate B-splines. Thereafter, in the following sections, bi-variate B-splines and T-splines are discussed.

**Definition 3.1.** Suppose  $U$  includes a set of  $e + p + 1$  non-decreasing numbers,  $u_1 \leq u_2 \leq u_3 \leq \dots \leq u_{e+p+1}$ . We call the  $u_i$ s knots, the set  $U$  the global knot vector, and the half-open interval  $[u_i, u_{i+1})$  the  $i^{\text{th}}$  knot span,  $p$  is the basis degree and  $e$  is the number of basis functions employed to create the B-spline curve.

Knots can be repeated. The multiplicity of  $u_i$  denoted by  $\#u_i$  is the maximum number repetition of  $u_i$  in  $U$ . The multiplicity of a knot affects the continuity of the corresponding basis function at this knots. In fact, if  $u_i$  is repeated  $\kappa$  times, then the corresponding basis function is  $C^{p-\kappa}$  continuous. Thus, a knot can be repeated at most  $p+1$  times. Since some knots may be repeated in  $U$ , some knot spans may not exist. For the sake of simplicity, if a knot  $u_i$  repeats  $\kappa > 1$  times, i.e.,  $u_i = u_{i+1} = \dots = u_{i+\kappa-1}$ , we write it as  $u_i(\kappa)$ . Otherwise,  $u_i$  with only one repetition is a simple knot. In a uniform global knot vector, knots are equally spaced, i.e.,  $u_{i+1} - u_i = c$  in which  $c$  is a constant for  $0 \leq i \leq e + p + 1$ . Otherwise, the global knot vector is non-uniform.

**Definition 3.2.** *If the knots at the beginning and the end have multiplicity  $p+1$ , i.e.,  $u_1 = u_2 = \dots = u_{p+1}$  and  $u_{e+1} = u_{e+2}, \dots = u_{e+p+1}$ , then the global knot vector is said to be open.*

A basis function built based on an open global knot vector is discontinuous at the beginning and the end.

**Definition 3.3.** *For a global knot vector  $U$ , the B-splines  $N_{i,p}(x)$  of degree  $p$  are defined by the Cox-de Boor recursion formula [19]:*

$$N_{i,0}(x) = \begin{cases} 1 & x \in [u_i, u_{i+1}) \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$N_{i,p}(x) = \frac{x - u_i}{u_{i+p} - u_i} N_{i,p-1}(x) + \frac{u_{i+p+1} - x}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(x) \quad (8)$$

In the above equations, if a denominator becomes zero, then the corresponding term is assigned zero. If  $p = 3$ , we call  $N_{i,3}$  a cubic-basis function. We proceed with the following properties about the set of basis functions [19]:

1. for all  $i, p$  and  $x$ ,  $N_{i,p}(x) \geq 0$ ,
2. if  $x \notin [u_i, u_{i+1})$ , then  $N_{i,p}(x) = 0$ ,
3. on any given knot span  $[u_i, u_{i+1})$ , at most  $p+1$  of the basis functions are non-zero, i.e.,  $N_{i-p,p}(x), N_{i-p+1,p}(x), N_{i-p+2,p}(x), \dots, N_{i,p}(x)$ ,
4. for any given knot span  $[u_i, u_{i+1})$ , we have that the summation of all non-zero basis functions is one, i.e.,  $\sum_{j=i-p}^i N_{j,p}(x) = 1$ , for all  $x \in [u_i, u_{i+1})$ ,
5. the number of knots in  $U$  is  $e+p+1$  in which  $p$  is the degree of basis functions and  $e$  is the number of basis functions,
6. at the interior of a given knot span  $[u_i, u_{i+1})$ , all derivatives of basis function  $N_{i,p}(x)$  exist. In fact,  $N_{i,p}$  is  $p - \kappa$  times continuously differentiable at a knot with multiplicity  $\kappa$ .

In what follows, without loss of generality, we assume that  $f : [a, b] \rightarrow \mathbb{R}$  is a computationally expensive black-box function and  $\widehat{f} : [a, b] \rightarrow \mathbb{R}$  is a surrogate function to be built. We discuss a general sketch of building  $\widehat{f}$  by B-spline basis functions.

**Definition 3.4.** For a given degree  $p$  and a global knot vector  $U$  including  $e + p + 1$  knots:

$$\widehat{f}(x) = \sum_{i=1}^e \alpha_i N_{i,p}(x), \quad (9)$$

where the coefficients  $\alpha_i$ ,  $i = 1, \dots, e$ , must be estimated for a given set of sample points  $(x^i, y^i)$ ,  $i = 1, \dots, n$ , ( $n \geq e$ ) and  $y^i = f(x^i)$ .

To estimate the coefficients, we solve the following least square problem:

$$\underset{(\alpha_1, \dots, \alpha_e)^T \in \mathbb{R}^e}{\text{minimize}} \sum_{i=1}^n (\widehat{f}(x^i) - y^i)^2. \quad (10)$$

**Example 3.5.** Suppose  $x \in [0, \pi]$  and  $f(x) = |\sin(3x)|$  in which  $|\cdot|$  is the absolute value sign. We consider  $p = 3$  and  $U = \{0, 0, 0, 0, \frac{\pi}{3}, \frac{\pi}{3}, \frac{\pi}{3}, \frac{2\pi}{3}, \frac{2\pi}{3}, \frac{2\pi}{3}, \pi, \pi, \pi, \pi\}$ . Since  $U$  has 14 knots, the number of basis functions i.e.,  $e = 14 - 3 - 1 = 10$ . We consider two kinds of sampling, i.e., fine grid and randomly. Regarding the fine grid sampling, we evaluate  $f$  at sample points  $x^i = \frac{i\pi}{9}$  for  $i = 0, \dots, 9$ . Random points are selected in MATLAB random points generator. Figures 1 (a) and (b), and Figure 2 (a) depict the original function, the surrogate function  $\widehat{f}$  by fine grid sampling and random sampling, respectively. As can be seen, this function is non-differentiable at  $x \in \{\frac{\pi}{3}, \frac{2\pi}{3}\}$ . Since the knot values  $\frac{\pi}{3}$  and  $\frac{2\pi}{3}$  are repeated 3 times in  $U$ , the surrogate function  $\widehat{f}$  is  $C^0$  according to property 6. We also approximate the function  $f$  using the selected sample points by RBF, SVR and Kriging. Figures 1 (c-e) and Figures 2 (b-d) illustrate the results, respectively. For instance, as can be seen in Figure 1 (b), RBF does not represent  $f$  accurately at  $x \in \{\frac{\pi}{3}, \frac{2\pi}{3}\}$  where  $f$  is not differentiable.

Next, we discuss bi-variate function approximation by B-splines.

## 4 Bi-variate function approximation

### 4.1 Bi-variate B-splines

The expansion of uni-variate B-splines regression to bi-variate B-splines one is straightforward.

**Definition 4.1.** Suppose  $f : [a, b]^2 \rightarrow \mathbb{R}$  is a computationally expensive black-box bi-variate function and  $\widehat{f} : [a, b]^2 \rightarrow \mathbb{R}$  is a surrogate function to be built. Then

$$\widehat{f}(x_1, x_2) = \sum_{i=1}^{e_1} \sum_{j=1}^{e_2} \alpha_{i,j} N_{i,j}^{p,q}(x_1, x_2), \quad (11)$$

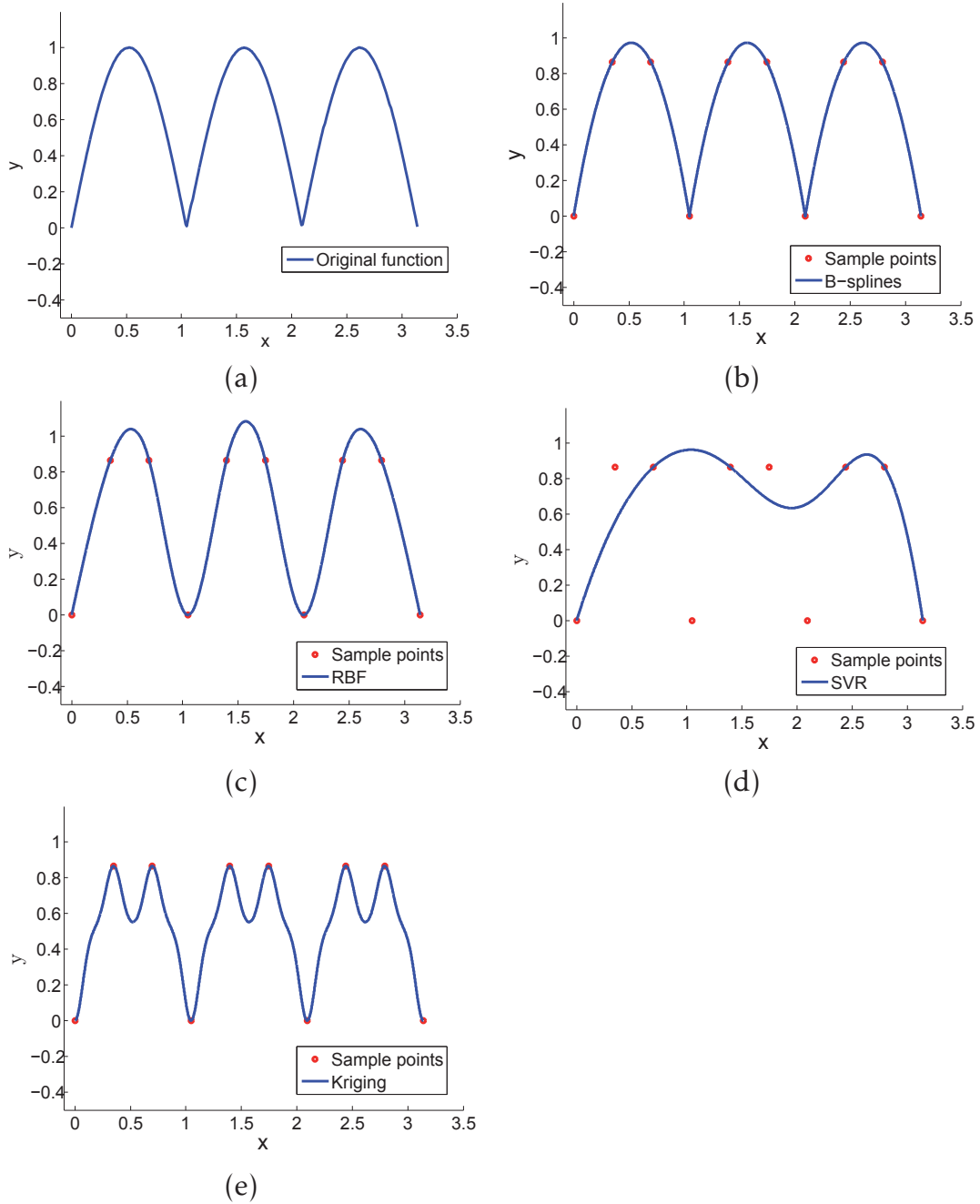


Figure 1: A uni-variate non-differentiable function (a) approximated by B-splines (b), RBF (c), SVR (d) and Kriging (e). Only B-splines represent non-differentiability of this function, properly. (Color online)

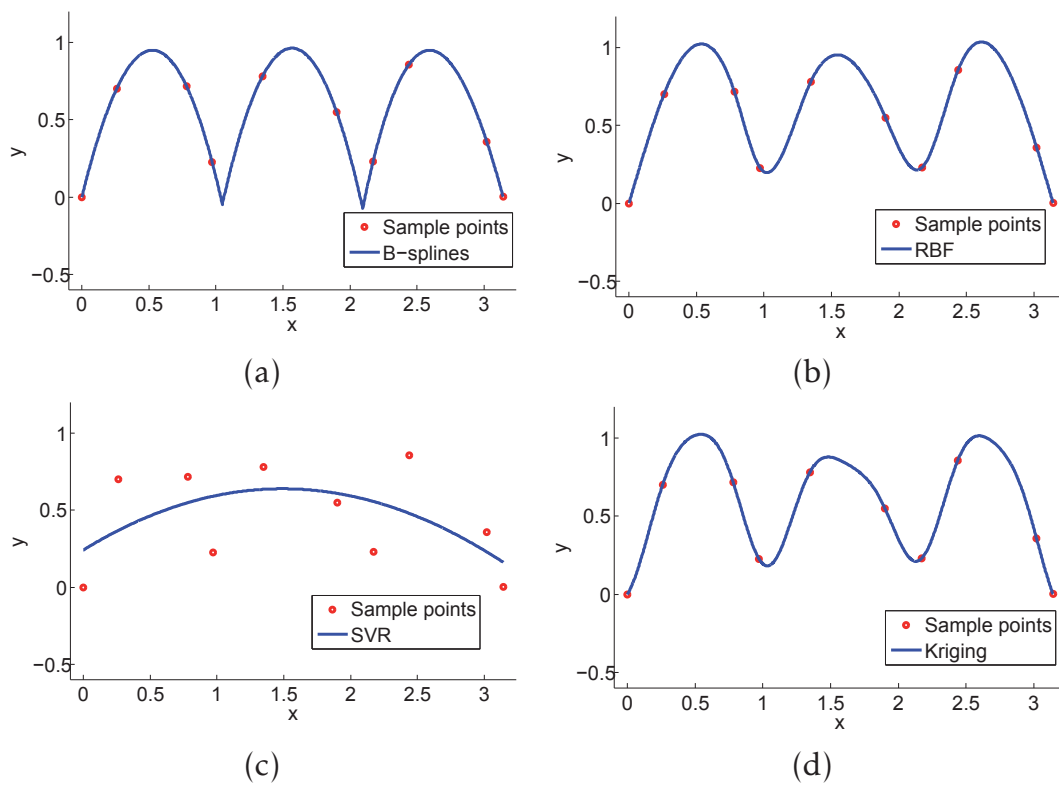


Figure 2: The uni-variate non-differentiable function represented in Figure 1 (a) is approximated by selecting samples randomly through B-splines (a), RBF (b), SVR (c) and Kriging (d). Only B-splines represent non-differentiability of this function, properly. (Color online)

in which  $N_{i,j}^{p,q}(x_1, x_2), i = 1, \dots, e_1, j = 1, \dots, e_2$ , are called tensor spline bases and formed as  $N_{i,j}^{p,q}(x_1, x_2) = N_{i,p}(x_1)N_{j,q}(x_2)$  where  $N_{i,p}, i = 1, \dots, e_1$ , are basis functions of degree  $p$  corresponding to a global knot vector  $U = \{u_1, u_2, \dots, u_{e_1+p+1}\}$ ,  $N_{j,q}, j = 1, \dots, e_2$ , are basis functions of degree  $q$  corresponding to a global knot vector  $V = \{v_1, v_2, \dots, v_{e_2+q+1}\}$  and  $\alpha_{i,j}, i = 1, \dots, e_1, j = 1, \dots, e_2$ , are coefficients to be calculated.

These coefficients for given  $U$  and  $V$  and a set of sample points  $(x_1^i, x_2^i, y^i), y^i = f(x_1^i, x_2^i), i = 1, \dots, \Delta, \Delta \geq e_1 \cdot e_2$ , are calculated through minimizing least square error by adopting problem (10) for a bi-variate case. One can see that bi-variate B-splines has a tensor product structure.

**Example 4.2.** Suppose  $(x_1, x_2) \in [0, \pi]^2$  and  $f(x_1, x_2) = |\sin(3x_1)|$ . By considering  $p = q = 3$  and  $U = \{0, 0, 0, 0, \frac{\pi}{3}, \frac{\pi}{3}, \frac{\pi}{3}, \frac{2\pi}{3}, \frac{2\pi}{3}, \frac{2\pi}{3}, \pi, \pi, \pi, \pi\}$  and  $V = \{0, 0, 0, 0, \pi, \pi, \pi, \pi\}$ , we have  $e_1 = 10$ ,  $e_2 = 4$  and 40 basis functions. Thus we select 40 sample points. The set of sample points in the decision space is  $\{(x_1^i, x_2^j) \mid x_1^i = \frac{i\pi}{9}, i = 0, \dots, 9, x_2^j = \frac{j\pi}{3}, j = 0, 1, 2, 3\}$ . Figures 3 (a) and (b) depict the original function  $f$  and the surrogate function  $\hat{f}$ . As can be seen, this function is non-differentiable on the lines  $x_1 = \frac{\pi}{3}$  and  $x_1 = \frac{2\pi}{3}$ . We also approximate the function  $f$  using the selected sample points by RBF, SVR and Kriging. Figures 3 (c-e) illustrate the results. As can be seen in these figures, none of them is able to represent  $f$  in regions where  $f$  is not differentiable.

**Remark 4.3.** The task in approximating a uni-variate (or a bi-variate) function by B-splines for a given degree  $p$  (or  $p$  and  $q$ ) is to identify the global knot vector  $U$  (or  $U$  and  $V$ ). Several methods have been developed in the literature to approximate uni-variate functions by B-splines, see e.g., [23, 24].

A drawback of bi-variate B-splines in function approximation is that, refining a set of basis functions for one the variables increases the number of tensor spline bases, exponentially. This may add some unnecessary basis functions after refinement. Next we discuss T-splines to rectify such an issue.

## 5 Review of T-splines

### 5.1 Preliminary

Before we discuss T-splines, we need to explain some basic concepts. In this subsection, we provide them.

**Definition 5.1.** For given global knot vectors  $U = \{u_1, u_2, \dots, u_{e_1+p+1}\}$  and  $V = \{v_1, v_2, \dots, v_{e_2+q+1}\}$ , we call the Cartesian product of  $U$  and  $V$ , i.e.,  $UV = U \times V$ , as the knot-value-coordinate system. The set of knot indices are  $I_1 = [1, e_1 + p + 1] \cap \mathbb{N}$  and  $I_2 = [1, e_2 + q + 1] \cap \mathbb{N}$ . The knot-index-coordinate system is defined as the Cartesian product of  $I_1$  and  $I_2$ , i.e.,  $I = I_1 \times I_2$  which is a subset of the indices space  $\mathbb{N}^2$  [25].

The following example shows the motivation of defining these systems.

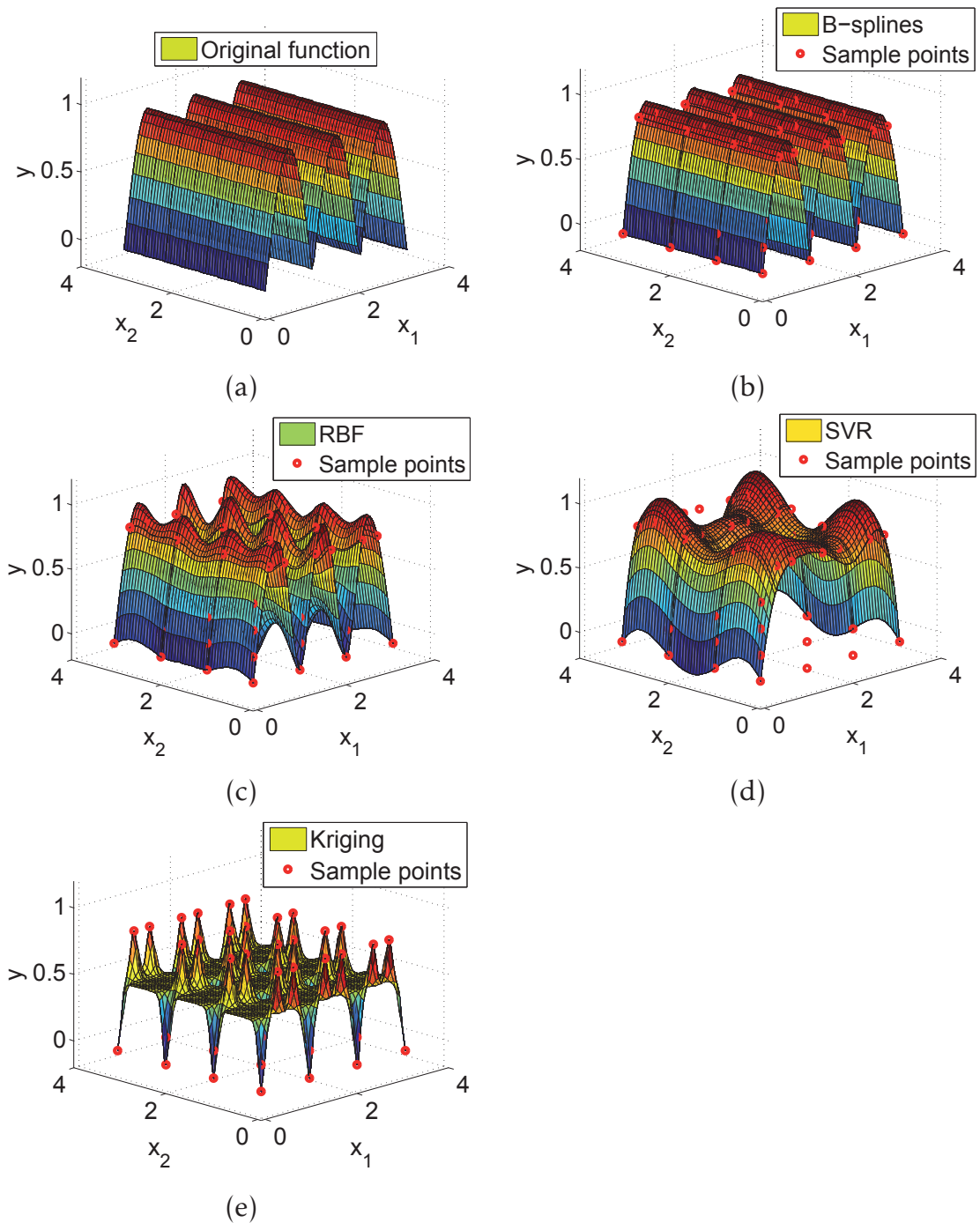


Figure 3: A bi-variate non-differentiable function (a) approximated by B-splines (b), RBF (c), SVR (d) and Kriging (e). Only B-splines represent non-differentiability of this function, adequately. (Color online)



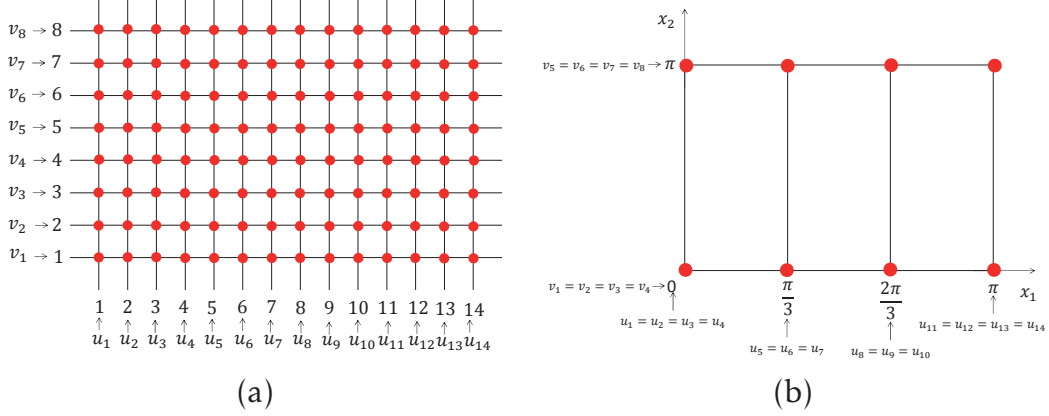


Figure 4: (a) Knot-Index-Coordinate System. (b) Knot-Value-Coordinate System corresponding to (a). (Color online)

**Example 5.2.** Suppose  $U, V, p$  and  $q$  are the same as given in Example 4.2. Then,  $UV = (0, 0), (\frac{\pi}{3}, 0), (\frac{2\pi}{3}, 0), (\pi, 0), (0, \pi), (\frac{\pi}{3}, \pi), (\frac{2\pi}{3}, \pi), (\pi, \pi)$ ,  $I_1 = [1, 14] \cap \mathbb{N}$ ,  $I_2 = [1, 8] \cap \mathbb{N}$  and  $I = I_1 \times I_2$ . Two elements among all elements in the knot-index-coordinate system  $I$  are  $(1, 1)$  and  $(3, 3)$ . These two elements correspond to  $(u_1, v_1)$  and  $(u_3, v_3)$ , respectively. In the knot-value-coordinate system, these two elements represent the same coordinate  $(0, 0)$ . As can be understood, the knot-index-coordinate system assists us to know which pair of knots we consider. One should note that by all elements in the knot-value-coordinate system  $UV$ , we form squares in the decision space as shown in Figure 4 (b). However, all elements in the knot-index-coordinate system  $I$  form a full grid in the indices space. Figures 4 (a) and 4 (b) depict the knot-value-coordinate system and the knot-index-coordinate system corresponding to Example 4.2, respectively.

The net grid in Figure 4 (a) is called *mesh* in the knot-index-coordinate system. In this figure, the red circles are called *anchors* in the knot-index-coordinate system.

## 5.2 T-splines

T-splines are first introduced in [21]. Following [21, 22, 26], we assume that  $p = q = 3$ . Suppose  $I_1 = [1, e_1 + 4] \cap \mathbb{N}$  and  $I_2 = [1, e_2 + 4] \cap \mathbb{N}$ . T-splines is defined based on global knot vectors  $U = \{u_1, \dots, u_{e_1+4}\}$  and  $V = \{v_1, \dots, v_{e_2+4}\}$  and a T-mesh. We define a T-mesh as mentioned in [27].

**Definition 5.3.** A T-mesh  $\mathfrak{T}$  is a rectangular partition of the knot-index-coordinate system  $I = I_1 \times I_2$ , such that all elements corners, i.e., vertices of  $\mathfrak{T}$ , have integer coordinates. The elements in  $\mathfrak{T}$  are open sets. The set of all vertices in  $\mathfrak{T}$  is denoted by  $\mathfrak{V}$ . A horizontal (respectively, vertical) edge between two vertices  $\mathbf{v}^1 = \{(i_1, j_1)\}$  and  $\mathbf{v}^2 = \{(i_2, j_2)\}$  with  $i_1 < i_2$  and  $j_1 = j_2$  (respectively,  $i_1 = i_2$  and  $j_1 < j_2$ ) is an open segment and is defined as  $h\epsilon = ]i_1, i_2[ \times \{j_1\}$  (respectively,  $v\epsilon = \{i_1\} \times ]j_1, j_2[$ ). The set of all horizontal (respectively, vertical) edges is denoted by  $h\mathfrak{E}$  (respectively,  $v\mathfrak{E}$ ). The set of all edges  $\mathfrak{E} = h\mathfrak{E} \cup v\mathfrak{E}$ . We denote the boundary of an element

$\mathcal{Q} \in \mathcal{T}$  by  $\partial\mathcal{Q}$  and denote the union of the two vertices  $\mathbf{v}^1, \mathbf{v}^2 \in \mathcal{V}$  which are endpoints of an edge  $\epsilon \in \mathcal{E}$  by  $\partial\epsilon$ . The valence of a vertex  $\mathbf{v} \in \mathcal{V}$  is the number of edges  $\epsilon \in \mathcal{E}$  such that  $\mathbf{v} \subset \partial\epsilon$ . A vertex with valence three is called a T-junction. The union of all horizontal (respectively, vertical) edges and vertices is called the horizontal (respectively, vertical) skeleton of the T-mesh and is denoted by  $h\mathcal{S}$  (respectively,  $v\mathcal{S}$ ). The skeleton of the T-mesh is defined as  $\mathcal{S} = h\mathcal{S} \cup v\mathcal{S}$ .

**Definition 5.4.** Following [27], the knot-index-coordinate system  $I$  is split into an active region  $AR$  and a frame region  $FR$  (see Figure 5 (a)), such that  $AR = [3, e_1 + 2] \times [3, e_2 + 2]$  and  $FR = ([1, 3] \cup [e_1 + 2, e_1 + 4]) \times [1, e_2 + 4] \cup [1, e_1 + 4] \times ([1, 3] \cup [e_2 + 2, e_2 + 4])$ .

One should note that  $AR$  and  $FR$  are closed regions.

**Definition 5.5.** Similar to [27], the T-mesh  $\mathcal{T}$  is said to be admissible (see Figure 5 (a)), if  $\mathcal{S} \cap FR$  includes the vertical segments  $\{i\} \times [1, e_2 + 4]$ , for  $i = 1, 2, 3$  and  $i = e_1 + 2, e_1 + 3, e_1 + 4$ , and the horizontal segments  $[1, e_1 + 4] \times \{j\}$ , for  $j = 1, 2, 3$ , and  $j = e_2 + 2, e_2 + 3, e_2 + 4$ , and all vertices  $\mathbf{v} \subset ]1, e_1 + 4[ \times ]1, e_2 + 4[ \cap FR$  have valence four.

If the T-mesh  $\mathcal{T}$  is admissible, we then say that  $\mathcal{T}$  belongs to  $AD$ . As mentioned in [27], by Definition 5.5, the T-mesh does not have T-junctions in the frame region. See Figure 5 (a).

**Definition 5.6.** Following [27], the T-mesh  $\mathcal{T} \in AD$  is said to belong to  $AD^+$  if, considering each couple of vertices  $\mathbf{v}^1 = \{(i_1, j_1)\}$  and  $\mathbf{v}^2 = \{(i_2, j_2)\}$  in  $\mathcal{V}$ , such that  $\mathbf{v}^1, \mathbf{v}^2 \subset \partial\mathcal{Q}$  for some  $\mathcal{Q} \in \mathcal{T}$ , and with  $i_1 = i_2$  (respectively,  $j_1 = j_2$ ), the open segment  $\{i_1\} \times ]j_1, j_2[$  (respectively,  $]i_1, i_2[ \times \{j_1\}$ ) is included in  $\mathcal{S}$ .

As pointed out in [27], by Definition 5.6, the T-mesh does not include two facing T-junctions as those shown in Figure 5 (b).

**Definition 5.7.** As mentioned in [27], for a given T-mesh  $\mathcal{T} \in AD$ , the set of anchors  $\mathfrak{A}(\mathcal{T}) = \{A \in \mathcal{V} : A \subset AR\}$  (see Figure 5 (a)). An anchor  $A \in \mathfrak{A}(\mathcal{T})$  is represented as  $a \times b$  where  $\{a\} \subset I_1$  and  $\{b\} \subset I_2$ .

**Definition 5.8.** Suppose  $\{a\} \subset \mathbb{N}$ . Following [27], we define  $h\mathcal{J}(a) := \{i \in I_1 : \{i\} \times a \subset v\mathcal{S}\}$  and  $v\mathcal{J}(a) := \{j \in I_2 : a \times \{j\} \subset h\mathcal{S}\}$ , and assume that these sets are ordered.

**Definition 5.9.** For a given anchor  $A = a \times b \in \mathfrak{A}(\mathcal{T})$ , following [27], we define the corresponding horizontal (vertical) index vector  $h\mathbf{v}(A)$  ( $v\mathbf{v}(A)$ , respectively) as a subset of  $h\mathcal{J}(b)$  ( $v\mathcal{J}(a)$ , respectively) such that  $h\mathbf{v}(A) = (i_1, \dots, i_5) \in \mathbb{N}^5$  which is made of the unique 5 consecutive indices in  $h\mathcal{J}(b)$  with  $\{i_3\} = a$ . Analogously,  $v\mathbf{v}(A) = (j_1, \dots, j_5) \in \mathbb{N}^5$  is constructed.

In Figure 5 (a), the green lines show the horizontal and vertical index vectors corresponding to the anchor  $A$  centered on  $(9, 7)$ .

**Definition 5.10.** Following [27], for a given T-mesh  $\mathcal{T} \in AD^+$ , a T-spline representation is given by

$$T(x_1, x_2) = \sum_{i=1}^{e_T} \alpha_i B_i(x_1, x_2), \quad (12)$$

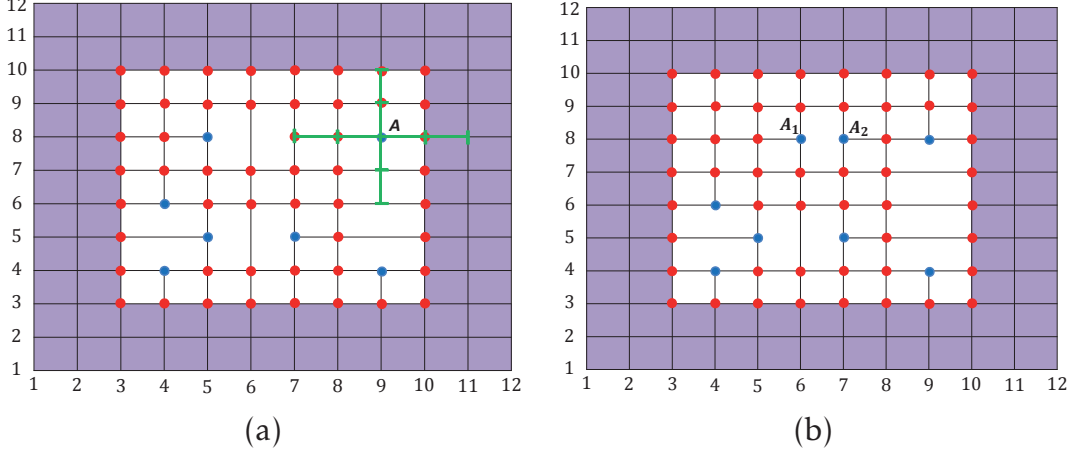


Figure 5: (a) An admissible T-mesh belonging to  $AD^+$  in the knot-index-coordinate system. The frame and active regions are shown in purple and white, respectively. Red circles are anchors. T-junctions are in blue. Green lines show the horizontal and vertical index vectors of the anchor (T-junction)  $A$ . (b) An admissible T-mesh with face-to-face T-junctions  $A_1$  and  $A_2$ . This T-mesh does not belong to  $AD^+$ . It can be turned into an  $AD^+$  T-mesh by adding an edge connecting T-junctions  $A_1$  and  $A_2$ . See Remark 5.15. (Color online)

in which  $\alpha_i \in \mathbb{R}$  are coefficients to be calculated,  $e_T$  is the number of anchors in the T-mesh and  $e_T \leq e_1 \cdot e_2$ . For  $i = 1, \dots, e_T$ ,  $B_i(x_1, x_2)$  is called a blending function corresponding to the  $i^{\text{th}}$  anchor point  $A_i$  and defined as

$$B_i(x_1, x_2) = N_i[\mathbf{u}_i](x_1)N_i[\mathbf{v}_i](x_2), \quad (13)$$

where  $\mathbf{u}_i = [u_{\delta_1^i}, u_{\delta_2^i}, u_{\delta_3^i}, u_{\delta_4^i}, u_{\delta_5^i}]$  and  $\mathbf{v}_i = [v_{\beta_1^i}, v_{\beta_2^i}, v_{\beta_3^i}, v_{\beta_4^i}, v_{\beta_5^i}]$  are sub-sequences of  $U$  and  $V$ , respectively, where  $h\nu(A_i) = (\delta_1^i, \delta_2^i, \delta_3^i, \delta_4^i, \delta_5^i)$  and  $\nu\nu(A_i) = (\beta_1^i, \beta_2^i, \beta_3^i, \beta_4^i, \beta_5^i)$ ,  $N_i[\mathbf{u}_i]$  and  $N_i[\mathbf{v}_i]$  are cubic-basis functions defined on  $\mathbf{u}_i$  and  $\mathbf{v}_i$ , respectively. We call  $\mathbf{u}_i$  and  $\mathbf{v}_i$  local knot vectors of  $N_i$  in  $x_1$  and  $x_2$  coordinates, respectively. As mentioned in [26], blending functions (13) in T-splines are not necessarily linearly independent. We shall discuss it in more details in Subsection 5.4. One should anchor that the coefficients  $\alpha_i, i = 1, \dots, e_T$ , for a given set of sample points  $(x_1^i, x_2^i, y^i), y^i = f(x_1^i, x_2^i), i = 1, \dots, \Delta, \Delta \geq e_T$ , and a given set of blending functions with their corresponding local knot vectors are calculated through minimizing least square error by adopting problem (10) for a bi-variate case.

If the T-mesh in Figure 5 (a) consists of all  $(i, j) \in I_1 \times I_2$ , then T-splines representation in 5.10 is equal with Definition 4.1.

One should note that there is a one-to-one correspondence between anchors in a T-mesh and blending function in (12). The advantage of T-splines (i.e., Definition 5.10) over bi-variate B-splines (i.e., Definition 4.1) lies in the fact that refinement has a local effect on T-splines, because T-splines does not follow a tensor product structure. Therefore a T-mesh can form an incomplete net grid as shown in Figure 5 (a). Figure6 (a) depicts an

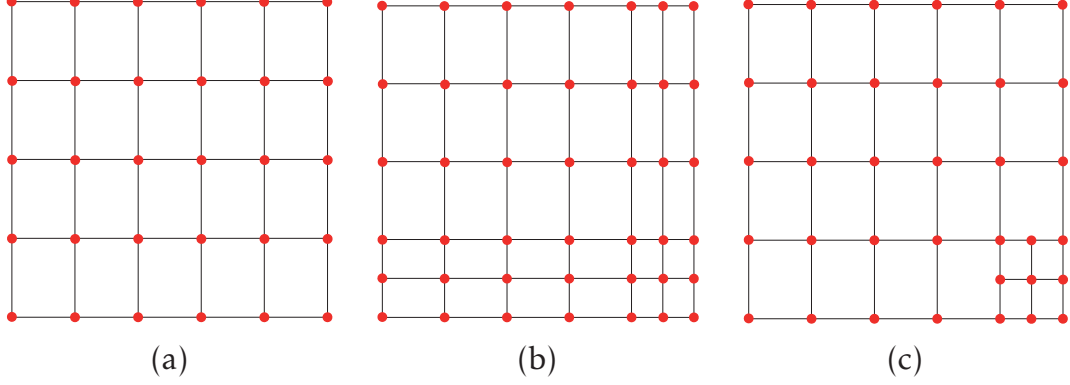


Figure 6: (a) Initial mesh. (b) bi-variate B-splines refinement. (c) T-spline refinement. (Color online)

initial mesh. Refining this mesh by bi-variate B-splines and T-splines leads to generating a new mesh and a T-mesh as shown in Figures 6 (b) and 6 (c), respectively. As can be seen, the refinement in Figure 6 (b) has a global effect whereas it is local in Figure 6 (c).

Suppose  $\mathbf{u}_i$  and  $\mathbf{v}_i$ ,  $i = 1, \dots, e_T$ , are given and  $\tilde{x}_{1,j} = x_1 - u_{\delta_j^i}$  and  $\tilde{u}_{j,k} = u_{\delta_j^i} - u_{\delta_k^i}$ . From Equations (7) and (8) with  $p = 3$ , we have

$$N_i[\mathbf{u}_i](x_1) = \begin{cases} 0 & x_1 = u_{\delta_1^i} = u_{\delta_2^i} = u_{\delta_3^i} = u_{\delta_4^i}, \\ \frac{(\tilde{x}_{1,1})^3}{\tilde{u}_{4,1}\tilde{u}_{3,1}\tilde{u}_{2,1}} & u_{\delta_1^i} \leq x_1 < u_{\delta_2^i}, \\ \frac{-(\tilde{x}_{1,1})^2\tilde{x}_{1,3}}{\tilde{u}_{4,1}\tilde{u}_{3,1}\tilde{u}_{3,2}} + \frac{-\tilde{x}_{1,4}\tilde{x}_{1,2}\tilde{x}_{1,1}}{\tilde{u}_{3,2}\tilde{u}_{4,2}\tilde{u}_{4,1}} + \frac{-\tilde{x}_{1,5}(\tilde{x}_{1,2})^2}{\tilde{u}_{5,2}\tilde{u}_{4,2}\tilde{u}_{3,2}} & u_{\delta_2^i} \leq x_1 < u_{\delta_3^i}, \\ \frac{\tilde{x}_{1,1}(\tilde{x}_{1,4})^2}{\tilde{u}_{4,1}\tilde{u}_{4,2}\tilde{u}_{4,3}} + \frac{\tilde{x}_{1,2}\tilde{x}_{1,4}\tilde{x}_{1,5}}{\tilde{u}_{4,3}\tilde{u}_{4,2}\tilde{u}_{5,2}} + \frac{(\tilde{x}_{1,5})^2\tilde{x}_{1,3}}{\tilde{u}_{5,2}\tilde{u}_{5,3}\tilde{u}_{4,3}} & u_{\delta_3^i} \leq x_1 < u_{\delta_4^i}, \\ \frac{-(\tilde{x}_{1,5})^3}{\tilde{u}_{5,2}\tilde{u}_{5,3}\tilde{u}_{5,4}} & u_{\delta_4^i} \leq x_1 < u_{\delta_5^i}, \\ 1 & x_1 = u_{\delta_2^i} = u_{\delta_3^i} = u_{\delta_4^i} = u_{\delta_5^i}, \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

Similarly,  $N_i[\mathbf{v}_i](x_2)$  is defined. In T-splines, local refinement means inserting a new anchor in the T-mesh and correspondingly adding a new blending function into the T-splines equation. In the next subsection we discuss it.

### 5.3 Local refinement of T-splines

A local refinement of T-splines consists of blending functions refinement and T-mesh refinement [22]. When inserting a new anchor in a T-mesh we refine those blending functions which their local knot vectors are changed by presence of the new anchor. Refining these blending functions may possibly result in generating some new blending functions. Since there is a one-to-one correspondence between anchors and blending functions, we

must add some new anchors corresponding to those new generated blending functions. We first discuss blending function and then refining T-mesh.

### 5.3.1 Blending function refinement

Suppose  $\mathbf{u} = [u_1, u_2, u_3, u_4, u_5]$  is a hypothetical local knot vector in coordinate  $x_1$  corresponding to  $N[\mathbf{u}](x_1)$ ;  $\tilde{\mathbf{u}}$  is a knot vector including  $l$  knots and  $\mathbf{u}$  is a sub-sequence of  $\tilde{\mathbf{u}}$ . Then, as mentioned in [22], cubic-basis function  $N[\mathbf{u}](x_1)$  is written as a linear combination of  $l - 4$  cubic-basis functions with local knot vector of length 5 which are sub-sequences of  $\tilde{\mathbf{u}}$ . We explain cubic-basis function refinement for  $l = 6$ . Cases with  $l > 6$  can be obtained repeatedly using the following equations.

If  $\tilde{\mathbf{u}} = [u_1, \tilde{u}, u_2, u_3, u_4, u_5]$ , then

$$N[\mathbf{u}](x_1) = \frac{\tilde{u} - u_1}{u_4 - u_1} N[u_1, \tilde{u}, u_2, u_3, u_4](x_1) + N[\tilde{u}, u_2, u_3, u_4, u_5](x_1), \quad (15)$$

if  $\tilde{\mathbf{u}} = [u_1, u_2, \tilde{u}, u_3, u_4, u_5]$ , then

$$N[\mathbf{u}](x_1) = \frac{\tilde{u} - u_1}{u_4 - u_1} N[u_1, u_2, \tilde{u}, u_3, u_4](x_1) + \frac{u_5 - \tilde{u}}{u_5 - u_2} N[u_2, \tilde{u}, u_3, u_4, u_5](x_1), \quad (16)$$

if  $\tilde{\mathbf{u}} = [u_1, u_2, u_3, \tilde{u}, u_4, u_5]$ , then

$$N[\mathbf{u}](x_1) = \frac{\tilde{u} - u_1}{u_4 - u_1} N[u_1, u_2, u_3, \tilde{u}, u_4](x_1) + \frac{u_5 - \tilde{u}}{u_5 - u_2} N[u_2, u_3, \tilde{u}, u_4, u_5](x_1), \quad (17)$$

if  $\tilde{\mathbf{u}} = [u_1, u_2, u_3, u_4, \tilde{u}, u_5]$ , then

$$N[\mathbf{u}](x_1) = N[u_1, u_2, u_3, u_4, \tilde{u}](x_1) + \frac{u_5 - \tilde{u}}{u_5 - u_2} N[u_2, u_3, u_4, \tilde{u}, u_5](x_1), \quad (18)$$

and if  $\tilde{u} \leq u_1$  or  $\tilde{u} \geq u_5$ , then  $N[\mathbf{u}](x_1)$  does not changed.

A blending function  $B(x_1, x_2)$  can be refined in either  $x_1$  or  $x_2$  coordinate. Therefore, the corresponding cubic-basis function as shown above is split into two cubic-basis functions. This means that two new blending functions are resulted which sum to the blending function  $B(x_1, x_2)$ . By adding new knots, a set of blending functions are obtained that sum to the original one. In Figure 7 (a), we refine the blending function  $B_1$  by first inserting a new knot in the corresponding local knot vector in  $x_1$  coordinate and then in the corresponding local knot vector in  $x_2$  coordinate. This leads to generating four blending functions as shown in Figure 7 (b). By applying equations (15)- (18), in this figure we have

$$B_1(x_1, x_2) = p_1 \tilde{B}_1(x_1, x_2) + p_2 \tilde{B}_2(x_1, x_2) + p_3 \tilde{B}_3(x_1, x_2) + p_4 \tilde{B}_4(x_1, x_2). \quad (19)$$

in which the coefficients  $p_i, i = 1, \dots, 4$ , are obtained from the above equations.

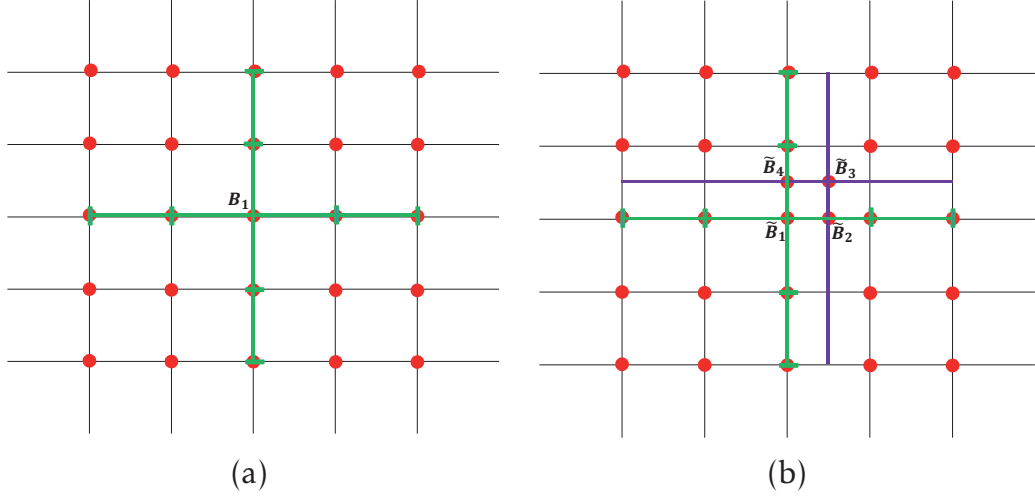


Figure 7: (a) Blending function  $B_1$ . (b) Refinement of  $B_1$  in  $x_1$  and  $x_2$  coordinates leads to generating new blending function  $\tilde{B}_i, i = 1, \dots, 4$ . (Color online)

### 5.3.2 T-mesh refinement

In what follows, we present a T-mesh refinement discussed in [22]. As mentioned earlier, there is a one-to-one correspondence between blending functions and anchors in a T-mesh. According to Definition 5.10, in a T-mesh, all anchors corresponding to all blending functions satisfy Definition 5.9. When adding a new anchor in the current T-mesh, three cases may occur:

- Case 1: The local knot vector of an existing blending function corresponding to an anchor does not include a knot dictated by Definition 5.9. For example, in Figure 8 (b), the local knot vector of the blending function  $B_1$  in  $x_2$  coordinate is  $[v_6, v_7, v_9, v_{10}, v_{11}]$ . This local knot vector is missing knot  $v_8$  to follow Definition 5.9.
- Case 2: The local knot vector of an existing blending function (corresponding to an existing anchor) contains a knot which violates Definition 5.9 in the current T-mesh. For instance, in Figure 8 (d), the local knot vector of the blending function  $\tilde{B}$  in  $x_1$  coordinate is  $[u_6, u_7, u_8, u_9, u_{10}]$ . Thus, the local knot vector violates Definition 5.9.
- Case 3: A new blending function is obtained which does not have any corresponding anchor in the T-mesh. For example, in Figure 8 (d), a new blending function with local knot vectors  $[u_7, u_8, u_9, u_{10}, u_{11}]$  and  $[v_6, v_7, v_8, v_9, v_{10}]$  is resulted. However, as can be seen, there is not any anchor in the T-mesh corresponding to this new blending function.

If by the result of blending function modification, none of the above cases occurs, then the T-mesh is valid. Otherwise, as discussed in [22], we conduct the following steps until the above cases are resolved:

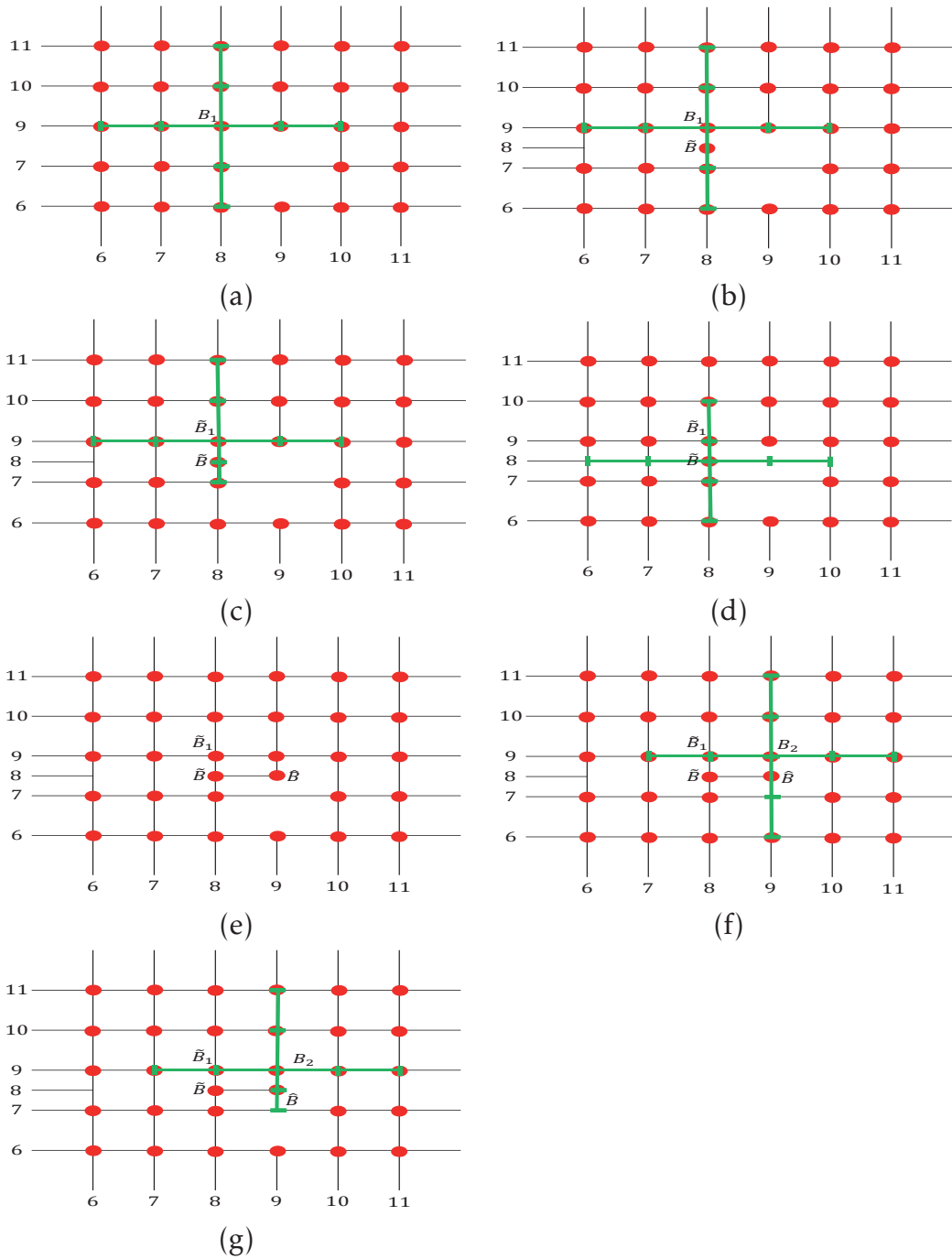


Figure 8: T-mesh refinement. (Color online)

- Step 1: Insert all desirable anchors into the T-mesh and modify the corresponding blending functions.
- Step 2: If Case 1 occurs, we insert necessary knots into that blending function.
- Step 3: If Case 2 occurs, we insert a new anchor in an appropriate place into the T-mesh.
- Step 4: Repeat Steps 2 and 3 until all cases are resolved.

Repeating Steps 2 and 3 automatically resolves Case 3. We explain the algorithm with an example.

**Example 5.11.** Figure 8 (a) depicts some part of AR of a hypothetical T-mesh to be refined. As can be seen, none of the above cases occurs. We wish to add a new anchor corresponding to a pair of knots  $(v_8, u_8)$ . This is equal with adding a new knot  $v_8$  in  $V$ . We denote the blending function corresponding to this new anchor by  $\widetilde{B}$ . When we add the new anchor, (Figure 8 (b)) without refining any blending function, some of the above cases occurs. Since the coordinate of the new anchor corresponding to  $\widetilde{B}$  is  $(8, 8)$ , Case 1 happens in four blending functions, i.e., those corresponding to the anchors at  $(8, 6)$ ,  $(8, 7)$ ,  $(8, 9)$  and  $(8, 10)$ . The blending functions corresponding to these anchors miss the new knot  $v_8$ . This case is resolved by adding the new knot  $v_8$  in these blending function through the blending function refinement discussed in Subsection 5.3.1. In Figure 8 (b), the blending function corresponding to the anchor  $(8, 9)$  is  $B_1 = N[u_6, u_7, u_8, u_9, u_{10}](x_1)N[v_6, v_7, v_9, v_{10}, v_{11}](x_2)$ . By adding the new knot  $v_8$  in the local knot vector of this blending function in  $x_2$  coordinate, it is decomposed into two new blending functions, i.e.,  $aN[u_6, u_7, u_8, u_9, u_{10}](x_1)N[v_6, v_7, v_8, v_9, v_{10}](x_2)$  and  $bN[u_6, u_7, u_8, u_9, u_{10}](x_1)N[v_7, v_8, v_9, v_{10}, v_{11}](x_2)$  in which the coefficients  $a$  and  $b$  are obtained through equation (16).

The blending function  $bN[u_6, u_7, u_8, u_9, u_{10}](x_1)N[v_7, v_8, v_9, v_{10}, v_{11}](x_2)$  (i.e.,  $\widetilde{B}_1$  in Figure 8 (c)) satisfies Definition 5.9. Analogously, the refinement of the blending functions corresponding to the anchors  $(8, 6)$ ,  $(8, 7)$  and  $(8, 10)$  follows Definition 5.9. Nevertheless, as shown in Figure 8 (d), Case 2 occurs for the blending function  $aN[u_6, u_7, u_8, u_9, u_{10}](x_1)N[v_6, v_7, v_8, v_9, v_{10}](x_2)$  (i.e.,  $\widetilde{B}$ ), since the local knot vector of this blending function in  $x_1$  coordinate does not satisfy Definition 5.9. We must resolve this situation by adding a new anchor in the current T-mesh.

By adding a new anchor with coordinate  $(9, 8)$  (and denoting the corresponding blending function by  $\widetilde{B}$ ) in the T-mesh in Figure 8 (d), the T-mesh shown in Figure 8 (e) is obtained. This leads to resolve Case 2, but create Case 1 in the blending function corresponding to the anchor  $(9, 9)$  as shown in Figure 8 (f). In this figure, the local knot vector of the blending function  $B_2$  in  $x_2$  coordinate misses the knot  $v_8$ , thus Definition 5.9 is not called for this local knot vector. This is rectified by inserting the knot  $v_8$  into this local knot vector as shown in Figure 8 (g). As can be seen in this figure, none of the above cases occurs. Thus the resulted T-mesh is a valid T-mesh.

As discussed in [22], it is guaranteed that the above described algorithm always terminates. In the worst situation, the algorithm converts a T-mesh into a full grid mesh. In a full grid mesh, all basis functions are linearly independent. However, in T-splines, cases may arise that the blending functions are not linearly independent (see e.g., [26]). Actually, the above algorithm does not guarantee that the blending functions obtained



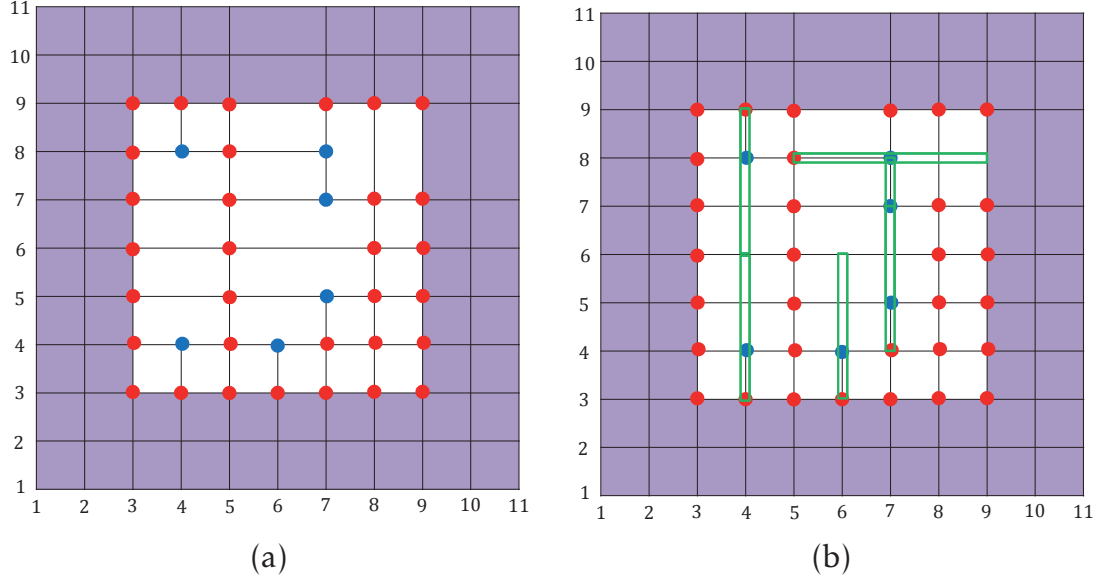


Figure 9: (a) A T-mesh. T-junctions are in blue color. (b) T-junction extensions (shown in green) in the T-mesh of (a). (Color online)

are linearly independent as discussed in [26]. In this reference, a sub-class of T-splines called *analysis-suitable T-splines* is introduced in which blending functions are linearly independent and form basis functions for T-splines. We discuss it in the next subsection.

## 5.4 Analysis-Suitable T-splines

The matter of linearly independent blending function has been investigated in [26]. As a result, a sub-class of T-splines termed *analysis-suitable T-splines (AST-splines)* is introduced. In AST-splines, blending functions are linearly independent and form basis functions [26]. Before defining AST-splines, we need to define *T-junction extension* for a T-junction. Let  $\mathcal{TJ}$  be the set of all T-junctions (i.e., vertices with valence three) in  $AR$ . Following [27], we adopt the notation  $\perp, \top, \vdash, \dashv$  to show the four possible orientations of the T-junctions.

**Definition 5.12.** For the sake of simplicity and following [27], suppose  $\tau = \{(\bar{i}, \bar{j})\} \in \mathcal{TJ}$  and is type of  $\dashv$ . It is obvious that  $\bar{i}$  is one of the components of  $hJ(\bar{j})$ . The 4 consecutive indices  $i_1, i_2, i_3, i_4$  are extracted from  $hJ(\bar{j})$  such that  $\bar{i} = i_2$ . Let  $\text{ext}^e(\tau) = [i_1, \bar{i}] \times \{\bar{j}\}$ ,  $\text{ext}^f(\tau) = [\bar{i}, i_4] \times \{\bar{j}\}$  and  $\text{ext}(\tau) = \text{ext}^e(\tau) \cup \text{ext}^f(\tau)$ , in which  $\text{ext}^e(\tau)$  is called edge-extension,  $\text{ext}^f(\tau)$  is called face-extension and  $\text{ext}(\tau)$  is called T-junction extension of  $\tau$ .

Analogously, other types of T-junction extensions are defined. T-junctions of types  $\vdash, \dashv$  (respectively,  $\perp, \top$ ) are called *horizontal* (respectively, *vertical*). Figures 9 (a) and (b) show a T-mesh and corresponding T-junction extensions, respectively.

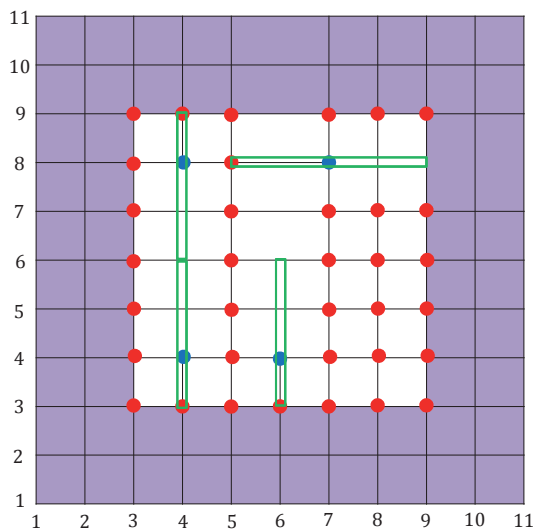


Figure 10: An AST-mesh. (Color online)

**Definition 5.13.** Following [28], an analysis-suitable T-mesh (AST-mesh)  $\mathcal{T} \in AD^+$  is a T-mesh in which no horizontal T-junction extension touches a vertical T-junction extension. The T-spline corresponding to an analysis-suitable T-mesh is called an analysis-suitable T-splines.

The authors in [26] prove that blending functions in AST-splines are linearly independent and form basis functions. Figures 10 shows an AST-mesh.

As discussed in Subsection 5.3, a new anchor can be added into a T-mesh and a new refined T-mesh can be obtained. However, this refinement may result a T-mesh whose corresponding blending functions are not linearly independent. This can be resolved by adding some new anchors in appropriate places in the T-mesh to become an AST-mesh. For instance, the T-mesh shown in Figure 9 (b) is not an AST-mesh. By adding a new anchor centered at (7, 6), it is converted into an AST-mesh as shown in Figure 10. In [29], a heuristic method is developed by which a T-mesh is converted into an AST-mesh by adding new anchors in appropriate places.

**Remark 5.14.** The definition of AST-mesh shows that a new anchor should be added on an existing edge. Thus, it is not possible to add an isolated anchor, i.e., a anchor that is not located on any edge; Because, according to Definition 5.13, for in an isolated anchor as shown in Figure 11, the vertical T-junction extension (which has two face extensions) touches the horizontal T-junction extension (which has two face extensions as well).

**Remark 5.15.** The restriction to  $AD^+$  (Definition 5.6) can be seen from Figures 12 (a) and (b). The AD T-mesh shown in Figure 12 (a) does not belong to  $AD^+$ , since it includes some facing T-junctions (shown in blue). It is not also an AST-mesh since, for instance, the horizontal T-junction extension corresponding to the T-junction centered on (7, 7) touches the vertical T-junction extension corresponding to the T-junction centered on (6, 6). However, by adding the missing edges (shown in green in Figure 12 (b)), it is turned into an  $AD^+$  T-mesh which is also an AST-mesh (see Figure 12 (b)).

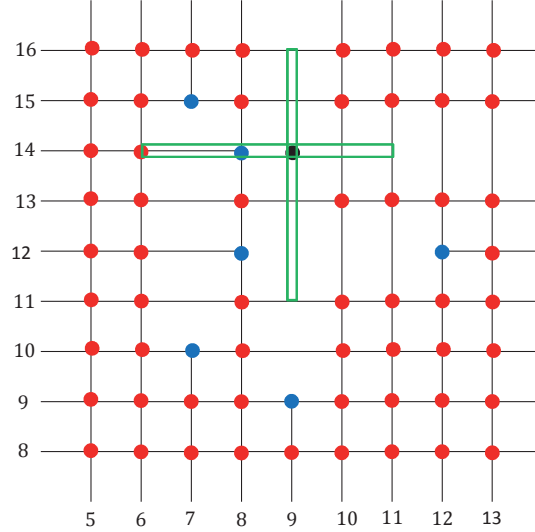


Figure 11: An isolated anchor shown in black. (Color online)

So far, we have introduced T-splines, AST-splines and matters related to linearly independent blending functions. In the next section, we discuss how AST-splines can be employed to developed surrogate-based methods for computationally expensive MOPs.

## 5.5 Application of T-splines in ANOVA-MOP

As discussed in [4], in the ANOVA-MOP method, a surrogate function is built for each computationally expensive function. We expect that for computationally expensive MOPs including non-differentiable functions, AST-splines can be considered as a potential method to conduct sensitivity analysis on such functions. In this way, sub-problems are formed accurately by ANOVA-MOP and solutions obtained can be reliable. The beauty of AST-splines lies in the fact that computationally expensive functions can be approximated with fewer bases functions in comparison to bi-variate B-splines. To be able to incorporate AST-splines in the ANOVA-MOP method, the following important challenges must be first tackled:

- Challenge 1: As mentioned earlier, the minimum number of required sample points is equal to the number of blending functions. By adding a new anchor into a T-mesh, some new anchors may also be added into the T-mesh as discussed in Subsection 5.3. Moreover, when converting the T-mesh into an AST-mesh, some new anchors are added as discussed in Subsection 5.4. Nevertheless, to the best of our knowledge, there is no discussion on where to add new anchors to improve the accuracy of a surrogate function such that the number of new added anchors is minimal.
- Challenge 2: In the literature, analytical properties of AST-splines such as linearly

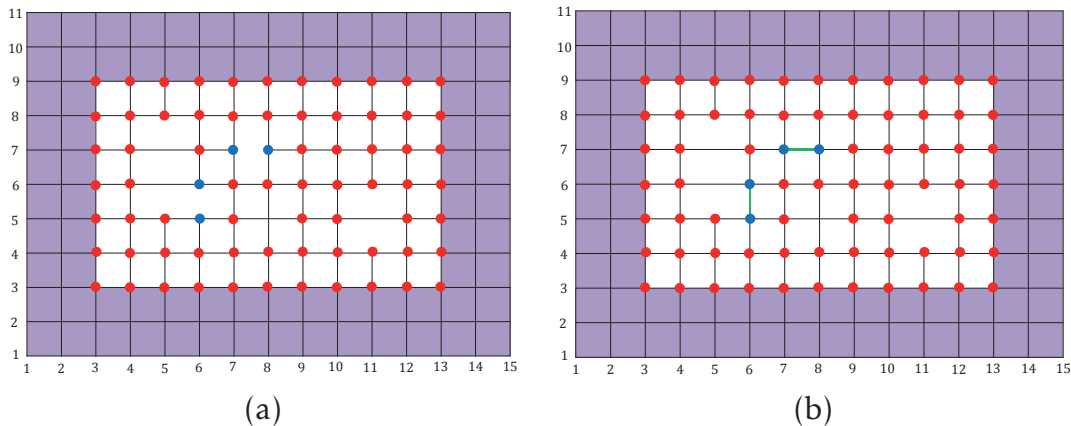


Figure 12: (a) An  $AD$  T-mesh which is neither  $AD^+$  nor an AST-mesh. T-junctions are in blue. (b) An  $AD^+$  T-mesh which is an AST-mesh obtained by adding missing edge (in green) in the T-mesh of (a). (Color online)

independent blending functions are limited for functions with two and three decision variables [18]. Computationally expensive MOPs typically have more than three decision variables. Thus, we should extend such analytical properties of AST-splines for functions with more than three decision variables.

## 6 Conclusions

Sensitivity analysis plays an important part in the ANOVA-MOP method developed in [4]. We have discussed how sensitivity analysis methods developed in the literature can be applied in ANOVA-MOP and paid attention to a method called T-splines. In particular, we discussed a sub-class of T-splines termed analysis-suitable T-splines (AST-splines) in which bases functions are linearly independent. Since a refinement in AST-splines is local, computationally expensive functions can be approximated with fewer bases functions in comparison to bi-variate B-splines. We expect that AST-splines can be considered as a potential metamodeling method to approximation terms in the functional ANOVA decomposition and to conduct sensitivity analysis on computationally expensive non-differentiable functions. Analytical properties of AST-splines (T-splines) have been investigated for functions with at most three variables. However, computationally expensive MOPs typically have more than three decision variables. Thus, to become applicable in ANOVA-MOP, analytical properties of T-splines for functions with more than three decision variables must be developed. Then, by incorporating T-splines in ANOVA-MOP, we expect that high-dimensional computationally expensive MOPs can be solved effectively.

## Acknowledgement

Mohammad Tabatabaei was supported by the financial support the Academy of Finland project No. 287496. Both authors were supported by the Early Career Scheme (ECS) project No. 21201414 sponsored by the Research Grants Council of Hong Kong.

## References

- [1] M. Tabatabaei, J. Hakanen, M. Hartikainen, K. Miettinen, and K. Sindhya. A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods. *Structural and Multidisciplinary Optimization*, 52(1):1–25, 2015.
- [2] A. Saltelli, K. Chan, and M. Scott, editors. *Sensitivity Analysis*. Wiley, New York, 2000.
- [3] J. Wu and S. Azarm. Metrics for quality assessment of a multiobjective design optimization solution set. *Journal of Mechanical Design*, 123(1):18–25, 2000.
- [4] M. Tabatabaei, A. Lovison, M. Tan, M. Hartikainen, and K. Miettinen. ANOVA-MOP: ANOVA decomposition for multiobjective optimization. *Submitted*, 2016.
- [5] M. Tan. Sequential Bayesian polynomial chaos model selection for estimation of sensitivity indices. *SIAM/ASA Journal on Uncertainty Quantification*, 3:146–168, 2015.
- [6] A. Saltelli, S. Tarantola, and K. P. S. Chan. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics*, 41(1):39–56, 1999.
- [7] I. M. Sobol. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55:271–280, 2001.
- [8] M. J. W. Jansen. Analysis of variance designs for model output. *Computer Physics Communications*, 117:35–43, 1999.
- [9] M. D. Morris, L. M. Moore, and M. D. McKay. Using orthogonal arrays in the sensitivity analysis of computer models. *Technometrics*, 50(2):205–215, 2008.
- [10] A. Owen. Variance components and generalized Sobol’ indices. *SIAM/ASA Journal on Uncertainty Quantification*, 1(1):19–41, 2013.
- [11] J. E. Oakley and A. O’Hagan. Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(3):751–769, 2004.
- [12] L. Le Gratiet, C. Cannamela, and B. Iooss. A Bayesian approach for global sensitivity analysis of (multifidelity) computer codes. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):336–363, 2014.
- [13] S. Touzani and D. Busby. Smoothing spline analysis of variance approach for global sensitivity analysis of computer codes. *Reliability Engineering & System Safety*, 112:67–81, 2013.

- [14] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering and System Safety*, 93(7):964–979, 2008.
- [15] S. Da Veiga, F. Wahl, and F. Gamboa. Local polynomial estimation for sensitivity analysis on models with correlated inputs. *Technometrics*, 51(4):452–463, 2009.
- [16] B. J. Reich, C. B. Storlie, and H. D. Bondell. Variable selection in bayesian smoothing spline ANOVA models: Application to deterministic computer codes. *Technometrics*, 51(2):110–120, 2009.
- [17] E. Borgonovo and E. Plischke. Sensitivity analysis: A review of recent advances. *European Journal of Operational Research*, 248(3):869 – 887, 2016.
- [18] P. Morgenstern. Globally structured three-dimensional analysis-suitable T-splines: Definition, linear independence and  $m$ -graded local refinement. *SIAM Journal on Numerical Analysis*, pages 2163–2186, 2016.
- [19] L. Piegl and W. Tiller. *The NURBS Book (2nd Ed.)*. Springer-Verlag New York, Inc., New York, NY, USA, 1997.
- [20] K. Höllig and J. Hrner. *Approximation and Modeling with B-Splines*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2013.
- [21] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCs. *ACM Trans. Graph.*, 22(3):477–484, 2003.
- [22] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. In *ACM SIGGRAPH 2004 Papers, SIGGRAPH '04*, pages 276–283, New York, NY, USA, 2004. ACM.
- [23] J. Z. Huang, C. O. Wu, and L. Zhou. Polynomial spline estimation and inference for varying coefficient models with longitudinal data. *Statistica Sinica*, 14(3):763–788, 2004.
- [24] M. Landajo, J. De Andrés, and P. Lorca. Measuring firm performance by using linear and non-parametric quantile regressions. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 57(2):227–250, 2008.
- [25] D. C. Thomas and M. A. Scott. Isogeometric analysis based on T-splines. In *Isogeometric Methods for Numerical Simulation*, pages 205–232. Springer Vienna, Vienna, 2015.
- [26] X. Li, J. Zheng, T. W. Sederberg, T. J. R. Hughes, and M. A. Scott. On linear independence of T-spline blending functions. *Computer Aided Geometric Design*, 29(1):63–76, 2012.
- [27] L. Beirão DA Veiga, A. Buffa, G. Sangalli, and R. Vázquez. Analysis-suitable T-splines of arbitrary degree: Definition, linear independence and approximation properties. *Mathematical Models and Methods in Applied Sciences*, 23(11):1979–2003, 2013.
- [28] L. Beirão DA Veiga, A. Buffa, D. Cho, and G. Sangalli. Analysis-suitable T-splines are dual-compatible. *Computer Methods in Applied Mechanics and Engineering*, 249-252:42–51, 2012.

- [29] M.A. Scott, X. Li, T.W. Sederberg, and T.J.R. Hughes. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering*, 213–216:206–222, 2012.