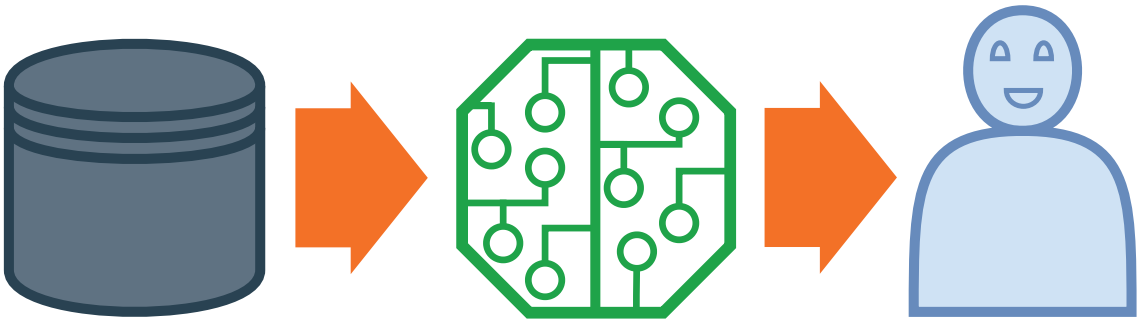
JYU DISSERTATIONS 456

Atanu Mazumdar

Novel Approaches for Offline Data-Driven Evolutionary Multiobjective Optimization



UNIVERSITY OF JYVÄSKYLÄ
FACULTY OF INFORMATION TECHNOLOGY

JYU DISSERTATIONS 456

Atanu Mazumdar

Novel Approaches for Offline Data-Driven Evolutionary Multiobjective Optimization

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston vanhassa juhlasalissa S212
joulukuun 9. päivänä 2021 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Seminarium, auditorium S212, on December 9, 2021 at 12 o'clock noon.

JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2021

Editors
Timo Männikkö
Faculty of Information Technology, University of Jyväskylä
Ville Korkiakangas
Open Science Centre, University of Jyväskylä

Cover picture by Atanu Mazumdar.

ISBN 978-951-39-8919-4 (PDF)
URN:ISBN:978-951-39-8919-4
ISSN 2489-9003

Permanent link to this publication: http://urn.fi/URN:ISBN:978-951-39-8919-4

ABSTRACT

Mazumdar, Atanu
Novel Approaches for Offline Data-Driven Evolutionary Multiobjective Optimization
Jyväskylä: University of Jyväskylä, 2021, 74 p. (+included articles)
(JYU Dissertations
ISSN 2489-9003; 456)
ISBN 978-951-39-8919-4 (PDF)

Most multiobjective evolutionary algorithms (MOEAs) assume that analytical functions or simulation models are available while solving a multiobjective optimization problem (MOP). However, in some cases we must start with data and build approximation models known as surrogates that are later used to solve the MOP by an MOEA. These types of problems are called data-driven MOPs. This thesis is devoted to solving so-called offline data-driven MOPs that are particularly challenging as no new data is available during the optimization process.

The author first presents approaches to utilize the uncertainty in the prediction of Kriging or Gaussian process (GP) surrogates as additional objectives. However, these approaches increase the complexity of the MOP being solved. Hence, the author proposes probabilistic selection approaches that can be embedded in a decomposition-based MOEA without further analytical derivations. These approaches utilize Monte Carlo sampling and kernel density estimation to calculate the probability of selection criterion of the MOEA and later select individuals based on them. Next, the author proposes an interactive optimization framework that utilizes decision maker's preferences for uncertainties in addition to preferences for objective values. The framework was further extended to use probabilistic selection approaches for a decomposition-based MOEA and a custom reference vector adaptation technique to consider uncertainty in the solutions during the adaptation process.

Building GPs with all the provided data becomes computationally expensive when the size of the data is large. Hence, the author finally proposes treed GP surrogates for multiobjective optimization (TGP-MO). They can be built with a relatively low computational cost and have a good accuracy exclusively in the regions around the optimal solutions. This thesis provides multiple novel approaches and detailed experimental studies for solving offline data-driven MOPs with decision support that will enhance real-world problem-solving capabilities.

Keywords: metamodelling, surrogates, Pareto optimality, Kriging, Gaussian processes, evolutionary algorithm, decision making, uncertainty, interactive methods, preference information

TIIVISTELMÄ (ABSTRACT IN FINNISH)

Mazumdar, Atanu
Uusia lähestymistapoja datapohjaiseen monitavoiteoptimointiin kun uutta data ei ole saatavilla
Jyväskylä: University of Jyväskylä, 2021, 74 s. (+artikkelit)
(JYU Dissertations
ISSN 2489-9003; 456)
ISBN 978-951-39-8919-4 (PDF)

Monet monitavoiteoptimointiin kehitetyt evoluutioalgoritmit olettavat, että optimoitavien funktioiden analyyttiset lausekkeet tai simulaatiomallit ovat ratkaisuprosessin aikana käytettävissä. Joissakin tapauksissa lähtökohta kuitenkin on aiemmin kerätty data, johon täytyy ensin sovittaa sijaismalleja. Näitä malleja voidaan sitten käyttää monitavoiteoptimointiongelman muotoilussa ja ratkaisemisessa. Tällaisia ongelmia kutsutaan datapohjaisiksi monitavoiteoptimointiongelmiksi. Tässä väitöskirjassa käsitellään ns. erillisiä datapohjaisia ongelmia, jotka ovat erityisen haastavia. Tässä erillisyys tarkoittaa sitä, ettei uutta dataa ole ratkaisuprosessin aikana saatavilla.

Ensin väittelijä esittelee lähestymistapoja, joissa käytetään Kriging- tai Gaussisten prosessien sijaismallien tuottamaa epävarmuustietoa optimoitavina lisäfunktioina. Ne kuitenkin lisäävät ratkaistavan monitavoiteoptimointiongelman kompleksisuutta. Tämän innoittamana väittelijä esittelee todennäköisuuteen perustuvia tapoja, jotka voidaan kätevästi sisällyttää hajotelmapohjaisiin evoluutioalgorimeihin. Tämän lisäksi väittelijä esittelee interaktiivisen optimointikehikon, jossa päätöksentekijä antaa preferenssi-informaatiota optimoitavien funktioiden lisäksi myös epävarmuuksille. Hän keskittyy hajontapohjaisiin evoluutioalgoritmeihin, joissa referenssivektoreita muokataan huomioimaan epävarmuus.

Gaussisten prosessien sovittaminen on laskennallisesti raskasta kun dataa on paljon. Tähän haasteeseen väittelijä vastaa lähestymistavalla, joka on laskennallisesti varsin edullinen mutta riittävän tarkka optimaalisten ratkaisujen läheisyydessä. Väitöskirja tarjoaa useita uusia lähestymistapoja ja niiden taustalle yksityiskohtaisia laskennallisia testejä erilliseen datapohjaiseen monitavoiteoptimointiin. Esitellyt päätöksenteon tukimenetelmät auttavat tekemään parempia datapohjaisia päätöksiä ja niitä voidaan soveltaa monilla käytännön sovellusaloilla.

Avainsanat: metamallinnus, sijaismallit, Pareto-optimaalisuus, Kriging, Gaussiset prosessit, evoluutioalgoritmit, päätöksenteko, epävarmuus, interaktiiviset menetelmät, preferenssi-informaatio

Author

Atanu Mazumdar
Faculty of Information Technology
University of Jyväskylä
Finland

Supervisors

Professor Kaisa Miettinen
Faculty of Information Technology
University of Jyväskylä
Finland

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

Dr. Tinkle Chugh
Department of Computer Science
University of Exeter
UK

Reviewers

Professor Joshua Knowles
Invenia Labs
Cambridge, UK

Professor Markus Olhofer
Honda Research Institute Europe
Germany

Opponent

Professor Richard Everson
Department of Computer Science
University of Exeter
UK

ACKNOWLEDGEMENTS

Firstly, I would like to sincerely thank my supervisors Prof. Kaisa Miettinen, Dr. Jussi Hakanen and Dr. Tinkle Chugh, for their constant support and guidance and for motivating me at every step of my PhD. Their expertise in data-driven multiobjective optimization and decision support helped me gain the experience and knowledge and develop the necessary skills in the field. In addition, I would like to thank Dr. Manuel Lopez Ibanez for providing his valuable support and collaboration in the field.

I would like to express my sincere gratitude to Prof. Joshua Knowles from Invenia Labs and Prof. Markus Olhofer from Honda Research Institute Europe for reviewing my dissertation and providing their valuable comments and suggestions that helped me improve my thesis. I also acknowledge Cuie Yang and Prof. Yaochu Jin for providing the codes for the transfer learning approach for offline data-driven multiobjective optimization. I also thank Prof. Jonathan E. Fieldsend for the codes for distance-based multi/many-objective point problems that were crucial for testing and benchmarking my proposed approaches.

My PhD thesis was supported by the University of Jyväskylä as a part of its profiling area (thematic research area) "Decision Analytics utilizing Causal Models and Multiobjective Optimization (DEMO)", for which I am incredibly thankful. Additionally, I am thankful for the computing resources provided by the Faculty of IT at my university and CSC – IT Center for Science that were crucial for my work. Overall I thank all the members of the Multiobjective Optimization Group who provided a conducive research environment through discussions and seminars. Last but not least, I thank my parents, Mrs. Anindita and Mr. Tapan Subhra Mazumdar, for their constant support and blessings.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

LIST OF INCLUDED ARTICLES

PI Atanu Mazumdar, Tinkle Chugh, Kaisa Miettinen, and Manuel López-Ibáñez. On Dealing with Uncertainties from Kriging Models in Offline Data-Driven Evolutionary Multiobjective Optimization. In Kalyanmoy Deb, Erik Goodman, Carlos A. Coello Coello, Kathrin Klamroth, Kaisa Miettinen, Sanaz Mostaghim, and Patrick Reed, editors, *Proceedings on Evolutionary Multi-Criterion Optimization*, pages 463–474. Springer, 2019.

PII Atanu Mazumdar, Tinkle Chugh, Jussi Hakanen, and Kaisa Miettinen. Probabilistic Selection Approaches in Decomposition-based Evolutionary Algorithms for Offline Data-Driven Multiobjective Optimization. *Conditionally accepted in IEEE Transactions on Evolutionary Computation*.

PIII Atanu Mazumdar, Tinkle Chugh, Jussi Hakanen, and Kaisa Miettinen. An Interactive Framework for Offline Data-Driven Multiobjective Optimization. In Bogdan Filipič, Edmondo Minisci, and Massimiliano Vasile, editors, *Proceedings on Bioinspired Optimization Methods and Their Applications*, pages 97–109. Springer, 2020.

PIV Atanu Mazumdar, Manuel López-Ibáñez, Tinkle Chugh, Jussi Hakanen, and Kaisa Miettinen. TGP-MO: Treed Gaussian Processes for Solving Offline Data-Driven Multiobjective Optimization Problems. *Submitted to a journal*.

1 INTRODUCTION

Most multiobjective evolutionary optimization algorithms (MOEAs) solve multiobjective optimization problems (MOPs) consisting of conflicting objectives, assuming that analytical functions or simulation models are available [34]. However, this assumption is not true for some real-world MOPs. Instead, we start the optimization process using the data obtained from a phenomenon, i.e. real-world processes, sensors, historical records or physical experiments [33]. These types of problems are known as data-driven optimization problems. The collected data may require pre-processing since it may have noise or errors. Later we build surrogate models (also known as meta-models) to approximate the *underlying* (or real) objective functions and/or constraint functions. To find the solutions of the MOP, we use the surrogates as objectives by embedding them in an MOEA. Using MOEAs is advantageous since they have proven beneficial in solving black-box MOPs [32], and can solve MOPs with non-convex and local optimal fronts. Since MOPs have conflicting objectives, we have multiple optimal solutions known as Pareto optimal solutions that represent the tradeoff between the objectives.

Data-driven optimization can be categorized into two types; *online* and *offline* optimization [33, 34]. In online data-driven optimization, new data can be generated during the optimization process by conducting further (expensive) function evaluations. Thus we can enhance the approximation quality of the surrogates and further improve the accuracy and objective values of the solutions. Online data-driven optimization is also known as surrogate assisted optimization in many pieces of literature, and there have been many works such as [11, 12, 13, 37, 70]. In offline data-driven optimization, we cannot actively sample or generate new data during the optimization process. Therefore, we cannot validate the accuracy of the surrogates and must utilize the available data in the best way possible. Thus the quality of the solutions is entirely dependent on the surrogates' approximation and the quality and quantity of the available offline data. The primary goals of this thesis are on the development of optimization approaches to make the best use of the offline data, and the development of decision support approaches for solving offline data-driven MOPs.

Most of the previous works in offline data-driven multiobjective optimization

were primarily focused on various surrogate modelling techniques and improving the approximation accuracy of the surrogates. For instance, in [10] the operating parameters of a blast furnace with eight objectives were optimized where the provided dataset was small in size and noisy. The work also reduced the number of decision variables using principal component analysis and used K-RVEA [11] for optimization. Another work that dealt with a small dataset was the optimization of fused magnesium furnaces in [28]. Synthetic data was generated using polynomial regression models, and Kriging models were later built using both the provided offline data and the generated synthetic data. The trauma system design problem is an example where the dataset is large, and function evaluations are expensive. A model management strategy was proposed in [67] to reduce the computation time by adjusting the fidelity of the surrogates using hierarchical clustering. A random forest assisted MOEA capable of handling constraints was proposed in [66] that solved the trauma system design problem. The work also proved its effectiveness in data-driven constrained multiobjective combinatorial optimization problems. In [72], an optimization approach based on transfer learning was proposed. The approach used two different surrogates for global and local search, and information was shared between the two. The approach found promising subregions using the coarse surrogate, and the knowledge about good solutions was transferred for improving the solutions further. An ensemble model was proposed in [68] where a large number of models were built using bootstrapped samples of the dataset. The proposed surrogate model was also used to solve an airfoil design optimization problem.

However, one of the primary challenges while solving an offline data-driven MOP is that the MOEAs generally do not have a mechanism to understand the error in the objective values of the solutions when they are evaluated with the underlying objectives. One of the ways to address the challenge is by using uncertainty in the prediction of surrogates such as Kriging. Previous works such as [31, 53] utilized the uncertainty information by calculating the probability of dominance and can be applied to dominance-based MOEAs. However, dominance-based MOEAs are not suitable for solving MOPs with a high of objectives ($>$ 3) [40, 63], also known as many-objective problems. A more practical way of solving many-objective problems is to use decomposition-based MOEAs [8, 17, 73]. A probabilistic selection criterion for decomposition-based MOEAs will be capable of handling uncertainties in the objectives and can solve many-objective problems. However, calculating the probabilities of selection criteria as proposed in [31] requires complex analytical derivations. Deriving the closed form for the probability of selection criterion becomes difficult for decomposition-based MOEAs that already employ complicated selection criteria.

Solving an MOP essentially means helping a human decision maker (DM) to find a solution with the most convincing objective values (also known as the most preferred solution) [49]. However, while using an MOEA to solve an offline data-driven MOP produces a set of solutions spread in the entire feasible region of the objective space. Decision making becomes a challenging task as the DM has to choose from a large set of solutions. Interactive multiobjective optimization

approaches are a suitable way to solve MOPs that enables the DM to find solutions in certain regions of the Pareto front. These approaches also enables the DM to learn about the problem [49]. There have been many developments in interactive MOEAs [30, 44, 52, 58], and interactive approaches such as [29, 41, 75] have been proposed for decomposition-based MOEAs. However, these approaches are not specifically built to solve offline data-driven MOPs and do not provide any mechanism to consider preferences for uncertainties from the DM.

Surrogates such as Kriging or Gaussian process regression (GP) [24] are a popular choice since they can provide uncertainty in their prediction [54]. However, they become computationally expensive to build when the size of the provided data is large. Existing offline data-driven approaches generally do not use Kriging surrogates (or use all the provided data) when the size of the dataset is large. A suitable alternative is to build sparse GPs [61, 62] as they use a small subset of data called support or inducing points to build approximation models. However, these surrogates are not tailored to solve offline data-driven MOPs and become computationally expensive to build for larger datasets.

The primary challenges of solving offline data-driven MOPs addressed in this thesis are:

1. how to enable MOEAs to utilize uncertainty information provided by surrogates,

2. how to support decision making by utilizing uncertainty information, and

3. how to handle offline datasets of large size?

This thesis addresses these challenges by proposing several approaches and embedding them in frameworks to solve offline data-driven MOPs. In our works, we used offline datasets generated from benchmark problems developed for multiobjective optimization. As the Pareto fronts and the objective functions are known, we could better understand the search behaviour of the proposed approaches for different types of problems. Using these test problems, we were able to investigate and analyze the effectiveness and performance of the proposed approaches. Considering further challenges of real-world offline data deserves more attention that could not be incorporated in a single thesis.

The thesis consists of four articles ([PI]-[PIV]) published and submitted in scientific journals and conference proceedings. A brief introduction of these articles is presented in Chapters 3-6. The first article [PI] about utilizing uncertainty information from Kriging surrogates for solving offline data-driven MOPs is introduced in Chapter 3. In [PI] the uncertainty in the predictions of Kriging surrogates was used as additional objectives. In other words, besides optimizing posterior predicted mean from Kriging surrogates fitted to the data, the uncertainty in the prediction of each surrogate was also minimized. The proposed approaches produced solutions closer to the Pareto front and with better accuracy compared to the generic approach. However, these approaches also increased the complexity of the MOP being solved by increasing the number of objectives.

In [PII] (introduced in Chapter 4) we proposed probabilistic selection approaches for decomposition-based MOEAs that explicitly considers the uncertainty from Kriging surrogates in the selection criterion. The probability of selection criterion of the decomposition-based MOEA is computed using numerical approximation methods such as Monte Carlo sampling [46] and kernel density estimation (KDE) [55]. The approaches are "plug and play" and can be applied to any decomposition-based MOEA without the need for further analytical derivations. We also demonstrated the capabilities of the probabilistic selection approaches for RVEA [8] and MOEA/D [73] by comparing them to their generic counterparts (approaches that used the original selection criteria of the MOEAs). Tests showed that the solutions found by the proposed probabilistic approaches were better in hypervolume and accuracy than their generic counterparts.

However, while solving real-world MOPs, the DM is interested in certain regions of the Pareto front. This also reduces the cognitive load of the DM as only certain solutions are shown to him/her [49]. Works such as [30, 44, 58] have proposed different ways to solve online data-driven MOPs interactively. However, to our knowledge, there have been no works to solve offline data-driven MOPs interactively. Hence, in [PIII] (introduced in Chapter 5) we proposed a framework to solve offline data-driven MOPs in an interactive way. The approach also utilized the preferences related to uncertainty provided by the DM. The approach deals with two types of preferences without significantly increasing the cognitive load in decision making. This framework was further extended to account for the uncertainty in objective values when the DM provides the preferences. The extended framework also utilized the probabilistic selection approaches proposed in [PII]. Tests conducted with artificial decision maker showed improved performance compared to the other interactive approaches.

In [PIV] (introduced in Chapter 6) a surrogate modelling approach called TGP-MO (treed GP surrogates for multiobjective optimization) was proposed to handle offline datasets of large size. Most of the traditional surrogate models [10, 28, 67] do not consider the tradeoff between the objectives while solving an offline data-driven MOP. In TGP-MO, we first split the decision space into sub-regions using regression trees. Then, GPs are built only in certain regions of the decision space that represent the tradeoff between the objectives. TGP-MO significantly outperformed full GPs [24] and sparse GPs [62] in surrogate building times for datasets of different sizes, sampling strategies, number of objectives and decision variables for various benchmark problems. It should be noted that in [PIV] and Chapter 6, we refer to Kriging as GP. (The article devoted to this work was submitted to a machine learning journal, and the readers are more familiar with the term GP.)

To summarize, in this thesis, we first studied the effects of utilizing uncertainty information from Kriging surrogates and proposed a few approaches in [PI]. The identified drawbacks in [PI] were tackled in [PII], where we proposed the probabilistic selection approaches for decomposition-based MOEAs. We proposed a framework in [PIII] to enable the DM to provide preferences for objectives and uncertainties. This framework was further extended to utilize the probabilistic

selection approach proposed in [PII] and adjust the diversity of the solutions based on the uncertainties in objective values. Finally, to explore the challenges in handling large datasets while solving offline data-driven MOPs, TGP-MO surrogates were proposed in [PIV].

2 BACKGROUND

In this chapter, the author introduces the key concepts and terminologies that were used in this thesis. A brief description of Kriging [24], decomposition-based MOEAs [63], probabilistic selection [31], and interactive decomposition-based MOEA [29] is provided as they were the primary building blocks of our works.

2.1 Offline Data-Driven MOP and Basic Concepts

While solving an offline data-driven MOP, analytical expression or simulation models are not available or cannot be accessed during the optimization. Instead, we are provided with (pre-collected) data of the phenomenon that is composed of *underlying objective functions*. We can denote the underlying MOP as follows:

$$\begin{aligned} &\text{minimize} && \{f_1(\mathbf{x}), \ldots, f_K(\mathbf{x})\} \\ &\text{subject to} && \mathbf{x} \in \Omega, \end{aligned} \tag{1}$$

where $K \geq 2$ is the total number of objectives, and Ω is the feasible region of the decision space $\mathbb{R}^n$. For a feasible decision vector $\mathbf{x}$, the corresponding objective vector is $\mathbf{f}(\mathbf{x})$, that comprises of the underlying objective (function) values $\{f_1(\mathbf{x}), \ldots, f_K(\mathbf{x})\}$.

A solution $\mathbf{x}_1 \in \Omega$ dominates another solution $\mathbf{x}_2 \in \Omega$ if $f_k(\mathbf{x}_1) \leq f_k(\mathbf{x}_2)$ for all $k = 1, \ldots, K$ and $f_k(\mathbf{x}_1) < f_k(\mathbf{x}_2)$ for at least one $k = 1, \ldots, K$. A solution of an MOP is called nondominated if no other feasible solution dominates it. An MOEA typically produces solutions that are nondominated within the set of solutions it has found. The solutions of (1) that are nondominated in Ω are also called Pareto optimal solutions. The solutions of MOEAs is referred to approximated Pareto optimal ones in this thesis. The set of solutions in the objective space is called the Pareto front, and the corresponding set of decision vectors is the Pareto set.

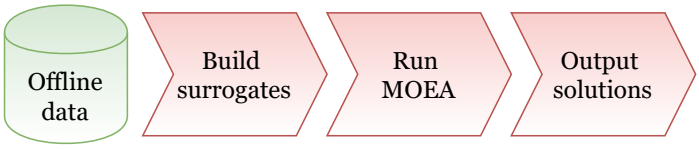


FIGURE 1 Flowchart of a generic offline data-driven multiobjective optimization approach.

Generic Offline Data-Driven Multiobjective Optimization

A generic approach to solve offline data-driven MOPs is shown in Figure 1. As described in [33, 68], the optimization process can be divided into three stages. They are: (a) data collection, (b) building surrogates, and (c) optimization using an MOEA. We start with an offline dataset consisting of N_D samples. Each sample consists of a decision vector $\mathbf{x}$ and its corresponding objective vector $\mathbf{f}(\mathbf{x})$ as a tuple of two matrices: (X, Y) where $X \in \mathbb{R}^{N_D \times n}$ and $Y \in \mathbb{R}^{N_D \times K}$. Each row in X represents a decision vector, whereas a row in Y represents its corresponding objective vector. Pre-processing the data can be done in the first stage if necessary. We then build surrogates (generally one per objective) using the provided offline data. Surrogates such as neural networks [6], Kriging [24], and support vector machines [6] are some of the popular choices for solving offline data-driven MOPs. Finally, we use an MOEA to solve the MOP by considering the surrogates as objectives. For simplicity, the objective values of the i^{th} objective is considered as $\mathbf{y}_i \in \mathbb{R}^{N_D \times 1}$ that is a vector of objective values $f_i(\mathbf{x})$ for all decision vectors $\mathbf{x} \in X$.

2.2 Terminologies

1. **Underlying objectives:** In an offline data-driven MOP, we cannot access the objectives of the process or phenomenon, and we start the optimization process using data that has been acquired. For simplicity of representation, we refer to the objective functions of the phenomenon being optimized as underlying objective functions.

2. **Surrogate:** Surrogate models, also known as metamodels, are used to approximate the underlying objectives of the offline data-driven MOP.

3. **Surrogate and underlying objective space:** While solving a real-world offline data-driven MOP, the objective values of the solutions obtained are predictions of the surrogates. These objective values may not be accurate and represent the real-world objective values of the underlying objectives. For benchmarking and understanding the various optimization approaches, we refer to two objective spaces: the surrogate and the underlying objective spaces.

4. **Tradeoff region:** In offline data-driven MOPs, the approximation accuracy in the neighbourhood region of the Pareto set is of utmost importance. In this thesis, we refer to this region of the decision space as the tradeoff region. The concept of tradeoff region was used in [PIV] where the goal was to build GP surrogates exclusively in this region of the decision space.

2.3 Kriging

Kriging or GP is one of the widely used surrogate models to approximate the underlying objectives. As mentioned previously, a Kriging model also provides uncertainty information along with the predicted value, which can be utilized for solving offline data-driven MOPs [PI, PII] and efficient decision making [PIII]. Kriging surrogates have been widely used in Bayesian optimisation [57], time-series modelling [50] and geo-statistics [14]. A Kriging model is a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix C:

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, C) \tag{2}$$

As we have multiple objectives, we generally build one Kriging model (2) for each objective function. For simplicity in calculations, we assume a mean of zero. The correlation between two samples is defined by the covariance matrix C that uses the kernel function. In our work in [PI], [PII], and [PIII], we used a Gaussian (or squared exponential) kernel to define a correlation between $\mathbf{x}$ and $\mathbf{x}'$. For squared exponential kernel, the correlation is:

$$\kappa(\mathbf{x}, \mathbf{x}', \boldsymbol{\Theta}) = \sigma_{\mathrm{f}}^2 \exp\left(-\frac{1}{2}\sum_{j=1}^{n}\frac{|x_j - x_j'|^2}{l_j^2}\right) + \sigma_t^2 \delta_{\mathbf{x}\mathbf{x}'}. \tag{3}$$

In [PIV] we used Matern 5/2 kernel and the correlation is defined as:

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

In the kernel equations (3) and (4), $\delta_{\mathbf{x}\mathbf{x}'}$ denotes the Kronecker delta function and $\boldsymbol{\Theta} = (\sigma_{\mathrm{f}}, l_1, \ldots, l_n, \sigma_t)$ is the set of parameters in the Kriging model. The Euclidean distance between x_j and x_j' is represented by $|x_j - x_j'|$. The amplitude, length scale of j^{th} variable, and noise parameters are represented by σ_{f}, l_j, and σ_t, respectively. These parameters are estimated by maximising the marginal likelihood function:

$$p(\mathbf{y}|X, \boldsymbol{\Theta}) = \frac{1}{\sqrt{|2\pi C|}} \exp\left\{-\tfrac{1}{2}\mathbf{y}^T C^{-1}\mathbf{y}\right\}. \tag{5}$$

After the parameters are estimated we can use the model to predict the posterior predictive distribution at a new sample (or decision vector) $\mathbf{x}^*$. The predicted posterior predictive distribution of the Kriging model is Gaussian and represented as follows:

$$\begin{aligned} p\left(y^*|\mathbf{x}^*, X, \mathbf{y}, \boldsymbol{\Theta}\right) =& \mathcal{N}\big(\kappa(\mathbf{x}^*, X, \boldsymbol{\Theta})C^{-1}\mathbf{y}, \\ & \kappa(\mathbf{x}^*, \mathbf{x}^*, \boldsymbol{\Theta}) - \kappa(\mathbf{x}^*, X, \boldsymbol{\Theta})^T C^{-1}\kappa(X, \mathbf{x}^*, \boldsymbol{\Theta})\big), \end{aligned} \tag{6}$$

where $\kappa(\mathbf{x}^*, X, \boldsymbol{\Theta})C^{-1}\mathbf{y}$ is the posterior mean. The uncertainty in the prediciton (or variance) is $\kappa(\mathbf{x}^*, \mathbf{x}^*, \boldsymbol{\Theta}) - \kappa(\mathbf{x}^*, X, \boldsymbol{\Theta})^T C^{-1}\kappa(X, \mathbf{x}^*, \boldsymbol{\Theta})$. In our works [PI]-[PIII] we used both the posterior mean and uncertainty where our goal was to utilize the uncertainty in the prediction to obtain more accurate solutions and better decision making. In [PIV] we only used the posterior mean since our focus was on building Kriging surrogates when the size of the data is large.

2.4 Decomposition-based MOEAs

The performance of traditional MOEAs such as MOGA [23], MO-CMA-ES [65], and NSGA-II [18], etc. deteriorates when the number of objectives increases [5, 8, 73]. Hence, decomposition-based MOEAs has been developed to solve MOPs with more than three objectives [40, 63] (also known as many-objective optimization problems). These algorithms generally decompose the problem into a number of single objective subproblems using *scalarizing functions* (e.g. MOEA/D [73]) or multiple MOPs (e.g. MOEA/D-M2M [43], NSGA-III [17] and RVEA [8]).

In decomposition-based MOEAs, we first create a set of N uniformly distributed unit reference vectors (or weight vectors) $\mathbf{v_j}$ $(j = 1, \ldots, N)$. The population of individuals generated by the MOEA is P and the objective vectors for the individuals are $F = \left\{\mathbf{f}_1, \ldots, \mathbf{f}_{|P|}\right\}$ consisting of $|P|$ individuals. Here the notation $|.|$ denotes the number of individuals. The i^{th} individual in P is denoted by I_i. The vector of minimum objective function values present in the given population is $\mathbf{z}^{min} = (z_1^{min}, \ldots, z_K^{min})$. In our works we have used RVEA in [PII]-[PIV] and MOEA/D in [PII] for the purpose of demonstration.

2.4.1 MOEA/D

In MOEA/D [73], we first define a neighbourhood of size N_{nh} of reference vectors by pairwise comparison of the Euclidean distances between them. Here $\mathbf{v}_j$ and $\mathbf{x}_j$ are the j^{th} reference and decision vector, respectively in the neighbourhood where $j = 1, \cdots, N_{nh}$. Next, an offspring, $\mathbf{x}'$, is generated in each neighbourhood using crossover and mutation. A scalarizing function is used as the selection criterion to compare and select the individuals. In MOEA/D, we use penalty boundary intersection (PBI), Tchebycheff, or weighted sum as scalarizing functions to compare the parents and the offspring. We used PBI as the scalarizing function

for developing the probabilistic MOEA/D in [PII]. The PBI for a decision vector $\mathbf{x}$ is:

$$g_{\mathbf{x}}^{PBI} = d_1 + \rho d_2, \tag{7}$$

where parameter ρ is the penalty term that balances between convergence ($d_1 = ||(\mathbf{z}^{min} - \mathbf{f}(\mathbf{x}))^T \mathbf{v}_j||/||\mathbf{v}_j||$) and diversity ($d_2 = ||\mathbf{f}(\mathbf{x}) - (\mathbf{z}^{min} - d_1 \mathbf{v}_j)||$). In MOEA/D we compare the PBI of the parents and the offspring by checking whether $g_{\mathbf{x}'}^{PBI} \leq g_{\mathbf{x}_j}^{PBI}$. If so, then we set $\mathbf{x}_j = \mathbf{x}'$ and $\mathbf{f}(\mathbf{x}_j) = \mathbf{f}(\mathbf{x}')$ and repeat this for $j = 1, \cdots, N_{nh}$. In MOEA/D the search process is performed in the neighbourhood of each reference vector and the the population is updated sequentially.

2.4.2 RVEA

In RVEA [8] the objective vectors of the individuals are first translated as $\mathbf{f}_i' = \mathbf{f}_i - \mathbf{z}^{min}$, where $i = 1, \ldots, |P|$. The individuals are then assigned to reference vectors by measuring the cosine of the angle between the reference vector and the translated objective vector. The cosine of the angle between the j^{th} reference vector $\mathbf{v}_j$ and the i^{th} translated objective vector $\mathbf{f}_i'$ is given by:

$$cos\, \theta_{i,j} = \frac{\mathbf{f}_i' \cdot \mathbf{v}_j}{\left\|\mathbf{f}_i'\right\|}, \tag{8}$$

where $\left\|\mathbf{f}_i'\right\|$ is the Euclidean norm. An individual I_i is included in the z^{th} subpopulation $\bar{P}_z$ if it has the lowest angle $\theta_{i,j}$ between $\mathbf{f}_i'$ and $\mathbf{v}_z$ (or highest $cos\, \theta_{i,j}$ value). The index of the z^{th} reference vector to which individual I_i is assigned is:

$$I_i|z = \underset{j \in \{1,\ldots N\}}{\operatorname{argmax}}\ cos\, \theta_{i,j}. \tag{9}$$

After the individuals have been assigned to subpopulations, RVEA selects the z^{th} individual from each subpopulation, which has the minimum APD between the i^{th} individual and the j^{th} reference vector according to:

$$I_z|z = \underset{i \in \left\{1,\ldots,|\bar{P}_j|\right\}}{\operatorname{argmin}}\ d_{i,j}, \tag{10}$$

where APD (or $d_{i,j}$) is defined as,

$$d_{i,j} = (1 + P(\theta_{i,j})) \cdot ||\mathbf{f}_i'||. \tag{11}$$

Here $P(\theta_{i,j}) = K \cdot (t/t_{max})^{\alpha} \cdot \theta_{i,j}/\gamma_{\mathbf{v}_j}$ is the penalty function depending on $\theta_{i,j}$, and $\gamma_{\mathbf{v}_j} = \min_{i \in 1,\ldots,N, i \neq j} \left\langle \mathbf{v}_i, \mathbf{v}_j \right\rangle$, is the smallest angle between reference vector $\mathbf{v}_j$ and the other reference vectors. Here t is the counter for generations, and t_{max} is the maximum number of generations. The parameter α controls the rate of change of $P(\theta_{i,j})$.

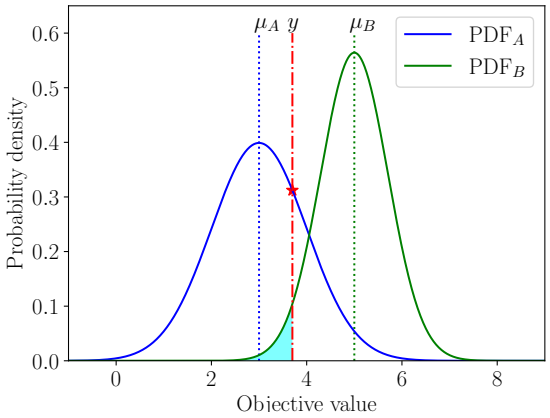


FIGURE 2 The PDF of two individuals with uncertain objective values (approximated by the surrogate) for a single objective minimization problem.

2.5 Probabilistic Selection in Single Objective Optimization

The probabilistic selection in single objective optimization proposed in [31] was further extended to solve offline data-driven MOPs using decomposition-based MOEAs in [PII]. We have a single objective offline data-driven minimization problem where the given data may have experimental or measurement noise. The total uncertainty estimated by the Kriging surrogate is due to the noise in the data and the uncertainty in the prediction. Thus, during the selection process, we may make an error by choosing an individual with a worse objective value.

In Figure 2, we have two individuals, A and B, with uncertain objective values. These two individuals have a normally distributed probability density function (PDF) as approximated by the Kriging surrogate. When we draw a random sample y, from PDF$_A$ (denoted by the red star), we may observe a higher value compared to another random sample drawn from PDF$_B$. Thus, we will make a wrong decision by selecting the individual B over A. The total probability of selecting the wrong individual B over A by observing a specific sample is the total area in the shaded region under PDF$_B$, or cumulative density function (CDF) of B (denoted by CDF$_B(y)$). The probability of drawing a random sample y is PDF$_A(y)$. Thus the total probability of a random sample drawn from PDF$_B$ being smaller than y is PDF$_A(y) \cdot$ CDF$_B(y)$. When the underlying objective value of A is smaller than that of B, the total probability of wrongly selecting B over A is [31]:

$$P_{wrong}(A > B) = \int_{-\infty}^{\infty} \text{PDF}_A(A - y) \cdot \text{CDF}_B((A - y) > (B - y))\ dy. \quad (12)$$

We can replace CDF$_B$ as an integral of PDF$_B$ as:

$$P_{wrong}(A > B) = \int_{-\infty}^{\infty} (\text{PDF}_A(y) \cdot \int_{-\infty}^{y} \text{PDF}_B(\mu) d\mu)\ dy. \quad (13)$$

For comparing a set of individuals with uncertain objective values and ranking

them based on their probabilities of wrong selection we use the following equation:

$$R_i = \sum_{n=1}^{|P|} P_{wrong}(I_n > I_i) - 0.5, \tag{14}$$

where R_i is the ranking score given to the i^{th} individual I_i. The population size is $|P|$ that is the total number of individuals to be compared. The probability of making a wrong decision in selection such that the fitness of I_i is smaller than the fitness of I_n is $P_{wrong}(I_n > I_i)$. We subtract a value of 0.5 from the ranking function as $P_{wrong}(I_i > I_i)$ is always 0.5. The individual achieving the smallest rank has the best fitness value or has the smallest probability of making the wrong selection.

2.6 Interactive Decomposition-based MOEAs

While solving real-world MOPs, the DM is generally interested in a subset of solutions or eventually a single solution to be implemented in practice. Interactive approaches find solutions based on the preferences for objectives provided by the DM. The interactive optimization process can be divided into two phases; learning and decision phase [1, 49]. In the learning phase, the DM explores the objective space to better understand the problem and identify a region of interest. In the decision phase, the DM provides preferences to further improve the solutions that are most interesting to him/her. The provided preferences, for example, can be in the form of a reference point, weights or preferred ranges for objectives. For more details, see [52, 69]. As decomposition-based MOEAs are able to solve many-objective problems, interactive methods using such MOEAs [48] were proposed in [29, 41, 75]. In [PIII] we utilized the approach in [29] to handle preferences for decomposition-based MOEAs.

In order to incorporate the preference information in an MOEA, we can adapt the reference vectors such that they follow the provided preferences for objectives [52]. One of the ways to incorporate preferences for objectives is to ask the DM about his/her satisfactory or desirable objective values (or aspiration levels). The vector formed by the aspiration levels for each objective is called a *reference point* [47]. In [PIII] (and the improved framework), we used a reference point as the preferences for objectives. However, these frameworks are not limited to only this type of preference information.

We initially have a set of uniformly distributed reference vectors $V = \{\mathbf{v}^i \in \mathbb{R}^K | i = 1, \ldots, N\}$, where N is the total number of reference vectors, and $\bar{\mathbf{z}} \in \mathbb{R}^K$ is a single reference point provided by the DM. The reference vectors are adapted as follows [8, 29]:

$$\overline{\mathbf{v}^i} = \frac{r \cdot \mathbf{v}^i + (1-r) \cdot \mathbf{v}^c}{\|r \cdot \mathbf{v}^i + (1-r) \cdot \mathbf{v}^c\|}, \tag{15}$$

where $\mathbf{v}^c = \bar{\mathbf{z}} / \|\bar{\mathbf{z}}\|$ and $r \in (0, 1)$. The reference point $\bar{\mathbf{z}}$ is projected on a unit hypersphere to form the central vector $\mathbf{v}^c$. The parameter r controls the spread of

the adapted reference vectors. If r is close to zero, the adapted reference vectors are close to $\mathbf{v}^c$. However, if r is close to one, the reference vectors are not changed much from their initial position.

3 HANDLING UNCERTAINTY FROM KRIGING SURROGATES FOR SOLVING AN OFFLINE DATA-DRIVEN MOP

As mentioned, in offline data-driven optimization, there is no way to update the surrogates during the optimization process, which restricts the possibility of verifying and improving the surrogate's prediction accuracy. Therefore, while solving offline data-driven MOPs, it is desirable that the surrogates have a low approximation error. However, in practice, surrogates always have approximation errors, and therefore the underlying objective values of the solutions found by the MOEA may be worse than the surrogate objective values.

The author started his research by understanding the effectiveness of the generic approach (as mentioned in Section 2.1) while solving offline data-driven MOPs with Kriging as surrogates. The dataset was generated by sampling from the DTLZ [19] benchmark problems with different sampling strategies. This also enabled the author to understand the search behaviour of the MOEA for different types of MOPs. One of the observations made during the initial tests was that solutions that have a higher uncertainty tend to have a higher approximation

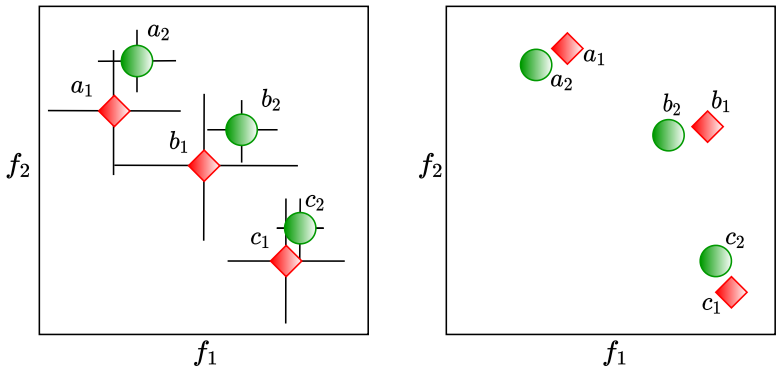


FIGURE 3 A few individuals with uncertainties in the surrogate objective space (left) and underlying objective space (right) for a biobjective minimization problem. The error bars show the 95% confidence interval of the objective values approximated by the Kriging surrogates.

error. In other words, the solutions with higher uncertainties in the surrogate space have a higher probability of having an error when they are evaluated with the underlying objectives. This phenomenon is illustrated in Figure 3 with the objective values and uncertainties of a few individuals approximated by Kriging surrogates and the objective values after they are evaluated with the underlying objectives. In the figure on the left, we can see a few solutions in the surrogate objective space of an MOEA with their uncertainties (here 95% confidence interval) while solving a biobjective minimization MOP. The red individuals dominate the green ones in the surrogate objective space. However, as can be observed, these non-dominated individuals also have higher uncertainties. On the right, we show the same individuals when they have been evaluated with the underlying objective functions. It can be observed that the green individuals dominate two out of three of the red individuals. Thus, while solving offline data-driven MOPs, utilizing just the surrogates' mean approximation (without including approximation error or uncertainty) can produce solutions with worse underlying objective values. One of the ways to tackle this problem is by using the uncertainty from the surrogates [31, 53].

In [PI] we proposed three approaches to incorporate uncertainties predicted by Kriging surrogates during the optimization process and addressed our first challenge as introduced in Chapter 1.

3.1 Approaches to utilize the uncertainty in approximation from Kriging surrogates

The primary concept of the approaches proposed in [PI] was to minimize the uncertainties in the solutions along with the objective values. Therefore the solutions found by the MOEA would have the best objective values with the least possible uncertainties. The three approaches proposed are as follows:

1. **Approach I**: The uncertainties in the prediction of Kriging surrogates are considered as additional objectives along with the predicted mean objective values. The modified MOP is:

$$\mathbf{f} = (f_1(\mathbf{x}), \ldots, f_K(\mathbf{x}), \sigma_1(\mathbf{x}), \ldots, \sigma_K(\mathbf{x})), \tag{16}$$

   where $f_i(\mathbf{x})$ and $\sigma_i(\mathbf{x})$ and are the predicted mean and the standard deviation values for the i^{th} objective from the Kriging models. In this approach, we have double the number of objectives being solved and the total number of objectives is $2K$.

2. **Approach II**: The average of the uncertainties predicted by Kriging surrogates is considered as an additional objective along with the predicted mean objective values. The modified MOP is:

$$\mathbf{f} = (f_1(\mathbf{x}), \ldots, f_K(\mathbf{x}), \bar{\sigma}(\mathbf{x})), \tag{17}$$

where $\bar{\sigma}(\mathbf{x})$ is the average of the standard deviations from Kriging models built for each objective function. In this approach, we have one more objective added to the original MOP and the total number of objectives is $K+1$.

3. **Approach III**: Expected improvement (EI) [35] is widely used for solving data-driven problems as it balances exploration and exploitation by utilizing uncertainty approximation. Therefore, we modified the MOP to find the tradeoff between the EI of the approximated objective functions as:

$$\mathbf{f} = (\mathrm{EI}_1(\mathbf{x}), \ldots, \mathrm{EI}_K(\mathbf{x}))\,, \tag{18}$$

where $\mathrm{EI}_i(\mathbf{x})$ is the expected improvement value for the i^{th} objective. This approach did not increase the number of objectives.

3.2 Results and Discussion

The three approaches, along with a generic approach (as introduced in Section 2.1) were tested with offline datasets generated using the DTLZ [19] benchmark problems (DTLZ2-7). This was done to understand the search behaviour of the different approaches for various types of problems. The problems had ten decision variables, and the initial dataset consisted of 109 samples (as per the recommended sample size of $11n-1$ in [37, 74]). The sampling strategies used were (a) Latin hypercube sampling (LHS) [45], (b) random sampling and (c) optimal random sampling (where 50% samples were randomly generated and 50% were Pareto optimal). The MOEA used for solving the MOPs was IBEA [76]. We chose IBEA for the purpose of demonstration and since it performed well in [5]. Ideally, an MOEA solving an offline data-driven MOP should not provide worse solutions compared to the provided dataset. The optimal random sampling provided us with a good understanding of the performance of the approaches. The solutions found by the MOEA should not be worse in inverse generational distance (IGD) compared to the initial dataset that already included Pareto optimal solutions. The solutions from each generation of the MOEA were stored in an archive. A non-dominated sorting was performed on the archive, and these sorted solutions were evaluated using the underlying objective functions. The IGD and RMSE (for measuring accuracy) of the evaluated solutions for all the tested approaches were calculated, and their means were compared. Overall, Approach I and Approach II performed the best in both IGD and RMSE compared to the generic approach and Approach III. Surprisingly, using expected improvement as objectives actually performed worse than the generic approach. This was primarily because EI tries to balance between convergence and diversity. Therefore the MOEA may select solutions that have higher uncertainty.

Out of the three approaches tested, we found that Approach I and Approach II outperformed the generic approach in both IGD and accuracy. Their strengths

were especially highlighted while solving MOPs with optimal-random datasets. These approaches were able to provide solutions with equally good or better IGD than the initial samples.

However, Approach I and Approach II increased the number of objectives, thereby increasing the complexity of the MOP. On the other hand, Approach III showed no improvements over the generic approach. In Approach I and Approach II, the number of solutions that were evaluated after performing non-dominated sorting on the archive was quite high. This may not be suitable while solving a real-world offline data-driven MOP. A fair comparison would be to consider the solutions of the final generation of the MOEA instead of the archive. A better way is to explicitly utilize the uncertainty information in the selection criterion of the MOEA. Therefore, in [PII] we proposed probabilistic selection approaches for decomposition-based MOEAs that do not increase the number of objectives and produces solutions with better accuracy than their generic counterparts.

4 PROBABILISTIC SELECTION IN DECOMPOSITION-BASED MOEAS

Utilizing uncertainty in the prediction of Kriging surrogates proved beneficial for solving offline data-driven MOP in [PI]. However, the approaches proposed in [PI] had a few drawbacks. These approaches increased the complexity of the MOP by increasing the number of objectives, thereby making them difficult to solve. Furthermore, they did not explicitly use the uncertainty information in the MOEA. Hence, we proposed improved approaches to utilize the uncertainty information in [PII].

In [31], an approach was proposed to utilize uncertainty in objective values in dominance-based MOEAs. The probability of making a wrong decision in selecting a solution with uncertain objective values was utilized in dominance-based MOEAs. However, dominance-based MOEAs are not suitable for solving MOPs with a high number of objectives.

As mentioned in the introduction, decomposition-based MOEAs [8, 17, 40, 63, 73] were designed specifically to solve many-objective problems ($K > 3$ objectives). Hence, probabilistic selection for decomposition-based MOEAs can be seen to be more suitable for handling offline data-driven many-objective problems.

Calculating the probability of selection criterion requires analytical derivations that are tailored for a specific selection criterion used in an MOEA. Deriving the closed form becomes difficult with complicated selection criteria such as angle penalized distance (APD) in RVEA [8]. One of the primary features of the probabilistic selection approaches proposed in [PII] is their adaptability or "plug and play" capability for any decomposition-based MOEA. In other words, the proposed approaches can be used in the selection criterion specific to the MOEA without any further modifications. The adaptive ability of the approaches was demonstrated by implementing and testing probabilistic versions of MOEA/D [73] and RVEA [8]. These two MOEAs were chosen primarily because the closed form of their probability of selection criterion does not exist. Thus the potential of the proposed probabilistic selection approaches can be better demonstrated.

4.1 Probabilistic decomposition-based MOEAs

In the probabilistic selection approaches for decomposition-based MOEAs, we utilized the uncertainty in the prediction of Kriging surrogates in the selection process of the individuals. This is done by modifying (12) and (13) to formulate a probabilistic selection criterion specific to the MOEA.

We first draw samples using Monte Carlo sampling [46] from the probability density function (PDF) of objective values approximated by the Kriging surrogates. Certain decomposition-based MOEAs assign individuals to subpopulations. However, since we are dealing with uncertain objective values, the process of assigning individuals is also probabilistic. Next, the values of the selection criterion (specific to the MOEA) for all the drawn samples are calculated. The PDF of the selection criterion is estimated using kernel density estimation (KDE) [55]. The probability of selection is then calculated using the estimated PDFs, and individuals are selected based on their probability values.

In Algorithm 1, we show the approach to implement probabilistic selection for a decomposition-based MOEA. The algorithm can be adapted for other selection criteria of the decomposition-based MOEA with minor modifications. The offline dataset consisting of N_D samples is provided as input along with other parameter settings. We assume that this dataset is provided or generated with some sampling strategy. We first build Kriging surrogates for each objective using the provided data. Next, we initialize the population and set the number of function evaluations FE to zero. A uniformly distributed set of unit reference vectors (or weight vectors), V_0 of size N is generated. Depending on the MOEA, the neighbourhood of each reference vector or sub-population is defined in a probabilistic way. The population undergoes crossover and mutation, and a total number of $|P_{offspring}|$ offspring individuals are generated. The new individuals are evaluated using surrogates, and the function evaluation counter, FE, is updated. Next, S samples are drawn from the PDF of objective values approximated by Kriging surrogates. To draw the samples we use the multivariate Gaussian PDF [24] of objective values for the individual I_i:

$$\mathrm{PDF}_{I_i} = \prod_{k=1}^{K} \frac{1}{\hat{\sigma}_{i,k}\sqrt{2\pi}} \exp\left(-\frac{(f_k - \hat{f}_{i,k})^2}{2\hat{\sigma}_{i,k}^2}\right), \tag{19}$$

where $\hat{f}_{i,k}$ is the approximated k^{th} objective function value for the i^{th} individual with $\hat{\sigma}_{i,k}$ as its standard deviation. The individuals are assigned to subpopulations according to the MOEA (if applicable). Next, individuals are selected by the probabilistic selection criterion of the MOEA and used as parents for the next generation. The MOEA is run for a predefined maximum number of function evaluation FE_{max}.

Algorithm 1: Probabilistic decomposition-based MOEA

Input: Offline data of size N_D; N = number of reference vectors; FE_{max} = maximum number of function evaluations using Kriging surrogates; S = number of samples to be used for estimating the distributions

Output: Approximated solutions

1 Build Kriging surrogates for each objective using the given offline data
2 Use the given data as the initial population; initialize the number of function evaluations $FE = 0$
3 Create a set of uniformly distributed unit reference vectors V_0 of size N
4 Find the neighbourhood for each unit reference vector
5 **while** $FE < FE_{max}$ **do**
6 Perform crossover and mutation on population and generate offspring
7 Evaluate the individuals using the Kriging surrogates and combine the parents and offspring
8 Update $FE = FE + |P_{offspring}|$
9 Draw S samples using Monte-Carlo from the distribution approximated by the surrogates
10 Perform algorithm specific sub-population assigning
11 Perform algorithm specific probabilistic selection
12 end

4.2 Probabilistic RVEA and MOEA/D

The proposed probabilistic selection approach can be adapted to any decomposition-based MOEA by modifying steps 10 and 11 in Algorithm 1. In [PII] we demonstrated this feature by implementing probabilistic selection for RVEA and MOEA/D.

4.2.1 Probabilistic APD

As described in Section 2.4.2, in RVEA, the individuals are first assigned to reference vectors using (9) to create subpopulations. As the individuals have uncertain objective values, they should be assigned to reference vectors in a probabilistic way. First, we assign each of the samples that were drawn from the PDF of objective values of the individuals to the reference vectors. Thus different samples may be assigned to different reference vectors depending on the distribution of objective values of the individual. An individual is assigned to the reference vector that has the most number of samples assigned to it.

After the subpopulations have been generated, the angle penalized distance (APD) values of the samples are calculated using (11) (described before in Section 2.4.2). These APD values are used to estimate the PDF of the APD for every individual using kernel density estimation (KDE) [55]. In our work, we used the

Gaussian kernel and utilized Silvermann's rule of thumb [59] for selecting the bandwidth parameter that controls the smoothness of the estimated distribution. Next, we rank the estimated PDFs of APD by modifying (10) utilizing (14) as:

$$P_{nextgen} = \left\{ I_z | z = \underset{i \in \{1,\ldots,|\bar{P}_j|\}}{\operatorname{argmin}} R'_{i,j} \right\}, \tag{20}$$

where

$$R'_{i,j} = \sum_{n=1}^{|\bar{P}_j|} P_{wrong}(d_{n,j} > d_{i,j}) - 0.5. \tag{21}$$

The i^{th} individual in the j^{th} subpopulation $\bar{P}_j$ is given the rank $R'_{i,j}$. The probability of wrong selection, P_{wrong}, is calculated by utilizing (14) and comparing the PDF of APDs ($\text{PDF}_{d_{i,j}}$) of the i^{th} individual in the j^{th} subpopulation. For the population of the next generation, $P_{nextgen}$ we select the z^{th} individual I_z from subpopulation $\bar{P}_j$, where $j = 1, \ldots, N$.

Calculating the rank $R'_{i,j}$ in (21) is a pairwise comparison between PDFs of APD of individuals in a subpopulation. Therefore, the computation cost of performing probabilistic selection is $O(|\bar{P}_j|^2)$, where $|\bar{P}_j|$ is the number of individuals in the j^{th} subpopulation. The double integral involved in calculating P_{wrong} as shown in (13), between PDF of APD of two individuals and computing KDE of $\text{PDF}_{d_{i,j}}$ makes probabilistic selection computationally expensive. Therefore, we proposed an efficient way to compute P_{wrong}. As computing $R'_{i,j}$ in (21) is a pairwise comparison, the probability P_{wrong} between an APD distribution with itself is always 0.5. Also, we can calculate the ranks between the same pairs once. The modified equation for calculating P_{wrong} is:

$$P_{wrong}(d_{n,j} > d_{i,j}) = \begin{cases} 0.5 \text{ if } n = i, \\ 1 - P_{wrong}(d_{i,j} > d_{n,j}) \text{ if } n > i. \end{cases} \tag{22}$$

Furthermore, the calculation of CDF using integral in (13) was changed to computing the empirical CDF from the APD samples. The lower limit of the integral in (13) was changed to zero instead of $-\infty$ since APD can never attain a value below zero. We further reduced the computation time by calculating P_{wrong} values in parallel for all the individuals in a subpopulation.

4.2.2 Probabilistic PBI

In MOEA/D, step 10 in Algorithm 1 was skipped as the assignment of solutions for each reference vector is performed by defining the neighbourhood of each vector [73]. A probabilistic selection utilizing the PBI selection criterion is similar to probabilistic APD. We draw S samples from the approximated distribution of objective values (from Kriging surrogates) of individuals in the neighbourhood and the offspring (see Section 2.4.1 for details on neighbourhood and offspring).

The PBI values of all the sampled objective values are calculated for the j^{th} individual in the neighbourhood and the offspring that we call as $g^{PBI}_{\mathbf{x}_{j,l}}$ and $g^{PBI}_{\mathbf{x}'_l}$, respectively, where $l = 1, \ldots, S$ and $j = 1, \ldots, N_{nh}$. The PDF of $g^{PBI}_{\mathbf{x}_j}$ and $g^{PBI}_{\mathbf{x}'}$ are approximated using KDE using the Gaussian kernel and bandwidth parameter as per Silvermann's rule. We update the solutions in the neighbourhood by checking if $P_{wrong}(g^{PBI}_{\mathbf{x}'} \leq g^{PBI}_{\mathbf{x}_j}) < 0.5$, then set $\mathbf{x}_j = \mathbf{x}'$ and $\mathbf{f}(\mathbf{x}_j) = \mathbf{f}(\mathbf{x}')$. It has to be noted that the comparison of PDFs of PBI is within a neighbourhood and hence one-to-many.

4.3 Analysis of the Probabilistic Selection

The closed-form of the distributions for APD and PBI is not available. The proposed probabilistic approaches are beneficial for such selection criteria and require no further analytical derivations. In Figure 4, we illustrate the estimated PDF of APD (a) and PBI (b) samples and the empirical CDF for an individual. The PDF of the selection criterion is approximated by KDE (shown by the red line). The histogram is represented by the blue bars, and the samples are shown in green. The blue line shows the empirical CDF. As can be observed, the approximated PDFs of the selection criterion are not well-known PDFs and closely follow the histogram.

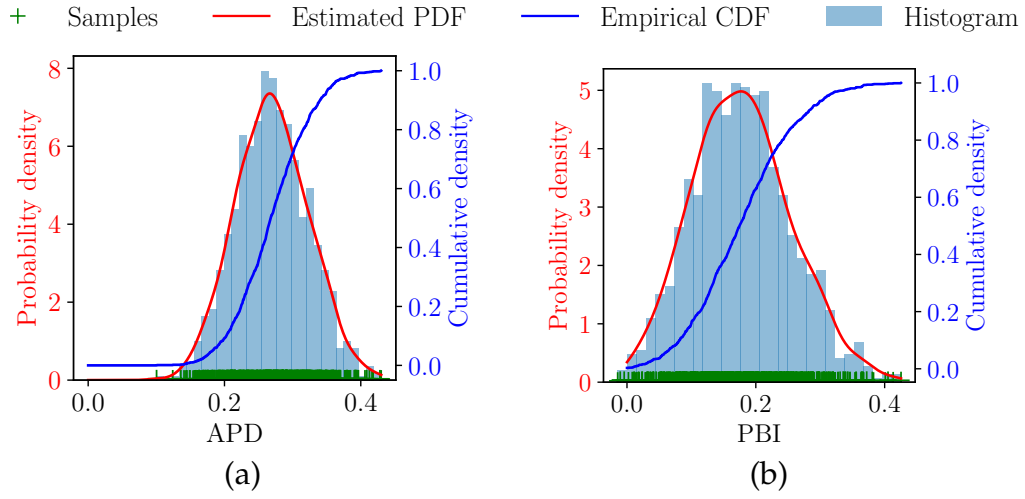


FIGURE 4 Illustration of samples drawn represented with '+', histogram, estimated PDF and empirical CDF of the selection criterion for (a) APD and (b) PBI for one individual.

The difference between the selection of individuals in generic and probabilistic RVEA is illustrated in Figure 5. The numbers on the individuals show the sub-population or reference vector they are assigned to. The individuals selected from a sub-population are shown in green, and those not selected are shown in red. It can be observed that the generic RVEA selects individuals with a better

objective value without considering the uncertainties (e.g. individuals assigned to reference vector '2'). On the other hand, probabilistic RVEA considers uncertainty in the individuals and selects the one with lower uncertainties.

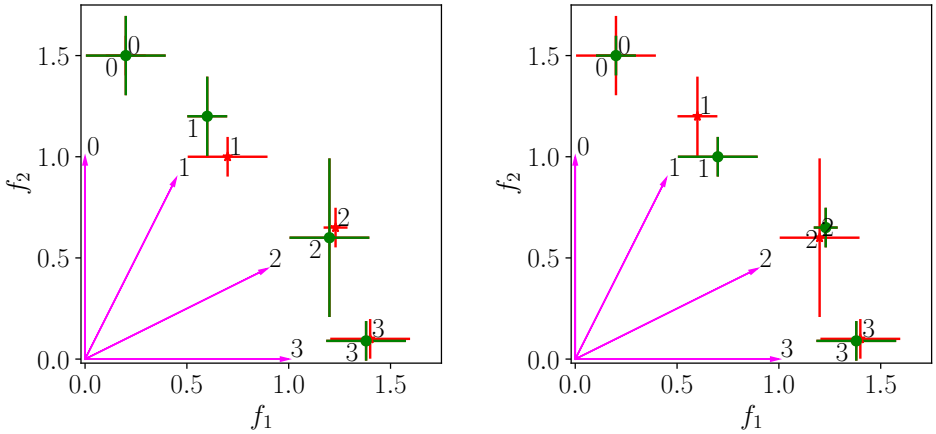


FIGURE 5 Individuals selected in a biobjective minimization problem using (a) generic RVEA and (b) probabilistic RVEA. Error bars show the 95% confidence interval of the distribution of objective values approximated by the surrogates. The individuals in 'green' are selected and 'red' are not selected from their subpopulation.

4.4 Hybrid Approach

The probabilistic approach produces solutions with better objective values and lower uncertainties (or high approximation accuracy). On the other hand, the generic approach does not consider uncertainty in the selection and produces solutions that are better in hypervolume (in the surrogate objective space). Hence, we proposed a hybrid selection approach that combines the benefits of both the selection approaches and produces solutions with a wider range of uncertainties and objective values. This will also prove to be beneficial to the DM as (s)he is provided with a wider variety of solutions with diverse objective and uncertainty values [PIII].

Figure 6 shows the flowchart of the proposed hybrid approach. We select individuals based on the generic selection criteria of the MOEA and the probabilistic selection criteria. An equal proportion of solutions from both the generic and probabilistic selection criteria are selected, and no additional parameters are necessary. Finally, the redundant copies of the individuals selected by both approaches are removed, and the combined population is used for crossover and mutation.

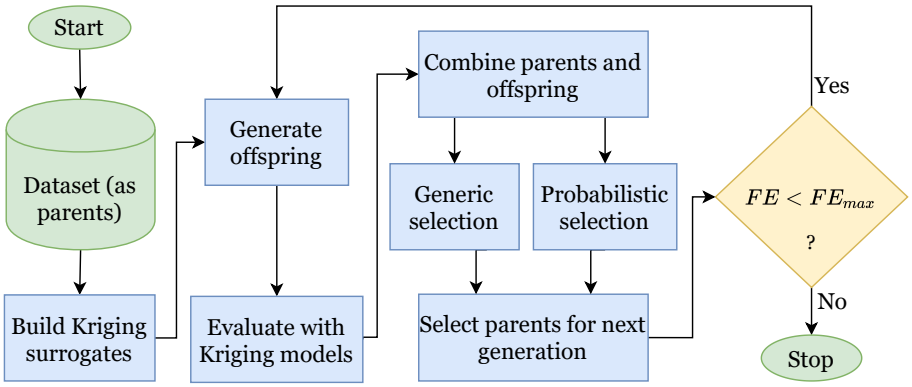


FIGURE 6 Flowchart for the hybrid approach.

4.5 Performance of the Probabilistic Approaches

We tested the proposed probabilistic and hybrid versions of RVEA (Prob-RVEA and Hyb-RVEA, respectively) and MOEA/D (Prob-MOEA/D and Hyb-MOEA/D, respectively) along with their generic counterparts (Gen-RVEA and Gen-MOEA/D) and a transfer learning (TL) approach [72]. The TL approach was considered in our tests as it was specifically designed to solve offline data-driven MOPs. Additionally, the initial dataset was also considered in the comparison to check whether the solutions found have actually improved in objective values. We chose two distance-based visualizable test problems (DBMOPP) [21] with different parameter settings utilizing the code in [22]. The two problems were P1 and P2 and their configuration can be found in Table 1 in Section 5.2.2. The problem instances had 2-10 number of objectives with ten decision variables. These problems are advantageous as we can visualize the progress of the solution process and better understand the behaviour of the optimizer. Further tests were also conducted with DTLZ [19] benchmark problems.

A total of 31 independent runs were executed with all the approaches, and the initial dataset consisting of 109 samples (as per the recommended sample size of $11n - 1$ in [37, 74]) was acquired by LHS [45] and multivariate normal sampling (MVNS). In MVNS, the samples were generated using a multivariate normal distribution. This type of sampling simulated real-world datasets in offline data-driven MOPs where the data cannot be assumed to have any predefined sampling strategy. For DBMOPP problems we considered, the objectives to be independent with mean at the mid-point of the decision space (or zero for all decision variables). The variance of the sampling distribution was set to 0.1 for all the objectives. The same settings were used for the MVNS datasets in Chapter 5 and 6.

For building the Kriging surrogates, we used Scikit-learn python library [51] with Gaussian kernel and BFGS [3] to maximize the marginal likelihood. The indicators used for comparing the performances of the different approaches were hypervolume and multivariate root mean squared error (RMSE) of the solutions

after evaluating them with the underlying objectives. The multivariate RMSE is calculated by:

$$\text{RMSE} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\sum_{k=1}^{K}(\hat{f}_{k,i} - f_{k,i})^2}, \tag{23}$$

where $\hat{f}_{k,i}$ and $f_{k,i}$ are the approximated objective value and the underlying objective value, respectively for the i^{th} solution and the k^{th} objective. In this thesis we refer to multivariate RMSE as RMSE for simplicity. We implemented all the proposed approaches in Python utilizing the DESDEO framework (desdeo.it.jyu.fi).[1]

To confirm the effectiveness of our proposed approaches, we performed a pairwise Wilcoxon two-tailed significance test [25]. The p-values were Bonferonni corrected, and we considered $\alpha = 0.05$ for rejecting the null hypothesis. The null hypothesis in our tests was when an approach is not significantly better or worse than another approach for a given indicator. If the calculated p-value is less than α, we compared the median values of the approaches to determine whether an approach is significantly better or worse than another one. The performance ranking of the different approaches was done using a scoring system. We gave an approach a score of $+1$ if it was significantly better than the other approach. A score of -1 was given to the approach if it was significantly worse than the other approach. We gave a score of zero to both approaches if the approach was not significantly better or worse than the other one. Finally, we calculated the sum of the obtained scores for ranking all the approaches (a higher score gives a better rank) for the indicator being compared. If an approach obtained a rank of '1', then that approach had performed significantly better than all other approaches. The approaches having equal ranks were not significantly different in their performances.

In our tests, we observed that Prob-RVEA and Hyb-RVEA performed the best in hypervolume, and Prob-MOEA/D performed the best in RMSE. The generic versions of RVEA and MOEA/D and the transfer learning approach performed the worst in both hypervolume and RMSE. This was primarily because the solution process did not consider the uncertainty in the prediction of the surrogates. Therefore, the solutions obtained using inaccurate surrogates were worse in objective values when evaluated with underlying objectives. This was the primary challenge the probabilistic selection approaches were designed to address.

The progress of the solution processes of the different approaches (the run with the median hypervolume) is shown in Figure 7. The plots show the projection of the solutions in 10-dimensional decision space to a 2-dimensional space. The distance between the points on the circle in different colours represents the minimized objectives. In the first column, we show the problem instance, and in the second column, we show the initial samples. The progress of the solution process is shown in the next seven columns showing the underlying objective

[1] Source code is available at https://github.com/industrial-optimization-group/offline_datadriven_moea

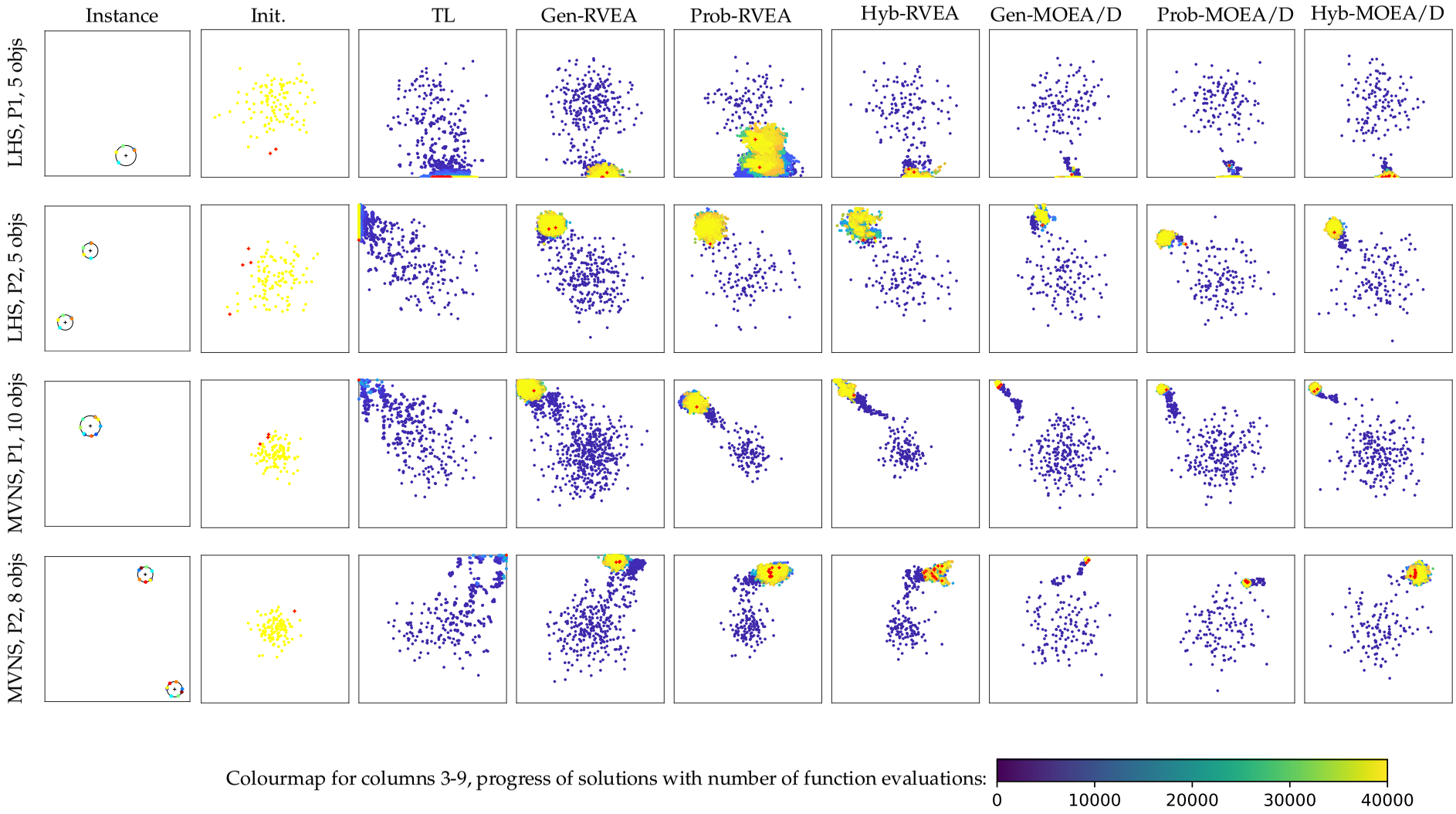


FIGURE 7 Progress of solution process for P2 with MVNS for the runs with median hypervolume values. The colour code represents the number of function evaluations of a solution during the optimization process.

values of the solutions. The colour code denotes the function evaluation count, and the non-dominated solutions/ samples are represented by the red '+'.

In Figure 7 we can observe that Prob-RVEA, Hyb-RVEA, Prob-MOEA/D and Hyb-MOEA/D converged much closer to the Pareto front compared to their generic counterparts. The solutions for Gen-RVEA, Gen-MOEA/D, and TL were further away from the Pareto front. One can also see that the non-dominated solutions in the initial sampling were much closer to the Pareto front compared to the generic approaches and TL. Therefore, the solutions obtained using generic approaches (or just using the predictive mean from surrogates) were worse than the provided dataset. We can also observe that all approaches failed to get solutions closer to both of the Pareto sets that are disconnected. This was because we did not have a sufficient amount of data for solving such MOPs.

As said, we demonstrated the adaptability of the probabilistic selection approaches with RVEA and MOEA/D. The first approach used the probability of selection, whereas the second approach used a hybrid selection. The accuracy of solutions is generally not considered in the field of data-driven optimization. The proposed approaches were more focused on improving the accuracy of the solutions. Detailed testing and analysis with offline datasets generated from benchmark problems were performed. Overall results showed that the proposed approaches produced solutions with a better hypervolume and accuracy compared to their generic counterparts. However, if accuracy in the solutions is the primary requirement, one should prefer the probabilistic approach over the hybrid approach.

In [PI] and [PII] we tacked the first challenge (as introduced in Chapter 1) by utilizing uncertainty information provided by surrogates in MOEAs. In the future, we intend to test other decomposition-based MOEAs with different selection criteria and normalization techniques. However, while solving real-world MOPs, the DM is interested in a smaller subset of solutions as per the preferences. In the next chapter, we discuss an interactive optimization framework for solving offline data-driven MOPs.

5 HANDLING PREFERENCES IN OFFLINE DATA-DRIVEN MOEA

While solving real-world MOPs, the DM is typically interested only in certain regions of the Pareto front or a single solution, depending on his/her preferences. This is particularly useful while solving computationally expensive MOPs (generally online data-driven MOPs) as using preferences in optimization reduces the overall computation time. The preference information is provided in generally three ways, (a) a priori, (b) a posteriori and (c) interactively, where the preference information is utilized before, after and during the solution process, respectively [47].

In interactive multiobjective optimization approaches, the DM is involved during the optimization process and provides his/her preferences to find desirable solutions. This enables the DM to learn about the problem and the feasibility of the preferences and adjust them in the later interactions. Additionally, it reduces the cognitive load on the DM due to the limited amount of information to be considered at a time [49]. Therefore, several interactive approaches for MOEAs [30, 44, 52, 58] have been proposed. Due to the recent developments in decomposition-based MOEAs, interactive approaches such as [29, 41, 75] have been proposed. Most of the interactive approaches use preference information from the DM in the form of, e.g. a reference point, pairwise comparison, weights, preferred ranges for objectives function values and selecting a preferred solution from a given set [48, 71]. However, as far as we know, there has been no work to address DM's preference for solving offline data-driven MOPs in decomposition-based MOEAs.

Generally, while solving an MOP with the involvement of the DM, the preferences for objective values are required. Also, during the decision making process, the DM is provided with the objective values of the solutions. However, while solving an offline data-driven MOP, the solutions have uncertainties due to the approximation error of the surrogates. The uncertainty information could be vital for the DM since it provides him/her the knowledge about the probable error in the solutions. For example, the DM may be willing to accept solutions with higher uncertainty in one particular objective and lower uncertainty in another. Providing this additional information to the DM and enabling him/her to provide

preferences for uncertainties is a challenging task. This is because, firstly, the DM is generally not familiar with what uncertainties in solutions mean. Secondly, the DM has to provide two types of preferences that increase the cognitive load. Thirdly, the DM provides the preferences for objectives by observing the solutions that have uncertainly in their objective values. This sometimes results in solutions that are not actually of the desired preferences for objectives when they are evaluated with the underlying objective function.

In our previous works [PI, PII] we proposed approaches that utilized the uncertainty information from Kriging surrogates. These approaches proved to be advantageous over their generic counterparts that did not utilize uncertainty in the solutions. Conveying the uncertainty information may be quite valuable for the DM as (s)he can better understand the solutions and provide preferences for objectives. Hence, in [PIII], we proposed a framework that enables the DM to understand and make decisions based on the uncertainties in the approximated solutions. The framework provides an additional option for the DM to provide his/her preferences for uncertainties along with the preferences for objective values. Furthermore, it does not significantly increase the cognitive load of the DM while handling two types of preferences. We further extend our framework by integrating it with the probabilistic selection approaches proposed in [PII]. We also proposed a reference vector technique to automatically adjust the diversity of the solutions based on their uncertainties. The new framework was tested with an artificial decision maker (ADM), and its performance was compared with other variants of interactive approaches. The frameworks proposed in this chapter were developed to tackle the second challenge as introduced in Chapter 1.

5.1 The Proposed Framework

In [PIII], we proposed a framework for interactively solving offline data-driven MOPs. The framework enables the DM to provide preferences for uncertainties (in the form of cutoff tolerances) along with preferences for objectives (in the form of a reference point). The framework involves two separate steps for providing the two types of preferences (for objectives and uncertainties). Therefore, the DM does not have to deal with two different types of preferences at the same time. Thus the cognitive load is not increased by much. Also, the approach to handle preferences for uncertainties does not affect the solution process of the MOEA and thus does not increase the computation cost. A simplified block diagram of the framework is shown in Figure 8. The framework consists of five steps that are as follows:

Step 1: Build Kriging surrogates for each objective using the provided dataset. This may require some domain expertise and the data may require some preprocessing. Next, a decomposition-based MOEA is initialized considering the built surrogates as objectives to be optimized. A uniformly distributed set of reference vectors is generated, and a population is initialized randomly.

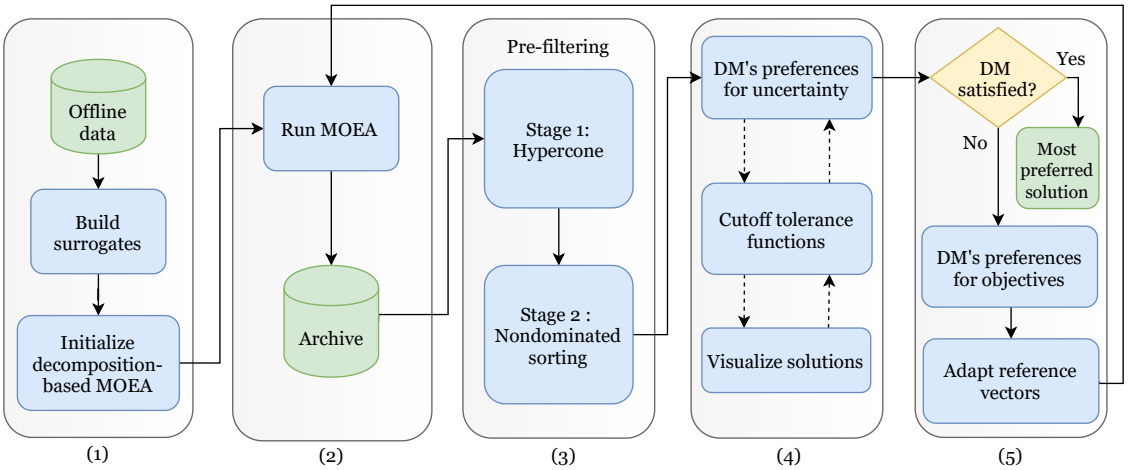


FIGURE 8 The proposed framework for solving offline data-driven MOPs interactively.

Step 2: The MOEA is run for a fixed number of generations. The objective and uncertainty values of the solutions from every generation are stored in an archive.

Step 3: A two-stage pre-filtering is applied on the archive consisting of objective and uncertainty values. The first stage is the *hypercone filtering*. The solutions in the archive that are within the hypercone formed by the extreme reference vectors are passed by the filter. This is done to ensure that the DM considers only the solutions in the archive that are within the preferences for objectives at the later step. In the first iteration, the reference vectors are not adapted to incorporate preferences for objectives (done in Step 5). Therefore, all the solutions pass the hypercone filter. In the second stage of the pre-filtering, a non-dominated sorting is performed by considering the objectives and uncertainties. Therefore, only the solutions that are the best in both the objective and uncertainty values pass the filter. The solutions at the different pre-filtering stages are shown in Figure 9. All the solutions in the archive are shown in Figure 9(a). The solutions after the hypercone filtering and non-dominated sorting are shown in Figure 9(b) and 9(c), respectively.

Step 4: The knowledge about uncertainty in the solutions is crucial to the DM while solving offline optimization problems. However, the DM is not always familiar with the uncertainty information while solving real-world problems. Generally, the DM has an idea about the permissible tolerances in objective values. For instance, cost and deflection are the two objectives to be minimized in the welded beam problem [16]. The permissible deviation in cost (or the maximum allowed cost) is known to the DM and can be regarded as is the preferred one-sided *tolerance* of the DM [38]. The one-sided tolerance information is regarded as the *preferences for uncertainties* provided by the DM and represent the maximum permissible variation in the underlying objective values of the solutions. In [PIII], we refer to one-sided tolerance as tolerance for simplicity.

The preferences for uncertainties are provided as indifference tolerances $\tau_k\%$ for the k^{th} objective. The indifference tolerance is provided in percentages and represents the 95% tolerance interval. The distribution of the predicted objective

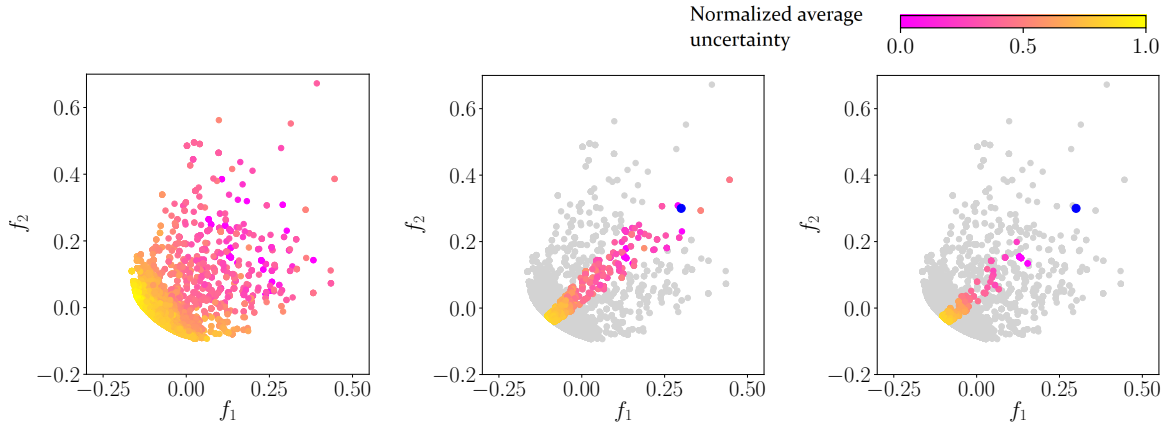


FIGURE 9 The solutions at the three pre-filtering stages while minimizing a bi-objective problem. The filtered out solutions at each stage are shown in grey, and the reference point provided by the DM is shown in blue.

values for the Kriging surrogates is Gaussian with a standard deviation of $\hat{\sigma}_k(\mathbf{x})$ for the k^{th} objective. The *cutoff tolerance functions* are used to show the solutions that follow the DM's preferences for uncertainties. The cutoff tolerance function for the k^{th} objective is:

$$g_k(\mathbf{x}) = 1.96\hat{\sigma}_k(\mathbf{x}) - \tau_k \cdot \hat{f}_k(\mathbf{x})/100 \leq 0, \tag{24}$$

where $k = 1, \ldots, K$. In the framework, the DM can provide the preferences for uncertainties as many times as (s)he wishes to, and view the solutions based on the preferences for uncertainties. This step may be skipped by the DM if (s)he chooses to do so.

Step 5: If satisfactory solutions are found, the DM can either stop the optimization process or provide new preferences for objectives to find more preferred solutions. The reference vectors are adapted according to (15) such that the solutions follow the provided preferences for objectives. If the DM wishes to continue the optimization process, we go to step 2.

Results and Discussions

Testing and benchmarking the performance of interactive approaches is still a research challenge [1]. It becomes even more difficult since the proposed framework utilizes preferences for uncertainties from the DM. Hence, we demonstrated the working and the advantages of the proposed framework by solving the general aviation aircraft (GAA) [56, 60] design problem. The GAA problem is an optimization design problem for recreational pilots to business executives and consists of 27 decision variables, ten objectives and one constraint. The offline data consisting of 1000 samples were generated using Latin hypercube sampling [45] using the Matlab implementation of the GAA problem in [7]. We used Kriging surrogates with a Gaussian kernel to approximate the underlying objectives. The MOEA

used was RVEA with parameter settings as recommended in [8] and executed it for 100 generations in each iteration (or interaction). We used a spread parameter $r = 0.2$ for reference vector adaptation in our tests. However, this can be changed as per the DM's requirement of diversity. Since our framework did not support constraint handling yet, we considered the constraint violation as an additional objective function for demonstration.

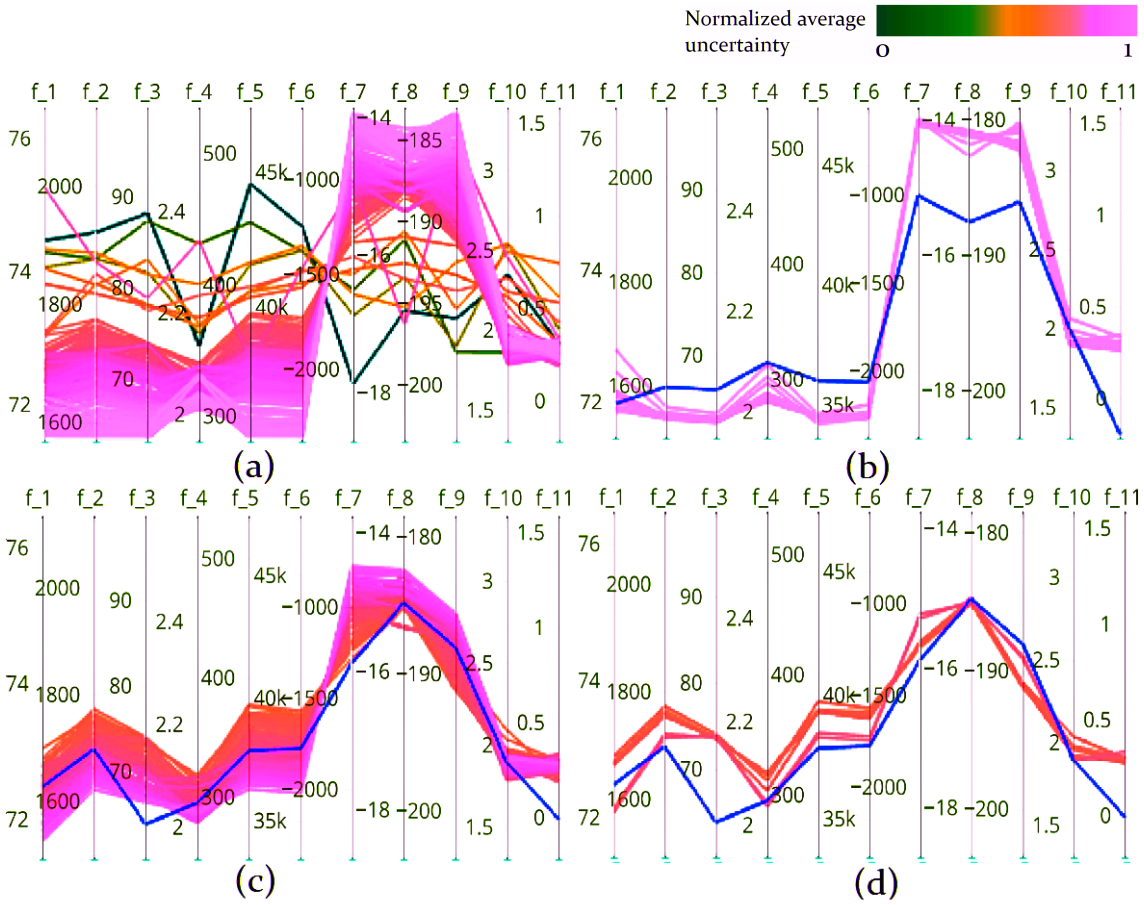


FIGURE 10 The solutions found by the interactive framework for two iterations when all the objectives are minimized. (a): solutions in the archive at the first iteration. (b) & (c): solutions obtained after the pre-filtering stage in the first and second iteration respectively with different preferences (blue line is the reference point). (d): solutions shown to the DM after (s)he provided the preferences for uncertainties.

In Figure 10, we show the solutions produced by the framework at various steps for two iterations. The colour of the parallel coordinate lines shows the normalized average uncertainty of the solutions (green is lowest and pink is highest). The solutions in the archive at the first iteration are shown in sub-figure (a). Here the author acted as the DM and provided the preferences for objectives in the form of reference points (shown in blue). The pre-filtered solutions in the first iteration are shown in sub-figure (b). However, as can be observed, these

solutions had a high uncertainty and were not desirable to the DM. Hence, the DM chose to skip the step of providing preferences for uncertainties and provide new preferences for objectives. It can be observed in sub-figure (c) that the pre-filtered solutions had a lower uncertainty in the second iteration. To demonstrate the ability of the framework to handle preferences for uncertainties, we provided hypothetical tolerances. The solutions within the preferred uncertainty are shown in sub-figure (d). The DM could select a solution as the final decision, provided it matches the preferences for objectives and uncertainties. (S)he could view a new set of solutions by resetting the cutoff tolerances. Alternatively, the DM could choose to change the preferences for objectives and continue the optimization if (s)he was not satisfied with any solution.

The primary goal of the framework is to enable the DM to visualize and get information about the uncertainty in the solutions while interactively solving an offline data-driven MOP. If decision making is solely based on the objective values of the solutions, it can be misleading (as shown in Figure 10). The proposed framework resolves these issues by enabling the DM to change the preferences for uncertainties in the form of tolerances and make decisions based on a wide range of solutions with various uncertainties. The two-step interaction process also does not excessively increase the cognitive load on the DM. This is primarily because the DM does not need to consider and provide two different types of preferences simultaneously.

However, one of the problems with the framework is that the adaptation of reference vectors does not consider the uncertainty in the solutions. Therefore, the DM misses out on solutions within the preference for objectives in the underlying objective space. It is also desirable that the MOEA used is also capable of handling uncertainty in the predictions of the surrogates. As we proposed the probabilistic selection approaches in [PII], we decided to incorporate them in the framework. Next, we discuss the extended framework that combines the probabilistic selection approach and a new reference vector adaptation technique.

5.2 An Improved Framework

While solving offline data-driven MOPs, the preferences for objectives provided by the DM are based on the observations made in the surrogate objective space. As we know, the objective values in the surrogate objective space are uncertain and hence do not necessarily reflect the objective values of the underlying objective functions. Therefore, the solutions found by the MOEA may not reflect the preferences for objectives provided by the DM. Hence, in the improved framework, we modify the reference vector adaptation in (15) to adjust the spread of the reference vector based on the uncertainty in the solutions. We further extend the framework proposed in [PIII] with an improved reference vector adaptation, population injection technique and probabilistic selection for decomposition-based MOEAs [PII].

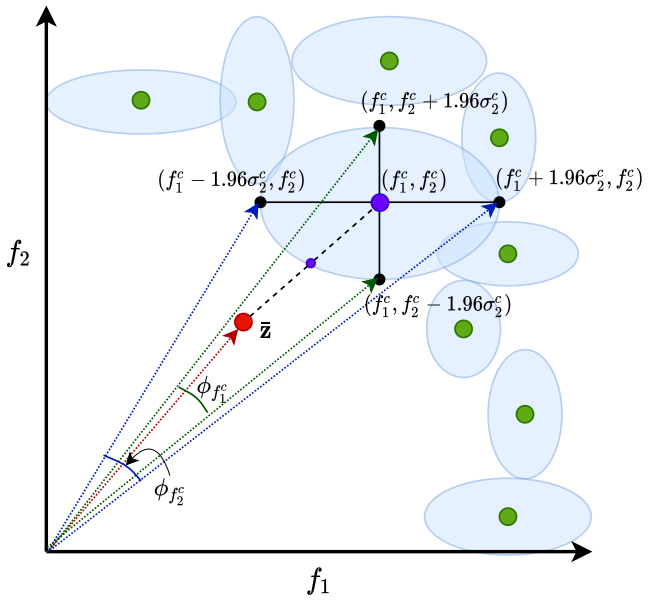


FIGURE 11 The nearest solution (in purple) to the reference point provided by the DM (in red). Blue and green dotted lines represent the vectors to the lower and upper bounds of the 95% confidence intervals of the objective values of the nearest solution.

We propose an approach for adapting the reference vectors that is dependent on the uncertainties of the solution nearest to the reference point ($\bar{\mathbf{z}}$) provided by the DM. We perform this by adapting the reference vectors for each component of objectives. If the closest solution has a high uncertainty in one particular objective, the spread of the reference vectors is increased only for that objective. We utilize (15) and selectively adapt the components of every vector depending on the uncertainty in the nearest solution.

An illustration of the proposed reference vector adaptation technique for a bi-objective minimization problem is shown in Figure 11. In the figure, we show several solutions in green. The shaded region in blue represents the distribution of objective values (95% confidence interval) predicted by Kriging surrogates. The DM provides the preferences for objectives as a reference point, $\bar{\mathbf{z}}$ (shown in red). As the objective values of the solutions have uncertainties, the distance between the reference point and the solutions is also uncertain. To find the closest solution from the reference point, we first draw S samples using Monte Carlo sampling [46] from the PDF of the objective values as previously defined in (19). Next, we calculate the mean Euclidean distance between all the samples drawn. The solution with the minimum mean Euclidean distance is the closest to the reference point with objective values $\{f_1^c, \cdots, f_k^c, \cdots, f_K^c\}$ and is shown in purple in Figure 11. The points in black over the distribution of the nearest solution are the lower and upper bounds of the 95% confidence interval for each objective. The lower

and upper bounds for the k^{th} objective are $f^c_{k\min}$ and $f^c_{k\max}$, respectively, where:

$$f^c_{k^{\min}} = f^c_k - 1.96\sigma^c_k f^c_k \ , \text{ and}$$
$$f^c_{k^{\max}} = f^c_k + 1.96\sigma^c_k f^c_k.$$

The vectors to the lower and upper bounds of the confidence interval of f^c_k are represented by:

$$\mathbf{f}^c_{k^{\min}} = \{f^c_{1^{\min}}, \cdots, f^c_{k^{\min}}, \cdots, f^c_{K^{\min}}\} \ , \text{and}$$
$$\mathbf{f}^c_{k^{\max}} = \{f^c_{1^{\max}}, \cdots, f^c_{k^{\max}}, \cdots, f^c_{K^{\max}}\},$$

respectively. These bounds of confidence intervals provide the spread amount required in each of the components of the reference vectors while adapting them. Therefore, we need to calculate the angle between these lower and upper bounds of the confidence interval with respect to the ideal point. In other words, we calculate the angle between the lower and upper bound vectors for each objective. We find the cosine of angles $\phi^{f^c_k}$ by a dot product between the extreme vectors (shown in dotted blue and green) for the k^{th} objective using the following equation:

$$\cos\phi^{f^c_k} = \frac{\mathbf{f}^c_{k^{\min}} \cdot \mathbf{f}^c_{k^{\max}}}{\left\|\mathbf{f}^c_{k^{\min}}\right\| \cdot \left\|\mathbf{f}^c_{k^{\max}}\right\|}. \tag{25}$$

Finally, we calculate the adapted vector component v^i_k for the k^{th} objective and i^{th} reference vector by using the following equation:

$$v^i_k = r_k \cdot v^i_k + (1 - r_k) \cdot v^i_k, \tag{26}$$

where r_k is the component-wise spread parameter that is adjusted as $r_k = \beta + \alpha(1 - \cos\phi^{f^c_k})$. The bias parameter, β, controls the minimum amount of change in the reference vectors' position when the uncertainty in the solution is zero. The weight parameter, α, controls the magnitude of the adaptation of reference vectors. So a higher $\phi^{f^c_k}$ means the spread parameter will be closer to one. In other words, there will be no significant change in the reference vectors' component for that objective. The final set of adapted reference vectors are found by:

$$\overline{\mathbf{v}^i} = \frac{\mathbf{v}^i}{\|\mathbf{v}^i\|} \ , \tag{27}$$

where $\mathbf{v}^i$ is the i^{th} reference vector. The adapted set of reference vectors is skewed to adjust for the solution's uncertainty in each objective.

In Figure 12 we show the steps to perform the adaptation of reference vectors. The approach starts after the DM has provided the preferences for objectives in the form of a reference point. First, we find the nearest solution in the archive from the provided reference point. The reference vectors are then adapted for every component of the objectives according to (26), with the predefined bias and weights parameters. Finally, the adapted reference vectors are normalized using (27) and used by the MOEA for the next iteration.

Additionally, in the new framework, we use a probabilistic selection approach as proposed in [PII]. The probabilistic selection approaches have proven to

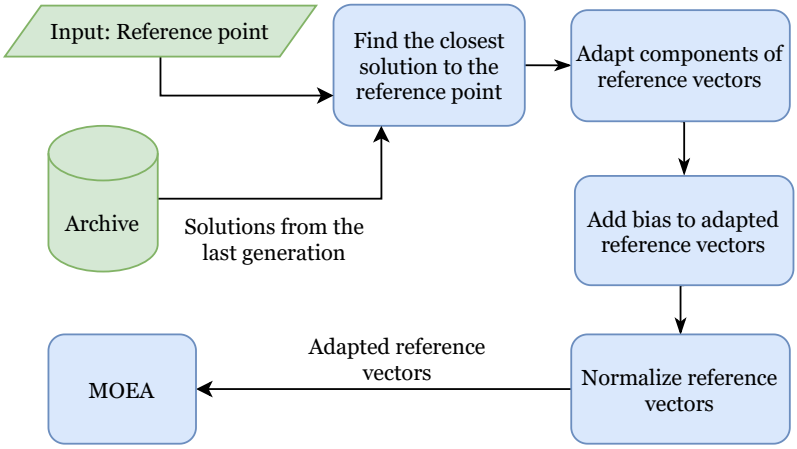


FIGURE 12 Flowchart for new reference vector adaptation embedded in the framework in Figure 8

provide solutions with better hypervolume and accuracy compared to the generic selection criterion. Therefore, even if the DM chooses to skip the step to provide preferences for uncertainties, the solutions found by the MOEA are better than using the generic selection criterion.

5.2.1 Population Injection

One of the problems faced in interactive decomposition-based MOEAs is the loss of population size when the preferences for objectives are changed drastically from the previously provided preferences for objectives. After the DM has provided new preferences for objectives far from the previously provided preferences, the reference vectors are adapted to a region in the objective space with very few or no individuals. This slows down the convergence of the MOEA leading to solutions with poorer objective values [39]. In the proposed framework, we inject a population from previous generations when the population size drops below a threshold. The injection is done in the first generation of the iteration of the MOEA, right after the DM has provided the preferences for objectives. The threshold population size in the framework is set to $P_t = 5K$ to have a sufficient number of individuals in the population. If the number of individuals falls below the threshold P_t, all the individuals from the last generation of all the previous iterations (from the archive) are included in the current population. This includes the objective values, uncertainty values and decision variable values of the individuals.

5.2.2 Experiment Setup

Testing and comparing different interactive methods is still a challenge in the field of multiobjective optimization. Here we utilized an ADM [1] for generating preferences for objectives and comparing the performance of the proposed framework. The ADM runs different interactive MOEAs simultaneously and provides

preferences for objectives in the form of a reference point. The solutions found by the different MOEAs are used to generate a composite front, and the preferences for objectives are provided for the learning and decision phases of the interaction (see Section 2.6). As the ADM has been designed for only reference point based MOEAs, it does not have the ability to provide preferences for uncertainties. Hence, we tested the various approaches only with the preferences for objectives. We implemented the proposed framework and the test setup for the ADM in Python utilizing the DESDEO framework (desdeo.it.jyu.fi).[1]

Approaches Tested

The setup of the ADM with the different interactive approaches is shown in Figure 13. In order to test the effects of different components of the new framework, we tested various setups of generic and probabilistic RVEA [PII] with APD as the selection criteria. The various interactive approaches with their components were as follows:

1. **Gen-RVEA V1**: The framework proposed in [PIII] with generic RVEA as MOEA.
2. **Prob-RVEA V1**: The framework proposed in [PIII] with probabilistic RVEA as MOEA [PII].
3. **Gen-RVEA V2**: The framework proposed in [PIII] with population injection and generic RVEA as the MOEA.
4. **Prob-RVEA V2**: The new framework that includes custom reference vector adaptation, population injection and probabilistic RVEA as MOEA [PII].

The bias and weight parameters for Prob-RVEA V2 were set as $\beta = 0.2$ and $\alpha = 0.8$, respectively. The spread parameter for reference vector adaptation for the original reference vector adaptation was kept the same as [PIII]. It should be noted that these parameters were set for demonstrating the approaches' potential. The analyst can change them depending on the requirements of the DM and the problem. The number of interactions for the learning and decision phase of the ADM was 10 and 5, respectively. We provided a higher number of interactions for the learning phase so that the objective space is well explored when the number of objectives is high. The remaining parameters for the ADM were kept the same as proposed in [1].

Dataset

In our tests, we generated the data using five distance-based visualizable test problems (DBMOPP) [21], P1-5, with various configurations as shown in Table 1. The problem instances and data were generated by the code provided in [22]. All

[1] Source code is available at https://github.com/industrial-optimization-group/Offline_IMOEA_Framework

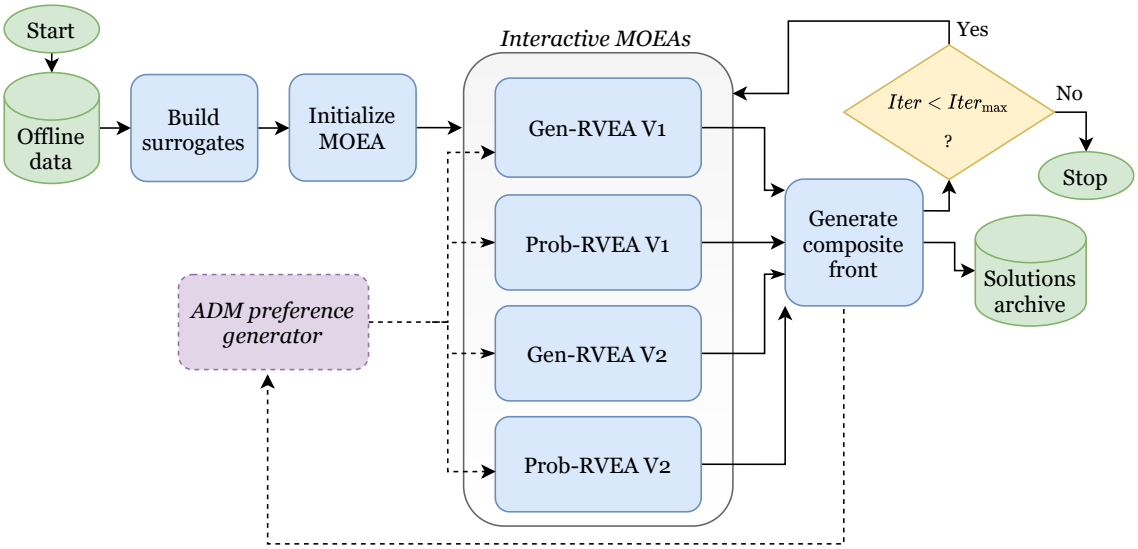


FIGURE 13 Structure for testing and comparing the approaches.

the combinations for numbers of objectives ($K \in \{5, 7, 9\}$) and number of decision variables ($n = 10$) were used for our tests. We used LHS and multivariate normal sampling (MVNS) for generating the data. The same setup was used for MVNS sampling as previously described in Section 4.5. The sample size of initial data was 109, and 11 sets of data were generated with a random seed. Each of these dataset were the starting point of the different interactive approaches that were tested. These individual runs were independent, and the results obtained were used to compare the surrogates' performances statistically.

Parameter settings of RVEA

We used RVEA with the default parameter settings as in [8]. The interaction with the ADM and reference vector adaptation was done after every 100 generations that were sufficient for the solutions found by the MOEA to converge. The stopping criterion was $Iter_{\max} = 15$ interactions (10 interactions for the learning phase and 5 for the decision phase).

Performance indicators

Measuring the performance of interactive methods is still a research challenge. Currently, we have indicators such as expanding hypercube (EH) [4] and R-metric hypervolume (R-HV) [42] to measure and compare the performances of a priori methods with DM's preferences for objectives. Hence, we decided to use both of the indicators to measure the performance of the different approaches. Additionally, we also used multivariate RMSE by utilizing (23) to measure the accuracy of the solutions. However, there is no well-defined way to aggregate these indicators for an interactive optimization method while using an ADM. In our tests, we used the median indicator value (EH, R-HV and RMSE) of the solutions for all the

TABLE 1 Configurations of the DBMOPP problems used.

Problem	Configuration	Dimension (n)	Objectives (K)
P1	*number of disconnected set regions* = 0, *number of local fronts* = 0, *number of dominance resistance regions* = 0, *number of discontinuous regions* = 0	10	5, 7, and 9
P2	*number of disconnected set regions* = 1, *number of local fronts* = 0, *number of dominance resistance regions* = 0, *number of discontinuous regions* = 0	10	5, 7, and 9
P3	*number of disconnected set regions* = 2, *number of local fronts* = 0, *number of dominance resistance regions* = 0, *number of discontinuous regions* = 0	10	5, 7, and 9
P4	*number of disconnected set regions* = 0, *number of local fronts* = 0, *number of dominance resistance regions* = 1, *number of discontinuous regions* = 0	10	5, 7, and 9
P5	*number of disconnected set regions* = 1, *number of local fronts* = 0, *number of dominance resistance regions* = 1, *number of discontinuous regions* = 0	10	5, 7, and 9

iterations in a given run.

5.2.3 Results and Discussion

The indicator values obtained from the 11 runs were considered for a statistical comparison between the four different approaches tested. We used the same scoring and ranking methodology as described previously in Section 4.5.

In Figure 14 we show the performance of the four different approaches. Lighter colour (or towards the yellow spectrum) indicates a better performance rank for that instance. It was observed that in terms of EH and RMSE, Prob-RVEA V1 and Prob-RVEA V2 performed equally well. However, in terms of R-HV, Gen-RVEA V2 performed the best, with Prob-RVEA V2 coming second. This proves that we cannot use a single indicator to measure the performance of interactive approaches. Thus, considering both EH and R-HV, we may say that Prob-RVEA V2 performed better than Prob-RVEA V1 as the former showed better performance than the latter in R-HV. Furthermore, Prob-RVEA V2 performed better than Gen-RVEA V2 in RMSE, thus making it more desirable for the DM when solutions with better accuracy are required.

In conclusion, we proposed an improved framework that uses custom reference vector adaptation to tackle uncertainty in the solutions while the DM provides preferences for objectives. The approach also utilizes a population injection technique and probabilistic selection for RVEA proposed in [PII].

Overall, the framework proposed in [PIII] and the improved framework successfully tacked the second challenge of supporting decision making by utilizing uncertainty information (as introduced in Chapter 1). However, we were unable to test the full potential of the framework since the ADM did not support preferences for uncertainties. In future, we plan to work on developing an ADM capable of

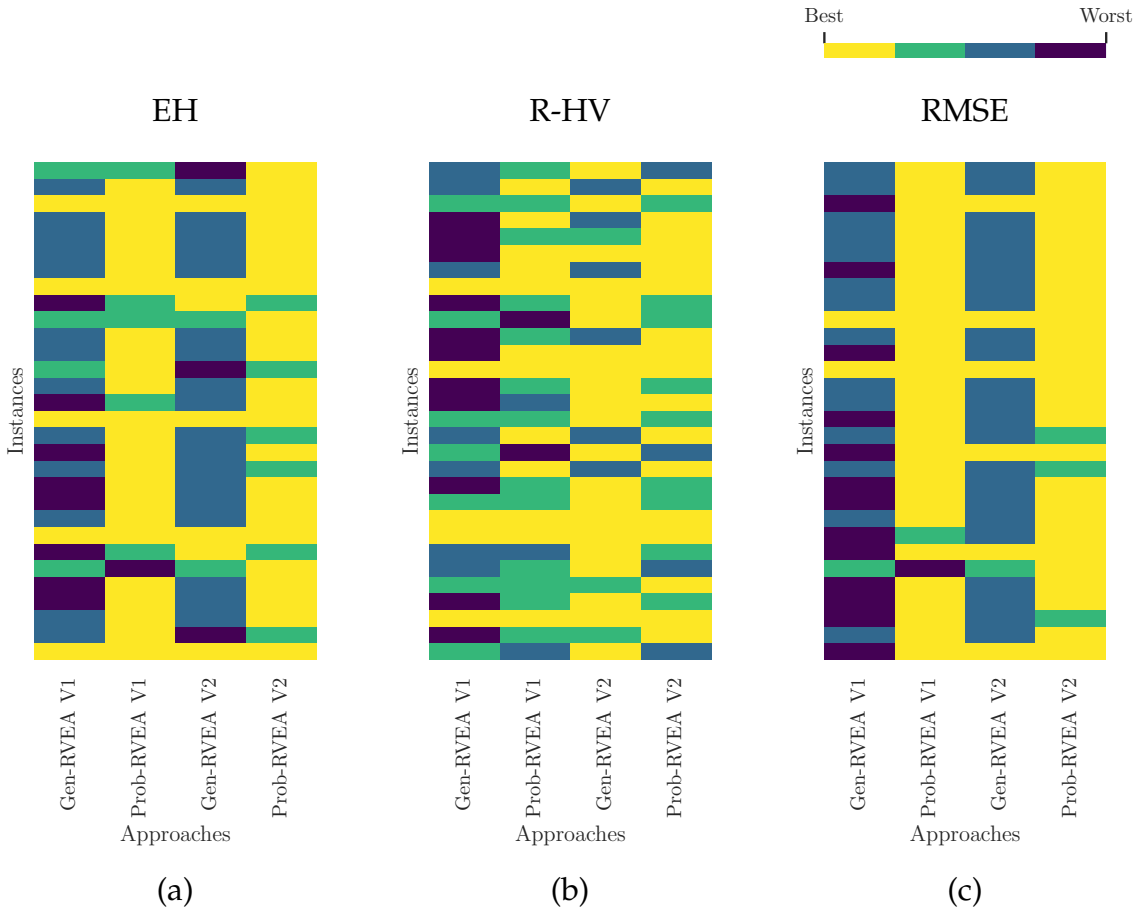


FIGURE 14 Heatmaps showing the rankings of the four interactive approaches tested for EH, R-HV and RMSE.

generating such types of preferences. As there are no reliable indicators to measure the performance of interactive methods, we had to rely on a combination of three indicators. The performance of the approaches varied with the indicator used. However, overall, the improved framework performed better than the other tested approaches. Developing a suitable indicator for measuring the performances of interactive methods will be one of our future works.

6 HANDLING LARGE DATASETS IN OFFLINE DATA-DRIVEN MULTIOBJECTIVE OPTIMIZATION

As previously described, Kriging or Gaussian process (GP) regression are widely used as surrogate for solving offline data-driven MOPs due to their ability to provide uncertainty information in the prediction. Previous works [PI, PII] have shown solutions with improved accuracy and hypervolume by utilizing the uncertainty in the prediction of GP surrogates. Most of the previous approaches build a global GP surrogate for each objective using all the provided data. However, building GPs in that way becomes computationally expensive when the size of the provided offline dataset is large. Considering N_D number of samples/items in the offline dataset, the complexity of building GP surrogates for an MOP with K objectives is $O(KN_D^3)$. Using sparse GP surrogates [61, 62] is a suitable alternative that reduces the complexity to $O(KN_D^2)$. However, searching the inducing points in sparse GPs become computationally expensive as well when the size of the dataset is large.

The surrogates used in most of the previously proposed approaches such as [10, 28, 67] are not specifically tailored for the purpose of solving offline data-driven MOPs. Approximating the objectives accurately in the entire decision space is not necessary as we are only interested in a good approximation accuracy at the neighbourhood of the Pareto set. In [PIV] we refer the neighbourhood of the Pareto set as the *tradeoff region* for simplicity. Previous works in [15, 36, 64] built local GPs by splitting the decision space into regions that reduced the computation cost. The concept of fitting GPs in each region or leaf node was extended in [27] by using a Bayesian treed model previously proposed in [9]. In [2] treed GPs were proposed to handle cases where the noise is different for different samples. However, these surrogates were not tailored to solve offline data-driven MOPs and did not take into account the tradeoff between the objectives during the building process. Furthermore, if the number of regions is quite high, several local GPs must be built, which increases the computation cost.

In our previous works [PI]-[PIII] we used GP surrogates, and considered datasets that were not large. However, building GP surrogates when the provided data is large in size still remained a challenge. Hence, in [PIV] we proposed

treed GP surrogates for offline data-driven MOPs (TGP-MO) to tackle the third challenge introduced previously in Chapter 1. These surrogates are specifically tailored to solve offline data-driven MOPs when the size of the offline dataset is large.

6.1 TGP-MO Surrogates

The building process of a generalized treed GP surrogate consists of building a regression tree first using all the provided data. Later, GPs are built using the subset of data at each leaf node. Thus, GPs are built at all the leaf nodes of the trees that accurately approximates the entire decision space of the MOP. However, while solving offline data-driven MOPs, the approximation accuracy of the tradeoff region is of utmost importance. Therefore, we proposed treed GP surrogates for multiobjective optimization (TGP-MO) that build GPs exclusively at leaf nodes consisting of the subset of data representing the tradeoff region.

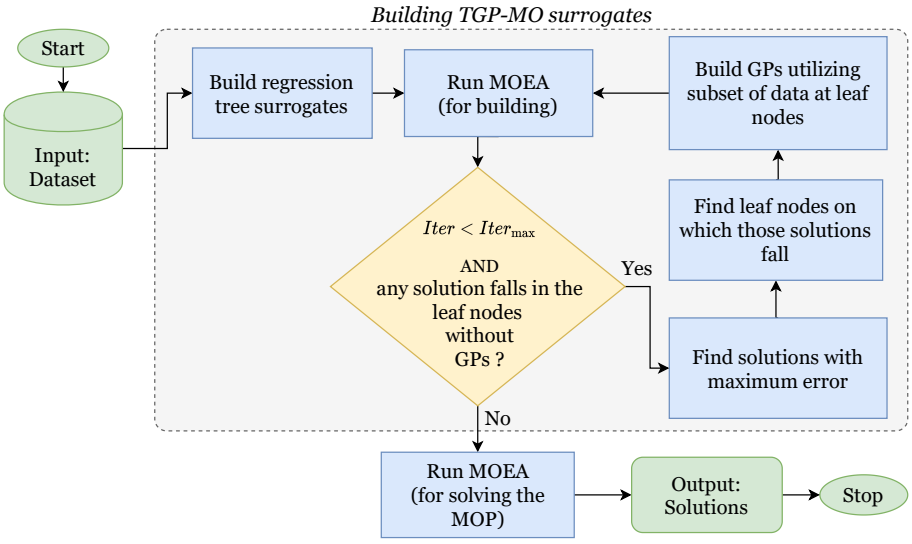


FIGURE 15 A flowchart of solving data-driven MOPs with TGP-MO surrogates with an MOEA.

The flowchart of the proposed approach of using TGP-MO surrogates is shown in Figure 15. We start by building regression trees using all the provided offline data. The depth of the trees is not controlled; however, we set the minimum number of samples for a split to occur at a node as N_{min}. Next, we initialize an MOEA that utilizes the prediction of the regression trees and solves the MOP. We examine the solutions after a certain number of generations or after completing an iteration (denoted by an iteration counter *Iter*). Next, the error in the leaf nodes (or MSE at each leaf during the building process) predicting these solutions is checked. The solutions with the maximum error for each objective are the ones where prediction accuracy needs to be improved. Thus, we build GPs at

those leaf nodes using the subset of samples at the leaf. This process of building GPs is repeated as long as; a) any solution of the MOEA is predicted by leaf nodes without a GP built, and b) $Iter < Iter_{\max}$. Thus a maximum of K GPs is built at each iteration with few samples at the leaves that inherently reduce the computation cost. After the building process is completed, the final prediction of the TGP-MO surrogates consists of the posterior predictive mean of GPs at the tradeoff region and regression trees' prediction in the rest of the decision space. The TGP-MO surrogates can then be used as objectives for another MOEA to find the final solutions of the MOP.

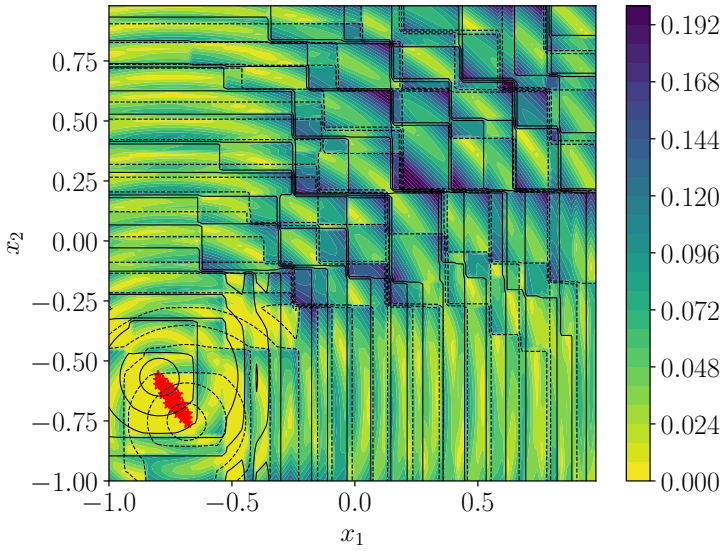


FIGURE 16 Contour plots of the objective landscape approximated by the proposed TGP-MO surrogates. The contour lines indicate the approximated objectives $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$. Colour shade represents the RMSE in the approximation (dark colour represents higher RMSE), and the red points show the Pareto set.

The regression trees perform two tasks: a) they split the decision space into sub-regions, and b) they provide an approximation (though not very accurate) of the global landscape of the underlying objectives. The solutions found by the MOEA at each iteration are located in the tradeoff region, and iteratively building GPs enables the MOEA to improve the solutions further. In later iterations, if the decision vector in the MOEA's population falls within the path of the leaf node where a GP is already built, the posterior predictive mean of the GP is used as the final prediction of the surrogate. Otherwise, the prediction is the mean value at the leaf node of the regression trees. Figure 16 shows the landscape of the approximated objectives for a bi-objective minimization problem. It can be observed that the contour lines representing the overall approximation of TGP-MO surrogates are continuous near the Pareto set (represented by red points) and discontinuous in the rest of the decision space. This is because at the tradeoff region, we use GPs that have a continuous prediction. However, in the rest of the decision space, we use the prediction from the trees. The colour shade representing

the approximation error also shows a high accuracy at the tradeoff region but low elsewhere.

The overall complexity of building the TGP-MO surrogates cannot be accurately calculated and varies with the dataset and landscape of the underlying objectives. In the worst case scenario, GPs are built at all the leaf nodes, with each leaf code consisting of a maximum of $2N_{\min} - 1$ samples. Thus the complexity of building a GP at a leaf node is $O((2N_{\min} - 1)^3)$ and the total number of leaf nodes (in the worst case) is $\frac{N_D}{2N_{\min}-1}$. The overall complexity of building GPs at all the leaf nodes is $O(N_D(2N_{\min} - 1)^2)$. For an MOP with K objectives, the worst case time complexity is $O(KN_D(2N_{\min} - 1)^2)$ with a memory complexity of $O(KN_D(2N_{\min} - 1))$.

6.2 Results and Discussion

Tests were conducted to validate and compare the quality of solutions obtained by using TGP-MO surrogates. The other two types of surrogates used were full GPs (that use all the provided data), and sparse GPs proposed in [62]. We used four configured problems from the DBMOPP benchmark [21] to generate 31 independent offline datasets of sample sizes $N_D \in \{2000, 10000, 50000\}$. The problem configurations used were P1-P4 (see Table 1 in Section 5.2.2). We used LHS [45] and multivariate normal (MVNS) sampling strategies and different number of objectives. For MVNS sampling we used the same configuration as previously described in Section 4.5. The parameters for TGP-MO surrogates were set as $N_{\min} = 10n$, where n is the number of decision variables, and $Iter_{\max} = N_D/N_{\min} = N_D/10n$. We used the sklearn Python package [51] for building the regression trees. For building GP surrogates we used the GPy package [26] with Matern 5/2 kernel. We implemented the proposed TGP-MO surrogates in Python utilizing the DESDEO framework (desdeo.it.jyu.fi).[1]

In our tests, we used RVEA [8] as an MOEA to build the TGP-MO surrogates and, later, to perform optimization primarily because of its ability to handle many-objective problems. For comparing the performance of optimization using different surrogates, we calculated the hypervolume of the solutions evaluated with the underlying objectives. For measuring the accuracy of the solutions, we used multivariate RMSE by utilizing (23). Finally, we measured the time (in seconds) taken to build the different surrogates to estimate the computation cost. All the obtained indicator values were statistically compared with Wilcoxon rank sum significance tests. For ranking the different approaches, we used the same methodology as previously described in Section 4.5.

It was observed that the building process for full GPs surrogates failed due to memory overflow for sample sizes of 10000 and 50000. This further asserted the need for surrogates capable of handling larger datasets. The proposed TGP-

[1] Source code is available at https://github.com/industrial-optimization-group/TreedGP_MOEA

MO surrogates performed the best in computation time in all the categories of sample sizes and sampling strategies. For sample sizes of 2000, full GPs produced solutions with the best hypervolume with TGP-MO surrogates coming second. For sample sizes of 10000 and 50000, TGP-MO surrogates performed the best in hypervolume for all sampling strategies. It also performed the best in terms of accuracy for LHS sampling strategy. However, the accuracy of sparse GPs was better than TGP-MO surrogates for MVNS sampling. This was primarily because sparse GPs find the inducing points by maximizing the entropy. Therefore, they are able to handle skewed datasets and hence do not suffer from overfitting [20].

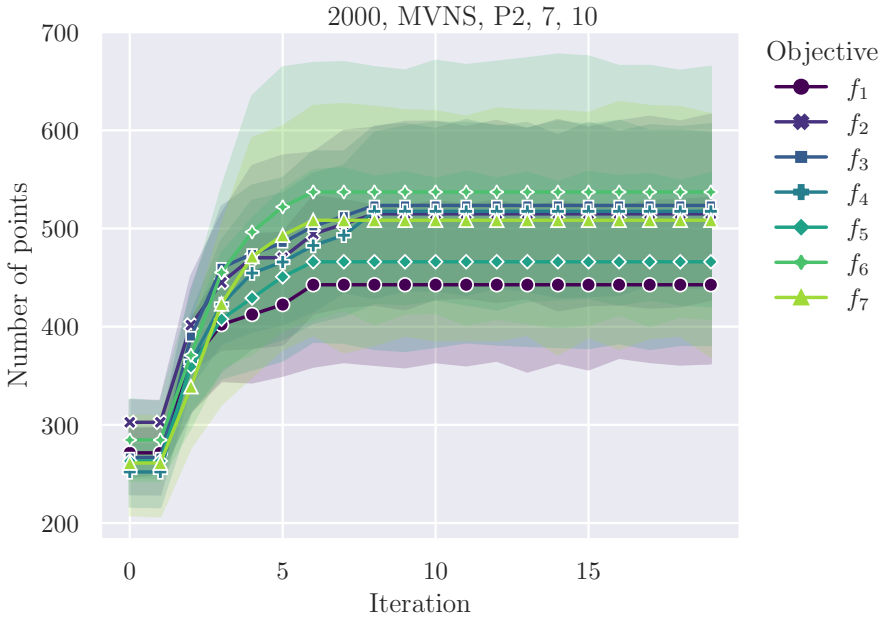


FIGURE 17 The total number of samples utilized (median and 95% confidence interval) to build GPs at leaf nodes with iterations for DBMOPP problem P2 with seven objectives, 10 decision variables and data size of 2000 generated using MVNS sampling. The iteration axis is extended to the maximum number of iterations, $\frac{N_D}{10n}$.

In Figure 17 we show the progress of the number of samples used while building TGP-MO surrogates for a DBMOPP problem instance with five objectives and ten decision variables. The offline dataset was sampled by MVNS sampling with a sample size of 2000. It can be observed that before the tenth iteration, the process of building GPs at the leaf nodes was stopped, and the number of samples utilized per objective converged. The number of samples utilized for each objective differed according to the tradeoffs between the objectives and their landscape.

The TGP-MO surrogates were proposed to address the third challenge of handling offline datasets of large size (as introduced in Chapter 1). These surrogates performed significantly better than full GPs and sparse GPs in computation time and produced solutions with a better hypervolume and accuracy. However, the prediction of the TGP-MO surrogates was not continuous at the splits of the regres-

sion trees. We can tackle this problem with the approach proposed in [64]. In the future, we plan to utilize non-linear tradeoff criteria to build the regression trees. We also plan to perform a sensitivity analysis of the hyperparameters and solve a real-world problem using TGP-MO surrogates. A leaf node GP approximates only a specific region of the decision space. This feature can be exploited while solving offline data-driven MOPs with preferences from the DM in an a priori or interactive fashion. Therefore in the future, we plan to utilize the TGP-MO surrogates to solve offline data-driven MOPs interactively. We will also test and compare the uncertainty in the predictions of TGP-MO surrogates with full GPs by applying them to the approaches proposed in [PI]-[PIII].

7 AUTHOR'S CONTRIBUTIONS

The research topics on offline data-driven multiobjective optimization and decision support were suggested by the author's supervisors. In the beginning, the author performed a literature survey on the existing works in the field. As the field is relatively new, with very little literature available, the author performed numerous tests. These tests included solving offline data-driven MOPs using Kriging as surrogates since they can provide uncertainty in its prediction. The data generated for these tests was from DTLZ [19] test problems with different sampling strategies. While testing, it was conformed that when the solutions obtained by the MOEA were evaluated using the underlying objectives, they had approximation errors. It was also observed that the uncertainty in the prediction of Kriging surrogates had a relationship with the approximation error of the surrogate objective values. This led the author to explore different ways of utilizing the uncertainty estimate from Kriging surrogates in the optimization process.

The development of the first set of approaches to utilize uncertainty from Kriging surrogates was proposed in [PI] and started in 2018. The author developed several approaches and performed extensive tests with benchmark problems. Dr. Manuel López-Ibáñez suggested checking the performance of expected improvement along with the other two approaches proposed. Using the hypervolume indicator exclusively to measure the performance of offline data-driven multiobjective optimization approaches is not sufficient. Hence, Dr. Tinkle Chugh and Prof. Kaisa Miettinen suggested using RMSE between the surrogate objective values and underlying objective values of the solutions. The author matured this, and the resulting article containing test results and analysis was accepted for publication at Evolutionary Multi-Criterion Optimization (EMO) 2019 conference.

The approaches proposed in [PI] inherently made the problem more complex by increasing the number of objectives compared to the original MOP. Furthermore, the population size had to be increased to have a sufficient number of solutions. Dr. Tinkle Chugh proposed to extend the concept of probability of dominance for decomposition-based MOEAs. Improving upon this idea, the author developed probabilistic selection approaches that could be easily adapted for any selection criteria of a decomposition-based MOEA. The author developed probabilistic

selection approaches for RVEA and MOEA/D and tested with various benchmark problems, and compared them with other state of the art offline optimization approaches. The author also utilized non-standard sampling that could better reflect real-world datasets and suggested the use of multivariate RMSE as an indicator for accuracy. In addition, the author put substantial effort in utilizing and integrating the approaches in the DESDEO framework and implementations of benchmark problems in different programming languages. The article [PII] consisting of all the results and analyses was submitted in the journal IEEE Transactions on Evolutionary Computation and is conditionally accepted.

There were two main motivations for the development of an interactive framework for offline data-driven optimization. Firstly, [PI] and [PII] proved that uncertainty information from Kriging surrogates improves the quality of the solutions. However, making decisions under uncertainty was still a challenge. Secondly, interactive approaches are beneficial since the DM is actively involved in the optimization process. Prof. Kaisa Miettinen and Dr. Jussi Hakanen suggested the use of indifference tolerance while providing preferences for uncertainties. While developing the framework, the author understood the difficulties in human cognition, and developed techniques for processing and visualizing the solutions. The framework with test results was published in the Proceedings of Bioinspired Optimization Methods and Their Applications (BIOMA) 2020 conference [PIII]. The author decided to further extend the framework by integrating it with the probabilistic selection approach proposed in [PII]. He further developed a reference vector adaptation technique that considers uncertainty in the solutions. The author performed tests using an artificial decision maker with various setups of benchmark problems and has presented the results in Section 5.2.

The author recognized the challenge of building Kriging or Gaussian process surrogates for offline datasets of large size in 2018. Various surrogate modelling techniques were developed and tested by the author without much success. In 2020, Dr. Manuel López-Ibáñez suggested exploring treed GPs as surrogate models. Various modelling approaches on the concept of treed GPs were developed and tested by the author, out of which the TGP-MO surrogates performed the best. The author conducted thorough tests with different benchmark problems, data sizes, sampling strategies and compared the performance of the TGP-MO surrogates with full GPs and sparse GPs. The resulting article was written in [PIV] and was submitted to a journal in 2021.

Overall the developments of the approaches, experiments and most of the writing in the articles were done by the author. The feedback and comments provided by the co-authors improved the author's writing ability, coherence and further refined the ideas for publishing the articles.

8 CONCLUSIONS

The primary goal of this thesis was to identify the challenges of solving offline data-driven MOPs and develop novel approaches to tackle them. The thesis is also aimed at providing better decision support approaches for solving real-world offline data-driven MOPs. In the first part of the thesis, the author described one of the major challenges faced while solving offline data-driven MOPs. Since we are unable to evaluate the underlying objectives and update the surrogates, the solutions found by the MOEA generally have low accuracy. Utilizing the uncertainty information from surrogates such as Kriging is one of the ways to find solutions with better accuracy and hypervolume when they are evaluated with the underlying objectives. A few different approaches were proposed in [PI] that utilized the uncertainty information from Kriging surrogates and treated them as additional objectives to the original MOP. The approaches produced promising results; however, they increased the complexity of the MOP being solved by increasing the number of objectives.

The concept of utilizing uncertainty information from the surrogates in the optimization process was further extended in [PII]. We proposed probabilistic selection approaches for decomposition-based MOEAs and utilized the uncertainty in the prediction of Kriging surrogates. The approaches utilize Monte Carlo sampling to estimate the distribution of the selection criterion of the decomposition-based MOEA using kernel density estimation. The probability of selection is later calculated by comparing the distribution of the selection criterion for different individuals. Finally, for selecting the individuals, the probabilities are compared or ranked based on the MOEA. The other approach proposed was a hybrid of the generic selection (without utilizing uncertainty) and the probabilistic selection. The probability of selection criterion selects solutions based on their objective values and simultaneously tries to reduce the uncertainties they have. Therefore, the solutions have a lower RMSE when they are evaluated with the underlying objectives or implemented in real life. In the experiments, we considered MOEA/D (PBI) and RVEA (APD) as our MOEAs and embedded the proposed probabilistic approaches in them. The probabilistic selection performed superior to their generic counterparts in hypervolume and RMSE for several distance-based multiobjective

visualizable test problems (DBMOPP). This further reinforced the theory that using the uncertainty information from surrogates is crucial while optimizing offline data-driven MOPs.

While solving real-world MOPs, the decision maker (DM) is generally interested in a subset of solutions (or a single solution) in the Pareto front. In [PIII] we proposed an interactive framework to solve offline data-driven MOPs that is capable of handling preferences for objectives along with preferences for uncertainties. The first challenge addressed was to enable the DM to understand and provide preferences for uncertainties during an interaction. The second challenge was not to drastically increase the cognitive load on the DM when two types of preferences are required. We proposed a two-stage interaction to split the cognitive load of the DM for providing preferences for objectives and uncertainties. The preferences for uncertainties are provided in the form of tolerances that enables the DM to visualize solutions within the threshold tolerances and preferences for objectives. The working and advantages of the framework were demonstrated on the general aviation aircraft problem.

However, as we are dealing with offline data-driven MOPs, the preferences for objectives might not guarantee the solutions to follow the DM's preferences for objectives when evaluated with the underlying objectives. Thus we further extended the framework in [PIII] by proposing a component-wise reference vector adaptation that adjusts the diversity of the solutions based on their uncertainties. We also tested a population injection technique to counter the loss of population size when preferences for objectives are changed drastically. The framework was also integrated with the probabilistic selection approaches proposed in [PII]. The framework was tested with an artificial decision maker (ADM) with various configurations of DBMOPP test problems and a number of objectives. Depending on the indicators used, we observed significant improvement with the proposed component-wise reference vector adaptation, probabilistic selection, and population injection. However, further work is needed in the development of better indicators for measuring the performances of interactive approaches and ADMs capable of providing preferences for uncertainties.

In [PIV] we proposed treed Gaussian processes for multiobjective optimization (TGP-MO). These surrogates consider the tradeoff region of the MOP being solved during the building process and are capable of handling large datasets. Test results showed that the TGP-MO surrogates outperformed full GPs and sparse GPs in computation time for the building process. The building process of full GPs failed for datasets of size (≥ 10000) due to memory overflow, thus making them impractical to use when the size of the dataset is large. The solutions obtained using TGP-MO as surrogates also had a better hypervolume and accuracy than using sparse GP surrogates for most of the instances of DBMOPP problems. Therefore using TGP-MO surrogates is a suitable choice while dealing with larger datasets and when uncertainty in the prediction of GPs is required.

The proposed approaches in this thesis are an important first step in developing tools to solve real-world offline data-driven MOPs. This includes critical processes such as traffic optimization, forest management and health management,

where performing experiments is difficult or not practical. These approaches can also be applied to solve online problems where the underlying objectives are computationally very demanding and hence can be regarded as offline. This includes design optimization problems (e.g. computational fluid dynamics) where a simulation may take multiple hours or days to complete. The approaches and frameworks are available on GitHub and implemented in the open-source framework, DESDEO (desdeo.it.jyu.fi), and therefore easily accessible to researchers and engineers.

In summary, this thesis highlights significant issues and challenges in solving offline data-driven MOPs and proposes various approaches and embeds them in frameworks to address them. The proposed approaches include utilizing uncertainty estimates from Kriging surrogates, utilizing preferences for uncertainties with decision support, and building computationally cheaper surrogates for large offline datasets. Therefore, all the three challenges introduced in this thesis were addressed in the form of articles [PI]–[PIV]. The various optimization and surrogate modelling approaches and frameworks in the articles included in this thesis will support the field of offline data-driven multiobjective optimization and improve the DM's ability to solve real-world offline data-driven MOPs.

In this thesis, the author has introduced various challenges and proposed approaches to tackle them. However, offline data-driven multiobjective optimization is still a relatively new field, and there remain substantial challenges that could not be addressed in this thesis. Hence, we intend to work on some of those areas in the future. For instance, the approaches proposed in [PI]–[PIII] rely on the uncertainty in the prediction of Kriging surrogates. However, in our tests, we observed that the uncertainty in the prediction of Kriging surrogates is underestimated. In other words, the underlying objective values lie outside the 95% confidence interval of the surrogate's approximation. This will be problematic for the DM as the variance in the underlying objective values from the surrogate's objective values is misleading. In the future, we plan to work specifically on enhancing the uncertainty in the prediction from surrogates. We also plan to investigate the uncertainty in the prediction provided by other types of models, i.e. random forest, neural networks and support vector regression. Utilizing co-Kriging methods to model parameter interactions is another interesting area we desire to work on. In future, we wish to develop probabilistic constraint handling techniques and test the effectiveness of the proposed approaches for noisy and real-world offline datasets with a real DM.

The TGP-MO surrogates only approximate certain regions of the decision space. We can exploit this feature for solving MOPs with preferences from the DM in an a priori or interactive way. The uncertainty in the prediction of TGP-MO surrogates should also be analyzed and tested for the probabilistic selection approaches in [PII]. The prediction from these surrogates have discontinuities that diminish its prediction quality for certain problems. In the future, we plan to develop non-linear splitting criteria for the regression trees to approximate the tradeoff region. This will further improve the approximation accuracy of the surrogates and require a lower number of samples and leaf node GPs.

YHTEENVETO (SUMMARY IN FINNISH)

Tämän väitöskirjan päätavoite oli kartoittaa sellaisten datapohjaisten monitavoiteoptimointiongelmien ratkaisemiseen liittyvät käytännön haasteet, joissa uutta dataa ei ole saatavilla optimoinnin aikana (ns. erilliset ongelmat) ja kehittää uusia menetelmiä näiden haasteiden ratkaisemiseksi. Väitöskirja tarjoaa myös parempia päätöksenteon tukimenetelmiä käytännön erillisten datapohjaisten monitavoiteoptimointiongelmen ratkaisemiseen. Kehitetyt menetelmät esitellään väitöskirjan tieteellisissä artikkeleissa PI-PIV ja niiden ideat ovat lyhyesti seuraavat.

Kehitetyt menetelmät hyödyntävät dataan pohjautuvia Kriging-sijaismalleja ja niiden tuottamaa tietoa mallien epävarmuudesta. Tätä epävarmuustietoa siis käytetään erillisten datapohjaisten optimointitehtävien ratkaisemisessa. Ensimmäisissä menetelmissä mallien epävarmuustieto muotoillaan optimoitaviksi lisäfunktioiksi. Seuraavissa menetelmissä epävarmuustietoa käytetään monitavoiteoptimoinnin hajotelmapohjaisten evoluutioalgoritmien valintaprosessissa tehostamaan näiden algoritmien toimittaa epävarmuuksien huomioonottamisessa. Laskennalliset tulokset kehitetyillä epävarmuuden huomioonottavilla MOEA/D (PBI) ja RVEA (APD) -menetelmillä olivat selkeästi parempia kuin vastaavilla menetelmillä, jotka eivät hyödynnä epävarmuustietoa sekä hypertilavuus- että RMSE-mittareiden suhteen useilla DBMOPP-sarjan testitehtävillä.

Väitöskirjassa esitetään myös interaktiivinen lähestymistapa tarkasteltujen erillisten datapohjaisten monitavoiteoptimointitehtävien ratkaisemiseen, jossa hyödynnetään päätöksentekijän preferenssi-informaatiota sekä optimoitaville funktioille että niihin liittyville epävarmuuksille. Lähestymistavan toimintaa ja sen tuomia hyötyjä havainnollistetaan ratkaisemalla kirjallisuudessa aiemmin esitetty päätösongelma, joka liittyy yleisilmailulentokoneen suunnitteluun. Kehitettyä lähestymistapaa laajennetaan hyödyntämään funktiokohtaista referenssivektorien päivittämistapaa, uutta populaation täydentämismenetelmää sekä artikkelissa PII esitettyä epävarmuustietoa hyödyntävää valintaa. Automaattista päätöksentekijää hyödyntävät testitulokset osoittivat laajennetun lähestymistavan paremmuuden käytettyjen mittareiden suhteen.

Lopuksi väitöskirjassa esitetään puurakenteinen Gaussisiin prosesseihin perustuva malli, joka soveltuu tilanteisiin, joissa dataa on paljon. Nämä mallit hyödyntävät optimoitavien funktioiden välistä vaihtosuhteiden joukkoa ja kykenevät käsittelemään rakennusvaiheessa suurta määrää dataa. Testitulokset osoittavat, että kehitetty TGP-MO -malli suoriutuu mallin rakentamiseen käytetyn ajan suhteen selvästi paremmin kuin vertailtavina olleet täysi GP-malli sekä harva GP-malli. Tuotetuilla ratkaisuilla on lisäksi parempi hypertilavuus ja tarkkuus verrattuna harvaan GP-malliin useimmissa DBMOPP-sarjan testitehvissä.

Väitöskijassa esitetyt uudet menetelmät muodostavat tärkeän ensiaskeleen kehitettäessä työkaluja käytännön datapohjaisten monitavoiteoptimointiongelmien ratkaisemiseen, kun uutta dataa ei ole käytettävissä. Sovelluskohteita näille menetelmille on esimerkiksi liikenteen, metsänhoidon suunnittelun ja terveydenhuollon aloilla, joissa kokeellisen tiedon hankkiminen on vaikeaa tai mahdotonta.

Kehitettyjä menetelmiä voidaan soveltaa myös laskennallisesti vaativien datapohjaisten optimointiongelmien ratkaisemiseen, joissa uuden tiedon hankkiminen on periaatteessa mahdollista mutta käytännössä mahdotonta kuten suunnitelutehtävissä (esimerkiksi laskennallisessa virtausdynamiikassa), joissa käytetään useita tunteja tai päiviä kestäviä simulaatioita.

Väitöskirjan menetelmät ovat saatavilla GitHubissa, ja ne on toteutettu osaksi avoimen lähdekoodin DESDEO-ohjelmistokehikkoa (desdeo.it.jyu.fi), mikä mahdollistaa niiden helpon saatavuuden ja edelleenkehittämisen sekä tutkijoille että käytännön soveltajille. Väitöskirjaan kuuluvissa tieteelisissä artikkeleissa esitetyt optimointi- ja mallinnusmenetelmät tukevat erillisen datapohjaisen monitavoiteoptimoinnin alaa ja parantavat päätöksentekijän mahdollisuuksia käytännön päätösten tekemisessä.

REFERENCES

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

[2] J.-A. M. Assael, Z. Wang, B. Shahriari, and N. de Freitas. Heteroscedastic treed Bayesian optimisation. *CoRR*, page arXiv:1410.7172, 2014.

[3] M. Avriel. *Nonlinear Programming: Analysis and Methods*. Dover Publishing, 2003.

[4] S. Bandaru and H. Smedberg. A parameterless performance metric for reference-point based multi-objective evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '19, pages 499–506. Association for Computing Machinery, 2019.

[5] L.C. T. Bezerra, M. López-Ibáñez, and T. Stützle. A large-scale experimental evaluation of high-performing multi- and many-objective evolutionary algorithms. *Evolutionary Computation*, 26:621–656, 2018.

[6] C. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

[7] M. Castro-Gama. mariocastrogama/GAA-Problem-MATLAB. 2017. Accessed August 31, 2020.

[8] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20:773–791, 2016.

[9] H. A. Chipman, E. I. George, and R. E. McCulloch. Bayesian CART model search. *Journal of the American Statistical Association*, 93:935–948, 1998.

[10] T. Chugh, N. Chakraborti, K. Sindhya, and Y. Jin. A data-driven surrogate-assisted evolutionary algorithm applied to a many-objective blast furnace optimization problem. *Materials and Manufacturing Processes*, 32:1172–1178, 2017.

[11] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya. A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 22:129–142, 2018.

[12] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen. A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing*, 23:3137–3166, 2019.

[13] T. Chugh, K. Sindhya, K. Miettinen, Yaochu J., T. Kratky, and P. Makkonen. Surrogate-assisted evolutionary multiobjective shape optimization of an air intake ventilation system. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 1541–1548, 2017.

[14] N. A. C. Cressie. *Statistics for Spatial Data, Revised Edition*. Wiley, 1993.

[15] K. Das and A. N. Srivastava. Block-GP: Scalable gaussian process regression for multimodal data. In *2010 IEEE International Conference on Data Mining*, pages 791–796, 2010.

[16] K. Deb. Evolutionary algorithms for multi-criterion optimization in engineering design. In K. Miettinen, P. Neittaanmäki, M. M. Mäkelä, and J. Périaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*. Wiley, 1999.

[17] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18:577–601, 2014.

[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.

[19] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Congress on Evolutionary Computation (CEC 2002)*, pages 825–830. IEEE Press, 2002.

[20] L. Feng, B. Wang, H.and Jin, H. Li, M. Xue, and L. Wang. Learning a distance metric by balancing KL-divergence for imbalanced datasets. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49:2384–2395, 2019.

[21] J. E. Fieldsend, T. Chugh, R. Allmendinger, and K. Miettinen. A feature rich distance-based many-objective visualisable test problem generator. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 541–549. ACM, 2019.

[22] J. E. Fieldsend, T. Chugh, R. Allmendinger, and K. Miettinen. fieldsend/DBMOPP_generator. 2019. Accessed June 15, 2021.

[23] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation discussion and generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 416–423. Morgan Kaufmann Publishers Inc., 1993.

[24] A. Forrester, A. Sobester, and A. Keane. *Engineering Design via Surrogate Modelling*. John Wiley & Sons, 2008.

[25] W. Frank. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.

[26] GPy. GPy: A gaussian process framework in python. http://github.com/SheffieldML/GPy, 2012. Accessed July 1, 2021.

[27] R. B. Gramacy and H. K. H Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103:1119–1130, 2008.

[28] D. Guo, T. Chai, J. Ding, and Y. Jin. Small data driven evolutionary multi-objective optimization of fused magnesium furnaces. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, 2016.

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

[30] J. Hakanen and J. D. Knowles. On using decision maker preferences with ParEGO. In H. Trautmann, G. Rudolph, K. Klamroth, O. Schütze, M. Wiecek, Y. Jin, and C. Grimme, editors, *Evolutionary Multi-Criterion Optimization, Proc.*, pages 282–297. Springer, 2017.

[31] E. J. Hughes. Evolutionary multi-objective ranking with uncertainty and noise. In E. Zitzler, L. Thiele, K. Deb, C. A. Coello Coello, and D. Corne, editors, *Evolutionary Multi-Criterion Optimization, Proc.*, pages 329–343. Springer, 2001.

[32] Y. Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1:61–70, 2011.

[33] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen. Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation*, 23:442–458, 2019.

[34] Y. Jin, H. Wang, and C. Sun. *Data-Driven Evolutionary Optimization*. Springer, 2021.

[35] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.

[36] H.-M. Kim, B. K Mallick, and C. C Holmes. Analyzing nonstationary spatial data using piecewise Gaussian processes. *Journal of the American Statistical Association*, 100:653–668, 2005.

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

[38] K. Krishnamoorthy and T. Mathew. *Statistical Tolerance Regions: Theory, Applications, and Computation*. John Wiley and Sons, 2009.

[39] G. Lai, M. Liao, and K. Li. Empirical studies on the role of the decision maker in interactive evolutionary multi-objective optimization. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 185–192, 2021.

[40] K. Li. Decomposition multi-objective evolutionary optimization: From state-of-the-art to future opportunities, 2021.

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

[42] K. Li, K. Deb, and X. Yao. R-metric: Evaluating the performance of preference-based evolutionary multiobjective optimization using reference points. *IEEE Transactions on Evolutionary Computation*, 22:821–835, 2018.

[43] H. Liu, F. Gu, and Q. Zhang. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Transactions on Evolutionary Computation*, 18:450–455, 2014.

[44] M. López-Ibáñez and J. Knowles. Machine decision makers as a laboratory for interactive EMO. In A. Gaspar-Cunha, C. Henggeler Antunes, and C. C. Coello, editors, *Evolutionary Multi-Criterion Optimization, Proc.*, pages 295–309. Springer, 2015.

[45] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[46] N. Metropolis. The beginning of the Monte Carlo method. *Los Alamos Science*, pages 125–130, 1987.

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

[48] K. Miettinen, J. Hakanen, and D. Podkopaev. *Interactive Nonlinear Multiobjective Optimization Methods*, pages 927–976. Springer, 2016.

[49] K. Miettinen, F. Ruiz, and A. P. Wierzbicki. *Introduction to Multiobjective Optimization: Interactive Approaches*, pages 27–57. Springer, 2008.

[50] W. Norbert. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The MIT Press, 1964.

[51] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011.

[52] R. C. Purshouse, K. Deb, M. M. Mansor, S. Mostaghim, and R. Wang. A review of hybrid evolutionary multiple criteria decision making methods. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 1147–1154, 2014.

[53] A. A. M. Rahat, C. Wang, R. M. Everson, and J. E. Fieldsend. Data-driven multi-objective optimisation of coal-fired boiler combustion systems. *Applied Energy*, 229:446–458, 2018.

[54] C. E. Rasmussen. *Gaussian Processes in Machine Learning*, pages 63–71. Springer, Berlin, Heidelberg, 2004.

[55] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27:832–837, 1956.

[56] R. A. Shah, P. M. Reed, and T. W. Simpson. Many-objective evolutionary optimisation and visual analytics for product family design. In L. Wang, A.H.C. Ng, and K. Deb, editors, *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*, pages 137–159. Springer, 2011.

[57] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104:148–175, 2016.

[58] S. M. Shavarani, M. López-Ibáñez, and J. Knowles. Realistic utility functions prove difficult for state-of-the-art interactive multiobjective optimization algorithms. In *Genetic and Evolutionary Computation Conference, Proc.*, pages 457–465. Association for Computing Machinery, 2021.

[59] B.W Silverman. *Density Estimation for Statistics and Data Analysis*. London: Chapman Hall/CRC, 1986.

[60] T. W. Simpson, J. K. Allen, W. Chen, and F. Mistree. Conceptual design of a family of products through the use of the robust concept exploration method. In *6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pages 1535–1545, 1996.

[61] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, pages 1257–1264. MIT Press, 2005.

[62] M. K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In D. van Dyk and M. Welling, editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, pages 567–574. PMLR, 2009.

[63] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh. A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation*, 21:440–462, 2017.

[64] B. van Stein, H. Wang, W. Kowalczyk, T. Bäck, and M. Emmerich. Optimally weighted cluster kriging for big data regression. In E. Fromont, T. De Bie, and M. van Leeuwen, editors, *Advances in Intelligent Data Analysis XIV*, pages 310–321. Springer, 2015.

[65] T. Voß, N. Hansen, and C. Igel. Recombination for learning strategy parameters in the MO-CMA-ES. In *Evolutionary Multi-Criterion Optimization, Proc.*, pages 155–168. Springer, 2009.

[66] H. Wang and Y. Jin. A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems. *IEEE Transactions on Cybernetics*, 50:536–549, 2020.

[67] H. Wang, Y. Jin, and J. O. Jansen. Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system. *IEEE Transactions on Evolutionary Computation*, 20:939–952, 2016.

[68] H. Wang, Y. Jin, C. Sun, and J. Doherty. Offline data-driven evolutionary optimization using selective surrogate ensembles. *IEEE Transactions on Evolutionary Computation*, 23:203–216, 2019.

[69] H. Wang, M. Olhofer, and Y. Jin. A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges. *Complex & Intelligent Systems*, 3:233–245, 2017.

[70] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer. An adaptive Bayesian approach to surrogate-assisted evolutionary multi-objective optimization. *Information Sciences*, 519:317–331, 2020.

[71] B. Xin, L. Chen, J. Chen, H. Ishibuchi, K. Hirota, and B. Liu. Interactive multiobjective optimization: A review of the state-of-the-art. *IEEE Access*, 6:41256–41279, 2018.

[72] C. Yang, J. Ding, Y. Jin, and T. Chai. Offline data-driven multiobjective optimization: Knowledge transfer between surrogates and generation of final solutions. *IEEE Transactions on Evolutionary Computation*, 24:409–423, 2020.

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

[74] Q. Zhang, W. Liu, E. Tsang, and B. Virginas. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14:456–474, 2010.

[75] J. Zheng, G. Yu, Q. Zhu, X. Li, and J. Zou. On decomposition methods in interactive user-preference based optimization. *Applied Soft Computing*, 52:952–973, 2017.

[76] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature - PPSN VIII*, pages 832–842. Springer, 2004.

ORIGINAL PAPERS

PI

ON DEALING WITH UNCERTAINTIES FROM KRIGING MODELS IN OFFLINE DATA-DRIVEN EVOLUTIONARY MULTIOBJECTIVE OPTIMIZATION

by

Atanu Mazumdar, Tinkle Chugh, Kaisa Miettinen, and Manuel López-Ibáñez
pages 463–474. Springer, 2019

In Kalyanmoy Deb, Erik Goodman, Carlos A. Coello Coello, Kathrin Klamroth, Kaisa Miettinen, Sanaz Mostaghim, and Patrick Reed, editors, *Proceedings on Evolutionary Multi-Criterion Optimization*

On Dealing with Uncertainties from Kriging Models in Offline Data-driven Evolutionary Multiobjective Optimization

Atanu Mazumdar[1], Tinkle Chugh[2], Kaisa Miettinen[1], and Manuel López-Ibáñez[3]

[1] University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland
[2] Department of Computer Science, University of Exeter, UK
[3] Alliance Manchester Business School, University of Manchester, UK

Abstract. Many works on surrogate-assisted evolutionary multiobjective optimization have been devoted to problems where function evaluations are time-consuming (e.g., based on simulations). In many real-life optimization problems, mathematical or simulation models are not always available and, instead, we only have data from experiments, measurements or sensors. In such cases, optimization is to be performed on surrogate models built on the data available. The main challenge there is to fit an accurate surrogate model and to obtain meaningful solutions. We apply Kriging as a surrogate model and utilize corresponding uncertainty information in different ways during the optimization process. We discuss experimental results obtained on benchmark multiobjective optimization problems with different sampling techniques and numbers of objectives. The results show the effect of different ways of utilizing uncertainty information on the quality of solutions.

Keywords: Machine learning · Gaussian process · Pareto optimality · Metamodelling · Surrogate

1 Introduction

Sometimes in real applications, multiple conflicting objectives should be optimized, but there is no mathematical or simulation model of the objectives involved. Instead, there is data, e.g., obtained via physical experiments. In such cases, surrogate models can be built using the given data and optimization is then performed with the surrogate models. In the literature, surrogate models such as Kriging [8], neural networks [18] and support vector regression [16] have been typically used for solving computationally expensive optimization problems [6,10]. If we may conduct new (expensive) function evaluations when needed, this process is called online data-driven optimization [20]. When we do not have access to additional data during the optimization, we call it *offline data-driven optimization* [11].

In using surrogate models, the main challenge is to manage the models for improving convergence and diversity without too much sacrifice in the accuracy of models. In online data-driven optimization problems, an infill criterion [6] is maximized or minimized for updating the models iteratively during the optimization process. However, this is not applicable for offline data-driven optimization when no further data is available during the optimization process.So far, little research has been conducted on solving optimization problems, where no new data is available for managing the surrogates [4,11,20]. In such case, the quality of the solutions obtained after using the surrogate models is entirely dependent on the accuracy of the models and optimizer used.

When solving an offline data-driven problem with multiple conflicting objectives, one can fit models using all the data available for each objective function. Then an evolutionary multiobjective optimization (EMO) algorithm can be used on these models to find a set of approximated nondominated solutions. Essentially, in that case, an offline data-driven multiobjective optimization problem (MOP) can be divided into two major parts: model building and using an EMO algorithm.

Some surrogate models, like Kriging, provide uncertainty information (or standard deviation) about the predicted values. A low standard deviation implies that the actual objective function value has a higher chance of being close to the predicted value (though the actual function may remain unknown and the only information is the data available). Therefore, one possible way to improve the accuracy of the model is to utilize uncertainty in the fitted model as an additional objective to be optimized.

In this article, we study different ways to deal with the uncertainty information provided by the Kriging models in offline data-driven multiobjective optimization. Moreover, we consider the effect of using different initial sampling techniques on some benchmark test problems. In this study, we simulate offline problems by generating data for problems with known optimal solutions to be able to analyze the results. The results show the effect of utilizing uncertainty information in the quality of solutions.

The rest of this article is organized as follows. We summarize the basic concepts of data-driven optimization and Kriging model in Section 2. In Section 3, we present different approaches of incorporating uncertainty information in the optimization problem and present and analyze the results in Section 4. Finally, we draw conclusions in Section 5.

2 Background

2.1 Generic Offline Data-Driven EMO

We consider MOPs of the following form :

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

with k (≥ 2) objective functions and the feasible set S is a subset of the decision space $\mathbb{R}^n$. For any feasible decision vector $\mathbf{x}$ we have a corresponding objective vector $f(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))$.

MOPs that are offline in nature can generally be solved by the approach given in Fig. 1. In what follows, we refer to it as a *generic approach*. As described in [11,21], the solution process can be split into three major components: (1) data collection, (2) model building and management, and (3) EMO method utilized. The collection of data may also incorporate data pre-processing, if it is required. Once the data has been obtained, the objectives and constraints of the MOP are formulated. The next stage is to build surrogate models (also known as meta-models) e.g. for each objective function using the available data. Finally, an EMO method is used to find nondominated solutions utilizing the surrogates as objective functions. As objectives to be optimized in (1) we have for $i = 1, \ldots, k$ the predicted means $\hat{f}_i$ of the surrogate of objective f_i and our objective vector is denoted by:

$$\hat{\mathbf{f}} = (\hat{f}_1(\mathbf{x}), \ldots, \hat{f}_k(\mathbf{x})). \tag{2}$$

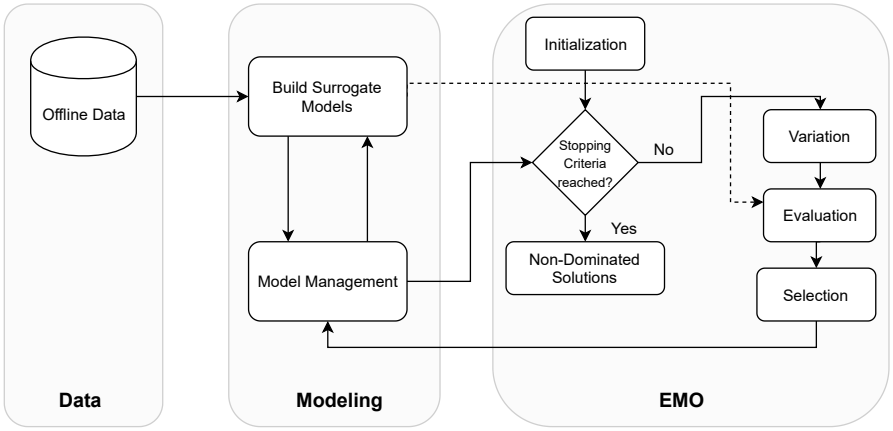


Fig. 1. Flowchart of a generic offline data-driven evolutionary multiobjective optimization approach.

Selecting proper surrogate models is a challenging task in model management. In online data-driven EMO, the quality of the surrogate models can be accessed and updated as new data becomes available during the optimization process. However, for offline data-driven EMO this is not possible. It becomes even more challenging with the data being noisy [22], skewed [23], time-varying [2] or heterogeneous [3]. Thus, it is crucial to build, before optimization, surrogates that are as good approximations as possible of the "true" objective functions. One way to improve the accuracy of the surrogates is to enhance the quality of the

data. In this research, our consideration is on a general level and we do not go into the characteristics of the data.

In offline data-driven EMO, the possible ways to improve the accuracy of the surrogate models are to have an effective data pre-processing for noise removal [4], creating synthetic data [23], transferring knowledge [15] or applying advanced machine leaning techniques [19,20]. However, it is quite possible that the surrogate models are not good representations of the true objectives. It may even happen that the solutions obtained are actually worse than the data used for fitting the models.

2.2 Kriging

Kriging or Gaussian process regression has been widely used as a surrogate model for solving expensive optimization problems [6]. The main advantage of using Kriging is its ability to provide uncertainty information of the predicted values. Given a Kriging model, the approximated mean value y^* and its variance s^2 for a sample (or decision variable value) $\mathbf{x}^*$ are as follows:

$$y^* = \mathbf{k}(\mathbf{x}^*, \mathbf{X})K(\mathbf{X}, \mathbf{X})^{-1}\mathbf{y}, \tag{3}$$

$$s^2 = \mathbf{k}(\mathbf{x}^*, \mathbf{x}^*) - K(\mathbf{x}^*, \mathbf{X})K(\mathbf{X}, \mathbf{X})^{-1}K(\mathbf{X}, \mathbf{x}^*), \tag{4}$$

where $\mathbf{X} \in \mathbb{R}^{N_I \times n}$ is the matrix of the given data with N_I items with n decision variables, $\mathbf{y} \in \mathbb{R}^{N_I}$ is the vector of given objective values corresponding to some decision vector, $K(\mathbf{X}, \mathbf{X})$ is the covariance matrix of $\mathbf{X}$ and $\mathbf{k}(\mathbf{x}^*, \mathbf{X})$ is a vector of covariances between $\mathbf{x}^*$ and $\mathbf{X}$. For more details about Kriging, see [17].

3 Approaches to Incorporate Uncertainty

As new data cannot be obtained in offline data-driven optimization, it is difficult to update the surrogates and enhance their accuracy. One approach is to build a very accurate surrogate model before the optimization process. Another possible approach is to provide a suitable metric in addition to final solutions after the optimization process, which can be used to measure the accuracy of solutions obtained. This approach can be beneficial when the surrogate models cannot provide a very exact representation of the true objective functions. One such instance can be when the data consists of optimal solutions. In such a case, the surrogate might not be a good representation of the actual objectives, which might lead to degraded final solutions. Providing a set of solutions together with the uncertainty information of predicted final solutions can be helpful in the decision making process.

As previously discussed, the two major components in offline data-driven optimization are building a surrogate model and using an EMO algorithm. In this research we have limited ourselves by focusing on a few variations of the optimization problem which try to minimize the uncertainty in the final solutions. As shown in Figure 2, the uncertainties in the predicted value of the Kriging

models are utilized as additional objective functions. By considering uncertainties in this way, the EMO method tries to minimize the predicted mean values from the fitted Kriging models by subsequently minimizing the standard deviations in the prediction. Thus, the final set of nondominated solutions will consist of solutions with different levels of uncertainty.

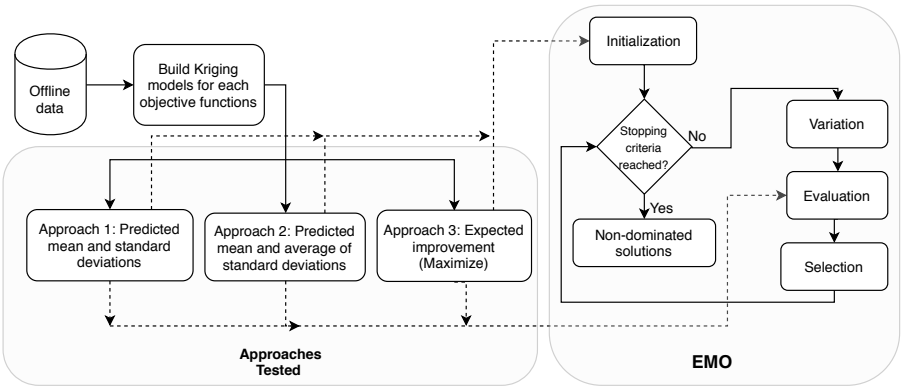


Fig. 2. Flowchart of offline data-driven optimization with uncertainty.

We have tested three different approaches for utilizing uncertainties in the optimization. Approach 1 uses all the standard deviations given by each surrogate model as additional objectives. The resulting objective vector in Approach 1 is:

$$\hat{\mathbf{f}} = (\hat{f}_1(\mathbf{x}), \ldots, \hat{f}_k(\mathbf{x}), s_1(\mathbf{x}), \ldots, s_k(\mathbf{x})), \tag{5}$$

where $\hat{f}_i(\mathbf{x})$ and $s_i(\mathbf{x})$ and are the predicted mean and the standard deviation values for the i^{th} objective. Final solutions are obtained by performing a non-dominated sort on the archive of predicted solutions (predicted mean values and standard deviations) stored while optimization. It might be possible that the solutions have different uncertainties for different objectives. We double the number of objectives which may increase the complexity of solving the resulting optimization problem.

Approach 2 utilizes the average of the standard deviations given by each of the surrogate models as an additional objective and the resulting objective vector is:

$$\hat{\mathbf{f}} = (\hat{f}_1(\mathbf{x}), \ldots, \hat{f}_k(\mathbf{x}), \bar{s}(\mathbf{x})), \tag{6}$$

where $\bar{s}(\mathbf{x})$ is the average of the standard deviations from Kriging models built for each objective function. This method has fewer objectives when compared to Approach 1, however, either of the approaches provide solutions with a range of uncertainty values. Both Approaches 1 and 2 can provide an option for filtering solutions based on the uncertainty information.

Algorithm 1 Uncertainties as additional objective functions

Input: k Kriging models, one for each objective function and an empty archive
Output: Final nondominated approximated solutions from the archive

1: Generate parent population
2: **while** *Stopping criteria are not reached* **do**
3: Generate offspring with crossover and mutation
4: Evaluate offspring using Kriging models and get the objective function values of either Eqs. (4), (5) or (6)
5: Combine offspring population with parent population
6: Select parents for the next generation
7: Store parents in the archive
8: Perform nondominated sorting of solutions in the archive

Approach 3 utilizes the expected improvement (EI) [12] for every surrogate model as objectives to be optimized by the EMO algorithm, see, e.g. [9]. Expected improvement can be expressed as $\mathrm{EI}(\mathbf{x}) = (f_{min} - \hat{f}(\mathbf{x}))\Phi\left(\frac{f_{min}-\hat{f}(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\phi\left(\frac{f_{min}-\hat{f}(\mathbf{x})}{s(\mathbf{x})}\right)$, where $\phi(\cdot)$ and $\Phi(\cdot)$ are the standard normal density and distribution function respectively, and f_{min} is a k-dimensional vector, where the i^{th} component represents the best values of the i^{th} objective function in the given data. The objective vector in this case is:

$$\hat{\mathbf{f}} = (\mathrm{EI}_1(\mathbf{x}), \ldots, \mathrm{EI}_k(\mathbf{x}))\,, \tag{7}$$

where $\mathrm{EI}_i(\mathbf{x})$ is the expected improvement value for the i^{th} objective. The EI criterion takes the predicted mean value and the standard deviation into account.

Now we have introduced three approaches for incorporating uncertainty information. Algorithm 1 shows the process of applying any of them in the offline optimization process, where k is the number of objectives and we can use the maximum number of evaluations using surrogate models as a stopping criterion.

4 Experimental Results

We compare the three different approaches to each other and also to a generic approach (as (2) in Subsection 2.1), using test problems DTLZ2, DTLZ4–DTLZ7 with 2, 3 and 5 objectives. As said, we generate data for these problems and fit Kriging models there. The dimension of the decision variable space n is fixed to 10.

The size of the data set used is 109 (corresponds to the $11n - 1$ [5,13,24]). The sampling techniques for creating the data sets were *Latin hypercube sampling* (LHS), *uniform random sampling* and a special case of sampling which we call *optimal-random sampling*. In the latter, 50% of the data are nondominated solutions and the remaining 50% are uniform random samples. This kind of hypothetical sampling might resemble a special case where most of the samples in the given data set are close to optimal, and thus the optimization process could

no longer improve the solutions further. However, in such a scenario the offline optimization technique should not compute final solutions which are worse than the provided samples. A total of 31 independent runs from each sampling were performed for each case.

We used indicator based evolutionary algorithm (IBEA) [25] as the EMO method as it has been demonstrated to perform well in [1] even for problems with a higher number of objectives. The selection criterion was $I_{\epsilon+}$ (Step 6 in Algorithm 1) with κ parameter values 0.51, 0.87 and 0.48 for $k = 2, 3$ and 5, respectively, and κ value of 0.5 for any other number of objectives. The population size was 100 and the maximum number of function evaluations was 40 000 according to [1]. We used Matlab implementation of Kriging models with first order polynomial functions and a Gaussian kernel function.

For measuring the performance of different approaches, we first performed a nondominated sort on the archive (also including the additional objective(s)). These nondominated solutions were then evaluated with the real objective function. After obtaining their true objective function values, dominated solutions were removed producing the final nondominated set. For comparing the quality of solutions for all the approaches, inverted generational distance (IGD) metric was utilized with 5000 points in the reference set for all problems.

Table 1 shows the comparison between the mean and standard deviation values of the IGD for all the three approaches and the generic approach. It was observed that Approaches 1 and 2 performed better than the generic approach for LHS and uniform random sampling for all the problems with various numbers of objectives with the exception of DTLZ6 and DTLZ7. However, while using optimal-random sampling, Approaches 1 and 2 performed better than the generic approach for DTLZ2, DTLZ4-5 and better for DTLZ6 and DTLZ7 for few of the objectives. Approach 3 did not produce good results for any of the problems, objectives or sampling technique.

Adding uncertainties as additional objectives pose a major problem in explaining the effect of optimization as the fitness landscape of the uncertainties is mostly unknown. A possible explanation that no noticeable performance improvement is observed in DTLZ6 when using Approaches 1 and 2 is because the problem consists has a non-uniform (or biased) [7] degenerated Pareto front. Adding additional uncertainty objectives makes the problem even harder to solve and fewer nondominated solutions are obtained. For DTLZ7, a possible explanation for the worse performance of Approaches 1 and 2 is that the objective functions are completely separable [14]. Thus, the additional objectives added by Approaches 1 and 2 only make the problem more difficult than the generic approach.

For optimal-random sampling the advantage of Approaches 1 and 2 was clearly visible. Despite the initial sampling including also nondominated solutions, the generic approach failed to provide good solutions. This is because the surrogate models do not provide a perfect representation of the true objectives. While utilizing EIs as objectives in Approach 3, the solutions were actually worse (comparing mean IGD values) for most of the cases. This is because EI tries to

balance between convergence and diversity. Therefore, it can select a solution with a high uncertainty for achieving its goal.

Figure 3 shows the root mean square error (RMSE) of the final solutions obtained by different approaches with LHS sampling on problems with two objectives. It can be observed that the solutions obtained by Approaches 1 and 2 are more accurate in most of the cases. This means that using uncertainty as additional objective(s) helps to find solutions with a low approximation error. Therefore, using uncertainty in the optimization process can be considered as an advantage in solving an offline data-driven EMO problem where there is no possibility for updating the surrogate models. An illustration of solutions obtained after evaluating them with real objectives for the DTLZ2 problem with LHS and optimal-random sampling is shown in Figure 4. Due to space limitations, further analysis is available at bluehttp://www.mit.jyu.fi/optgroup/extramaterial.html as additional material. The performance of the proposed approaches on other test problems (i.e., DTLZ1, DTLZ3, WFG1-WFG3, WFG5 and WFG9) can also be found at the above-mentioned website.

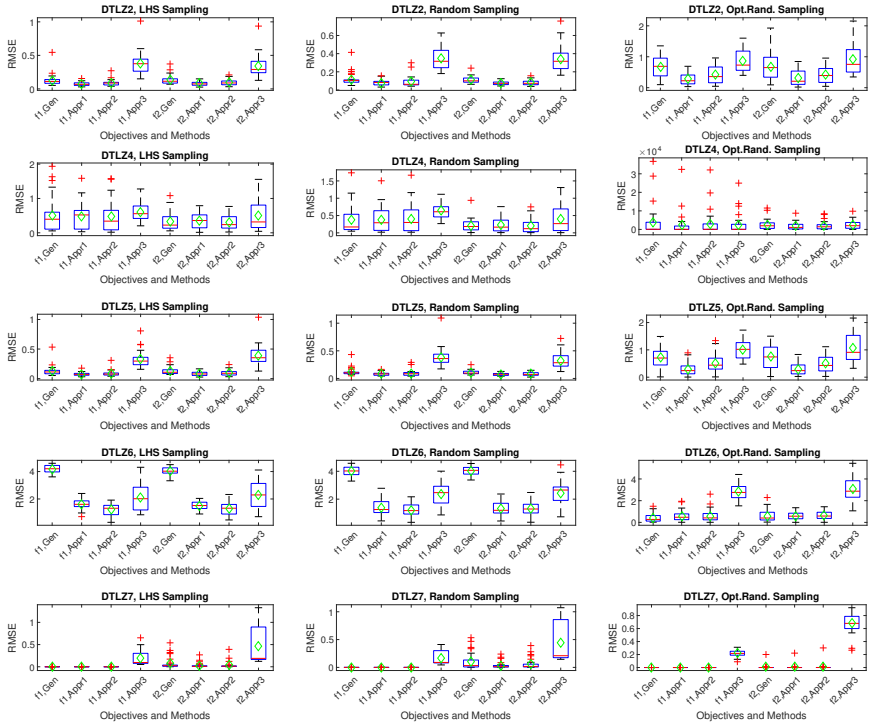


Fig. 3. RMSE of the final solutions for bi-objective problems. Here f1 and f2 are the objectives and "Gen","Appr1","Appr2" and "Appr3" are the generic and Approaches 1, 2 and 3, respectively. Opt.Rand is optimal-random sampling.

Table 1. Means and standard deviations of IGD values of the final archive, evaluated on the true objective functions, obtained by each approach, for various problems and sampling techniques. (Best values are in bold)

Sampling	Problems	k	Generic		Approach 1		Approach 2		Approach 3	
			Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.
LHS	DTLZ2	2	0.0989	0.1260	**0.0722**	**0.0431**	0.0770	0.0651	0.3377	0.0477
		3	0.2027	0.0910	0.1787	0.0530	**0.1665**	0.0539	0.3471	**0.0365**
		5	0.2708	0.0873	0.2689	**0.0343**	**0.2574**	0.0396	0.3993	0.0395
	DTLZ4	2	0.6311	**0.1619**	**0.3951**	0.1935	0.4919	0.1852	0.6467	0.2098
		3	0.7306	0.2021	**0.5309**	0.1413	0.5867	0.1467	0.7166	**0.1162**
		5	0.6929	0.0766	**0.5640**	0.0653	0.6062	0.0545	0.7173	**0.0514**
	DTLZ5	2	0.1030	0.1326	0.1032	0.0905	**0.0814**	**0.0570**	0.3716	0.0580
		3	0.1191	0.0982	**0.0684**	**0.0315**	0.0701	0.0452	0.2676	0.0388
		5	0.0934	0.0606	**0.0655**	**0.0277**	0.0805	0.0453	0.1486	0.0387
	DTLZ6	2	**0.1570**	**0.1078**	1.6188	0.7635	2.4518	0.5797	3.5210	1.1369
		3	**0.9871**	**0.2737**	1.7564	0.7308	1.5561	0.7159	3.2847	1.1907
		5	**0.8207**	**0.2158**	2.3859	0.4822	1.3725	0.3734	2.8157	1.0211
	DTLZ7	2	**0.0023**	**0.0049**	0.0292	0.0095	0.0095	0.0086	0.6157	0.1767
		3	**0.0549**	**0.0120**	0.1791	0.1721	0.0956	0.1449	0.6529	0.1016
		5	**0.2800**	**0.0541**	0.5254	0.2175	0.3675	0.1234	0.7169	0.0888
Random	DTLZ2	2	0.0947	0.0893	0.0879	0.0468	**0.0828**	0.0493	0.3673	**0.0395**
		3	0.2315	0.0712	0.1907	0.0534	**0.1692**	**0.0316**	0.3591	0.0433
		5	0.2843	0.0790	0.2593	**0.0268**	**0.2514**	0.0335	0.4188	0.0289
	DTLZ4	2	0.5986	0.1857	**0.4461**	0.1850	0.4665	**0.1735**	0.4935	0.2243
		3	0.7885	0.1465	**0.5354**	0.1474	0.5682	**0.1320**	0.7680	0.1544
		5	0.7064	0.1731	**0.5487**	0.1021	0.6034	0.1127	0.7391	**0.0697**
	DTLZ5	2	0.1144	0.1211	0.0949	0.0495	**0.0889**	0.0506	0.3590	**0.0481**
		3	0.1114	0.0367	**0.0610**	0.0291	0.0615	**0.0283**	0.2823	0.0350
		5	0.0644	0.0447	**0.0498**	**0.0169**	0.0542	0.0254	0.1521	0.0319
	DTLZ6	2	**0.2826**	**0.3739**	1.8949	1.0420	2.6166	0.7696	4.6779	1.2463
		3	**1.2833**	**0.2710**	2.9273	0.4893	1.2966	0.4552	3.0290	0.9259
		5	**0.7897**	**0.2869**	2.5206	0.6990	1.6732	0.6577	2.9527	1.1470
	DTLZ7	2	**0.0081**	**0.0113**	0.0444	0.0254	0.0260	0.0382	0.5942	0.1295
		3	**0.0500**	**0.0261**	0.1635	0.1030	0.0853	0.0443	0.6159	0.0980
		5	**0.2821**	**0.0235**	0.5763	0.2356	0.4916	0.3096	0.7254	0.0781
Optimal-Random	DTLZ2	2	0.4220	0.2079	**0.0053**	**0.0020**	0.0090	0.0029	0.1244	0.1827
		3	0.3152	0.2285	**0.0517**	**0.0101**	0.0554	0.0120	0.2088	0.1247
		5	0.1619	0.0604	0.1582	0.0143	**0.1404**	0.0253	0.2758	**0.0078**
	DTLZ4	2	0.8335	0.8480	**0.0194**	**0.0160**	0.0526	0.0351	0.5851	0.4683
		3	0.7853	0.1831	**0.2662**	**0.0738**	0.2966	0.0857	0.5575	0.1704
		5	0.5789	0.1020	**0.4319**	0.1062	0.4730	0.0904	0.6047	**0.0801**
	DTLZ5	2	0.7489	0.4255	**0.0086**	**0.0024**	0.0094	0.0032	0.2086	0.2516
		3	0.3323	0.3085	**0.0064**	0.0018	0.0076	**0.0017**	0.1010	0.0845
		5	0.1890	0.2090	**0.0049**	**0.0019**	0.0055	0.0021	0.0251	0.0232
	DTLZ6	2	**0.0064**	0.0031	0.0077	**0.0013**	0.0081	0.0019	0.0147	0.0019
		3	0.0556	0.0868	**0.0075**	**0.0021**	0.0085	0.0029	0.0198	0.0104
		5	0.0396	0.0986	**0.0069**	0.0014	0.0085	**0.0012**	0.0171	0.0078
	DTLZ7	2	**0.0005**	0.0004	0.0013	**0.0003**	0.0020	0.0007	0.0177	0.0033
		3	0.0397	0.0093	**0.0365**	**0.0043**	0.0388	0.0058	0.1012	0.0124
		5	0.1910	0.0179	**0.1855**	**0.0141**	0.1825	0.0220	0.3404	0.0367

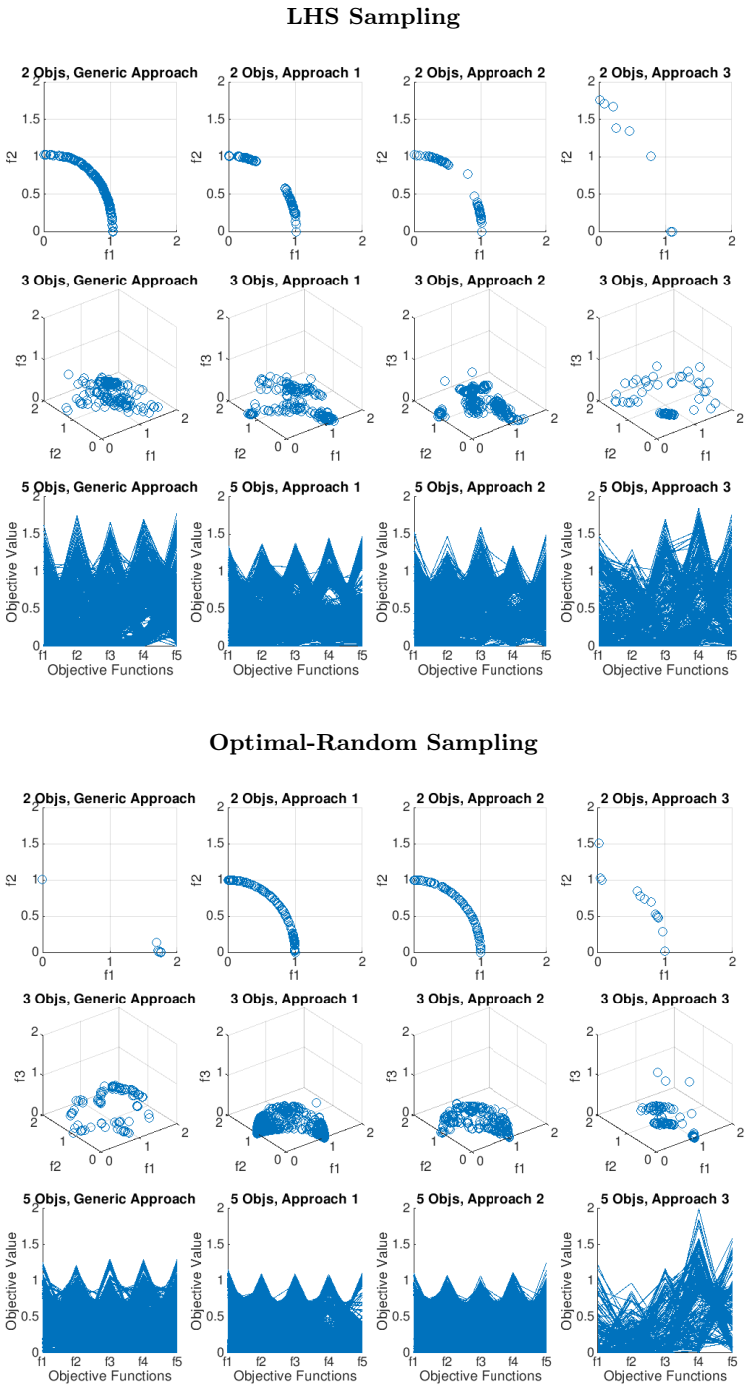


Fig. 4. Final solutions obtained of the run with the median IGD value using different approaches for LHS sampling (top three rows) and optimal-random sampling (bottom three rows) for the DTLZ2 problem.

5 Conclusions

We have considered offline data-driven optimization with evolutionary multiobjective optimization. We used Kriging to fit surrogate models to data and proposed and tested three approaches to utilize uncertainty information from Kriging models in the optimization. A comparison was done with several benchmark problems, sampling techniques and varying the number of objectives in solving offline data-driven multiobjective optimization problems. Adding uncertainty as one or more objectives showed improvements in the final solutions for certain problems in our benchmark testing. However, utilizing expected improvements as objectives (in Approach 3) did not seem to be effective in solving this kind of problems. The analysis also revealed that the solutions obtained in Approaches 1 and 2 are more accurate compared to the ones obtained using a generic approach (without uncertainty information).

Future work will include comparing the performance of the proposed approaches with bigger initial sample sizes, higher number of decision variables and higher number of objectives. Aiding the decision making process by giving a decision maker an option to select a final solution using the uncertainty information is another direction to work on. Moreover, filtering techniques can be applied to remove solutions with higher uncertainties. Testing on real-world data sets and exploring different ways to deal with uncertainties using other surrogate models will also be future research topics.

Acknowledgements

This research is related to the thematic research area Decision Analytics utilizing Causal Models and Multiobjective Optimization (DEMO) at the University of Jyvaskyla. This work was partially supported by the Natural Environment Research Council [NE/P017436/1].

References

1. Bezerra, L.C.T., López-Ibáñez, M., Stützle, T.: A large-scale experimental evaluation of high-performing multi- and many-objective evolutionary algorithms. Evolutionary Computation **26**(4), 621–656 (2018)
2. Blackwell, T., Branke, J.: Multiswarms, exclusion, and anti-convergence in dynamic environments. IEEE Transactions on Evolutionary Computation **10**(4), 459–472 (2006)
3. Castano, S., Antonellis, V.D.: Global viewing of heterogeneous data sources. IEEE Transactions on Knowledge and Data Engineering **13**(2), 277–297 (2001)
4. Chugh, T., Chakraborti, N., Sindhya, K., Jin, Y.: A data-driven surrogate-assisted evolutionary algorithm applied to a many-objective blast furnace optimization problem. Materials and Manufacturing Processes **32**(10), 1172–1178 (2017)
5. Chugh, T., Jin, Y., Miettinen, K., Hakanen, J., Sindhya, K.: A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. IEEE Transactions on Evolutionary Computation **22**(1), 129–142 (2018)

6. Chugh, T., Sindhya, K., Hakanen, J., Miettinen, K.: A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. Soft Computing (to appear, DOI: 10.1007/s00500-017-2965-0)
7. Coello, C., Lamont, G., Veldhuizen, D.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, New York, 2nd edn. (2007)
8. Forrester, A., Sobester, A., Keane, A.: Engineering Design via Surrogate Modelling. John Wiley & Sons (2008)
9. Jeong, S., Obayashi, S.: Efficient global optimization (EGO) for multi-objective problem and data mining. In: 2005 IEEE Congress on Evolutionary Computation. vol. 3, pp. 2138–2145 Vol. 3 (2005)
10. Jin, Y.: Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm and Evolutionary Computation **1**, 61–70 (2011)
11. Jin, Y., Wang, H., Chugh, T., Guo, D., Miettinen, K.: Data-driven evolutionary optimization: An overview and case studies. IEEE Transactions on Evolutionary Computation (to appear, DOI: 10.1109/TEVC.2018.2869001)
12. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. Journal of Global Optimization **13**(4), 455–492 (1998)
13. Knowles, J.: ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Transactions on Evolutionary Computation **10**(1), 50–66 (2006)
14. Li, K., Omidvar, M., Deb, K., Yao, X.: Variable interaction in multi-objective optimization problems. In: Handl, J., et al. (eds.) Parallel Problem Solving from Nature – PPSN XIV. pp. 399–409. Springer (2016)
15. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering **22**(10), 1345–1359 (2010)
16. Pilat, M., Neruda, R.: Aggregate meta-models for evolutionary multiobjective and many-objective optimization. Neurocomputing **116**, 392–402 (2013)
17. Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press (2005)
18. Regis, R.G.: Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions. IEEE Transactions on Evolutionary Computation **18**(3), 326–347 (2014)
19. Sun, X., Gong, D., Jin, Y., Chen, S.: A new surrogate-assisted interactive genetic algorithm with weighted semisupervised learning. IEEE Transactions on Cybernetics **43**(2), 685–698 (2013)
20. Wang, H., Jin, Y., Jansen, J.O.: Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system. IEEE Transactions on Evolutionary Computation **20**(6), 939–952 (2016)
21. Wang, H., Jin, Y., Sun, C., Doherty, J.: Offline data-driven evolutionary optimization using selective surrogate ensembles. IEEE Transactions on Evolutionary Computation (to appear, DOI: 10.1109/TEVC.2018.2834881)
22. Wang, H., Zhang, Q., Jiao, L., Yao, X.: Regularity model for noisy multiobjective optimization. IEEE Transactions on Cybernetics **46**(9), 1997–2009 (2016)
23. Wang, S., Minku, L.L., Yao, X.: Resampling-based ensemble methods for online class imbalance learning. IEEE Transactions on Knowledge and Data Engineering **27**(5), 1356–1368 (2015)
24. Zhang, Q., Liu, W., Tsang, E., Virginas, B.: Expensive multiobjective optimization by MOEA/D with Gaussian process model. IEEE Transactions on Evolutionary Computation **14**(3), 456–474 (2010)
25. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Parallel Problem Solving from Nature - PPSN VIII. pp. 832–842. Springer (2004)

PII

PROBABILISTIC SELECTION APPROACHES IN DECOMPOSITION-BASED EVOLUTIONARY ALGORITHMS FOR OFFLINE DATA-DRIVEN MULTIOBJECTIVE OPTIMIZATION

by

Atanu Mazumdar, Tinkle Chugh, Jussi Hakanen, and Kaisa Miettinen

Conditionally accepted in IEEE Transactions on Evolutionary Computation

Probabilistic Selection Approaches in Decomposition-based Evolutionary Algorithms for Offline Data-Driven Multiobjective Optimization

Atanu Mazumdar
University of Jyvaskyla
Faculty of Information Technology
P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla
Finland

Tinkle Chugh
Department of Computer Science
University of Exeter, UK

Jussi Hakanen
University of Jyvaskyla
Faculty of Information Technology
P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla
Finland

Kaisa Miettinen
University of Jyvaskyla
Faculty of Information Technology
P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla
Finland

***Abstract*—In offline data-driven multiobjective optimization, no new data is available during the optimization process. Approximation models, also known as surrogates, are built using the provided offline data. A multiobjective evolutionary algorithm can be utilized to find solutions by using these surrogates. The accuracy of the approximated solutions depends on the surrogates and approximations typically involve uncertainties. In this paper, we propose probabilistic selection approaches that utilize the uncertainty information of the Kriging models (as surrogates) to improve the solution process in offline data-driven multiobjective optimization. These approaches are designed for decomposition-based multiobjective evolutionary algorithms and can, thus, handle a large number of objectives. The proposed approaches were tested on distance-based visualizable test problems and the DTLZ suite. The proposed approaches produced solutions with a greater hypervolume, and a lower root mean squared error compared to generic approaches and a transfer learning approach that do not use uncertainty information.**

***Index Terms*—Kriging, Gaussian processes, Metamodelling, Surrogate, Kernel density estimation, Pareto optimality**

1. Introduction

Sometimes, real-world multiobjective optimization problems (MOPs) consisting of conflicting objectives do not have analytical functions or simulation models. Instead, the starting point for optimization is data obtained by, e.g. physical experiments, sensors or real-life processes. The available data can be used to build surrogates (also known as metamodels) that approximate the *underlying* objective functions involved in the phenomenon. Optimization can be performed using these surrogates by embedding them in multiobjective evolutionary algorithms (MOEAs) which have proven to be suitable in solving black-box optimization problems [1]. Data-driven optimization can be categorized as *online* or *offline*. In online data-driven optimization problems, getting more data e.g. by conducting further (expensive) function evaluations is possible, which enables updating the surrogates [2]. However, if no additional data can be acquired during the optimization process, it is known as offline data-driven optimization [3], which is the main focus of this paper.

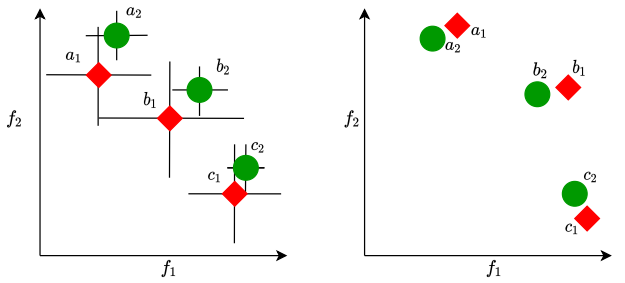


Figure 1: Individuals in the surrogate objective space (left) and underlying objective space (right) for a minimization problem.

Updating the surrogates is not possible in offline data-driven optimization [4]. Thus, while solving an offline MOP, the approximation accuracy and hypervolume of the solutions obtained are entirely dependent on the optimization algorithms used and the surrogates (that involve uncertainty).

In Figure 1, we show an illustration of the objective values and uncertainties of a few individuals approximated by the surrogates (e.g. Kriging). For simplicity, we call the objective space of the individuals evaluated with the surrogates and the underlying objective functions as *surrogate*

objective space and *underlying objective space*, respectively. The red individuals dominate the green in the surrogate objective space. However, as can be observed, these nondominated individuals also have higher uncertainties. On the right, we show the same individuals when they are evaluated with the underlying objective functions. It can be observed that the green individuals dominate most of the red individuals. Thus, while solving offline data-driven MOPs, utilizing just the surrogates' mean approximation can lead us to worse solutions. One of the ways to tackle this problem is by using the uncertainty from the surrogates.

Most of the previous works on offline data-driven optimization such as [2], [3], [5] do not consider uncertainty information provided by the surrogates.

In [6], optimization is assisted by coarse and fine surrogates. A coarse surrogate is used by the MOEA to find a promising subregion in the search space. Later, the knowledge about good solutions from the coarse search is transferred to the fine search. On the other hand, the works in [7], [8], [9] utilize the approximated uncertainty information provided by Kriging surrogates. The works in [7], [9] use probability of dominance that can be applied to dominance-based MOEAs. The approach in [8] utilizes the approximated uncertainties as additional objective function(s), thereby minimizing the objective values along with the uncertainties in the solutions. However, this approach increases the number of objectives and thus increases the complexity of the optimization problem.

A *generic* approach to solve offline MOPs using MOEA utilises the mean approximations of surrogates as objectives (without uncertainty). The MOEA finds a set of approximated nondominated solutions that represents the trade-offs between the objectives. However, the performance of traditional MOEAs such as MOGA [10], MO-CMA-ES [11], and NSGA-II [12], etc. deteriorates when the number of objectives increases [13], [14], [15]. Decomposition-based MOEAs (e.g. MOEA/D [15], NSGA-III [16] and RVEA [14]) have explicitly been developed to handle a large number of objectives (> 3).

In this paper, we propose a probabilistic selection approach and an extended version of it that incorporate uncertainty information in the solution process of offline data-driven MOPs. The proposed approaches utilize techniques such as Monte-Carlo sampling [17] and kernel density estimation (KDE) [18] to estimate the probability of selection criterion in decomposition-based MOEAs. This is particularly advantageous when the closed form of the probability is not available. The proposed approaches are novel in their adaptability or "plug and play" feature for any decomposition-based MOEAs without the requirement of further analytical derivations specific to the MOEA.As an example, we incorporate the proposed probabilistic selection approaches in RVEA and MOEA/D for solving offline MOPs.

The numerical experiments show that the first probabilistic selection approach produces solutions with a better accuracy compared to the generic approach. The second approach proposed is a hybrid selection approach that employs a combination of both the probabilistic and the generic selection approaches. The hybrid approach produces solutions better in hypervolume when compared to the parent approaches. To summarize, the main contributions are:

- Uncertainty information from the surrogates are utilized in the selection process of decomposition-based MOEAs.
- Easy adaptability to any decomposition-based MOEA without any need of analytical derivations.

As we use decomposition-based MOEAs, the proposed approaches are capable of handling a large number of objectives.

The rest of the paper is organized as follows. The basic notations, the background of the generic approach, decomposition-based MOEAs, and probabilistic selection are discussed in Section II. The proposed probabilistic and hybrid selection approaches are presented in Section III. The results of our experiments with analyses are compiled in Section IV. Finally, conclusions and future research perspectives are discussed in Section V.

2. Background

In offline data-driven MOPs, there exists no functional form or simulation model which can be accessed during the optimization process. The available (pre-collected) data is the output of a process or phenomenon. As mentioned in the introduction, we refer to the process generating the offline data as *underlying* objective functions. We consider the underlying MOPs of the following form:

$$\begin{aligned} &\text{minimize } \{f_1(\mathbf{x}), \ldots, f_K(\mathbf{x})\}, \\ &\text{subject to } \mathbf{x} \in \Omega, \end{aligned} \tag{1}$$

where $K \geq 2$ is the total number of objectives, and Ω is the feasible region of the decision space $\mathbb{R}^n$. For a feasible decision vector $\mathbf{x}$, the corresponding objective vector is $\mathbf{f}(\mathbf{x})$, that comprises of the underlying objective (function) values $(f_1(\mathbf{x}), \ldots, f_K(\mathbf{x}))$.

A solution $\mathbf{x}_1 \in \Omega$ dominates another solution $\mathbf{x}_2 \in \Omega$ if $f_k(\mathbf{x}_1) \leq f_k(\mathbf{x}_2)$ for all $k = 1, \ldots, K$ and $f_k(\mathbf{x}_1) < f_k(\mathbf{x}_2)$ for at least one $k = 1, \ldots, K$. A solution of an MOP is nondominated if it is not dominated by any other feasible solution. An MOEA typically produces solutions that are nondominated within the set of solutions it has found. The solutions of (1) that are nondominated in Ω are also called Pareto optimal solutions. In what follows, we refer to solutions of MOEAs as approximated Pareto optimal ones. The set of solutions in the objective space is called the Pareto front and the corresponding set of decision vectors is the Pareto set.

2.1. Generic Offline Data-Driven Multiobjective Optimization

The generic approach to solve offline MOPs is shown in Figure 2. As described in [3], [4], the solution process can

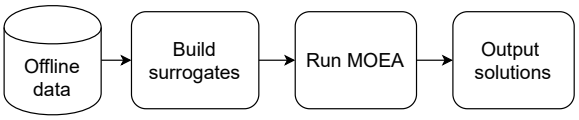


Figure 2: Flowchart of a generic offline data-driven multi-objective optimization approach.

be split into three stages which are (a) data collection, (b) formulating the MOP and surrogate building, and (c) optimization using an MOEA. The initial step of data collection may include pre-processing if necessary. Surrogates are then built using the provided offline data. Some of the popular surrogates that are used for solving offline data-driven MOPs are Kriging [19], neural networks [20], and support vector machines [20]. Finally, an MOEA is used to solve the MOP with the surrogates as objectives.

2.2. Decomposition-based MOEAs in Brief

Decomposition-based MOEAs are designed to solve MOPs with more than three objectives [21], [22]. In general, they decompose the problem into a number of single objective subproblems using scalarizing functions (e.g. MOEA/D [15]) or multiple MOPs (e.g. MOEA/D-M2M [23], NSGAIII [16] and RVEA [14]). In this paper, we use RVEA and MOEA/D as two decomposition-based algorithms and apply our proposed approaches in them. These algorithms start by creating a set of N uniformly distributed unit reference vectors (or weight vectors) $\mathbf{v_j}$ $(j = 1, \ldots, N)$. The population of individuals is P and the objective vectors for the individuals are $F = \{\mathbf{f}_1, \ldots, \mathbf{f}_{|P|}\}$ consisting of $|P|$ individuals. The i^{th} individual in P is denoted by I_i. The vector of minimum objective function values present in the given population is $\mathbf{z}^{min} = (z_1^{min}, \ldots, z_K^{min})$.

2.2.1. MOEA/D in brief. MOEA/D [15] performs search in the neighbourhood of each reference vector, it updates the population sequentially. In each neighbourhood, the offspring population is generated using crossover and mutation, which is then compared with the parent population. A selection criterion, e.g. PBI or Tchebycheff scalarizing function, is used to select the population for the next generation. In this article, we use PBI as the selection criterion and evaluate it for $\mathbf{x}$ as:

$$g_{\mathbf{x}}^{PBI} = d_1 + \rho d_2, \tag{2}$$

where parameter ρ is the penalty term that balances between convergence and diversity, $d_1 = ||(\mathbf{z}^{min} - \mathbf{f}(\mathbf{x}))^T \mathbf{v}_j||/||\mathbf{v}_j||$ and $d_2 = ||\mathbf{f}(\mathbf{x}) - (\mathbf{z}^{min} - d_1 \mathbf{v}_j)||$, respectively. Here $\mathbf{v}_j$ is the j^{th} reference vector in the neighbourhood. MOEA/D updates solutions in their neighbourhood by checking if $g_{\mathbf{x}'}^{PBI} \leq g_{\mathbf{x}_j}^{PBI}$, then set $\mathbf{x}_j = \mathbf{x}'$ and $\mathbf{f}(\mathbf{x}_j) = \mathbf{f}(\mathbf{x}')$. Here, $\mathbf{x}'$ is the offspring and $\mathbf{x}_j$ is the j^{th} solution in the neighbourhood.

2.2.2. RVEA in brief. The RVEA algorithm first translates the objective vectors as $\mathbf{f}_i' = \mathbf{f}_i - \mathbf{z}^{min}$, where $i = 1, \ldots, |P|$. It then splits the population into subpopulations by assigning individuals to reference vectors by measuring the cosine between the reference vector and the translated objective vector. The cosine value between the j^{th} reference vector $\mathbf{v}_j$ and the i^{th} translated objective vector $\mathbf{f}_i'$ is given by:

$$cos\, \theta_{i,j} = \frac{\mathbf{f}_i' \cdot \mathbf{v}_j}{\|\mathbf{f}_i'\|}, \tag{3}$$

where $\|\mathbf{f}_i'\|$ is the Euclidean norm. An individual I_i is included in the z^{th} subpopulation $\bar{P}_z$ if it has the lowest angle $\theta_{i,j}$ between $\mathbf{f}_i'$ and $\mathbf{v}_z$ (or highest $cos\, \theta_{i,j}$ value). The index of the z^{th} reference vector to which individual I_i is assigned is:

$$I_i | z = \underset{j \in \{1,\ldots N\}}{\operatorname{argmax}} \; cos\, \theta_{i,j}. \tag{4}$$

After the individuals are assigned to subpopulations, RVEA selects the z^{th} individual from each subpopulation, which has the minimum APD between the i^{th} individual and the j^{th} reference vector according to:

$$I_z | z = \underset{i \in \{1,\ldots,|\bar{P}_j|\}}{\operatorname{argmin}} \; d_{i,j}, \tag{5}$$

where APD (or $d_{i,j}$) is defined as,

$$d_{i,j} = (1 + P(\theta_{i,j})) \cdot ||\mathbf{f}_i'||. \tag{6}$$

Here $P(\theta_{i,j}) = K \cdot (t/t_{max})^{\alpha} \cdot \theta_{i,j}/\gamma_{\mathbf{v}_j}$ is the penalty function depending on $\theta_{i,j}$, and $\gamma_{\mathbf{v}_j} = \min_{i \in 1,\ldots,N, i \neq j} \langle \mathbf{v}_i, \mathbf{v}_j \rangle$, is the smallest angle between reference vector $\mathbf{v}_j$ and the other reference vectors. Here t is the generation counter, t_{max} is the maximum number of generations and α controls the rate of change of $P(\theta_{i,j})$. For more details, see [14].

2.3. Probabilistic Selection in Single Objective Optimization

We here provide an overview of probabilistic selection in single objective optimization, which is then further extended to solve offline data-driven MOPs. Let us consider a single objective offline data-driven minimization problem where the given data may have noise due to e.g., experimental or measurement error. The total uncertainty is due to the noise in the data (that can be estimated by Kriging surrogate) and the uncertainty in the approximation. Due to this, an individual with worse (greater) underlying objective values may be selected.

For example, in Figure 3, we have two individuals A and B with uncertain objective values. These two individuals have a normally distributed probability density function (PDF). The red star shows an example of a random sample, y, drawn from PDF_A. When a random sample is drawn from PDF_B, we may observe a smaller value than y thus making us select the individual with a worse objective value. The total probability of selecting the wrong individual B over A by observing a specific sample is the total area in the shaded region under PDF_B, or cumulative density function

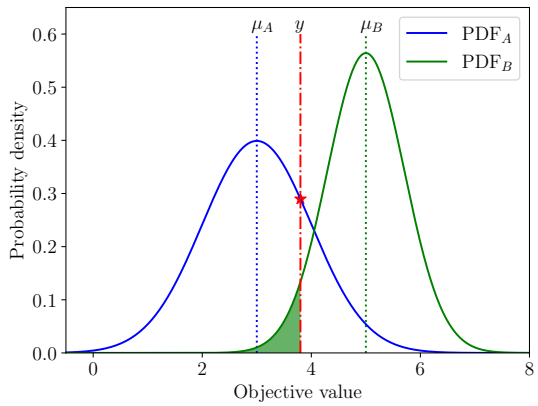


Figure 3: Probability of choosing the individual with worse underlying objective value in selection between two individuals with uncertain objective values (approximated by the surrogate) for a single objective minimization problem.

(CDF) of B (denoted by $\text{CDF}_B(y)$). The probability of drawing a random sample y is $\text{PDF}_A(y)$. Thus the total probability of a random sample drawn from PDF_B being smaller than y is $\text{PDF}_A(y) \cdot \text{CDF}_B(y)$.

The probability of wrongly choosing B over A when the underlying objective value of A is smaller than that of B according to [7] is:

$$P_{wrong}(A > B) = \int_{-\infty}^{\infty} \text{PDF}_A(A-y) \cdot \text{CDF}_B((A-y) > (B-y))\ dy, \quad (7)$$

We can replace CDF_B as an integral of PDF_B as:

$$P_{wrong}(A > B) = \int_{-\infty}^{\infty} (\text{PDF}_A(y) \cdot \int_{-\infty}^{y} \text{PDF}_B(\mu) d\mu)\ dy. \quad (8)$$

The work in [7] used the following equation for comparing a set of individuals with uncertain objective values and ranking them based on their probabilities:

$$R_i = \sum_{n=1}^{|P|} P_{wrong}(I_n > I_i) - 0.5, \quad (9)$$

where R_i is the ranking score given to the i^{th} individual I_i. The total number of individuals to be compared is $|P|$, and $P_{wrong}(I_n > I_i)$ is the probability of making a wrong decision in selection such that the fitness of I_i is smaller than the fitness of I_n. A value of 0.5 is subtracted from the ranking function as $P_{wrong}(I_i > I_i)$ is always 0.5. The individual with the best fitness value will have the smallest rank or has the smallest probability of making the wrong selection.

3. Probabilistic decomposition-based MOEAs

We utilize uncertainty in decomposition-based MOEAs and propose approaches to solve offline data-driven MOPs. Hence, the original selection process in decomposition-based MOEAs has to be modified to utilize the uncertainty approximated by the surrogates. This can be done by utilizing (7) and (8) to formulate a probabilistic selection criterion specific to the MOEA. The probability of the selection criterion can be computed analytically. However, this is quite complex in decomposition-based MOEAs, and the selection criterion has to be tailor-made for every variant of MOEA. Next, we describe our plug and play probabilistic approaches.

We summarize the generalized steps of the proposed probabilistic approach in Algorithm 1. We start with offline data of size N_D. Next, we build a Kriging surrogate for each underlying objective and initialize N uniformly distributed unit reference vectors [24].

For initializing the population we use the provided offline data set. We generate offspring using crossover and mutation in the neighbourhood (a fixed number in MOEA/D and the maximum number of reference vectors in RVEA). We then use Kriging models for approximating the objective values of the offspring. As the approximation distribution of Kriging surrogate is Gaussian, the multivariate PDF [19] for an individual I_i is:

$$\text{PDF}_{I_i} = \prod_{k=1}^{K} \frac{1}{\hat{\sigma}_{i,k}\sqrt{2\pi}} \exp\left(-\frac{(f_k - \hat{f}_{i,k})^2}{2\hat{\sigma}_{i,k}^2}\right), \quad (10)$$

where $\hat{f}_{i,k}$ is the approximated k^{th} objective function value for the i^{th} individual with $\hat{\sigma}_{i,k}$ as its standard deviation.

We draw S samples using Monte-Carlo sampling [17] from the distribution in (10) for the i^{th} individual. Individuals are then assigned to sub-populations in a probabilistic manner depending on the decomposition-based MOEA. The selection criterion is then calculated for all the generated samples of objective values. In the next step, we apply Kernel density estimation (KDE) [18] to approximate the distribution of the selection criterion depending on the decomposition-based MOEA. However, we may skip this step if the closed form distribution of the selection criterion is available (e.g. weighted sum and Tchebycheff as shown in the supplementary material). We then use these estimated PDFs to select individuals in a probabilistic way. The details of KDE are provided in the supplementary material. The reference vectors are then adapted after a certain number of generations (or function evaluations) to obtain a uniformly distributed set of solutions [14]. The stopping criterion is the maximum number of function evaluations performed with surrogates.

3.1. Probabilistic Selection in RVEA

The two major modifications to incorporate uncertainties in approximated objective values in RVEA are, a) the assignment of individuals to reference vectors and b) the selection of an individual using probabilistic APD (steps 10 and 11, respectively in Algorithm 1).

Algorithm 1: Probabilistic decomposition-based MOEA

Input: Offline data of size N_D; N = number of reference vectors; FE_{max} = maximum number of function evaluations using Kriging surrogates; S = number of samples to be used for estimating the distributions

Output: Approximated solutions

1 Build Kriging surrogates for each objective using the given offline data
2 Use the given data as the initial population; initialize the number of function evaluations $FE = 0$
3 Create a set of uniformly distributed unit reference vectors V_0 of size N
4 Find the neighbourhood for each unit reference vector
5 **while** $FE < FE_{max}$ **do**
6 Perform crossover and mutation on population and generate offspring
7 Evaluate the individuals using the Kriging surrogates and combine the parents and offspring
8 Update $FE = FE + |P_{offspring}|$
9 Draw S samples using Monte-Carlo from the distribution approximated by the surrogates
10 Perform algorithm specific sub-population assigning
11 Perform algorithm specific probabilistic selection
12 end

3.1.1. Probabilistic Assigning to Reference Vectors. As mentioned in Section II, assigning individuals to respective reference vectors in RVEA depends on the objective values as in (3). However, when the objective values (provided by the Kriging models) have uncertainties, assigning individuals can not be deterministic. Hence, in step 10 we perform a probabilistic assigning of individuals to reference vectors by using the distribution of the objective values provided by Kriging surrogates.

As explained previously, we draw S samples using Monte-Carlo sampling from the distribution in (10) for the i^{th} individual in the current population. The vector of samples is used to calculate $cos\,\theta_{i,j,l}$ for the i^{th} individual and j^{th} reference vector using (3), where $l = 1, \ldots, S$ is the sample number. These samples can then be used to estimate the PDF of $cos\,\theta_{i,j}$ using KDE. We use (9) for ranking the PDFs of $cos\,\theta_{i,j}$ that gives us the rank $R_{i,j}$. We use the following modification of (4) to include an individual to a subpopulation:

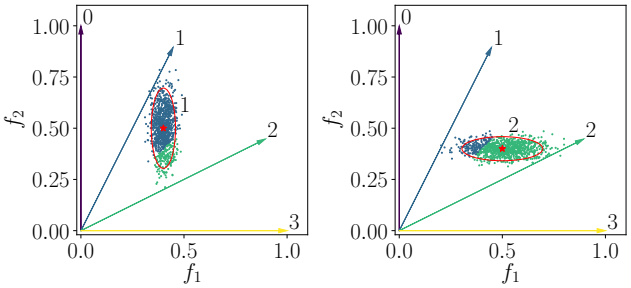
$$\bar{P}_z = \left\{ I_i | z = \underset{j \in \{1,\ldots,N\}}{\operatorname{argmax}} \; R_{i,j} \right\}, \tag{11}$$

where

$$R_{i,j} = \sum_{n=1}^{N} P_{wrong}(\, cos\,\theta_{i,n} > \; cos\,\theta_{i,j}) - 0.5 \tag{12}$$

and $\bar{P}_z$ is the z^{th} subpopulation and $R_{i,j}$ is the probabilistic rank of assigning an individual to a subpopulation.

Computing P_{wrong} in (8) is computationally expensive as it involves double integration. Hence, calculating $R_{i,j}$ in (12) for all the individuals becomes computationally expensive with a complexity of $O(N \cdot |P|)$. This is especially high when the numbers of individuals and reference vectors are large. To reduce the computation time, we calculate how many samples out of S for every individual are assigned to different reference vectors instead of performing numerical integration. By such a voting mechanism, the i^{th} individual is assigned to the reference vector to which most samples are assigned.



(a) (b)

Figure 4: Distribution of samples drawn for two individuals with different mean objective values (indicated by the red star) and standard deviations (indicated by ellipses). The colour code indicates the reference vector a sample drawn from the PDF of the individual is assigned to. The reference vector the individual is assigned to is indicated by the number.

We provide an illustration of the probabilistic assigning of individuals to reference vectors in Figure 4. It shows the samples drawn for two individuals in the objective space for the approximated distribution and the corresponding reference vectors they are assigned to. Samples assigned to a reference vector are colour coded. The red stars show the mean of the distributions of objective values for the individuals as approximated by the surrogate. The ellipses show the distributions of objective values for two standard deviations. It can be observed in sub-figure (a) that most of the samples are 'blue' or get assigned to reference vector '1'. Hence we can assign the individual to reference vector '1'. However, in sub-figure (b) when the distributions of objective values are changed, the individual gets assigned to reference vector '2'.

3.1.2. Probabilistic Angle Penalized Distance. After assigning individuals to reference vectors, we select one individual from every subpopulation in step 11 of Algorithm 1. Given a subpopulation and its associated individuals with

their PDFs, we utilize the samples for each individual and calculate their APD values using (6). These samples of APD values are used to estimate the distribution of APD for every individual, $\mathrm{PDF}_{d_{i,j}}$ using KDE. In this work, we used Gaussian kernel and adopted the Silvermann's rule [25] for selecting the bandwidth parameter that controls the smoothness of the estimated distribution. The estimated PDFs of APD are ranked by modifying (5) utilizing (9) as:

$$P_{nextgen} = \left\{ I_z | z = \underset{i \in \{1,\ldots,|\bar{P}_j|\}}{\operatorname{argmin}} R'_{i,j} \right\}, \tag{13}$$

where

$$R'_{i,j} = \sum_{n=1}^{|\bar{P}_j|} P_{wrong}(d_{n,j} > d_{i,j}) - 0.5. \tag{14}$$

The rank of the i^{th} individual in the subpopulation $\bar{P}_j$ is $R'_{i,j}$. The z^{th} individual I_z is selected from subpopulation $\bar{P}_j$, where $j = 1, \ldots, N$, for population of the next generation $P_{nextgen}$.

To find $R'_{i,j}$ in (14), a pairwise comparison between $\mathrm{PDF}_{d_{i,j}}$ has to be performed. Thus, the computation cost of performing the probabilistic selection is $O(|\bar{P}_j|^2)$, where $|\bar{P}_j|$ is the number of individuals in the j^{th} subpopulation. The overall computation cost becomes high due to the double integral involved while finding P_{wrong} between $\mathrm{PDF}_{d_{i,j}}$ of two individuals. Besides, there is an additional computation cost involved while performing the KDE of $\mathrm{PDF}_{d_{i,j}}$.

We propose an approach to compute P_{wrong} in an efficient way. As we know from (13), the calculation of the rank matrix $R'_{i,j}$ in (14) is a pairwise comparison. Thus P_{wrong} between an APD distribution with itself is always 0.5. Also, the computation needs to be done just once for the same pair. The following equation describes the alteration in the calculation of P_{wrong}:

$$P_{wrong}(d_{n,j} > d_{i,j}) = \begin{cases} 0.5 \text{ if } n = i, \\ 1 - P_{wrong}(d_{i,j} > d_{n,j}) \text{ if } n > i. \end{cases} \tag{15}$$

The double integration in (8) with the inner integral responsible for calculating the CDF from the approximated PDF contributes the most to the computation cost. Instead, we can use a coarse approximation of the CDF by computing the empirical CDF from the APD samples which would reduce the computation cost. To compute P_{wrong} in (8), the lower limit during integration can be changed to zero instead of $-\infty$. This is because APD can never attain a value below zero, and thus the PDF should be adjusted to estimate the probability density for APD values lower than zero. An illustration of the estimated PDF and empirical CDF calculated from the APD samples is shown in Figure 5(a). To further reduce the computation cost, P_{wrong} values for all the subpopulation individuals are computed in parallel. Applying all the proposed cost reduction approaches reduced the computation cost for computing the rank $R'_{i,j}$ for every generation.

3.2. Probabilistic Selection in MOEA/D

In MOEA/D, the assignment of solutions for each reference vectors is performed by defining the neighborhood of each vector [15]. Thus, step 10 in Algorithm 1 can be skipped. Next we demonstrate how to implement probabilistic selection with PBI as selection criterion (in step 11 of Algorithm 1).

Similar to the probabilistic selection in RVEA, we first draw S samples from the approximated distribution of objectives for the individuals in the neighbourhood and the offspring. The vector of sampled objective values is used to calculate PBI samples for the j^{th} individual in the neighbourhood and the offspring that we call as $g^{PBI}_{\mathbf{x}_{j,l}}$ and $g^{PBI}_{\mathbf{x}'_l}$, respectively, where $l = 1, \ldots, S$. Next, we approximate the PDF of $g^{PBI}_{\mathbf{x}_j}$ and $g^{PBI}_{\mathbf{x}'}$ using KDE. Again, similar to the probabilistic RVEA, we use Gaussian kernel and Silvermann's rule to select the bandwidth parameter. Since we are comparing PDFs of PBI, we need to modify the update operation in generic MOEA/D and calculate the P_{wrong} utilizing (7). We update the solutions in the neighbourhood by checking if $P_{wrong}(g^{PBI}_{\mathbf{x}'} \leq g^{PBI}_{\mathbf{x}_j}) < 0.5$, then set $\mathbf{x}_j = \mathbf{x}'$ and $\mathbf{f}(\mathbf{x}_j) = \mathbf{f}(\mathbf{x}')$.

It has to be noted that the comparison of PDFs of PBI is one to many, whereas for PDFs of APD its pairwise. In Figure 5(b), we show the estimated PDF of PBI samples for one of the individuals and the empirical CDF.

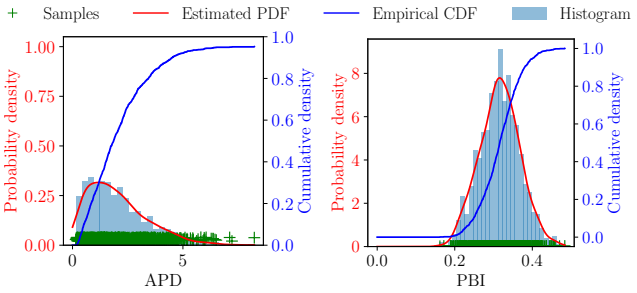


Figure 5: Samples drawn, histogram, estimated PDF and empirical CDF of the selection criterion for (a) RVEA (APD) and (b) MOEA/D (PBI) for one individual.

We illustrate the potential of the proposed probabilistic selection approach for a bi-objective minimization problem in Figure 6. It shows different individuals with uncertain objective values, with error bars representing the 95% confidence interval of the approximated distributions. The numbers represent the reference vector / sub-population the individuals are assigned to. Green individuals are the ones selected from a sub-population. It can be observed that for the individuals assigned to reference vector '2', the original selection criterion (generic approach) selects the individual that is better in objective values even though it has a much higher uncertainty. On the other hand, the probabilistic approach selects the individuals that are worse in terms of objective values but have comparatively lower uncertainties.

For the individuals assigned to reference vector '0', both the individuals have the same approximated mean values. However, the probabilistic approach selects the one with lower uncertainties. For individuals assigned to reference vector '3' or in its neighbourhood, both the individuals have the same uncertainties with slightly different objective values. One can see that both the generic and the probabilistic approach select the same individual with a better value of the selection criterion.

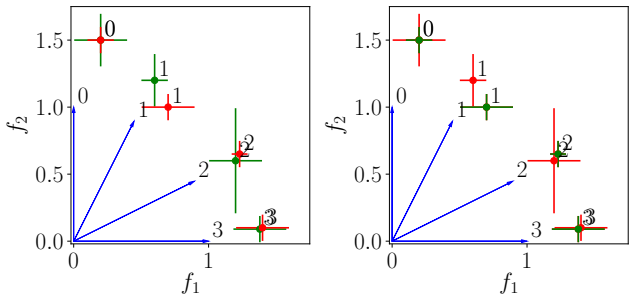


Figure 6: Selection of individuals using (a) generic approach and (b) probabilistic approach for a minimization problem. Error bars show the 95% confidence interval of the distribution of objective values approximated by the surrogates. Colours 'green' and 'red' show the individuals that are selected and not selected by the approaches, respectively.

3.3. Hybrid of Probabilistic and Generic Approaches

The proposed probabilistic approach tends to select individuals with better objective values and lower uncertainties (or high approximation accuracy). On the other hand, the generic approach, without incorporating any uncertainty, produces solutions with better hypervolume. Hence, a hybrid of the two approaches can provide the benefits of both and produce a set of solutions with a wider range of uncertainties and objective values. This is especially advantageous for decision making due to the wider choices it provides [26].

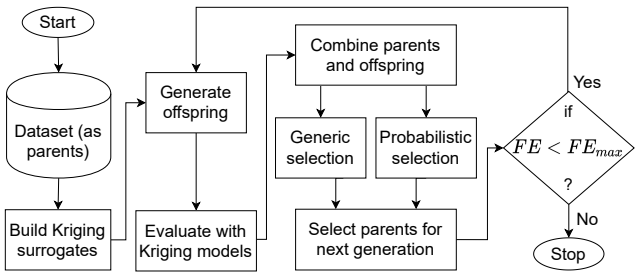


Figure 7: Flowchart for the hybrid approach.

A flowchart of the hybrid approach is shown in Figure 7. In the proposed hybrid approach, we select individuals based on both the criteria. An equal proportion of solutions from both the generic and probabilistic selection criteria are selected. This choice also helped in avoiding the need of introducing extra parameters. The redundant copies of the individuals (that were selected by both the approaches) are removed and the entire selected population is used for further crossover and mutation steps. It has to be noted that all the solutions obtained by the generic and the probabilistic approach (that are equal in number) are used, and no further parameters are required.

4. Experimental Results

In this section, we demonstrate the potential of the proposed approaches embedded in RVEA and MOEA/D by solving distance-based multiobjective visualizable test problems (DBMOPP) [27] and DTLZ [28] test problems with different numbers of objectives. DBMOPP problems have certain advantages over traditional benchmark suites. First, we can simultaneously visualize the solutions in both decision and objective spaces. Second, we can visualize the search behaviour with the number of function evaluations. Third, these problems do not have a limitation of having the same values of many decision variables on the Pareto set as in the DTLZ problems.

Experimental setup

- Benchmark problems: Two sets of DBMOPP problems denoted as P1 and P2 (details are in the supplementary material) utilizing the code in [29]. Test results with DTLZ suite are provided in the supplementary material.
- Number of objectives (K): 2-10
- Number of decision variables (n): 10
- Termination (FE_{max}): 40000 function evaluations with surrogates
- Kriging parameters: Scikit-learn python library [30] for Kriging with Gaussian kernel and BFGS [31] to maximize the marginal likelihood.
- Approach specific parameters: Number of samples for Monte-Carlo sampling $S = 1000$. For KDE we used Gaussian kernel with a bandwidth parameter set by Silvermann's rule (details are provided in the supplementary material).
- Other algorithms: We compared the proposed probabilistic approaches for RVEA and MOEA/D with three generic approaches that use the approximated values (posterior mean in this case) denoted as generic (Gen-RVEA and Gen-MOEA/D) and transfer learning (TL) approach [6] and the initial samples (Init). The probabilistic approaches for RVEA and MOEA are denoted by Prob-RVEA and Prob-MOEA/D, respectively. The hybrid approaches for RVEA and MOEA/D are denoted by Hyb-RVEA and Hyb-MOEA/D, respectively.
- MOEA parameter settings: The reference vectors were generated by simplex lattice design [24]. The number of reference vectors was varied with the

number of objectives as in [24]. All the approaches except TL used reference vector adaptation after every 10th generation as in [14] for all test instances. Other settings such as crossover, mutation, and parameters of the MOEAs are provided in the supplementary material.

- Number of independent runs for each instance: 31
- Performance metrics: We used hypervolume and RMSE as metrics to measure the performance of the tested approaches. Further details regarding metrics and reference points are provide in the supplementary material.
- Size of the initial data set (N_D): 109
- Sampling of the initial data set: Latin hypercube sampling (LHS) and multivariate normal sampling (MVNS). In MVNS sampling, the objectives were considered to be independent with mean at the midpoint of the decision space (0 for P1 and P2). The variance of the sampling distribution was set to 0.1 for all the objectives. Similarly, the mean of the distribution for DTLZ instances was set to 0.5 with variance of 0.1 for all objectives.

All the approaches were implemented in Python utilizing the DESDEO framework (desdeo.it.jyu.fi) [1]. For the TL approach, we used the Matlab implementation in [6].

It should be noted that we evaluated the approximated solutions obtained with different approaches with underlying objective function values to calculate hypervolume and RMSE. This is only for experimentation and such evaluations may not be possible while solving real-life offline MOPs. We refer to the hypervolume of the solutions evaluated with the surrogates as *surrogate hypervolume* for simplicity.

A pairwise Wilcoxon significance test [32] was conducted between the different approaches and the calculated p-values were later Bonferroni corrected. The threshold $\alpha = 0.05$ was considered for rejecting the null hypothesis. The overall ranking of the approaches was done by a scoring system where the approach considered as the alternate hypothesis is given a score +1 when it is significantly better than the null hypothesis. A score of zero is given if the alternate approach is not significantly better or worse than the null hypothesis. If the alternate approach is significantly worse than the null hypothesis, it gets a score -1. The sum of the scores of the hypothesis testings is used for ranking all the approaches in a descending order of the score.

We show the median hypervolumes and RMSEs (along with the standard deviations in different runs) for a few DBMOPP test instances in Table 1. The items in bold represent the best performing approaches. We also show the heatmap for all the tested DBMOPP instances comparing hypervolume and RMSE in Figure 8. The rankings of the approaches are colour-coded from best (yellow) to worst (purple) using the 'viridis' colourmap. It can be observed

1. Python source code can be accessed from https://github.com/industrial-optimization-group/offline_data_driven_moea

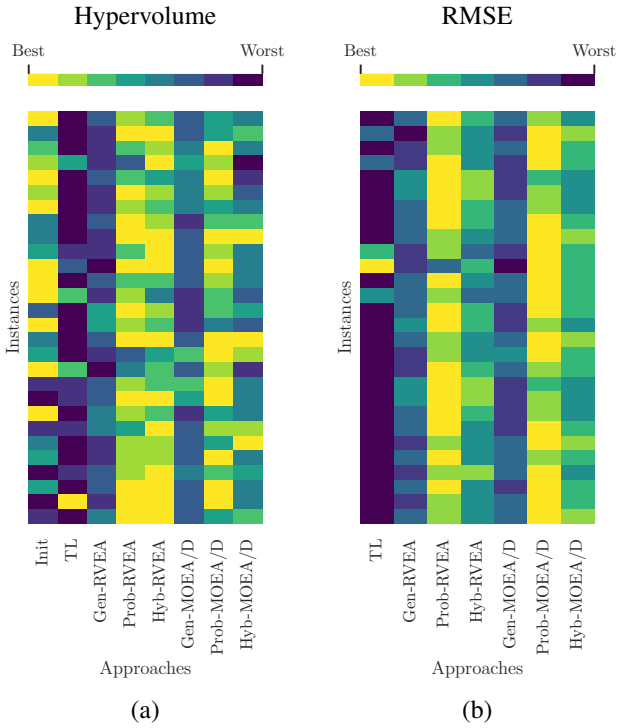


Figure 8: Heatmaps of: (a) hypervolume and (b) RMSE of solutions obtained by initial sampling, transfer learning and the generic, probabilistic and hybrid approaches for RVEA and MOEA/D respectively for DBMOPP problem instances.

that most of the yellow colours are around the Prob-RVEA, Hyb-RVEA and Prob-MOEA/D approaches.

As we can observe, Hyb-RVEA and Prob-RVEA performed best in terms of hypervolume followed by Prob-MOEA/D. In terms of RMSE, Prob-MOEA/D performed the best with Prob-RVEA coming second. For certain test instances, the generic approaches and TL produced worse solutions (both in hypervolume and RMSE) compared to the initial sampling. This is because they do not consider uncertainty in approximations when selecting solutions and thus converged far from the Pareto front. Detailed results on other DBMOPP and DTLZ instances are provided in the supplementary material.

Overall, the probabilistic approaches outperformed their generic counterparts, TL and initial sampling in both hypervolume and RMSE. However, we found that Hyb-MOEA/D did not perform better than Prob-MOEA/D. This is because of the extremely poor performance of Gen-MOEA/D in terms of both hypervolume and RMSE.

We also show the progress or search behaviour of different approaches of the runs with median hypervolume values in Figure 9. The first column shows the problem instance and the initial samples (red '+' showing the non-dominated ones). The problem instance in the top row has five objectives, and in the bottom row eight objectives with two disconnected Pareto sets. The black circles with centers

TABLE 1: Hypervolume (HV) and RMSE for a few DBMOPP test instances.

Sampling	Problem	K	Metric	Init.	TL	Gen-RVEA	Prob-RVEA	Hyb-RVEA	Gen-MOEA/D	Prob-MOEA/D	Hyb-MOEA/D
LHS	P1	8	HV	**9.09E+05** (2.65E+04)	5.22E+05 (1.65E+05)	5.98E+05 (1.00E+05)	7.38E+05 (1.01E+05)	6.88E+05 (9.63E+04)	6.17E+05 (9.67E+04)	6.52E+05 (9.70E+04)	6.18E+05 (8.83E+04)
			RMSE	-	1.76E+00 (3.92E-01)	1.11E+00 (4.01E-01)	**1.59E+00** (3.69E-01)	1.55E+00 (3.87E-01)	1.73E+00 (4.16E-01)	1.53E+00 (3.93E-01)	1.61E+00 (3.84E-01)
	P2	8	HV	7.33E+05 (7.11E+04)	3.86E+05 (1.24E+05)	7.77E+05 (5.27E+04)	**8.34E+05** (3.88E+04)	**8.16E+05** (4.24E+04)	7.24E+05 (3.33E+04)	7.78E+05 (3.90E+04)	7.73E+05 (4.01E+04)
			RMSE	-	1.56E+00 (2.76E-01)	1.34E+00 (3.48E-01)	**1.48E+00** (2.96E-01)	1.39E+00 (2.23E-01)	1.67E+00 (2.81E-01)	**1.34E+00** (3.21E-01)	1.46E+00 (3.23E-01)
MVNS	P1	6	HV	6.13E+03 (4.31E+02)	7.22E+03 (1.82E-12)	7.73E+03 (3.35E+02)	**8.62E+03** (3.23E+02)	**8.52E+03** (3.24E+02)	7.78E+03 (4.50E+02)	**8.71E+03** (3.76E+02)	7.78E+03 (2.53E+02)
			RMSE	-	1.85E+00 (1.31E-01)	1.77E+00 (1.57E-01)	**1.81E+00** (1.19E-01)	1.80E+00 (1.53E-01)	1.89E+00 (1.19E-01)	1.79E+00 (1.30E-01)	1.85E+00 (1.22E-01)
	P2	10	HV	4.52E+07 (2.92E+06)	3.37E+07 (1.10E+07)	8.05E+07 (8.61E+06)	8.62E+07 (9.34E+06)	**9.14E+07** (4.53E+06)	6.96E+07 (7.29E+06)	8.85E+07 (2.74E+06)	8.70E+07 (4.60E+06)
			RMSE	-	1.11E+00 (3.50E-01)	5.00E-01 (3.59E-01)	**9.16E-01** (3.22E-01)	9.43E-01 (3.64E-01)	1.29E+00 (3.48E-01)	**4.50E-01** (3.00E-01)	8.81E-01 (3.35E-01)

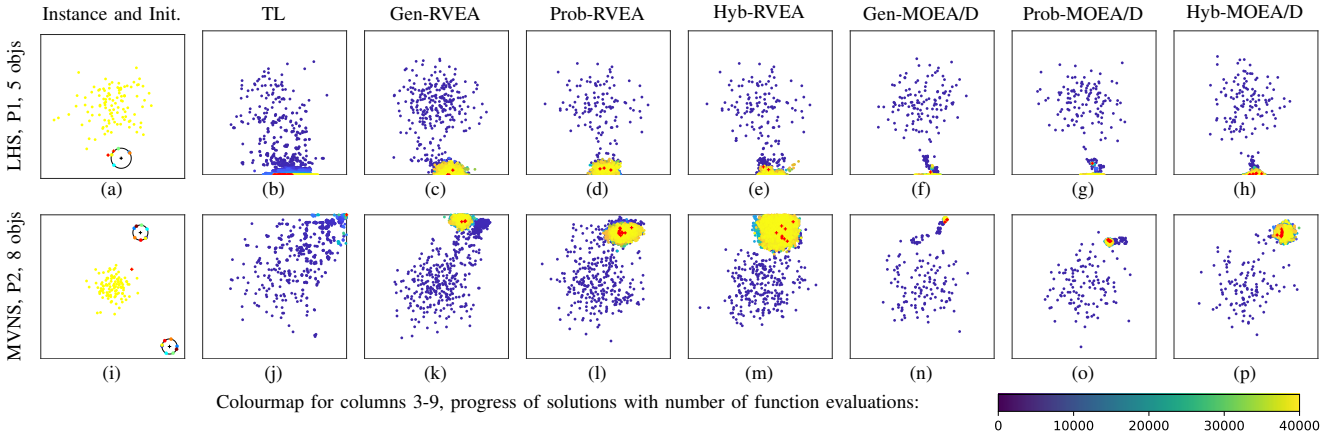
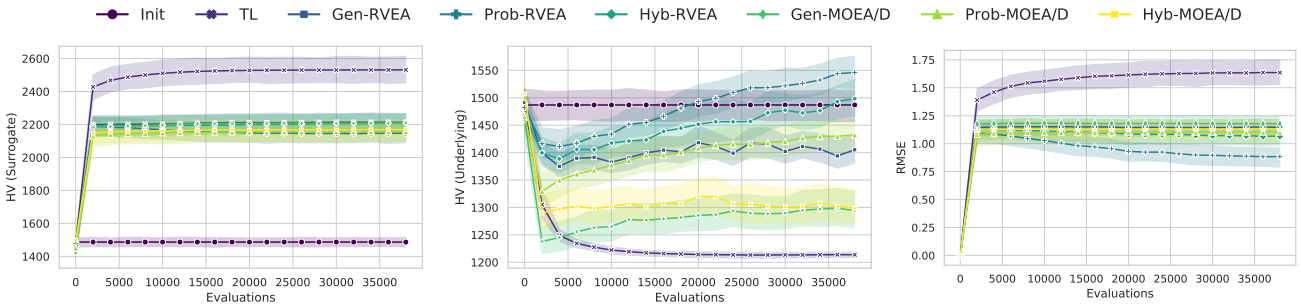


Figure 9: Progress of the solution process with median hypervolume values. DBMOPP P1 with $K = 5$ with LHS and P2 with $K = 8$ with MVNS. The colour code represents the number of function evaluations during the solution process.

denoted by '+' are the Pareto set and dots on the circle with different colors represent the points to which the objectives or distances are minimized. In other words, if solutions are on or in the circle, they are on the Pareto front. Therefore, with these visualizations, we can see how close the solutions from an approach can get to the Pareto front. The next seven columns show the progress of the solutions by evaluating with the underlying objectives for the different approaches tested. The color of the solutions represent the function evaluations count, and the non-dominated solutions of the last generations are shown in red '+'. For visualizing the solution of the tested DBMOPP problem instances, a 10-dimensional decision space is projected to 2-dimensional.

It can be observed that Prob-RVEA in (d) and (l), Hyb-RVEA in (e) and (m), Prob-MOEA/D in (g) and (o) and Hyb-MOEA/D in (h) and (p) converged much closer to the Pareto front. However, the solutions for the Gen-RVEA in (c) and (k), Gen-MOEA/D in (f) and (n), and TL in (b) and (j) converged further away from the Pareto front. One can also see that the non-dominated solutions in the initial sampling in (a) and (i) are much closer to the Pareto front compared to the generic approaches and TL. We also observed that all approaches failed to get solutions closer to both disconnected Pareto sets in the bottom row. This is because of the lack of adequate offline data for solving such MOPs.

In Figure 10, we show the surrogate hypervolume, hypervolume in the underlying objective space and RMSE after every 1000 function evaluations. As can be seen, the probabilistic and hybrid approaches improved the hypervolume in the underlying objective space with function evaluations. However, the generic approaches and TL could not further improve the hypervolume. On the contrary, the surrogate hypervolume for TL improved with function evaluations and the probabilistic approaches it gradually decreased. This is because the probabilistic approaches reject solutions with better objective values if they have high uncertainties. The RMSE increased for all the approaches in the initial function evaluations. However, in the later function evaluations, it gradually reduced for the probabilistic approaches (espe-



(a) (b) (c)

Figure 10: Progress of the solution process with median hypervolume values. DBMOPP P1 with $K = 5$ and LHS showing the variation of hypervolume (HV) in surrogate and underlying objective spaces, and RMSE with function evaluations.

cially for Prob-RVEA). The final RMSE in TL was the worst.

The performances of the generic approaches and TL were poor because they did not consider uncertainty in the solutions during the optimization process. Hence, the solutions, when evaluated with the underlying objectives, produced worse objective values compared to what was observed in the surrogate objective space. TL performed the worst because the sparse search adopted in the approach led to faster convergence in the surrogate objective space but far from the Pareto front.

5. Conclusions

We have proposed two probabilistic selection approaches for solving offline data-driven MOPs. These approaches are designed for decomposition-based MOEAs and utilize uncertainty in the approximations of Kriging surrogates. The first approach estimates the distribution of selection criterion embedding in a decomposition-based MOEA. The second one is a hybrid of the first approach and a generic approach (that does not use any uncertainty in the approximations). We demonstrated the potential of the proposed approaches with RVEA and MOEA/D, that also showed its adaptability. For benchmarking, we used several distance-based multi-objective visualizable test problems and DTLZ problems. We used different sampling techniques and numbers of objectives in our tests.

Considering the accuracy of the solutions is often neglected in offline data-driven optimization. The proposed probabilistic approaches are more focused on improving the accuracy of the solutions. A detailed analysis of quality measures (hypervolume and RMSE), visualization of the solution process and comparison with existing approaches clearly showed the advantages of the proposed approaches. For the problem instances tested, we can conclude that depending on the indicators used, MOEAs and problem characteristics, both probabilistic and hybrid approaches have their strengths. However, if more accurate solutions are required, the probabilistic approach should be preferred.

In future, we plan to test the proposed approaches in other decomposition-based MOEAs with different normalization techniques. We will also test on real-world problems and develop approaches to handle constraints.

Acknowledgements

This research was partly supported by the Academy of Finland (grant number 311877) and is related to the thematic research area DEMO (Decision Analytics utilizing Causal Models and Multiobjective Optimization, www.jyu.fi/demo) of the University of Jyväskylä. We also acknowledge Prof. Y. Jin and C. Yang for providing the transfer learning codes.

References

[1] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evol. Comput.*, vol. 1, pp. 61–70, 2011.

[2] H. Wang, Y. Jin, and J. O. Jansen, "Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system," *IEEE Trans. Evol. Comput.*, vol. 20, pp. 939–952, 2016.

[3] H. Wang, Y. Jin, C. Sun, and J. Doherty, "Offline data-driven evolutionary optimization using selective surrogate ensembles," *IEEE Trans. Evol. Comput.*, vol. 23, pp. 203–216, 2019.

[4] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Trans. Evol. Comput.*, vol. 23, pp. 442–458, 2019.

[5] H. Wang and Y. Jin, "A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems," *IEEE Trans. Cyber.*, vol. 50, pp. 536–549, 2020.

[6] C. Yang, J. Ding, Y. Jin, and T. Chai, "Offline data-driven multi-objective optimization: Knowledge transfer between surrogates and generation of final solutions," *IEEE Trans. Evol. Comput.*, vol. 24, pp. 409–423, 2020.

[7] E. J. Hughes, "Evolutionary multi-objective ranking with uncertainty and noise," in *Evolutionary Multi-Criterion Optimization, Proc.*, E. Zitzler, L. Thiele, K. Deb, C. A. Coello Coello, and D. Corne, Eds. Springer, 2001, pp. 329–343.

[8] A. Mazumdar, T. Chugh, K. Miettinen, and M. López-Ibáñez, "On dealing with uncertainties from Kriging models in offline data-driven evolutionary multiobjective optimization," in *Evolutionary Multi-Criterion Optimization, Proc.*, K. Deb, E. Goodman, C. A. Coello Coello, K. Klamroth, K. Miettinen, S. Mostaghim, and P. Reed, Eds. Springer, 2019, pp. 463–474.

[9] A. A. M. Rahat, C. Wang, R. M. Everson, and J. E. Fieldsend, "Data-driven multi-objective optimisation of coal-fired boiler combustion systems," *Applied Energy*, vol. 229, pp. 446–458, 2018.

[10] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation discussion and generalization," in *Proceedings of the 5th International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., 1993, pp. 416–423.

[11] T. Voß, N. Hansen, and C. Igel, "Recombination for learning strategy parameters in the MO-CMA-ES," in *Evolutionary Multi-Criterion Optimization, Proc.* Springer, 2009, pp. 155–168.

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, pp. 182–197, 2002.

[13] L. C. T. Bezerra, M. López-Ibáñez, and T. Stützle, "A large-scale experimental evaluation of high-performing multi- and many-objective evolutionary algorithms," *Evol. Comput.*, vol. 26, pp. 621–656, 2018.

[14] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, pp. 773–791, 2016.

[15] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, pp. 712–731, 2007.

[16] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, pp. 577–601, 2014.

[17] N. Metropolis, "The beginning of the Monte Carlo method," *Los Alamos Science*, pp. 125—130, 1987.

[18] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *Ann. Math. Stat.*, vol. 27, pp. 832–837, 1956.

[19] A. Forrester, A. Sobester, and A. Keane, *Engineering Design via Surrogate Modelling*. John Wiley & Sons, 2008.

[20] C. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

[21] K. Li, "Decomposition multi-objective evolutionary optimization: From state-of-the-art to future opportunities," 2021.

[22] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 21, pp. 440–462, 2017.

[23] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, pp. 450–455, 2014.

[24] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using gaussian process-based inverse modeling," *IEEE Trans. Evol. Comput.*, vol. 19, pp. 838–856, 2015.

[25] B. Silverman, *Density Estimation for Statistics and Data Analysis*. London: Chapman Hall/CRC, 1986.

[26] A. Mazumdar, T. Chugh, J. Hakanen, and K. Miettinen, "An interactive framework for offline data-driven multiobjective optimization," in *Bioinspired Optimization Methods and Their Applications, Proc.*, B. Filipič, E. Minisci, and M. Vasile, Eds. Springer, 2020, pp. 97–109.

[27] J. E. Fieldsend, T. Chugh, R. Allmendinger, and K. Miettinen, "A feature rich distance-based many-objective visualisable test problem generator," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2019, pp. 541–549.

[28] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, A. Abraham, L. Jain, and R. Goldberg, Eds. Springer, 2005, pp. 105–145.

[29] J. E. Fieldsend, T. Chugh, R. Allmendinger, and K. Miettinen, "fieldsend/DBMOPP_generator," 2019, accessed June 15, 2021. [Online]. Available: "https://github.com/fieldsend/DBMOPP_generator"

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[31] M. Avriel, *Nonlinear Programming: Analysis and Methods*. Dover Publishing, 2003.

[32] W. Frank, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, pp. 80–83, 1945.

PIII

AN INTERACTIVE FRAMEWORK FOR OFFLINE DATA-DRIVEN MULTIOBJECTIVE OPTIMIZATION

by

Atanu Mazumdar, Tinkle Chugh, Jussi Hakanen, and Kaisa Miettinen pages 97–109. Springer, 2020

In Bogdan Filipi˘c, Edmondo Minisci, and Massimiliano Vasile, editors, *Proceedings on Bioinspired Optimization Methods and Their Applications*

An Interactive Framework for Offline Data-Driven Multiobjective Optimization

Atanu Mazumdar[1], Tinkle Chugh[2], Jussi Hakanen[1], and Kaisa Miettinen[1]

[1] University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland
[2] Department of Computer Science, University of Exeter, UK

Abstract. We propose a framework for solving offline data-driven multiobjective optimization problems in an interactive manner. No new data becomes available when solving offline problems. We fit surrogate models to the data to enable optimization, which introduces uncertainty. The framework incorporates preference information from a decision maker in two aspects to direct the solution process. Firstly, the decision maker can guide the optimization by providing preferences for objectives. Secondly, the framework features a novel technique for the decision maker to also express preferences related to maximum acceptable uncertainty in the solutions as preferred ranges of uncertainty. In this way, the decision maker can understand what uncertainty in solutions means and utilize this information for better decision making. We aim at keeping the cognitive load on the decision maker low and propose an interactive visualization that enables the decision maker to make decisions based on uncertainty. The interactive framework utilizes decomposition-based multiobjective evolutionary algorithms and can be extended to handle different types of preferences for objectives. Finally, we demonstrate the framework by solving a practical optimization problem with ten objectives.

Keywords: Decision support · Decision making · Decomposition-based MOEA · Metamodelling · Surrogate · Kriging · Gaussian processes.

1 Introduction

Sometimes while solving data-driven multiobjective optimization problems (or MOPs) additional data can not be acquired during the solution process. Instead, we may have pre-collected data of the phenomenon of interest that was obtained beforehand, e.g. by conducting physical experiments. This type of optimization problems are termed as *offline* data-driven MOPs [3, 8, 17]. For formulating the optimization problem, we can build surrogate models using the given data to approximate the behaviour of the phenomenon. Optimization can then be performed utilizing these surrogates as objective functions e.g. by a multiobjective evolutionary algorithm (MOEA). However, approximation error in the surrogates' prediction can not be avoided. Certain surrogate models such as Kriging also provide information about the uncertainty (e.g. as standard deviation) in

predictions. This uncertainty information can be utilized in the optimization process to improve the quality of the solutions [11].

Previous works on offline multiobjective optimization such as [3,8,11,17] approximate the entire Pareto front. This makes decision making a difficult task as the decision maker (DM) has to choose from a large set of solutions. Interactive multiobjective optimization approaches allow the DM to find solutions in an interesting region of the Pareto front and learn about the problem and the feasibility of one's preferences and adjust the latter. They also provide limited amount of information at a time thereby reducing the cognitive load (see [13] for more information). There have been many developments in interactive MOEAs [14] and decomposition based MOEAs have become quite popular because of their capability of solving MOPs with a large number of objectives [2, 4, 20]. Hence, interactive approaches such as [7, 10, 21] have been proposed for decomposition-based MOEAs. However, as far as we know, addressing DM's preferences while solving offline MOPs in decomposition-based MOEAs has not been considered.

Utilizing the uncertainty information in interactive optimization may be quite valuable to the DM for a better understanding of the solutions and better decision making while solving offline MOPs. The major challenge in utilizing uncertainty in an interactive optimization process is conveying this extra information to the DM as (s)he may not be familiar with it.

In this paper, we propose a framework for solving offline data-driven MOPs interactively using decomposition-based MOEAs. It enables the DM to understand and make decisions based on the uncertainties present in the approximated solutions. The framework does not increase the cognitive load of the DM significantly while providing preference information for uncertainties along with the preferences for objectives.

2 Background

We consider the underlying MOP that has to be solved of the following form:

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

where $K \geq 2$ is the number of objectives and S is the feasible region in the decision space $\mathbb{R}^n$. For a feasible decision vector $\mathbf{x}$, the corresponding objective vector $\mathbf{f}(\mathbf{x})$ comprises of the underlying objective (function) values $(f_1(\mathbf{x}), \ldots, f_K(\mathbf{x}))$.

A solution $\mathbf{x}^1 \in S$ dominates another solution $\mathbf{x}^2 \in S$ if $f_k(\mathbf{x}^1) \leq f_k(\mathbf{x}^2)$ for all $k = 1, \ldots, K$ and $f_k(\mathbf{x}^1) < f_k(\mathbf{x}^2)$ for at least one $k = 1, \ldots, K$. If a solution of an MOP is not dominated by any other feasible solutions, it is called nondominated. Solving an MOP using an MOEA typically produces solutions that are nondominated within the set of solutions it has found. The solutions of Eq. (1) that are nondominated in S are also called Pareto optimal solutions. Next, we discuss a generic approach to solve an offline data-driven MOP.

2.1 Generic Approach for Offline Data-Driven Multiobjective Optimization

A generic way for offline data-driven optimization using an MOEA described in [8, 18] is shown in Fig. 1. The solution process can be divided into three parts: a) data collection, b) formulating the MOP and building surrogate models, and c) running an MOEA. The first step involves performing experiments to acquire the data and pre-processing it if necessary. Next, surrogate models are built to approximate the behaviour of the underlying objective functions using the provided data. The prediction vector of the fitted surrogate models can be represented as $\hat{\mathbf{f}}(\mathbf{x}) = (\hat{f}_1(\mathbf{x}), \ldots, \hat{f}_K(\mathbf{x}))$, where $\hat{f}_k$ is the surrogate's prediction for f_k. Surrogate models such as Kriging also provide the uncertainty in the model's prediction generally in the form of standard deviation. The predicted uncertainty vector is represented as $\hat{\boldsymbol{\sigma}}(\mathbf{x}) = (\hat{\sigma}_1(\mathbf{x}), \ldots, \hat{\sigma}_K(\mathbf{x}))$, where $\hat{\sigma}_k$ is the uncertainty in prediction for the k^{th} objective function. In the third step, an MOEA is run to solve the optimization problem with the surrogates as objective functions.

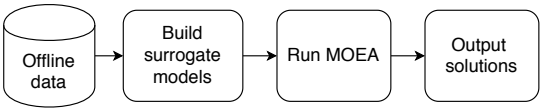


Fig. 1. A generic approach for offline data-driven multiobjective optimization.

Next, we briefly discuss an interactive approach for decomposition-based MOEAs which is a building block of the framework proposed in this paper.

2.2 Interactive Decomposition-Based MOEA

Decomposition-based MOEAs use reference (or weight) vectors to decompose the objective space into a number of sub-spaces. In general, they solve several simpler sub-problems that represent an aggregate of the objective functions by using a scalarizing function. Some examples of the scalarizing functions used are Chebyshev [20], penalty based boundary intersection distance (PBI) [20] and angle penalized distance (APD) [2]. The solutions obtained by solving these sub-problems jointly represent the approximated Pareto front of the MOP in the objective space.

Interactive decomposition-based MOEAs find solutions only in certain regions of the Pareto front. These approaches utilize preference information from the DM in the form of, e.g. a reference point, weights and preferred ranges for objectives. For more details, see, e.g. [14, 19]. In this paper, we adopt the interactive approach proposed in [7] for decomposition-based MOEAs and briefly describe its main ideas as follows.

Converting Preference Information to Reference Vectors: One of the ways to incorporate preference information into decomposition-based MOEAs is by adapting the reference vectors to follow the DM's preferences [14]. We here demonstrate how to utilize a reference point which consists of the DM's desired value for each objective. However, the framework proposed later in this paper is not limited to only this type of preference information.

Consider a set of uniformly distributed reference vectors $V = \{\mathbf{v}^i \in \mathbb{R}^k | i = 1, \ldots, m\}$, where m is the total number of reference vectors, and $\bar{\mathbf{z}} \in \mathbb{R}^k$ is a single reference point provided by the DM. Each reference vector can be adapted as follows [2, 7]:

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

where $\mathbf{v}^c = \bar{\mathbf{z}} / \|\bar{\mathbf{z}}\|$ and $r \in (0, 1)$. The central vector $\mathbf{v}^c$ is the projection of $\bar{\mathbf{z}}$ on a unit hypersphere and the spread of the adapted reference vectors is determined by the parameter r. The adapted reference vectors are close to $\mathbf{v}^c$ if r is close to zero and if r is close to one, the reference vectors are not changed much.

3 The Proposed Framework

As mentioned, since no new data is available in offline data-driven optimization, the approximation accuracy of the surrogate models determines the quality of solutions. In reality, the surrogate models' approximation involve uncertainty. As mentioned, Kriging surrogates [6] also provide an estimate of the uncertainty in its prediction. A solution with a higher uncertainty indicates that the objective values predicted by the surrogates have a lower probability of being close to the values of the underlying objective function. In other words, the uncertainty predicted by the surrogate models can represent the accuracy of the solutions when evaluated using the underlying objective functions. In [11], utilizing the predicted uncertainties from the surrogates as additional objective(s) produced solutions with a better hypervolume and accuracy in root mean squared error (RMSE) compared to the generic approach. This was because the approach simultaneously minimized the objective functions and their respective uncertainties. The solutions generated represented the trade-off between objective values and uncertainties. However, this results in an increase in both computational and cognitive load with a large number of objectives. Overall, it is desirable for the DM to get solutions that have a low uncertainty in order to achieve better accuracy.

As explained before, interactive approaches are quite advantageous as the DM can guide the optimization process through preferences for *objectives* and also learn about the problem. To incorporate preferences for *uncertainties* while solving an MOP interactively, the DM should first understand what uncertainty really means in regards to the MOP. Giving the DM an opportunity to provide preferences for uncertainties is desirable but may increase cognitive load.

The proposed framework aims at solving offline data-driven MOPs interactively by considering preferences for both objectives and uncertainties. The

framework is based on a decomposition-based MOEA and preference information for objectives in the form of reference points. The first and primary challenge faced is the DM's understanding of uncertainty, specifically the uncertainty in the surrogates' approximation. Secondly, the cognitive load should not drastically increase when the DM wants to provide preferences for uncertainties along with the preferences for objectives. The proposed framework tackles both of the challenges and aims at providing an improved decision support for the DM during the solution process. Next, we discuss two steps which are the primary building blocks of the proposed framework.

3.1 Pre-Filtering Solutions following DM's Preferences

Generally, in offline data-driven MOPs, there exists a trade-off between the quality of solutions (e.g. hypervolume) and the accuracy of the solutions (e.g. RMSE) [11]. To have a diverse range of uncertainty and objective values, we first store the solutions from all the generations of an MOEA in an archive. This allows us to filter and make decisions from a pool of solutions having various objective and uncertainty values. However, only the solutions representing the DM's preferences for objectives are interesting to him/her. Hence, the archive needs some amount of pre-filtering before we can present it to the DM. We have to further filter these solutions such that only the solutions that simultaneously achieve the best objective values and the lowest uncertainties are shown to the DM. Hence, we propose a two-stage pre-filtering approach as follows.

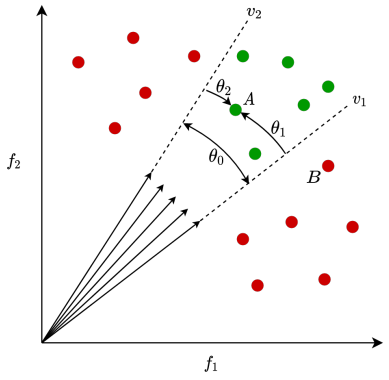


Fig. 2. Pre-filtering solutions: Green dots are kept and red dots are rejected.

The first stage is to find solutions in the archive that follow the DM's preferences for objectives, i.e., reference points. As described in Section 2.2, at first, the uniformly distributed set of reference vectors are adapted using Eq. (2) that reflects the DM's preferences for objectives. Next, we find adapted reference vectors that have the highest component in one of the objectives and the lowest in

all other objectives and call them *edge vectors*. The uniformly distributed vectors have just one vector at each axis (objective). Hence, we find just one edge vector for each objtive when adaptation in performed by a linear transformation in Eq. (2). Thus, we have K edge vectors. The multidimensional volume enclosed by the hyperplanes formed by the edge vectors is termed as the *hypercone*. A solution is accepted by the first stage pre-filter if it lies inside the hypercone. Fig. 2 shows the idea of the pre-filtering for a bi-objective minimization problem. The edge vectors are v_1 and v_2 and the angle between the edge vectors is θ_0. The angle between solution A and the edge vectors v_1 and v_2 is θ_1 and θ_2, respectively. A solution is accepted for the next pre-filtering stage if both θ_1 and θ_2 are smaller than θ_0. In the figure, the solutions in green (e.g. A) are accepted by this pre-filtering stage, and the solutions shown in red (e.g. B) are rejected. The rejected solutions do not follow the preferences and hence are not of interest to the DM. In general, with K objectives, the angle θ_0 between any two edge vectors is the same. This is because the set of uniformly distributed reference vectors is adapted by using a linear transformation. Hence, a solution is inside the hypercone if $\theta_k^i < \theta_0$ for all $k = 1, \ldots, K$, where θ_k^i is the angle between the kth edge vector and the ith solution.

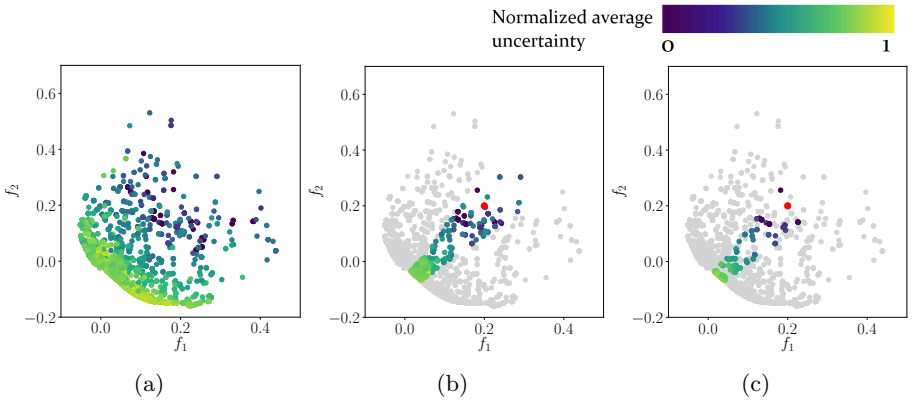


Fig. 3. The sub-figures show the solutions in different pre-filtering stages while solving a bi-objective minimization problem. The grey solutions are the ones filtered out at each stage. The red point denotes the reference point provided by the DM.

The archive contains objective vectors and their respective uncertainties from all the generations. However, only the solutions with the smallest uncertainties and objective values are interesting for the DM. Hence, we propose a second pre-filtering stage that performs nondominated sorting on the solutions filtered by the first stage and include uncertainties as additional components in the vectors while sorting (as done in [11]). Considering uncertainty while performing nondominated sorting finds the solutions representing the trade-off between objective values and uncertainty.

These two stages are applied sequentially in the pre-filtering stage of our proposed framework. The functioning of the pre-filtering stage can be understood from Fig. 3, which shows solutions in the archive for a bi-objective minimization problem. The colour code represents the normalized average of the uncertainty vector for the solutions. Sub-figure (a) shows all the solutions in the archive before the pre-filtering. Sub-figure (b) shows the solutions after the first stage pre-filtering. It can be observed that only the solutions following the preferences for objectives (here the reference point in red) are filtered. Sub-figure (c) shows the solutions obtained after the second stage pre-filtering. The solutions after the pre-filtering stage follow the DM's preferences for objectives and represent the trade-off between objective values and uncertainties in the solutions. The grey solutions are the ones that are rejected at each pre-filtering stage.

3.2 DM's Understanding of Uncertainty

As discussed before, knowledge of uncertainty is an essential aspect while solving offline optimization problems. However, while solving real-life problems, the DM is not always familiar with uncertainty in the solutions. Depending on the problem, the DM can be assumed to have an idea of permissible tolerances in objective values. For example, in the welded beam problem [5], cost and end deflection are minimized. Considering just the DM's preference regarding cost, (s)he is also aware of the highest permissible cost. Thus, the permissible deviation in the objective is the preferred one-sided *tolerance* of the DM [9]. In other words, one-sided tolerance information can be considered as a cutoff over the probable variation in the objective values of the solutions. In our case, the variation in objective values is available in the form of uncertainty in the surrogates. Preferred one-sided tolerances are *preferences for uncertainties* provided by the DM and represent the maximum permissible variation in the solutions when they are evaluated by the underlying objectives. In this paper, we refer to one-sided tolerance as tolerance for simplicity.

For the proposed framework (and later in the tests), we consider indifference tolerances. They are provided as a percentage for every objective and represent the 95% tolerance interval [9]. Let us consider the indifference tolerance provided by the DM for the k^{th} objective function as $\tau_k\%$, where $k = 1, \ldots, K$. The distribution of the predicted objective value is Gaussian while using Kriging surrogates and the predicted standard deviation of the k^{th} objectives' surrogate is $\hat{\sigma}_k(\mathbf{x})$. Thus, *cutoff tolerance functions* can be formulated such that the solutions do not violate the DM's preferences for uncertainties and thus are of interest to the DM. The k^{th} cutoff tolerance function is:

$$g_k(\mathbf{x}) = 1.96\hat{\sigma}_k(\mathbf{x}) - \tau_k \cdot \hat{f}_k(\mathbf{x})/100 \leq 0, \tag{3}$$

where $\mathbf{x}$ is the decision vector and $k = 1, \ldots, K$. A solution is interesting to the DM if the objective value of the k^{th} objective function does not exceed $1.96\hat{\sigma}_k(\mathbf{x})$ or 95% confidence interval of the Gaussian distribution. Thus, the DM can change the preferences for uncertainties and visualize the solutions that do

not violate the cutoff tolerance functions in Eq. (3). However, it has to be noted that the cutoff tolerance function can be modified depending on the prediction distribution of the surrogate.

3.3 Steps of the Framework

Fig. 4 shows the simplified structure of the proposed interactive offline data-driven MOEA framework. The framework can be broadly divided into five steps:

1. Building surrogate models and initializing the MOEA.
2. Running the MOEA and storing the solutions in an archive.
3. Applying two-stage pre-filtering on the archive.
4. Interactively visualizing the solutions based on the preferences for uncertainties provided by the DM.
5. Asking for preference information for objectives from the DM and adapting the reference vectors.

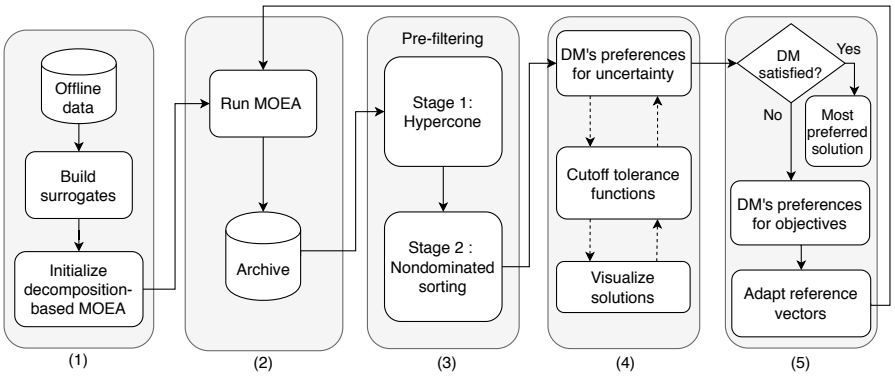


Fig. 4. The proposed framework for interactive offline data-driven multiobjective optimization.

Step 1: We formulate the MOP by utilizing the provided data. The expertise of the DM may be required in this. We build Kriging surrogate models for every objective function using the data (as in the generic approach in Section 2). Next, we initialize a decomposition-based MOEA and generate a uniformly distributed set of reference vectors and create the initial population.
Step 2: We run an MOEA for a fixed number of generations. The objective values and uncertainties for the individuals from every generation are stored in an archive that serves as a database for Step 3.
Step 3: At the end of Step 2, we have an archive containing objective vectors and uncertainties of different individuals. We apply the pre-filtering techniques as in Section 3.1. Note that for the first iteration, we do not have any preferences

for objectives, and the reference vectors (that includes the edge vectors) are not adapted. Hence, the hypercone constitutes the entire objective space and the first pre-filtering stage accepts all the solutions.
Step 4: The DM provides preferences for uncertainties (indifference tolerances) $\tau_k\%$ and the pre-filtered solutions from Step 3 qualifying the cutoff tolerance functions in Eq. (3) are shown.

The DM can provide preferences for uncertainties as many times (s)he wishes thereby enabling him/her to view different solutions within the provided tolerances. For a better understanding of uncertainties while visualizing, solutions can be colour coded. This can be done by the normalized average of the uncertainty vector (in percentage) or by the maximum uncertainty of a solution for any of the objective functions. The DM may skip this step entirely if solution uncertainties are not interesting. As this step consists of just filtering solutions obtained after Step 3, it can be repeated with a very low computational cost.
Step 5: In this step, the DM can stop the optimization process if (s)he has found a satisfactory solution. Otherwise, (s)he is asked for new preference information. We adapt the reference vectors according to Eq. (2) so that solutions follow the preferences for objectives. After adapting the reference vectors, we go to Step 2.

The interaction process is split into Steps 4 and 5, where the DM provides preferences for uncertainties and objectives, respectively. Due to this, the cognitive load on the DM does not increase significantly. The DM can provide different preferences for uncertainties and view the corresponding solutions and repeat this as long as one wishes. The proposed way of providing preferences for uncertainties does not modify the selection process of the MOEA. Hence, the solution process is not affected.

4 Numerical Results

Assessing and comparing the performance of interactive approaches is still a research challenge. Hence, we demonstrate and discuss the advantages of the proposed framework by solving the general aviation aircraft (GAA) [15, 16] design problem. Due to space limitations, further analysis on benchmark problems is available at `http://www.mit.jyu.fi/optgroup/extramaterial.html` as additional material.

The GAA problem refers to designing an aircraft for recreational pilots to business executives. We solved the problem as in [15] with 27 decision variables, ten objectives and one constraint. As we are dealing with offline optimization problems, we generated data using the implementation [1]. We used Latin hypercube sampling [12] to generate 1000 samples for decision variables and evaluated them using the GAA functions to obtain the offline data. To approximate the underlying objective functions, we used Kriging with a radial basis function kernel as our surrogate models. We used RVEA as the MOEA with standard parameter settings as in [2] and executed it for 100 generations in each iteration with standard crossover and mutation parameters. The spread parameter r was set as 0.2. However, it can be increased if the DM's wants a more diverse set of solu-

tions. As our framework does not support constraint handling, we considered the constraint violation as an additional objective function for the demonstration.

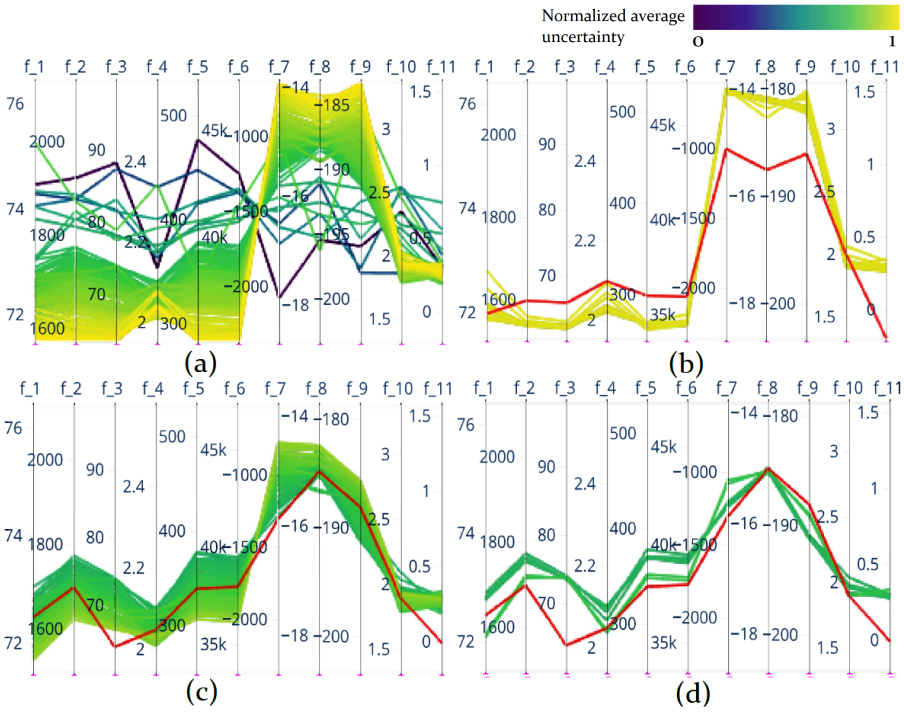


Fig. 5. The Solutions obtained for two iterations of the interactive framework (all objectives are minimized). (a): solutions in the archive after the first iteration. (b) & (c): solutions after pre-filtering in the first and second iteration respectively with different reference points (red line). (d): solutions after DM provides preferences for uncertainties.

Fig. 5 shows solutions produced by the framework for two iterations. The colour coding represents the normalized average of the uncertainty vector for the solutions (blue is lowest and yellow is highest). Sub-figure (a) shows the solutions in the archive at the first iteration when there are no preferences for objectives available. In sub-figure (b) the DM provides the reference point (in red) and gets the pre-filtered solutions. It can be observed that the solutions produced follow the DM's preferences for objectives. However, (s)he chooses to skip the step of providing preferences for uncertainties as none of the solutions has a low uncertainty (as represented by the colour). In the next iteration, the DM changes the preferences for objectives. The solutions after pre-filtering, as shown in sub-figure (c) not only follow the DM's preferences for objectives but also have a lower uncertainty. We now provide hypothetical tolerances to demonstrate the

framework's ability to consider preferences for uncertainties. In sub-figure (d) only a few solutions that are within the preferred uncertainty of the DM are shown. Finally, one of the solutions that matches the preferences for objectives and uncertainties may be chosen by the DM. (S)he may choose to reset the cutoff tolerances again to view a different set of solution to make decisions. Alternatively, if the DM is not satisfied with any of the solutions, (s)he may choose to change the preferences for objectives and continue the optimization.

If the DM is unaware of the uncertainties in the solutions, (s)he may be deprived of valuable knowledge regarding the acceptability of the solutions. In certain situations such as Fig. 5 (b), judging the goodness of a solution based on the objective values alone may be misleading. By observing the uncertainties, the DM avoids making a worse decision and can modify preferences for objectives. The DM may choose to provide preferences for uncertainties and see solutions within different tolerances with a low computational cost. As the DM can see the solutions pre-filtered from the archive that have various uncertainties, (s)he has a wide range of solutions to make decisions if so desired.

5 Conclusions

In this paper, we proposed a framework for interactively solving offline data-driven MOPs. It enabled the DM to understand and provide preferences for uncertainties during an interaction. By using preferences for objectives, the DM can guide the solution process. The solutions generated follow the DM's preferences for objectives and have a variety of uncertainties. By preferences for uncertainties, the DM can control which solutions (s)he can see. The two-step interaction proposed in the framework does not significantly increase the cognitive load on the DM. We also demonstrated it by solving the GAA problem that proved its capability in solving many-objective problems. The visualization in the framework enabled the DM to provide preferences for uncertainties interactively. However, more work should be done in the field of reference vectors adaptation and development of comparison metrics for interactive approaches. We also need to perform tests with different types of preferences for objectives. Furthermore, the framework is not designed to handle constraints. Handling constraints for offline data-driven problems deserves further attention.

Acknowledgements

This research was supported by the Academy of Finland (grant no 311877) and is related to the thematic research area DEMO (Decision Analytics utilizing Causal Models and Multiobjective Optimization, jyu.fi/demo) of the University of Jyväskylä. This work was partially supported by the Natural Environment Research Council [NE/P017436/1].

References

1. Castro-Gama, M.: mariocastrogama/GAA-Problem-MATLAB (2017), `https://github.com/mariocastrogama/GAA-problem-MATLAB`, accessed April 21, 2020
2. Cheng, R., Jin, Y., Olhofer, M., Sendhoff, B.: A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Transactions on Evolutionary Computation **20**(5), 773–791 (2016)
3. Chugh, T., Chakraborti, N., Sindhya, K., Jin, Y.: A data-driven surrogate-assisted evolutionary algorithm applied to a many-objective blast furnace optimization problem. Materials and Manufacturing Processes **32**(10), 1172–1178 (2017)
4. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (2014)
5. Deb, K.: Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. In: Miettinen, K., Neittaanmäki, P., Mäkelä, M.M., Périaux, J. (eds.) Evolutionary Algorithms in Engineering and Computer Science. Wiley (1999)
6. Forrester, A., Sobester, A., Keane, A.: Engineering Design via Surrogate Modelling. John Wiley & Sons (2008)
7. Hakanen, J., Chugh, T., Sindhya, K., Jin, Y., Miettinen, K.: Connections of reference vectors and different types of preference information in interactive multiobjective evolutionary algorithms. In: Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1–8 (2016)
8. Jin, Y., Wang, H., Chugh, T., Guo, D., Miettinen, K.: Data-driven evolutionary optimization: An overview and case studies. IEEE Transactions on Evolutionary Computation **23**(3), 442–458 (2019)
9. Krishnamoorthy, K., Mathew, T.: Statistical Tolerance Regions: Theory, Applications, and Computation. John Wiley and Sons (2009)
10. Li, K., Chen, R., Min, G., Yao, X.: Integration of preferences in decomposition multiobjective optimization. IEEE Transactions on Cybernetics **48**(12), 3359–3370 (2018)
11. Mazumdar, A., Chugh, T., Miettinen, K., López-Ibáñez, M.: On dealing with uncertainties from Kriging models in offline data-driven evolutionary multiobjective optimization. In: Deb, K., Goodman, E., Coello Coello, C.A., Klamroth, K., Miettinen, K., Mostaghim, S., Reed, P. (eds.) Evolutionary Multi-Criterion Optimization, 10th International Conference, Proceedings. pp. 463–474. Springer (2019)
12. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics **21**(2), 239–245 (1979)
13. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers (1999)
14. Purshouse, R.C., Deb, K., Mansor, M.M., Mostaghim, S., Wang, R.: A review of hybrid evolutionary multiple criteria decision making methods. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC). pp. 1147–1154 (2014)
15. Shah, R.A., Reed, P.M., Simpson, T.W.: Many-Objective Evolutionary Optimisation and Visual Analytics for Product Family Design. In: Wang, L., Ng, A.H.C., D.K. (eds.) Multi-objective Evolutionary Optimisation for Product Design and Manufacturing. pp. 137–159. Springer (2011)
16. Simpson, T.W., Allen, J.K., Chen, W., Mistree, F.: Conceptual design of a family of products through the use of the robust concept exploration method. In: 6th

AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization. pp. 1535–1545 (1996)
17. Wang, H., Jin, Y.: A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems. IEEE Transactions on Cybernetics **50**(2), 536–549 (2020)
18. Wang, H., Jin, Y., Sun, C., Doherty, J.: Offline data-driven evolutionary optimization using selective surrogate ensembles. IEEE Transactions on Evolutionary Computation **23**(2), 203–216 (2019)
19. Wang, H., Olhofer, M., Jin, Y.: A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges. Complex & Intelligent Systems **3**(4), 233–245 (2017)
20. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation **11**(6), 712–731 (2007)
21. Zheng, J., Yu, G., Zhu, Q., Li, X., Zou, J.: On decomposition methods in interactive user-preference based optimization. Applied Soft Computing **52**, 952 – 973 (2017)

PIV

TGP-MO: TREED GAUSSIAN PROCESSES FOR SOLVING OFFLINE DATA-DRIVEN MULTIOBJECTIVE OPTIMIZATION PROBLEMS

by

Atanu Mazumdar, Manuel López-Ibáñez, Tinkle Chugh, Jussi Hakanen, and Kaisa Miettinen

Submitted to a journal

TGP-MO: Treed Gaussian Processes for Solving Offline Data-Driven Multiobjective Optimization Problems

Atanu Mazumdar
University of Jyvaskyla
Faculty of Information Technology
P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla
Finland

Manuel López-Ibáñez
University of Málaga
School of Computer Science
Bulevar Louis Pasteur 35, Campus de Teatinos,
29071 Málaga, Spain

Tinkle Chugh
Department of Computer Science
University of Exeter, UK

Jussi Hakanen
University of Jyvaskyla
Faculty of Information Technology
P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla
Finland

Kaisa Miettinen
University of Jyvaskyla
Faculty of Information Technology
P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla
Finland

Abstract*—Solving offline data-driven multiobjective optimization problems (MOPs) is challenging as we do not have new data available during the optimization process. We first build approximation models (or surrogates) using the provided offline data. An optimizer, e.g. a multiobjective evolutionary algorithm, can then be utilized to find Pareto optimal solutions to the problem with surrogates as objectives. Gaussian process (GP) models are widely used as surrogates because of their ability to provide uncertainty information. However, building GPs becomes computationally expensive when the size of the dataset is large. Using sparse GPs is an alternative as they reduce the computation cost of building. However, they are not tailored to solve offline MOPs, where good accuracy of the surrogates is needed near Pareto optimal solutions. In this paper, we propose treed GP surrogates (TGP-MO) for offline MOPs. We first split the decision space into subregions using regression trees and build GPs sequentially in regions close to Pareto optimal solutions in the decision space to accurately approximate the tradeoffs between the objectives. TGP-MO surrogates are computationally inexpensive as we build GPs only in a smaller region of the decision space utilizing a subset of the data. We tested TGP-MO surrogates on distance-based visualizable problems with various data sizes, sampling strategies, numbers of objectives, and decision variables. Experimental results showed that TGP-MO surrogates are computationally cheaper and can handle datasets of large size. Furthermore, TGP-MO surrogates produced solutions close to Pareto optimal solutions compared to full GPs and sparse GPs.

***Index Terms*—Gaussian processes, Kriging, Regression trees, Metamodelling, Surrogate, Pareto optimality**

1. Introduction

A multiobjective optimization problem (MOP) consists of two or more conflicting objective functions that are optimized simultaneously. The solutions of an MOP are called Pareto optimal when we are unable to improve the value of any objective without degrading some of the others; thus, we have tradeoffs among the objectives. Not all real-world MOPs have analytical functions or simulation models available. Instead, the MOP may need to be formulated by utilizing the data acquired from a phenomenon, i.e. real-life processes, physical experiments, sensors, etc. To solve the MOP, we first approximate the *underlying objective functions* by fitting surrogates (also known as metamodels) on the data available. Next, we apply a multiobjective optimization algorithm to the problem with surrogates as objectives to approximate the set of Pareto optimal solutions. The optimization algorithm can be a multiobjective evolutionary algorithm (MOEA) since they have shown benefits in solving black-box optimization problems [1].

We can classify data-driven optimization problems into *online* and *offline* ones [2]. In online data-driven optimization, new data can be acquired during the optimization process, i.e. by conducting further experiments or simulations to update the surrogates [3]. In contrast, in offline data-driven optimization problems, no new data can be acquired,

and the optimization has to proceed by only using the existing data [4]. Ideally, the underlying objective values of the solutions should be better than the objective values in the provided dataset. Thus, making the best use of all the available data is desirable while solving offline MOPs. However, when the size of the data set is large, e.g. in [5], [6], building certain surrogates like Gaussian processes, can become computationally expensive. As we cannot update the surrogates while solving offline MOPs, the approximation accuracy of surrogates directly influences the closeness of the solutions to the Pareto optimal. Most of the previous works on offline data-driven optimization, such as [4], [5], [6], [7] considered different surrogate modelling techniques for solving offline data-driven optimization problems. However, these works did not consider the tradeoffs between the underlying objectives while building the surrogates. In this paper, we propose an approach to build computationally cheap surrogates tailored towards solving offline data-driven MOPs. The proposed surrogates use regression trees and Gaussian processes and can handle large datasets (tested for a maximum size of 50,000).

To study offline data-driven MOPs, we sample data from benchmark problems for which the Pareto optimal set is known. This enables quality comparisons of the solutions obtained by different optimization approaches and surrogates. While solving an offline data-driven MOP, we generally utilize two metrics to quantify the quality of the solutions [8], [9]. We measure the approximation accuracy in root mean squared error (RMSE) between the approximated objective values of the surrogates and the underlying objective values. In addition, we use hypervolume [10] to measure how close the approximated Pareto front (after evaluating with the underlying objectives) is to the Pareto front of the underlying MOP.

Kriging or Gaussian process (GP) regression [11] is a popular choice of surrogate [12] for solving offline problems since it also provides information about the uncertainty in the approximation. Previous works [8], [13], [14] have shown that utilizing the uncertainty prediction from GP surrogates produces solutions with improved accuracy and hypervolume. This makes GPs an attractive choice of surrogates for solving offline MOPs. Most of the approaches for solving offline MOPs build a global GP surrogate for each objective using all the provided data. The time and memory complexities of building global GPs (or full GPs) are $O(N^3)$ and $O(N^2)$, respectively, where N is the size of the dataset (or sample size). Thus, building full GP surrogates makes the process computationally expensive when the size of the dataset is large. Using sparse GPs [15], [16] is a suitable alternative as they build approximation models using a small subset of data called support or inducing points. Using M (with $M < N$) inducing points lowers the computational cost to $O(NM^2)$ for building the surrogates. In [16] a variational method was proposed to select the inducing points by gradient descent or greedy search algorithm. However, this process becomes computationally expensive when the dataset is large. On the other hand, using fewer points for building the surrogates reduces the approximation accuracy of the surrogates compared to full GPs [16]. Other works such as [17] proposed building GP using only K-nearest neighbour samples from the point that is to be predicted.

While performing multiobjective optimization using surrogates, it is desirable to obtain a good approximation accuracy in the neighbourhood of the Pareto set (that we refer to as the *tradeoff region*). The accuracy of the surrogates in other regions of the decision space (excluding the tradeoff region) is not of utmost importance. Thus, a possible approach is to build local GP surrogates for the underlying objectives exclusively in the tradeoff region. This shall reduce the overall computation cost of building GPs while preserving the accuracy at the tradeoff region. Previous works [18], [19], [20] partitioned the decision space into regions and fitted GPs in each region. Partitioning the decision space is relatively inexpensive, and each GP utilizes smaller sets of data. This concept was extended by [21] to fit GPs in each region or leaf nodes partitioned by a Bayesian treed model previously proposed by [22]. In [23] treed GPs was proposed to deal with cases where the noise is different for different samples. All the previous works build GPs in all the regions of the decision space. These types of GPs become computationally expensive as well, depending on the number of regions and amount of data in each region. Moreover, these surrogates are not explicitly designed to solve offline MOPs because they do not consider the tradeoff region of the MOP during the building process.

In this paper, we propose treed GP surrogates for multiobjective optimization (TGP-MO) surrogates that have a high accuracy around the tradeoff region and are tailored for solving offline MOPs. We first build computationally inexpensive regression tree surrogates using the provided dataset. The regression tree surrogates provide a less accurate approximation of the underlying objectives (compared to GPs) [24] and split the decision space into regions. Next, we run an MOEA considering the regression trees as objectives. The solutions of the MOEA are not very accurate but provide the approximate location of the tradeoff region. After a certain number of generations of the MOEA, we improve the accuracy of the trees' prediction by building GPs at leaf nodes within the tradeoff region. This improves the accuracy of the solutions in the region of the decision space corresponding to the leaf node that they replace. The final surrogates consist of regression trees with GPs at a few leaf nodes providing accurate approximations exclusively in the tradeoff region.

The TGP-MO surrogates were tested on several instances of distance-based visualizable test problems (DBMOPP) with different numbers of decision variables and objectives. Various sizes of the initial dataset with different sampling strategies were used in the tests. Numerical experiments showed that TGP-MO surrogates significantly reduced the building time for surrogates when using large size data while solving offline data-driven MOPs. TGP-MO surrogates also produced solutions with a better hypervolume compared to sparse GP surrogates.

The rest of the paper is arranged as follows. The background of offline data-driven MOPs, GPs, and regression

Figure 1: Flowchart of a generic offline data-driven multi-objective optimization approach.

trees is given in Section 2. The proposed TGP-MO surrogates are detailed in Section 3. The experimental results consisting of a comparison of TGP-MO surrogates with other surrogates are presented in Section 4. In Section 5, we conclude and discuss future research perspectives.

2. Background

In an offline data-driven MOP, the starting point for solving the problem is (pre-collected) data. In this paper, we assume that the available data is the output of a process or phenomenon. Here we refer to the process generating the data as the *underlying* objective functions of an MOP. We consider the underlying MOP that has to be solved of the following form:

$$\begin{aligned} &\text{minimize} \quad (f_1(\mathbf{x}), \ldots, f_K(\mathbf{x})) \\ &\text{subject to} \quad \mathbf{x} \in \Omega, \end{aligned} \tag{1}$$

where $K \geq 2$ is the number of objectives and Ω is the feasible region in the decision space $\Re^n$. For a feasible decision vector $\mathbf{x}$ (consisting of n decision variables), the corresponding objective vector is $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_K(\mathbf{x}))$, where $f_1(\mathbf{x}), \ldots, f_K(\mathbf{x})$ are the objective (function) values.

A solution $\mathbf{x}^1 \in \Omega$ dominates another solution $\mathbf{x}^2 \in \Omega$ if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for all $i = 1, \ldots, K$ and $f_i(\mathbf{x}^1) < f_i(\mathbf{x}^2)$ for at least one $i = 1, \ldots, K$. If a solution of an MOP is not dominated by any other feasible solutions, it is called nondominated. Solving an MOP using a multiobjective optimization algorithm e.g. an MOEA typically produces a set of mutually nondominated solutions. The solutions of (1) that are nondominated in the whole set Ω are called Pareto optimal solutions.

A generic way to solve an offline data-driven MOP with an MOEA as the optimizer is shown in Figure 1. As explained in [2], [4], we can divide the solution process into three components: (a) data collection, (b) formulating the MOP and surrogate building, and (c) optimization with surrogates.

We start with an offline dataset consisting of N samples. A sample consists of a decision vector $\mathbf{x}$ and its corresponding objective vector $\mathbf{f}(\mathbf{x})$ as a tuple of two matrices: (X, Y) where $X \in \Re^{N \times n}$ and $Y \in \Re^{N \times K}$. Each row in X and Y is a decision vector and its corresponding objective vector, respectively. Next, we build surrogates using all or a subset of the data. The surrogates are considered as objective functions by an MOEA to solve the MOP. For simplicity of describing a surrogate model for a single objective i, let $\mathbf{y}_i \in \Re^{N \times 1}$ be the vector of objective values $f_i(\mathbf{x})$ for all decision vectors $\mathbf{x} \in X$.

2.1. Gaussian Process Regression

In this work, we build a surrogate model for each objective function. For simplicity in terminologies, we consider $\mathbf{y}$ instead of $\mathbf{y}_i$ to introduce the Gaussian processes (GP) and regression trees. GP regression (also known as Kriging) has been widely used in surrogate assisted optimization, time-series analysis, etc. [2], [25]. The major advantage of using a GP is its ability to provide the distribution about its prediction (or the uncertainty). A GP is a multivariate normal distribution with a mean $\boldsymbol{\mu}$ and a covariance matrix C:

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, C). \tag{2}$$

For simplicity in calculations, we consider a mean of zero without loss of generality. The covariance matrix C uses a covariance (or kernel) function to define correlation between two samples, $\mathbf{x}$ and $\mathbf{x}'$. For Matern 5/2 kernel the correlation is:

$$\begin{aligned} \kappa(\mathbf{x}, \mathbf{x}', \boldsymbol{\Theta}) =& \sigma_{\mathrm{f}}^2 (1 + \sqrt{5} \sum_{j=1}^{n} r_j + \frac{5}{3} \sum_{j=1}^{n} r_j^2) \exp(-\sqrt{5} \sum_{j=1}^{n} r_j) \\ &+ \sigma_{\mathrm{t}}^2 \delta_{\mathbf{x}\mathbf{x}'} \end{aligned}$$

where $r_j = \frac{|x_j - x_j'|}{l_j}$, $\boldsymbol{\Theta} = (\sigma_{\mathrm{f}}, l_1, \ldots, l_n, \sigma_{\mathrm{t}})$ is the set of parameters in the GP model and $\delta_{\mathbf{x}\mathbf{x}'}$ is the Kronecker delta function. The notation $|x_j - x_j'|$ represents the Euclidean distance between x_j and x_j'. The parameters σ_{f}, l_j and σ_{t} represent the amplitude, length scale of the j^{th} variable and noise in the data, respectively. For more details on the significance of these parameters, see [12].

To build a GP model, the parameters mentioned above can be estimated by maximising the marginal likelihood function:

$$p(\mathbf{y} \mid X, \boldsymbol{\Theta}) = \frac{1}{\sqrt{|2\pi C|}} \exp\left(-\tfrac{1}{2} \mathbf{y}^\top C^{-1} \mathbf{y}\right). \tag{3}$$

After estimating the parameters, the model (2) can be used for the posterior predictive distribution at a new decision vector $\mathbf{x}^*$. The GP model provides a posterior predictive distribution which is also Gaussian:

$$\begin{aligned} p\left(y^* | \mathbf{x}^*, X, \mathbf{y}, \boldsymbol{\Theta}\right) =& \mathcal{N}\big(\kappa(\mathbf{x}^*, X, \boldsymbol{\Theta}) C^{-1} \mathbf{y}, \kappa(\mathbf{x}^*, \mathbf{x}^*, \boldsymbol{\Theta}) - \\ & \kappa(\mathbf{x}^*, X, \boldsymbol{\Theta})^\top C^{-1} \kappa(X, \mathbf{x}^*, \boldsymbol{\Theta})\big). \end{aligned} \tag{4}$$

The posterior mean in (4) is $\kappa(\mathbf{x}^*, X, \boldsymbol{\Theta}) C^{-1} \mathbf{y}$ and the variance representing the uncertainty is $\kappa(\mathbf{x}^*, \mathbf{x}^*, \boldsymbol{\Theta}) - \kappa(\mathbf{x}^*, X, \boldsymbol{\Theta})^\top C^{-1} \kappa(X, \mathbf{x}^*, \boldsymbol{\Theta})$.

2.2. Regression Trees

A regression tree recursively partitions the decision space such that the provided samples with similar objective function values are grouped together [24]. Each node ϕ of the tree contains data $Q^\phi = (X^\phi, \mathbf{y}^\phi)$ with N^ϕ samples, where each row of $X^\phi \in \Re^{N^\phi \times n}$ is a decision vector $\mathbf{x}_i$ and each row of $\mathbf{y}^\phi \in \Re^{N^\phi \times 1}$ is its corresponding

objective value y_i, $i = 1, \ldots, N^\phi$. Node ϕ is split into nodes ϕ^{left} and ϕ^{right} according to parameter $\theta = (j, t)$, which specifies a j^{th} decision variable ($1 \leq j \leq n$) and a threshold t, by partitioning Q^ϕ into two disjoint subsets $Q_\theta^{\phi^{\text{left}}} = \{(X^{\phi^{\text{left}}}, \mathbf{y}^{\phi^{\text{left}}}) \mid (\mathbf{x}_i, y_i) \in Q^\phi \wedge x_{ij} \leq t\}$ and $Q_\theta^{\phi^{\text{right}}} = Q^\phi \setminus Q_\theta^{\phi^{\text{left}}}$. That is, split parameter θ partitions the samples (rows) in Q^ϕ according to the value of variable x_j of each decision vector $\mathbf{x} \in X^\phi$.

Given a node ϕ and a split parameter θ, the quality of the split is calculated as:

$$G(Q^\phi, \theta) = \frac{N^{\phi^{\text{left}}}}{N^\phi} H(Q_\theta^{\phi^{\text{left}}}) + \frac{N^{\phi^{\text{right}}}}{N^\phi} H(Q_\theta^{\phi^{\text{right}}}), \quad (5)$$

where $N^{\phi^{\text{left}}}$ and $N^{\phi^{\text{right}}}$ are the number of samples in $Q_\theta^{\phi^{\text{left}}}$ and $Q_\theta^{\phi^{\text{right}}}$, respectively, and $H(Q^\phi)$ is a loss function. For instance, the loss function for mean-squared error is:

$$H(Q^\phi) = \frac{1}{N^\phi} \sum_{y_i \in Q^\phi} (y_i - \bar{y}^\phi)^2, \quad (6)$$

where $\bar{y}^\phi$ is the mean objective value in Q^ϕ. We can find an optimal split θ^* of a node ϕ by minimizing (5) using a single objective optimization algorithm. The splitting process is recursed for both $Q_{\theta^*}^{\phi^{\text{left}}}$ and $Q_{\theta^*}^{\phi^{\text{right}}}$ until either a predefined maximum depth of the tree is reached or the node to split contains no more than a predefined minimum number of samples, N_{min}.

For predicting any given decision variable value, we traverse the regression tree to the respective leaf node. The prediction of the tree at the leaf node l is $\bar{y}^l = \frac{1}{N^l} \sum_{y_i \in Q^l} y_i$ that contains training subset Q^l with N^l number of samples.

3. Treed GPs for Multiobjective Optimization (TGP-MO)

In a generalized treed GP surrogate, we first build a regression tree model using the provided data as described in Section 2.2. The splitting at the nodes is done by minimizing (5) with loss function as in (6). The subset of data at the l^{th} leaf node is Q^l, where $l = 1, \ldots, L$ and L is the total number of leaf nodes in the regression tree built. A GP is fitted at every leaf node of the regression tree using the data $Q^l = (X^l, \mathbf{y}^l)$. The GPs are built in a similar fashion as described in Section 2.1 by maximizing (3) at the leaf node l. The predictive distribution of the l^{th} GP is $p\left(y^* | \mathbf{x}^*, X^l, \mathbf{y}^l, \mathbf{\Theta}\right)$ using (4). Building GPs with smaller subsets of data reduces the overall cost for building surrogates compared to building a GP with the entire dataset. However, building GPs at all the leaf nodes becomes expensive when there are too many of them, and the data subset at each leaf is large. Moreover, for solving an offline data-driven MOP, we do not need to accurately approximate the global landscape of the underlying objective functions, but only the local landscape near the tradeoff region.

While performing multiobjective optimization using surrogates, the approximation accuracy in the tradeoff region is crucial and directly influences the quality of the approximated Pareto optimal solutions. Hence, to obtain better quality solutions, it is desirable to have accurate approximations in the tradeoff region. While building the regression trees, the decision space is partitioned into regions. In addition, the initial approximation of regression tree surrogates (though not highly accurate) provides information about the tradeoffs between the objectives. After building the regression trees with all the provided data, we run an MOEA considering them as objectives. The solutions found by the MOEA are not accurate, but they provide an approximation of the tradeoff region. Later, we build local GPs exclusively in the leaf nodes representing the tradeoff region and achieve an accurate approximation only in the neighbourhood of the Pareto set. Now, we introduce the algorithm to build the treed GP surrogates for multiobjective optimization (TGP-MO) tailored for the purpose of solving offline data-driven MOPs.

Algorithm 1: Build process of TGP-MO surrogates

Input: Offline data of sample size N with n decision variables and K objectives; N_{min} = minimum number of samples at a leaf node I_{max} = maximum number of iterations of building GPs at leaves; G_{max} = maximum number of generations of the MOEA per iteration

Output: TGP-MO surrogates

1 For each objective $j = 1, \ldots, K$, initialize a treed GP surrogate by building a regression tree using the given offline data and N_{min}
2 Initialize MOEA population
3 Initialize iteration counter, $I = 0$
4 **while** $I < I_{\max}$ **and** *any solution falls in the leaf nodes without GPs* **do**
5 Set counter of generations per iteration, $G = 0$
6 **while** $G < G_{\text{max}}$ **do**
7 Perform crossover and mutation and generate offspring
8 Evaluate the individuals using the treed GP surrogates and combine the parents and offspring
9 Perform MOEA specific selection
10 $G = G + 1$
11 For each treed GP surrogate $j = 1, \ldots, K$, find the leaf node that has the maximum loss function value in the prediction of the population of s solutions: $l_{\text{ML}}^j = \arg\max_{i=1,\ldots,s} H(Q^{l_i^j})$ where $j = 1, \ldots, K$
12 Build GPs using subset $Q^{l_{\text{ML}}^j}$ for $j = 1, \ldots, K$
13 Replace the prediction of the trees' leaf nodes with the built GPs
14 $I = I + 1$

In Algorithm 1, we start with a dataset containing N samples with K objectives and n decision variables. We first build K regression trees (one per objective) with all the provided data. Just the parameter N_{min} is adjusted, and we do not control the depth of the trees. After building the regression trees, we can have a maximum of $\frac{N}{N_{\text{min}}}$ leaf nodes. Initially, there are no GPs present at the leaf nodes, and the predictions are from the regression trees (i.e., $\bar{y}^l$). Next, we initialize a population and run an MOEA that iteratively builds GPs at specific leaf nodes. Each iteration consists of running the MOEA for G_{max} generations (that is a predefined parameter) and building a GP at one leaf node in every tree. In every generation, offspring individuals are produced using crossover and mutation operators and evaluated using the treed GPs. Selection of individuals is then performed using MOEA-specific selection criterion, and the process is repeated for G_{max} generations within an iteration.

After the MOEA completes G_{max} generations, the approximation errors of the solutions (for each objective) are compared. This is done by calculating the loss function value, here mean squared error as shown in (6), of the leaf node predicting the objective values of the solutions found by the MOEA. The leaf nodes belonging to the j^{th} objective's treed GP surrogate containing the solutions are $\left\{l_1^j, \ldots, l_s^j\right\}$, where s is the total number of solutions found by the MOEA and $j = 1, \ldots, K$. The loss function values in these leaf nodes are $\left\{H(Q^{l_1^j}), \ldots, H(Q^{l_s^j})\right\}$. Next, we find the leaf nodes that have the maximum loss function value for the j^{th} treed GP, $l_{\text{ML}}^j = \arg\max_{i=1,\ldots,s} H(Q^{l_i^j})$. We build one GP for the j^{th} treed GP using the subset of samples $Q^{l_{\text{ML}}^j}$. This process is repeated for $j = 1, \ldots, K$ treed GPs. When multiple solutions fall within the same leaf node, we calculate the loss value once and build one GP for that leaf node. We repeat the process of adding GPs to the leaf nodes until any of the following criteria is met:

- Number of iterations has reached the predefined maximum of I_{max}.
- All the solutions in the MOEA's population fall in the leaf nodes that have GPs built.

The regression trees perform two tasks: a) they provide an approximation of the underlying objectives, and b) they split the decision space into smaller regions. In the initial iterations, the MOEA first finds solutions using the approximation provided by the regression tree surrogates (that may have a poor accuracy). These solutions have a higher approximation error (compared to while using GP surrogates), but they provide information about the tradeoff between the objectives. To improve the approximation accuracy, we build GPs for solutions provided by the trees that have the maximum approximation error. This ensures that GPs are built exclusively in the region of the Pareto set and where the tree's approximation is the worst, simultaneously. In later iterations, if the decision vector in the MOEA's population falls within the path of the leaf node where a GP is already built, the posterior predictive mean of the GP is used as the final prediction of the surrogate. Otherwise, the prediction is the mean value at the leaf node of the regression trees. We build local GPs sequentially because building a local GP at a leaf node improves the accuracy of the solutions in the following iteration. Thus, many solutions found in the previous iteration (belonging to different leaf nodes of the trees) may be eliminated as newer and better solutions are discovered. This reduces the number of local GPs required to be built for approximating the underlying objectives.

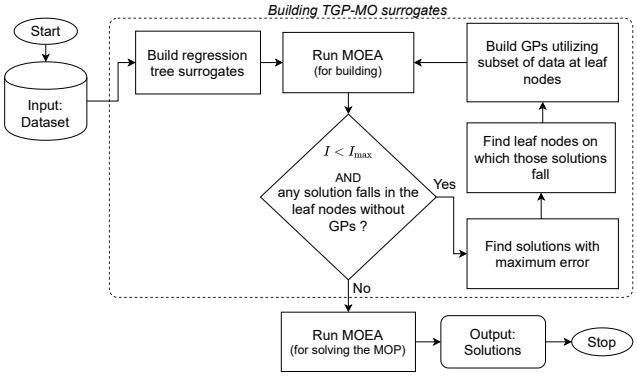


Figure 2: A flowchart of solving data-driven MOPs with TGP-MO surrogates with an MOEA.

The flowchart of solving offline data-driven MOPs with TGP-MO surrogates is shown in Figure 2. We start with a dataset and build regression trees for each objective. The blocks within the dotted lines represent the building process of TGP-MO surrogates as described in Algorithm 1. After the building is completed, we have surrogates composed of regression trees with GPs at some of the leaf nodes. The built TGP-MO surrogates are then used for multiobjective optimization using a second MOEA (that may have the same configuration as used in Algorithm 1).

However, instead of using two MOEAs, one may use the same MOEA to build the TGP-MO surrogates and solve the MOP. We can stop building the GPs at the leaves when the two mentioned stopping criteria are met. After building the TGP-MO surrogates, the MOEA continues to run and use the last generation's population for further evolution and improve the objective values. However, to test and compare the computation time exclusively for building the surrogates, we used the framework in Figure 2.

3.1. Accuracy Analysis

We illustrate the fitness landscape of a bi-objective DB-MOPP problem with $n = 2$ (with configuration same as P1 in Table 1) in Figure 3 to analyze the accuracy of the approximation of TGP-MO surrogates. The accuracy is measured as the RMSE (described later in subsection 4.1) between the approximated objective values of the surrogates and the evaluated values of the underlying objectives. The number of samples in the dataset was $N = 2000$ with LHS. The contour lines represent the underlying objectives' landscape in Figure 3a and the approximated objectives' landscape in

Figure 3b–3d. Further close up of the tradeoff regions for the sparse GP and TGP-MO surrogates are shown in Figure 3e and 3f, respectively. The color grading represents the RMSE between the approximated and the underlying objectives, and the blue dots represent the Pareto set.

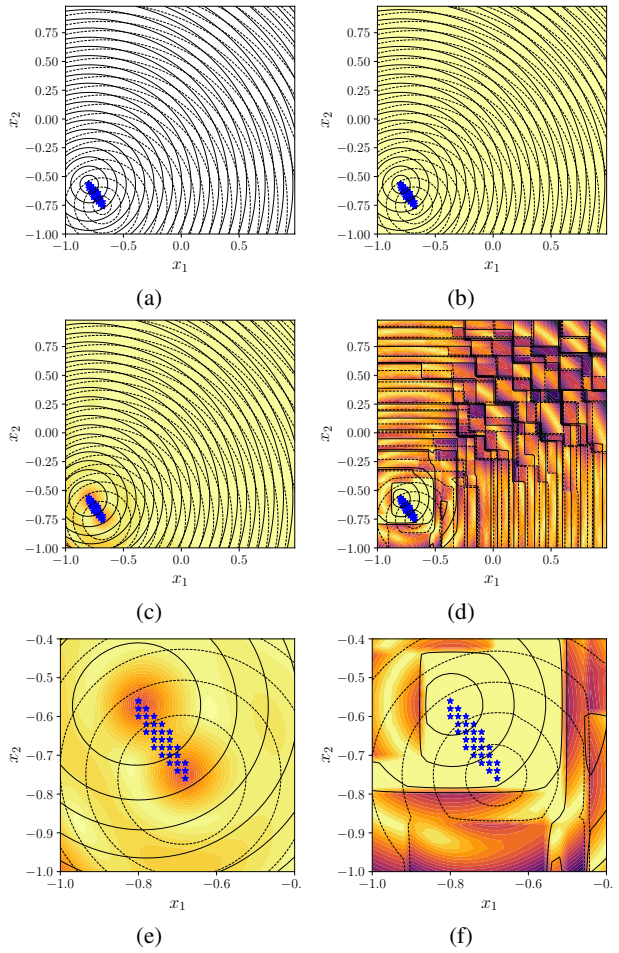


Figure 3: Contour plots of the (a) underlying objective function, landscape approximated by (b) full GPs, (c) sparse GPs, and (d) the proposed TGP-MO surrogates. Close up of the Pareto set for (e) sparse GPs and (f) the proposed TGP-MO surrogates. The contour lines (in solid and dotted) indicate the value of objectives $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$, respectively. Colour shade represents the RMSE in the approximation (darker is higher RMSE), and the blue dots show the Pareto set.

It can be observed that full GPs (that uses all the data) in Figure 3b has the highest accuracy in all the approximated regions of the decision space. For the sparse GP surrogates in Figure 3c, we can observe good accuracy in all the regions with deterioration near the Pareto set. In the case of the proposed TGP-MO surrogates in Figure 3d, the accuracy is low in most of the regions of the decision space except near the Pareto set. It can also be observed that the contour lines are linear in most regions because of the prediction provided by the leaf nodes of the trees. However, near the Pareto set the GPs at the leaf nodes are used for predictions, thus making the contours non-linear. Comparing the accuracy near the Pareto set for sparse GP and TGP-MO surrogates in Figure 3e and Figure 3f, it is evident that TGP-MO surrogates provides a better approximation accuracy compared to sparse GP surrogates.

3.2. Complexity

While building TGP-MO surrogates, we build a maximum of K number of GPs in every iteration. These GPs can have N_{min} to $2N_{\text{min}} - 1$ samples. This is because a split at the node of a tree occurs when the subset size is larger than or equal to $2N_{\text{min}}$. Thus every leaf node has at least N_{min} samples and a maximum of $2N_{\text{min}} - 1$ samples. As these local GPs have a smaller number of samples, the computational cost is low. The complexity of building TGP-MO surrogates can vary depending on the function landscape, the number of decision variables, and the provided data. The worst-case complexity will be achieved when GPs are built at all the leaf nodes of the regression tree with $2N_{\text{min}} - 1$ samples at each leaf node. The complexity of building a GP at a leaf node is $O((2N_{\text{min}} - 1)^3)$. As the total number of leaf nodes (in the worst case) is $\frac{N}{2N_{\text{min}}-1}$, the total complexity of building GPs at all the leaf nodes is $O(N(2N_{\text{min}} - 1)^2)$. For an MOP with K objectives, the worst case time complexity is $O(KN(2N_{\text{min}} - 1)^2)$ with a memory complexity of $O(KN(2N_{\text{min}} - 1))$.

Even in the worst case, the computational cost of building TGP-MO surrogates is significantly smaller than building full GPs with all the provided data with a complexity of $O(KN^3)$. For building sparse GPs, the complexity is $O(KNM^2)$ (where M is the number of induction points). However, as mentioned previously, finding the induction points uses a gradient descent or greedy search algorithm that becomes expensive when the sample size is large, increasing the overall computational cost. In contrast, the complexity of TGP-MO surrogates during optimization is primarily due to evaluating the individuals using the regression trees and GPs at the leaves, which is significantly lower. In the next section, we show some optimization results and time taken to build TGP-MO surrogates and compare them with other surrogate models.

4. Experimental Results

The goal of the proposed TGP-MO surrogates is to build computationally cheaper surrogates when dealing with large datasets for solving offline data-driven MOPs. We perform tests to find whether TGP-MO surrogates have significant improvement over sparse GPs and full GPs in computation time and the quality of the solutions by measuring hypervolume and accuracy. The starting point of the experiments was data generated from distance-based visualizable test problems (DBMOPP) [26] for a better understanding of the behaviour of the surrogates. In the DBMOPP test problems, we can simultaneously visualize the solutions in the decision

and objective spaces. This gives us the ability to visualize the search behaviour with the progress of the optimization process. These features of DBMOPP test problems prove to be more advantageous compared to DTLZ [27] benchmark problems.

All the approaches for solving the MOP were coded in Python utilizing the DESDEO framework (`http://desdeo.it.jyu.fi`).[1] The experiments were executed on one node of a HPC cluster equipped with AMD Rome CPUs, each node having 128 cores running at 2.6 GHz, with 256 GiB of memory. Each individual run was executed on one CPU core and allocated a maximum memory usage of 2 GiB. The regression tree was built using the sklearn Python package [28]. For building the GPs at the leaf nodes, we used the GPy [29] Python package.

4.1. Experiment Setup

4.1.1. Benchmark problems:. We used four configured DBMOPP problems P1-4 as shown in Table 1. The problem instances and data were generated by the code provided by [30]. All combinations of number of objectives ($K \in \{3, 5, 7\}$) and number of decision variables ($n \in \{2, 5, 7, 10\}$) were used for our tests.

TABLE 1: Configurations of the DBMOPP problems used.

Problem	Configuration	Dimension (n)	Objectives (K)
P1	*number of disconnected set regions* = 0, *number of local fronts* = 0, *number of dominance resistance regions* = 0, *number of discontinuous regions* = 0	2, 5, 7 and 10	3, 5, and 7
P2	*number of disconnected set regions* = 1, *number of local fronts* = 0, *number of dominance resistance regions* = 0, *number of discontinuous regions* = 0	2, 5, 7 and 10	3, 5, and 7
P3	*number of disconnected set regions* = 2, *number of local fronts* = 0, *number of dominance resistance regions* = 0, *number of discontinuous regions* = 0	2, 5, 7 and 10	3, 5, and 7
P4	*number of disconnected set regions* = 0, *number of local fronts* = 0, *number of dominance resistance regions* = 1, *number of discontinuous regions* = 0	2, 5, 7 and 10	3, 5, and 7

4.1.2. Dataset:. For generating the data, we used Latin hypercube sampling (LHS) and multivariate normal sampling (MVNS) [11]. In MVNS sampling, the objectives were considered independent with mean at the mid-point of the decision space, that is, zero for DBMOPP problems. The variance of the sampling distribution was set to 0.1 for all the objectives. Sample sizes of initial data ($N \in \{2000, 10000, 50000\}$) were chosen for the tests. For each problem configuration, we generated 31 sets of data with a random seed. Each of these dataset were the starting point of the three different surrogate models that were tested. These individual runs were independent, and the results were used to compare the surrogates' performances statistically.

1. Source code available at https://github.com/industrial-optimization-group/TreedGP_MOEA

4.1.3. Settings for TGP-MO surrogates:. We used RVEA [31], a decomposition-based MOEA [31], [32], [33] while building TGP-MO surrogates. Decomposition-based MOEAs have shown to be effective in solving MOPs with more than three objectives. However, the approach is not limited to RVEA and one can use any MOEA of choice. The parameters for RVEA were kept the same as suggested by [31]. The parameter G_{max} was set to 50. The maximum number of iterations was set to $I_{\text{max}} = \frac{N}{N_{\text{min}}} = \frac{N}{10n}$. Such a value of I_{max} was chosen considering we build one GP at a leaf node for all the trees in each iteration (as maximum number of leaf nodes are $\frac{N}{N_{\text{min}}}$). The loss function used for building the trees was MSE (Eq. 6) with $N_{\text{min}} = 10n$. We chose this setting because we want to have sufficient design points for building GPs at the leaves. Based on recommendations from the GP literatures [34], [35], we should have a minimum of $10n$ points at each leaf node. The kernel used for building the GPs at the leaf nodes was Matern 5/2 with automatic relevance determination enabled.

4.1.4. Other surrogates tested:. We compared the proposed TGP-MO surrogates with sparse GP and full GP surrogates. We used the same GPy package and kernel for building both of these surrogates as for the TGP-MO surrogates. For sparse GP surrogates, the number of induction points was set to $M = 10n$. Here we refer to the overall process of solving an MOP with a specific surrogate as *approach* for simplicity.

4.1.5. Parameter settings of MOEA (for solving the MOP):. For solving the MOP with surrogates as objectives, we also used RVEA with the same default parameter settings. Termination criteria for RVEA was 1000 generations that was sufficient for all the approaches tested to converge. To generate a uniform Pareto front, the reference vectors are rearranged or adapted after a certain number of generations in RVEA. In our experiments, we set the reference vector adaptation rate to once every 100 generations.

4.1.6. Performance metrics:. We measure the quality of the solutions obtained by the MOEA in terms of their hypervolume (HV) after evaluating them with the underlying objectives of the test problems. For computing the HV metric, we used the reference point $\mathbf{y}^* = (y_1^*, y_2^*, \ldots, y_K^*)$ in the objective space, such that is dominated by all the solutions. In this paper we used $\mathbf{y}^* = (2\sqrt{K}, 2\sqrt{K}, \ldots, 2\sqrt{K})$ as this point is always dominated in DBMOPP problems. We calculated the multivariate RMSE of the objective values of the solutions obtained by the MOEA with their respective underlying objective values to measure the accuracy. The multivariate RMSE is the Euclidean distance between the approximated and evaluated underlying objective function values of the solutions and is given by $\frac{1}{s}\sum_{i=1}^{s}\sqrt{\sum_{j=1}^{K}(\hat{f}_{j,i} - f_{j,i})^2}$, where s is the number of solutions, $\hat{f}_{j,i}$ and $f_{j,i}$ are the approximated and the underlying objective value, respectively, for the i^{th} solution and j^{th} objective. Finally, we measure the computational cost of the various methods as the time in seconds required to build the surrogates.

In this paper we refer to multivariate RMSE as RMSE for simplicity. The hypervolume and RMSE metrics are used here to benchmark the performance of TGP-MO surrogates for solving an offline MOP. Evaluating the solutions with the underlying objectives in an offline data-driven MOP may not be possible in real life.

4.2. Results and Discussions

While running the experiments, we observed that building full GP surrogates with 10 000 and 50 000 sample sizes consistently gave us out-of-memory errors. The storage complexity of full GPs is $O(KN^2)$ and each element of the array is a 64-bit float. Hence, the memory requirement for three objectives and sample sizes of 10 000 and 50 000 is 2.3 GiB and 56 GiB, respectively. This shows that building GP surrogates using all the provided offline data will become almost impossible with readily available computing resources when the sample size is large. Hence the results for full GP surrogates are not included for sample sizes of 10000 and 50000.

TABLE 2: Summary of pairwise comparison of the hypervolume (HV), RMSE, and time (seconds) for the different approaches. The number of instances in which full GP and sparse GP surrogates perform better, worse, and not significantly different compared to the TGP-MO surrogates is indicated by "+", "-" and "≈", respectively. The full GP runs failed due to memory overflow in the instances marked "—". The approaches' performance is ranked by the color code green, yellow, and red in the order of best to worst.

Sample size	Sampling strategy	Metric	Surrogate type		
			Full GP + / - / ≈	Sparse GP + / - / ≈	TGP-MO +
2000	LHS	HV	9 / 9 / 30	4 / 38 / 6	9
		RMSE	43 / 2 / 3	19 / 21 / 8	1
		Time (s)	0 / 48 / 0	1 / 46 / 1	46
	MVNS	HV	17 / 14 / 17	14 / 28 / 6	12
		RMSE	34 / 7 / 7	20 / 16 / 12	2
		Time (s)	0 / 48 / 0	1 / 47 / 0	47
10000	LHS	HV	—	4 / 33 / 11	33
		RMSE	—	19 / 23 / 6	23
		Time (s)	—	0 / 48 / 0	48
	MVNS	HV	—	11 / 29 / 8	29
		RMSE	—	21 / 16 / 11	16
		Time (s)	—	0 / 48 / 0	48
50000	LHS	HV	—	3 / 35 / 10	35
		RMSE	—	15 / 24 / 9	24
		Time (s)	—	0 / 48 / 0	48
	MVNS	HV	—	10 / 29 / 9	29
		RMSE	—	23 / 19 / 6	19
		Time (s)	—	0 / 48 / 0	48

To compare the performance of the different approaches, we conducted a pairwise Wilcoxon two-tailed significance test. The calculated p-values were Bonferonni corrected, and $\alpha = 0.05$ was considered for rejecting the null hypothesis (an approach is not significantly better or worse than another approach). For determining whether an approach is significantly better or worse than another one, we compare their median values provided that the p-value is less than α. For ranking the approaches, we used a pairwise comparison of the approaches with a scoring system. An approach is given a score of $+1$ if it is significantly better than the other approach. A score of -1 is given to the approach if it is significantly worse than the other approach. If the approach is not significantly better or worse than the other approach, a score of zero is given to both approaches. The sum of the scores is used for ranking all the approaches (a higher score gives a better rank) for the metric being compared. A rank of '1' indicates that an approach has performed significantly better than all other approaches. Equal ranks indicate that those approaches are not significantly different in their performance.

In Table 2 we summarize the approaches' performance rankings by the colour code green, yellow, and red in the order of best to worst for three different metrics. It should be noted that these rankings are categorized with respect to the sample size and sampling strategies. The total number of instances is 96 (48 for LHS and MVNS each) for each sample size. Each instance consists of the combination of the various problem settings, i.e. the number of samples (N), sampling strategy, number of objectives (K), number of decision variables (n) and problem configuration. The number of instances in which full GPs or sparse GPs perform better, worse, or not significantly differently compared to the proposed TGP-MO surrogates is denoted by "+", "−" and "≈" respectively. For TGP-MO surrogates, we only show the number of instances where it performed significantly better than both full GPs and sparse GPs (or it ranked the best).

The rows measuring "Time" in Table 2 show that the proposed TGP-MO surrogates was computationally the cheapest compared to the full GP and sparse GP for different sample sizes and sampling strategies. The number of instances TGP-MO performed significantly better than the other two surrogates in each sub-category is close to 48(that was the total number of instances in each sub-category). The TGP-MO surrogates performed better than sparse GP surrogates in hypervolume for all sample sizes and sampling strategies. However, full GPs performed the best in hypervolume for sample size of 2000. For 2000 samples sparse GPs outperformed TGP-MO surrogates for both sampling strategies in RMSE and full GP performed the best. The RMSE of the solutions for TGP-MO surrogates was better than sparse GPs for samples sizes of 10000 and 50000 for LHS sampling only. Whereas, for MVNS sampling the sparse GP slightly outperformed TGP-MO surrogates for 10000 and 50000 samples.

For a smaller sample size one may choose full GPs as they give the best performance in hypervolume and RMSE. However, full GPs becomes almost impossible to build due to their high computation cost when the sample size is large. One can use TGP-MO surrogates for larger sample sizes as it performs better in hypervolume and RMSE and excellently in computation time. However, for non-uniform sampling strategies, i.e. MVNS, TGP-MO surrogates suffers

TABLE 3: Comparison of selected test instances showing the median hypervolume, RMSE and time and their standard deviation (in italics) of the 31 test runs. The best performing approaches are shown in bold.

Sample size	Sampling strategy	Problem	K	n	Hypervolume			RMSE			Time (s)		
					Full GP	Sparse GP	TGP-MO	Full GP	Sparse GP	TGP-MO	Full GP	Sparse GP	TGP-MO
2000	LHS	P3	3	5	**7.54E+01** *4.02E+00*	5.49E+01 *9.35E+00*	**7.48E+01** *2.73E+00*	**4.18E-02** *2.74E-01*	1.04E+00 *5.72E-01*	**2.67E-01** *1.38E-01*	1.36E+02 *1.85E+01*	8.64E+01 *2.16E+01*	**1.40E+01** *1.20E+01*
		P4	5	10	1.46E+04 *6.73E+02*	1.27E+04 *1.23E+03*	**1.64E+04** *5.59E+02*	1.75E+00 *3.69E-01*	1.79E+00 *4.38E-01*	**9.33E-01** *1.66E-01*	6.16E+02 *1.40E+02*	5.51E+02 *1.43E+02*	**2.62E+01** *1.12E+01*
		P3	7	5	**9.21E+06** *1.00E+05*	6.55E+06 *1.24E+06*	**9.23E+06** *9.32E+04*	**5.11E-02** *8.26E-02*	2.16E+00 *9.70E-01*	1.66E-01 *1.11E-01*	3.20E+02 *5.38E+01*	2.10E+02 *3.46E+01*	**2.83E+01** *1.53E+01*
	MVNS	P3	3	10	5.71E+01 *4.14E+00*	5.65E+01 *6.35E+00*	**6.27E+01** *5.10E+00*	**1.08E+00** *4.65E-01*	1.52E+00 *6.09E-01*	**8.55E-01** *1.99E-01*	3.25E+02 *6.76E+01*	3.56E+02 *1.19E+02*	**1.31E+01** *2.45E+00*
		P2	5	10	**1.26E+04** *2.08E+03*	1.06E+04 *2.10E+03*	**1.30E+04** *1.20E+03*	**1.86E+00** *1.95E-01*	2.76E+00 *1.59E-01*	2.04E+00 *2.78E-01*	5.34E+02 *1.01E+02*	5.51E+02 *7.64E+01*	**1.39E+01** *8.00E+00*
		P2	7	7	7.81E+06 *5.36E+05*	6.42E+06 *2.62E+05*	**8.82E+06** *5.30E+05*	**7.80E-01** *2.07E-01*	3.76E+00 *8.55E-01*	9.80E-01 *2.38E-01*	3.57E+02 *4.40E+01*	4.36E+02 *6.32E+01*	**1.27E+01** *1.69E+01*
10000	LHS	P3	3	5	—	6.26E+01 *9.78E+00*	**7.58E+01** *5.93E-01*	—	8.95E-01 *4.85E-01*	**1.50E-01** *5.05E-02*	—	1.45E+03 *2.22E+02*	**2.49E+01** *8.80E+00*
		P4	5	10	—	1.27E+04 *6.09E+02*	**1.65E+04** *8.26E+02*	—	2.10E+00 *2.22E-01*	**7.09E-01** *2.28E-01*	—	6.52E+03 *1.24E+03*	**5.93E+01** *2.11E+01*
		P1	7	10	—	6.71E+06 *1.90E+05*	**8.59E+06** *2.61E+05*	—	1.56E+00 *1.84E-01*	**9.73E-01** *8.82E-02*	—	9.37E+03 *1.38E+03*	**3.78E+01** *1.32E+01*
	MVNS	P3	3	10	—	4.97E+01 *5.40E+00*	**6.27E+01** *3.31E+00*	—	1.68E+00 *3.40E-01*	**9.09E-01** *1.52E-01*	—	4.12E+03 *6.69E+02*	**1.42E+01** *1.52E+00*
		P3	5	5	—	1.23E+04 *2.50E+03*	**1.75E+04** *3.15E+02*	—	2.32E+00 *1.10E+00*	**1.83E-01** *1.22E-01*	—	2.38E+03 *4.65E+02*	**4.73E+01** *1.19E+01*
		P3	7	5	—	6.55E+06 *1.22E+06*	**9.24E+06** *1.01E+05*	—	2.91E+00 *1.34E+00*	**1.22E-01** *1.42E-01*	—	3.51E+03 *6.54E+02*	**8.94E+01** *3.02E+01*
50000	LHS	P3	3	10	—	6.04E+01 *7.28E+00*	**7.30E+01** *2.62E+00*	—	8.29E-01 *6.08E-01*	**3.91E-01** *8.63E-02*	—	2.16E+04 *1.77E+03*	**3.16E+01** *9.25E+00*
		P3	5	10	—	1.45E+04 *4.10E+02*	**1.65E+04** *4.04E+02*	—	8.64E-01 *2.10E-01*	**6.18E-01** *1.20E-01*	—	3.59E+04 *6.29E+03*	**3.90E+01** *1.10E+01*
		P1	7	10	—	7.49E+06 *1.12E+05*	**8.65E+06** *1.65E+05*	—	9.20E-01 *9.23E-02*	**7.44E-01** *8.59E-02*	—	5.05E+04 *9.41E+03*	**8.77E+01** *3.44E+01*
	MVNS	P3	3	5	—	5.27E+01 *1.14E+01*	**7.59E+01** *4.41E-01*	—	1.77E+00 *9.11E-01*	**9.73E-02** *5.04E-02*	—	8.19E+03 *6.35E+02*	**5.29E+01** *2.07E+01*
		P3	5	5	—	1.25E+04 *2.04E+03*	**1.76E+04** *3.03E+02*	—	2.73E+00 *1.13E+00*	**1.43E-01** *1.49E-01*	—	1.34E+04 *7.96E+02*	**8.68E+01** *4.14E+01*
		P3	7	5	—	6.55E+06 *1.05E+06*	**9.26E+06** *1.18E+05*	—	3.75E+00 *1.45E+00*	**9.81E-02** *1.36E-01*	—	1.87E+04 *2.24E+03*	**1.91E+02** *7.59E+01*

slightly in RMSE compared to sparse GP surrogates. This is because in sparse GP surrogates, the variational parameters are selected by minimizing the KL divergence. Thus, the inducing inputs selected are not skewed even if the provided dataset is i.e., in MVNS sampling. Therefore the RMSE is slightly better than TGO-MO surrogates that does not have an ability to handle skewed datasets.

The performances of a few selected instances are shown in Table 3. The table shows the median and the standard deviation of hypervolume, RMSE, and time taken to build the surrogates for the three different approaches for 31 runs. The instances shown in the table were chosen based on the maximum difference between the median hypervolume of the best and the second best performing surrogates. One instance was selected from each sample size, sampling strategy and number of objectives. The figures in bold represent the best performing approaches. We can observe that the proposed TGP-MO surrogates performed the best in building time in the instances shown. We can also also observe an improved hypervolume and RMSE for sample sizes of 10000 and 50000 compared to sparse GP surrogates. It can be observed that the building time of TGP-MO surrogates are lesser than sparse GP by an order of about 10^2 and 10^3 for samples size of 10000 and 50000 respectively. The building time also increases with the number of objectives for sparse GP surrogates.

The quantity of data available at the tradeoff region affects the accuracy and hypervolume of the solutions obtained that is a general challenge while solving offline data-driven MOPs. The proposed TGP-MO surrogates split the decision space into sub-regions and approximate the underlying objectives at the tradeoff region. This is best suitable when the Pareto set is located in a smaller region of the decision space. When the Pareto set is in a larger region, the prediction of TGP-MO surrogates is from multiple leaf node GPs and has discontinuities near the splits of the regression trees. Therefore the approximated Pareto front has some discontinuities compared to sparse GPs or full GPs. Further

tests with the DTLZ [27] benchmark problems are given in the supplementary material.

5. Conclusions

In this paper, we proposed surrogates that can be built with lesser computational cost compared to sparse GPs and full GPs for solving offline data-driven MOPs when dealing with large datasets. The proposed TGP-MO surrogates performed significantly better than sparse GP surrogates in hypervolume, RMSE, and computational time for most of the instances for DBMOPP problems. The full GP surrogates failed to complete the runs for larger sample sizes due to memory restrictions. Thus it can be concluded that the proposed TGP-MO surrogates are best suited for solving offline data-driven MOPs with a large size dataset.

A GP at a leaf node of a tree approximates certain regions of the decision space. This feature can be exploited while solving MOPs with preferences from the decision maker in an a priori or interactive fashion. While building the GPs at the leaf nodes, we can utilize the provided preferences for objectives and approximate only solutions that follow the preferences. Developing an interactive framework utilizing the TGP-MO surrogates will our one of our future works.

We will also test and compare the uncertainty prediction provided by the proposed TGP-MO surrogates while solving offline MOPs. Sensitivity analysis of the hyperparameters, especially the minimum number of leaf node samples will also be tested. One of the major drawbacks of the TGP-MO surrogates are the discontinuity between the leaf node GPs. Tackling discontinuity in the prediction [20] at the partition of the decision space (as provided by the tree) will be a future task. Further improvements can be made in the way the trees partition the decision space. Instead of using a traditional loss function, we can formulate a non-linear tradeoff criteria to split the nodes. This will enable the TGP-MO surrogates to approximate the tradeoff region with fewer number of samples. Testing the TGP-MO surrogates for solving real life offline data-driven MOPs will be a future task.

Acknowledgements

This research was partly supported by the Academy of Finland (grant number 311877) and is related to the thematic research area DEMO (Decision Analytics utilizing Causal Models and Multiobjective Optimization, `http://www.jyu.fi/demo`) of the University of Jyväskylä. M. López-Ibáñez is a "Beatriz Galindo" Senior Distinguished Researcher (BEAGAL 18/00053) funded by the Spanish Ministry of Science and Innovation (MICINN). This research is partially funded by TAILOR ICT-48 Network (No 952215) funded by EU Horizon 2020 research and innovation programme.

References

[1] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, pp. 61–70, 2011.

[2] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Transactions on Evolutionary Computation*, vol. 23, pp. 442–458, 2019.

[3] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, pp. 148–175, 2016.

[4] H. Wang, Y. Jin, C. Sun, and J. Doherty, "Offline data-driven evolutionary optimization using selective surrogate ensembles," *IEEE Transactions on Evolutionary Computation*, vol. 23, pp. 203–216, 2019.

[5] H. Wang, Y. Jin, and J. O. Jansen, "Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system," *IEEE Transactions on Evolutionary Computation*, vol. 20, pp. 939–952, 2016.

[6] H. Wang and Y. Jin, "A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems," *IEEE Transactions on Cybernetics*, vol. 50, pp. 536–549, 2020.

[7] C. Yang, J. Ding, Y. Jin, and T. Chai, "Offline data-driven multiobjective optimization: Knowledge transfer between surrogates and generation of final solutions," *IEEE Transactions on Evolutionary Computation*, vol. 24, pp. 409–423, 2020.

[8] A. Mazumdar, T. Chugh, K. Miettinen, and M. López-Ibáñez, "On dealing with uncertainties from Kriging models in offline data-driven evolutionary multiobjective optimization," in *Evolutionary Multi-Criterion Optimization, Proceedings*, K. Deb, E. Goodman, C. A. Coello Coello, K. Klamroth, K. Miettinen, S. Mostaghim, and P. Reed, Eds. Springer, 2019, pp. 463–474.

[9] A. Mazumdar, T. Chugh, J. Hakanen, and K. Miettinen, "An interactive framework for offline data-driven multiobjective optimization," in *Bioinspired Optimization Methods and Their Applications, Proceedings*, B. Filipič, E. Minisci, and M. Vasile, Eds. Springer, 2020, pp. 97–109.

[10] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms — A comparative case study," in *Parallel Problem Solving from Nature*, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds. Springer, 1998, pp. 292–301.

[11] A. Forrester, A. Sobester, and A. Keane, *Engineering Design via Surrogate Modelling*. John Wiley & Sons, 2008.

[12] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[13] A. A. M. Rahat, C. Wang, R. M. Everson, and J. E. Fieldsend, "Data-driven multi-objective optimisation of coal-fired boiler combustion systems," *Applied Energy*, vol. 229, pp. 446–458, 2018.

[14] E. J. Hughes, "Evolutionary multi-objective ranking with uncertainty and noise," in *Evolutionary Multi-Criterion Optimization, Proceedings*, E. Zitzler, L. Thiele, K. Deb, C. A. Coello Coello, and D. Corne, Eds. Springer, 2001, pp. 329–343.

[15] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Proceedings of the 18th International Conference on Neural Information Processing Systems*. MIT Press, 2005, pp. 1257–1264.

[16] M. K. Titsias, "Variational learning of inducing variables in sparse gaussian processes," in *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, D. van Dyk and M. Welling, Eds. PMLR, 2009, pp. 567–574.

[17] M. Emmerich, K. Giannakoglou, and B. Naujoks, "Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 421–439, 2006.

[18] H.-M. Kim, B. K. Mallick, and C. C. Holmes, "Analyzing nonstationary spatial data using piecewise Gaussian processes," *Journal of the American Statistical Association*, vol. 100, pp. 653–668, 2005.

[19] K. Das and A. N. Srivastava, "Block-GP: Scalable gaussian process regression for multimodal data," in *2010 IEEE International Conference on Data Mining*, 2010, pp. 791–796.

[20] B. van Stein, H. Wang, W. Kowalczyk, T. Bäck, and M. Emmerich, "Optimally weighted cluster kriging for big data regression," in *Advances in Intelligent Data Analysis XIV*, E. Fromont, T. De Bie, and M. van Leeuwen, Eds. Springer, 2015, pp. 310–321.

[21] R. B. Gramacy and H. K. H. Lee, "Bayesian treed Gaussian process models with an application to computer modeling," *Journal of the American Statistical Association*, vol. 103, pp. 1119–1130, 2008.

[22] H. A. Chipman, E. I. George, and R. E. McCulloch, "Bayesian CART model search," *Journal of the American Statistical Association*, vol. 93, pp. 935–948, 1998.

[23] J.-A. M. Assael, Z. Wang, B. Shahriari, and N. de Freitas, "Heteroscedastic treed Bayesian optimisation," *CoRR*, p. arXiv:1410.7172, 2014.

[24] W.-Y. Loh, "Classification and regression trees," *WIREs Data Mining and Knowledge Discovery*, vol. 1, pp. 14–23, 2011.

[25] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms," *Soft Computing*, vol. 23, pp. 3137–3166, 2019.

[26] J. E. Fieldsend, T. Chugh, R. Allmendinger, and K. Miettinen, "A feature rich distance-based many-objective visualisable test problem generator," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2019, pp. 541–549.

[27] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, A. Abraham, L. Jain, and R. Goldberg, Eds. Springer, 2005, pp. 105–145.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[29] GPy, "GPy: A gaussian process framework in python," http://github.com/SheffieldML/GPy, 2012, accessed July 1, 2021.

[30] J. E. Fieldsend, T. Chugh, R. Allmendinger, and K. Miettinen, "A feature rich distance-based many-objective visualisable test problem generator," https://github.com/fieldsend/DBMOPP_generator, 2019, accessed November 1, 2021.

[31] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, pp. 773–791, 2016.

[32] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, pp. 712–731, 2007.

[33] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, pp. 577–601, 2014.

[34] W. Chapman, W. Welch, K. Bowman, J. Sacks, and J. Walsh, "Arctic sea ice variability: Model sensitivities and a multidecadal simulation," *Journal of Geophysical Research: Oceans*, vol. 99, pp. 919–935, 1994.

[35] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, pp. 455–492, 1998.

Supplementary Material: TGP-MO: Treed Gaussian Processes for Solving Offline Data-Driven Multiobjective Optimization Problems

Atanu Mazumdar
University of Jyvaskyla
Faculty of Information Technology
P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla
Finland

Manuel López-Ibáñez
University of Málaga
School of Computer Science
Bulevar Louis Pasteur 35, Campus de Teatinos,
29071 Málaga, Spain

Tinkle Chugh
Department of Computer Science
University of Exeter, UK

Jussi Hakanen
University of Jyvaskyla
Faculty of Information Technology
P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla
Finland

Kaisa Miettinen
University of Jyvaskyla
Faculty of Information Technology
P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla
Finland

1. Building Process of TGP-MO Surrogates

The building process of TGP-MO surrogates are illustrated in Figure 1. The figures show the solutions obtained by the MOEA in some of the iterations of Algorithm 1 (in the main manuscript) for two bi-objective test problems. The two problems were DTLZ2 ($n = 10$) Figure 1a-1c from the DTLZ [1] test suite and a distance-based visualizable test problem (DBMOPP) [2] ($n = 2$) Figure 1d-1f. The problem configuration for the DBMOPP problem was the same as problem P1 that can be found in Table 1 (in the main manuscript). For both the cases we used a sample size (size of the data) of $N = 2000$ with Latin hypercube (LHS) sampling. The first column shows the solutions obtained by the MOEA in the first iteration before any GPs are built at the leaf nodes of the trees. As the objective values of the solutions are predicted by the regression trees, the obtained solutions do not form a smooth Pareto front. In the second column, we can observe the solutions obtained after four more iterations. Few GPs are built at the leaf nodes, and the approximated Pareto front gradually becomes more smooth. At this iteration, the objective values of the solutions are predicted by the regression trees and GPs at leaves for either or both the objectives. In the last column, we show the final iteration of the surrogate building process, when all solutions in the MOEA's population are predicted by leaf nodes that contain GPs. Building the surrogates for the DTLZ2 problem consumes more iterations than the DBMOPP problem instance. This behavior is due to the Pareto set of DTLZ problems that is located in a larger region of the decision space. Hence, more GPs are required to be built at the leaves of the trees to approximate the tradeoff region accurately.

In Figure 2 we show the total number of samples utilized to build TGP-MO surrogates with the number of iterations for six different problem instances with sample sizes of 2000, 10000, and 50000. The solid line shows the mean number of samples used, and the shaded region denotes the 95% confidence interval of the runs for each treed GP surrogate (for each objective). The plots have been extended in the iteration axis to the maximum allowed iterations, $I^{\max} = \frac{N}{10n}$. The iteration axis is broken when $I^{\max}$ is large to reduce the width of the plots. It can be observed that the number of samples utilized converged before the maximum iterations and varies with the objective. Certain problem instances required more samples during the building process than others, especially due to the number of decision variables and the characteristics of the problem. If the tradeoff region is located in a smaller region of the decision space, the building process converges quickly and therefore consumes less samples.

2. Tests with DTLZ Problems

Further tests were conducted with the DTLZ benchmark problems. The setup for the tested surrogates, i.e. full GPs, sparse GPs and TGP-MO surrogates were kept the same as mentioned in the main manuscript. The setup for the multiobjective evolutionary algorithm (MOEA) were also

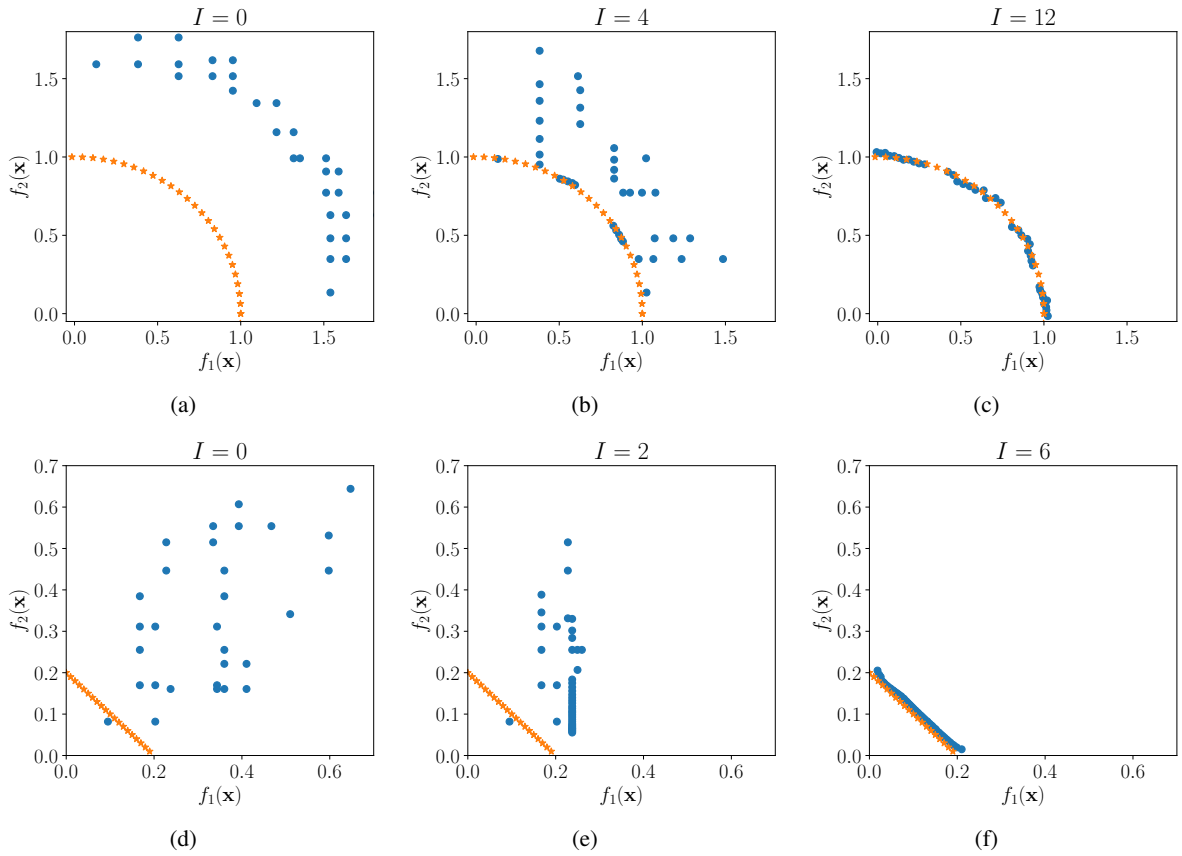


Figure 1: Progress of the building process of TGP-MO surrogates for a bi-objective DTLZ2 (a)–(c) and a DBMOPP (d)–(f) problem. The plots show the solutions of the MOEA at different iterations I of the building process. The Pareto front of the underlying MOP is shown in orange '$\star$'.

kept the same. The problems used were DTLZ2 and DTLZ4-7 with $K \in (3, 5, 7)$. For DTLZ problems the number of decision variables considered were $n = K + 1$. Therefore, we tested fewer problem instances for DTLZ compared to DBMOPP problems. The total number of problem instances were 60 (30 for LHS and MVNS each). DTLZ1 and DTLZ3 were not included in the tests as they have local optimal fronts. Thus it is extremely difficult to solve especially when the surrogates cannot be updated as in offline data-driven MOPs. For MVNS sampling, the mean of the distribution for DTLZ instances was set to 0.5 with a variance of 0.1 for all objectives. The reference points for calculating the hypervolume of solutions for DTLZ instances are provided in Table 1.

Table 2 shows that sparse GP performed better than TGP-MO surrogates for all sample sizes and sampling strategies in hypervolume. Full GP surrogates performed the best in hypervolume for sample size of 2000 for both sampling strategies. Sparse GP also outperformed TGP-MO surrogates for sample sizes of 10000 and 50000. However, TGP-MO surrogates outperformed sparse GP for 2000 samples in RMSE. The poor performance of TGP-MO surro-

TABLE 1: Reference point for calculating hypervolume for different test cases of DTLZ instances.

Problem	**Objectives** (K)	**Reference point**
DTLZ2	3	(6, 6, 6)
	5	(6, 6, 6, 6, 6)
	7	(6, 6, 6, 6, 6, 6, 6)
DTLZ4	3	(6, 6, 6)
	5	(6, 6, 6, 6, 6)
	7	(6, 6, 6, 6, 6, 6, 6)
DTLZ5	3	(6, 6, 6)
	5	(6, 6, 6, 6, 6)
	7	(6, 6, 6, 6, 6, 6, 6)
DTLZ6	3	(20, 20, 20)
	5	(20, 20, 20,20, 20)
	7	(20, 20, 20, 20, 20, 20, 20)
DTLZ7	3	(1, 1, 40)
	5	(1, 1, 1, 1, 50)
	7	(1, 1, 1, 1, 1, 1, 70)

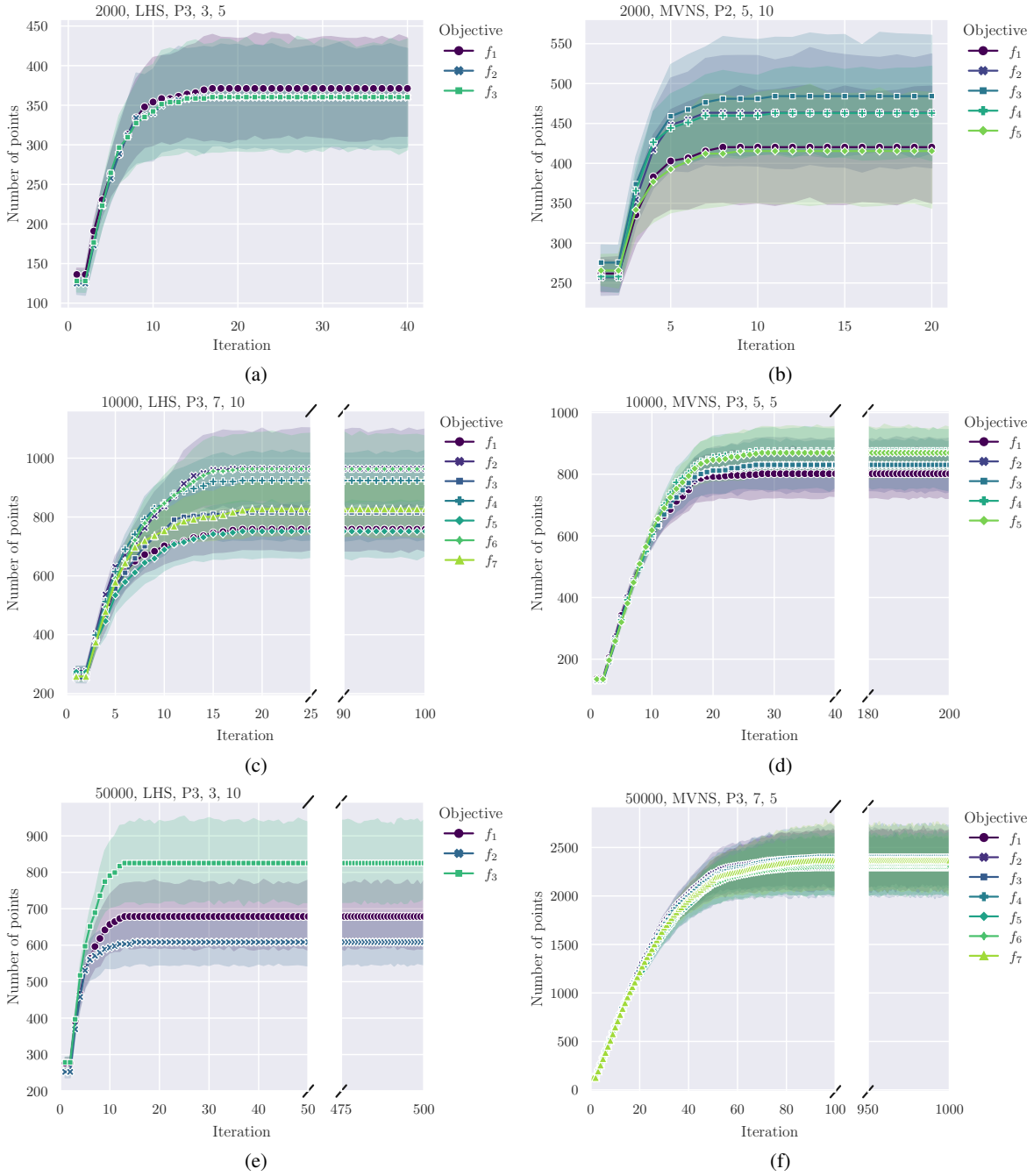


Figure 2: The total number of samples utilized (median and 95% confidence interval) to build GPs at leaf nodes with iterations for three different DBMOPP problem instances as displayed according to the following format (N, sampling strategy, problem, K, n). The iteration axis is extended to the maximum possible iteration $\frac{N}{10n}$. The iteration axis for plots 2c-2f are broken to accommodate the plots.

gates compared to sparse GP is due to the nature of the Pareto set for DTLZ benchmark problems. As the Pareto set is not locally situated in the decision space (like the DBMOPP problems), the TGP-MO surrogates require many leaf nodes GPs to approximate the tradeoff region. Therefore the final approximated Pareto front has some discontinuities compared to while using sparse GP (or full GP) surrogates that approximates the global function landscape.

TABLE 2: Summary of pairwise comparison of the hypervolume, RMSE, and time (seconds) for the different approaches for DTLZ problems. The number of instances in which full GP and sparse GP surrogates perform better, worse, and not significantly different compared to the TGP-MO surrogates is indicated by "+", "-" and "≈", respectively. The full GP runs failed due to memory overflow in the instances marked "—". The approaches' performance is ranked by the color code in the order of best to worst green, yellow, and red.

Sample size	Sampling strategy	Metric	Surrogate type		
			Full GP + / - / ≈	Sparse GP + / - / ≈	TGP-MO +
2000	LHS	HV	18 / 4 / 8	18 / 7 / 5	3
		RMSE	21 / 5 / 4	13 / 15 / 2	5
		Time (s)	0 / 30 / 0	2 / 28 / 0	28
	MVNS	HV	18 / 7 / 5	16 / 8 / 6	6
		RMSE	20 / 5 / 5	10 / 15 / 5	3
		Time (s)	0 / 30 / 0	2 / 27 / 1	27
10000	LHS	HV	—	14 / 9 / 7	9
		RMSE	—	16 / 13 / 1	13
		Time (s)	—	0 / 30 / 0	30
	MVNS	HV	—	13 / 8 / 9	8
		RMSE	—	15 / 11 / 4	11
		Time (s)	—	0 / 30 / 0	30
50000	LHS	HV	—	18 / 6 / 6	6
		RMSE	—	18 / 7 / 5	7
		Time (s)	—	0 / 30 / 0	30
	MVNS	HV	—	14 / 7 / 9	7
		RMSE	—	16 / 11 / 3	11
		Time (s)	—	0 / 30 / 0	30

References

[1] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, A. Abraham, L. Jain, and R. Goldberg, Eds. Springer, 2005, pp. 105–145.

[2] J. E. Fieldsend, T. Chugh, R. Allmendinger, and K. Miettinen, "A feature rich distance-based many-objective visualisable test problem generator," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2019, pp. 541–549.