

**METAMODELING IN EVOLUTIONARY MULTI-OBJECTIVE
OPTIMIZATION FOR CONSTRAINED AND
UNCONSTRAINED PROBLEMS**

By

Rayan Hussein

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering - Doctor of Philosophy

2022

ABSTRACT

**METAMODELING IN EVOLUTIONARY MULTI-OBJECTIVE
OPTIMIZATION FOR CONSTRAINED AND
UNCONSTRAINED PROBLEMS**

By

Rayan Hussein

One of the main difficulties in applying an optimization algorithm to a practical problem is that the evaluation of objectives and constraints often involve computationally expensive procedures. To handle such problems, a metamodel (or surrogate model, or response surface approximations) is first formed from a few exact (high-fidelity) solution evaluations, and then optimized by an algorithm in a progressive manner. However, there has been lukewarm interest in finding multiple trade-off solutions for multi-objective optimization problems using surrogate models. The literature on surrogate modeling for constrained optimization problems is also rare. The difficulty lies in the requirement of building and solving multiple surrogate models, one for each Pareto-optimal solution.

In this study, we propose a taxonomy of different possible metamodeling frameworks for multi-objective optimization and provide a comparative study by discussing advantages and disadvantages of each framework. Also, we argue that it is more efficient to use different metamodeling frameworks at different stages of the optimization process. Thereafter, we propose a novel adaptive method for switching among different metamodeling frameworks. Moreover, we observe the convergence behavior of the proposed approaches is better with a trust regions method applied within the metamodeling frameworks.

The results presented in this study are obtained using the well-known Kriging metamodeling approach. Based on our extensive simulation studies on proposed frameworks, we

report new and interesting observations about the behavior of each metamodeling framework, which may provide salient guidelines for further studies in this emerging area within evolutionary multi-objective optimization. Results of this study clearly show the efficacy and efficiency of the proposed adaptive switching approach compared to three recently-proposed other metamodeling algorithms on challenging multi-objective optimization problems using a limited budget of high-fidelity evaluations.

To my Beloved Family

ACKNOWLEDGMENTS

I would like to thank my family especially my wife, Salma Qasim, my children, and my parents for many things that go beyond listing.

I would also like to express my sincere gratitude to my advisor Prof. Kalyanmoy Deb for his support, patience, motivation and immense knowledge. I learnt from him a lot and could not have imagined a better advisor.

A big "Thank You!" also goes to all my COIN lab colleagues. It has been an honor working with each of them. A PhD is a lot easier when you are surrounded with friends.

Last but not least, I would like to thank the Higher Committee for Education Development (HCED) in Iraq, and Beacon Center for making my graduate studies financially feasible.

TABLE OF CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ALGORITHMS	xii
Chapter 1 Introduction	1
1.1 Overview	1
1.1.1 Metamodels (Surrogate Models)	1
1.1.2 Kriging	2
1.1.3 Evolutionary multiobjective optimization (EMO)	4
1.1.4 Achievement Scalarization Function (ASF) Method	6
1.2 Summary of Research Contributions	8
1.2.1 A Generative Kriging Surrogate Model	9
1.2.2 A Taxonomy for Metamodeling Frameworks for Evolutionary Multi-Objective Optimization	9
1.2.3 Adaptive Switching Between Metamodeling Frameworks	10
1.3 Dissertation Organization	11
Chapter 2 A Taxonomy for Metamodeling Frameworks for Evolutionary Multi-Objective Optimization	13
2.1 Introduction	13
2.2 The Proposed Taxonomy	16
2.2.1 A Taxonomy for Multi-objective Constraint Problems	16
2.2.2 A Taxonomy for Multi-Objective Unconstraint Problems	20
2.2.3 A Taxonomy for Single-Objective Constraint Problems	21
2.3 Summary	22
Chapter 3 Metamodeling Frameworks M1-1, M2-1, M1-2, M3-1, M4-1, M5, and M6	23
3.1 Metamodeling Frameworks M1-1 and M2-1	23
3.2 Metamodeling Frameworks M1-2	25
3.3 Metamodeling Frameworks M3-1 and M4-1	27
3.4 Metamodeling Framework M5	30
3.4.1 The Expected Improvement Procedure	32
3.4.2 Proposed Multi-objective EGO Method	35
3.4.3 Diversity_Preserver Procedure	36
3.4.4 Optimization Procedure	38
3.5 Metamodeling Framework M6	39
3.6 Summary	40

Chapter 4	Results and Comparison Between Metamodeling Frameworks	42
4.1	Two-Objective Unconstrained Problems	43
4.2	Two-Objective Constrained Problems	49
4.3	Three-Objective Constrained and Unconstrained Problems	52
4.4	Five-Objective Constrained and Unconstrained Problems	56
4.5	Summary	58
Chapter 5	Adaptive Switching Between Frameworks and Use of Trust Regions	62
5.1	Introduction	62
5.2	Literature Review	63
5.3	Trust Regions	65
5.4	Results of Trust Region	67
5.4.1	Two-Objective Unconstrained Problems	68
5.4.2	Two-Objective Constrained Problems	69
5.5	Adaptive Switching between Frameworks	70
5.5.1	Performance Metric for Framework Selection	71
5.5.2	Identifying a Suitable Framework for Next Epoch	73
5.6	Results	74
5.6.1	Two-Objective Unconstrained Problems	76
5.6.2	Two-Objective Constrained Problems	76
5.6.3	Three-Objective Constrained and Unconstrained Problems	79
5.6.4	Five-Objective Constrained and Unconstrained Problems	81
5.6.5	Summary of Results	84
5.6.6	Comparative Study	84
5.7	Summary	85
Chapter 6	Future Studies	88
Chapter 7	Conclusions	90
BIBLIOGRAPHY		92

LIST OF TABLES

Table 2.1:	Classification of existing metamodeling-based multi-objective optimization studies into six frameworks of this study. Framework M6 is proposed for the first time here.	15
Table 3.1:	Number of metamodels needed for each of the six metamodeling frameworks.	41
Table 4.1:	Computed IGD values for test problems. Best performing framework and other statistically similar frameworks are marked in bold.	48
Table 4.2:	Computed GD values for test problems. Best performing framework and other statistically similar frameworks are marked in bold.	49
Table 4.3:	Computed IGD values for ZDT problems. Best performing framework and other statistically similar frameworks are marked in bold.	50
Table 4.4:	Computed GD values for ZDT problems. Best performing framework and other statistically similar frameworks are marked in bold.	50
Table 5.1:	Computed IGD values for test problems. Best performing are marked in bold. W/TR: Without Trust Region.	70
Table 5.2:	Parameter settings used in the study.	75
Table 5.3:	Computed IGD values for test problems. Best performing framework is marked in bold and other statistically similar frameworks are marked in bold and italic.	86
Table 5.4:	Ranking of frameworks summarized. A ‘1’ indicates the best performing framework.	87
Table 5.5:	Mean IGD comparison among G-ASM, MOEA/D-EGO, K-RVEA, and CSEA algorithms on unconstrained problems.	87

LIST OF FIGURES

Figure 1.1:	A principle of set based multi-objective optimization procedure is illustrated.	6
Figure 1.2:	ASF procedure of finding a Pareto-optimal solution is illustrated. . .	7
Figure 2.1:	The proposed taxonomy of six different metamodeling frameworks for multiple and many-objective optimization.	17
Figure 2.2:	Degenerated taxonomy for multi-objective unconstrained optimization.	20
Figure 2.3:	Degenerated taxonomy for single-objective optimization for finding a single optimal solution.	21
Figure 3.1:	The proposed surrogate-based generative EMO.	37
Figure 4.1:	Obtained non-dominated solutions for problem ZDT1 using frameworks M1-2, M3, M5 and M6 from left to right.	44
Figure 4.2:	Obtained non-dominated solutions for problem ZDT2 using frameworks M1-2, M3, M5 and M6 from left to right.	45
Figure 4.3:	Obtained non-dominated solutions for problem ZDT3 using frameworks M1-2, M3, M5 and M6 from left to right.	46
Figure 4.4:	Obtained non-dominated solutions for problem ZDT6 using frameworks M1-2, M3, M5 and M6 from left to right.	47
Figure 4.5:	Obtained non-dominated solutions for BNH using frameworks M1-2, M2-2, M3, M4, M5 and M6 from left to right and top to bottom. . .	51
Figure 4.6:	Obtained non-dominated solutions for the SRN problem using frameworks M1-2, M2-2, M3, M4, M5 and M6 from left to right and top to bottom.	52
Figure 4.7:	Obtained non-dominated solutions for the TNK problem using frameworks M1-2, M2-2, M3, M4, M5 and M6 from left to right and top to bottom.	53

Figure 4.8:	Obtained non-dominated solutions for OSY problem using frameworks M1, M2, M3, M4, M5 and M6 from left to right and top to bottom.	54
Figure 4.9:	Obtained non-dominated solutions for the welded-beam problem using frameworks M1-2, M2-2, M3, M4, M5 and M6 from left to right and top to bottom.	55
Figure 4.10:	Obtained non-dominated solutions for problem DTLZ2 using frameworks M1-2, M3, M5 and M6 from left to right.	56
Figure 4.11:	Obtained non-dominated solutions for problem DTLZ4 using frameworks M1-2, M3, M5 and M6 from left to right.	57
Figure 4.12:	Obtained non-dominated solutions for problem DTLZ5 using frameworks M1-2, M3, M5 and M6 from left to right.	58
Figure 4.13:	Graphical results for problem C2DTLZ2 using frameworks M1-2, M3, M5 and M6 from left to right.	59
Figure 4.14:	Obtained non-dominated solutions for problem car-side impact using frameworks M1-2, M2-2, M3, M4 M5 and M6 from left to right and top to bottom.	60
Figure 5.1:	Proximal and Trust Region.	67
Figure 5.2:	Non-dominated solutions for ZDT1, ZDT2, ZDT3, and ZDT6 problems using framework M1-2 from left to right.	68
Figure 5.3:	Non-dominated solutions for TNK, OSY, BNH, and SRN problems using framework M1-2 from left to right.	69
Figure 5.4:	Flowchart of the proposed G-ASM method.	72
Figure 5.5:	Selection Error Probability (SEP) concept is illustrated.	73
Figure 5.6:	Obtained non-dominated solutions of the final archive for the median G-ASM run for ZDT problems.	77
Figure 5.7:	Adaptive switching method of six frameworks among average runs for two objectives unconstrained problems.	77
Figure 5.8:	Adaptive switching method of six frameworks for median IGD run for two objectives unconstrained problems.	78

Figure 5.9:	Obtained non-dominated solutions of the final archive for the median G-ASM run for two-objective constrained problems.	79
Figure 5.10:	Proportion of framework usage in 11 runs for two-objective constrained problems.	80
Figure 5.11:	Switching of frameworks for the median G-ASM run for SRN, BNH and welded beam problems.	80
Figure 5.12:	Obtained non-dominated solutions for Adaptive switching method of six frameworks for three objectives constrained and unconstrained problems.	81
Figure 5.13:	Proportion of framework usage in 11 runs for three and five-objective problems.	82
Figure 5.14:	Adaptive switching method of six frameworks for median IGD run for three and five objectives problems.	83

LIST OF ALGORITHMS

Algorithm 1	Metamodeling Frameworks M1-1 and M2-1	26
Algorithm 2	Metamodeling Frameworks M3-1 and M4-1	31
Algorithm 3	Metamodeling Framework M5	33

Chapter 1

Introduction

In this chapter, we provide a brief overview of metamodels (or surrogate models) and Evolutionary Multi-objective Optimization (EMO). Then, we highlight main research contributions, which include, a taxonomy of metamodeling frameworks, switching between frameworks, and use of a trust regions method to achieve a better convergence behavior.

1.1 Overview

1.1.1 Metamodels (Surrogate Models)

In most practical optimization problems, the evaluation of a solution involves computationally expensive softwares and procedures, which remain as one of the main bottlenecks of completing an optimization run within a reasonable amount of time. However, since solutions created at the beginning of an optimization simulation are expected to be far from the optimal region, an exact and precise evaluation of the solution may not be necessary. As the solutions progress towards the optimal region, a more precise evaluation may be invoked.

The recent developments of optimization methods have led to an increasing interest of approximation models or *surrogate* models [1, 2]. The use of metamodels (or surrogate models) to approximate the functional form of exact objectives and constraints by using a few high-fidelity solution evaluations is a common approach [3]. Among various methods,

the Kriging method is one of the widely used metamodels, which can provide an estimated function value and also simultaneously provide an error estimate of the approximation [4].

The optimization process is then continued using the surrogate model until the model is deemed to be accurate enough in the current region of focus. Due to this reason, surrogate models are effective in reducing the overall computational time required in real-world design optimization.

Despite the existence of numerous surrogate modeling studies in the context of single-objective optimization problems, there are not many efforts spent on extending the ideas to multi-objective optimization. The reasons for few studies [5,6] is that in multi-objective optimization, the aim is to find multiple trade-off Pareto-optimal solutions, instead of a single optimal solution. Most surrogate modeling studies have not considered modeling a multi-modal function in the spirit of locating multiple optimal solutions simultaneously. Hence, there is not much that can be borrowed from the single-objective literature. However, Emmerich et al. [5] have generalized the probability of improvement and the expected improvement concept to multi-objective optimization. Single-objective Evolutionary Algorithms (EA) are used for maximizing one of these metrics for determining which point should be evaluated next. Although one can select multiple test points with high-metric values at each iteration, these scalar metrics on their own could not predict the most likely shape and location of the whole Pareto-front (PF). Therefore, they may not be stochastically sound for locating multiple test points.

1.1.2 Kriging

Kriging, or the Gaussian process regression, has been one of the most popular choices in surrogate techniques used mainly because of its ability to provide uncertainty information

of the approximated values. The term Kriging was proposed by Matheron in 1963 [7] in honor of the South African mining engineer Danie G. Krige [8]. His research was focused on the distribution of gold samples found in mines and the correlation between these samples. He implemented a statistical technique based on a limited amount of samples, which is now known as Kriging. The first work in using Kriging for an approximation of simulation-based or computer experiments which was proposed in 1989 by Sacks et al. [9]. However, the most cited algorithm in using Kriging is efficient global optimization (EGO) proposed in 1998 by Jones et al. [10] for single-objective optimization problems. EGO uses a criterion called expected improvement (EI) to select samples for training the Kriging model.

The Kriging approach treats the function of interest as a realization of a random function (stochastic process) $y(\mathbf{x})$. For this reason, the mathematical model of the Kriging method has been presented as a linear combination of a global model, plus its departure:

$$y(\mathbf{x}) = f(\mathbf{x}) + Z(\mathbf{x}) \quad (1.1)$$

where $y(\mathbf{x})$ is the unknown deterministic response, $f(\mathbf{x})$ is a known function of $\mathbf{x}$, and $Z(\mathbf{x})$ is a realization of a stochastic process with zero mean, σ^2 variance, and having non-zero covariance values. The procedure starts with obtaining a sample data of limited size (i.e. n -design sets each having k -variables), $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\}^T$, and a corresponding vector of scalar responses, $\mathbf{Y} = \{y^{(1)}, y^{(2)}, \dots, y^{(n)}\}^T$. It is assumed that if any two design points, e.g. $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$, are positioned close together in the design space, their respective function values $y^{(i)}$ and $y^{(j)}$ are also expected to be similar. Usually, the Latin hypercube technique is used to create an initial point, ensuring a diverse set of points along each variable dimension [1].

Without going into the detailed mathematics, here, we provide the Kriging predictor, as follows:

$$\hat{y}(\mathbf{x}) = \hat{\mu} + r(\mathbf{x}^*, \mathbf{x})^T R^{-1}(y(\mathbf{x}) - \mathbf{1}\hat{\mu}) \quad (1.2)$$

where $r(\mathbf{x}^*, \mathbf{x})$ is the linear vector of correlation between the unknown point $\mathbf{x}$ to be predicted and the known sample points $\mathbf{x}^*$. R denotes the $n \times n$ matrix with (i, j) whose entry is $Corr[y^{(i)}, y^{(j)}]$, and $\mathbf{1}$ denotes an n -vector of ones. The optimal values of $\hat{\mu}$ and $\hat{\sigma}^2$, expressed as a function of R are given below [10]:

$$\hat{\mu} = \frac{\mathbf{1}^T R^{-1} y}{\mathbf{1}^T R^{-1} \mathbf{1}} \quad (1.3)$$

$$\hat{\sigma}^2 = \frac{(\vec{y} - \mathbf{1}\hat{\mu})^T R^{-1} (\vec{y} - \mathbf{1}\hat{\mu})}{n} \quad (1.4)$$

Moreover, Kriging is attractive because of its ability to provide error estimates of the predictor:

$$s^2(\mathbf{x}) = \hat{\sigma}^2 \left[1 - r^T R^{-1} r + \frac{(1 - r^T R^{-1} r)^2}{\mathbf{1}^T R^{-1} \mathbf{1}} \right] \quad (1.5)$$

1.1.3 Evolutionary multiobjective optimization (EMO)

Evolutionary multi-objective optimization (EMO) methods based on evolutionary algorithms (EAs) have become popular and been widely used in the past few decades [11, 12]. EMO methods start with a population of individuals, which evolve by using genetic operators (e.g., reproduction, crossover, and mutation). This study focuses on using evolutionary algorithms because of their wide applicability and certain advantages. For example, they do not assume any convexity or differentiability of the objective functions involved and can easily deal with the problems with locally optimal solutions.

Instead of finding a single Pareto-optimal solution at a time, set based multi-objective optimization methods attempt to find a set of Pareto-optimal solutions in a single simulation run [11]. The goals of an EMO are:

1. Find a well-converged set of trade-off solutions, and
2. Find a well-diversified set of solutions across the entire efficient front.

The population approach of an EMO and the flexibilities in its operators allow both the above goals to be achieved for multi-objective problems. EMO algorithms, such as NSGA-II [13] and SPEA2 [14], are popular methods for handling two and three-objective problems. Recent extensions, such as NSGA-III [15] and MOEA/D [16], are able to handle four or more objectives (even more than ten objectives). The topic is so important today that handling such large numbers of objectives is given a separate name, evolutionary many-objective optimization. Figure 1.1 illustrates the principle of set based multi-objective optimization.

In each iteration, a set of points gets operated by an EMO procedure and a new set of points—hopefully towards the strict efficient front and with more diversity—are generated. This process is expected to continue with iterations before converging on the true efficient front covering its entire range. When a set based multi-objective optimization algorithm transits from one set of non-dominated points to hopefully a better set of non-dominated points, there is a need to evaluate the convergence and diversity of the new set of solutions vis-a-vis the old solutions. It is because, if we can measure both aspects well through one or more performance metrics, they can also be used to determine the termination of the optimization algorithm.

Despite several advantages, EAs have one major limitation: they can consume many function evaluations to find an approximated set of Pareto-optimal (PO) solutions [17]. This

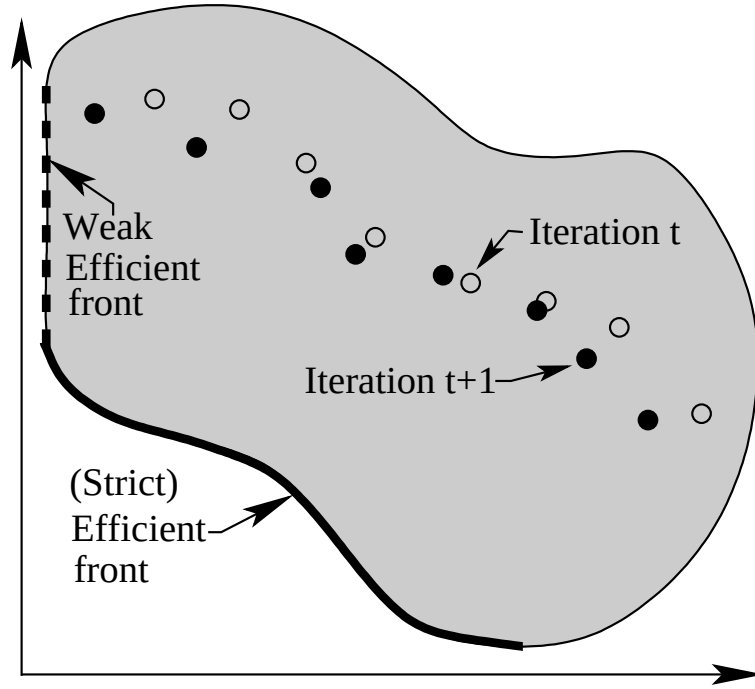


Figure 1.1: A principle of set based multi-objective optimization procedure is illustrated.

issue becomes more prominent when the problem to be solved involves computationally expensive functions which is another challenge in solving industrial optimization problems. To obtain solutions for expensive problems in a limited number of expensive function evaluations, surrogates (or metamodels) have been used in the literature as an alternative to expensive evaluations.

1.1.4 Achievement Scalarization Function (ASF) Method

One of the common ways to solve the generic multi-objective optimization problem is to solve a parameterized *achievement scalarization function (ASF)* optimization problem repeatedly for different parameter values. The ASF approach was originally suggested by Wierzbicki [18]. For a specified reference point $\mathbf{z}$ and a weight vector $\mathbf{w}$ (parameters of the ASF problem),

$((f_1 - z_1)/w_1$ and $(f_2 - z_2)/w_2)$ is then chosen as the ASF value of the point $\mathbf{x}$. For the point G, the above two quantities are identical, that is, $(f_1 - z_1)/w_1 = (f_2 - z_2)/w_2 = p$ (say). For points on line GH, the first term dominates and the ASF value is identical to p . For points on line GK, the second term dominates and the ASF value is also identical to p . Thus, it is clear that for any point on lines GH and GK, the corresponding ASF value will be the same as p . This is why an iso-ASF line for an objective vector at F traces any point on lines GK and GH. For another point A, the iso-ASF line is shown in dashed lines passing through A, but it is important to note that its ASF value will be larger than p , since its iso-ASF lines meet at the $\mathbf{w}$ -line away from the direction of minimum of both objectives. Since the ASF function is minimized, point F is considered better than A in terms of the ASF value. Similarly, point A will be considered better than point B. Simple logic will reveal that when the ASF problem is optimized the point O will correspond to the minimum ASF value in the entire objective space (marked with a shaded region), thereby making point O (an efficient point) as the final outcome of the optimization problem stated in Equation 1.6 for the chosen $\mathbf{z}$ and $\mathbf{w}$ vectors. By keeping the reference point $\mathbf{z}$ fixed and by changing the weight vector $\mathbf{w}$ (treating it like a parameter of the resulting scalarization process), different points on the efficient front can be generated by the above ASF minimization process.

1.2 Summary of Research Contributions

The main research contributions of this study is summarized in the following subsections.

1.2.1 A Generative Kriging Surrogate Model

Surrogate models are effective in reducing the computational time required for solving optimization problems. Nevertheless, interest in and literature on surrogate models and its effectiveness in finding multiple trade-off solutions has been lacking. This is largely because there is difficulty in building and solving multiple surrogate models, one for each Pareto-optimal solution. In this study, we suggest a computationally fast, Kriging-based, and generative procedure for finding multiple near Pareto-optimal solutions in a systematic manner [20]. The approach is computationally fast due to the interlinking of building multiple surrogate models, and in its systematic sequencing methodology for assisting one model with another. The proposed framework takes only a few hundreds of high-fidelity solution evaluations to find widely distributed solutions near Pareto-optimal solutions compared to the standard EMO methods requiring tens of thousands of high-fidelity solution evaluations. The framework is generic and can be extended to utilize other surrogate modeling methods easily.

1.2.2 A Taxonomy for Metamodeling Frameworks for Evolutionary Multi-Objective Optimization

An increased interest in metamodeling efforts has grown from recent developments in optimization methods. Some researchers have made efforts to classify different metamodeling approaches, but only in the realm of single-objective optimization. Most metamodeling efforts in multi-objective optimization, so far, seem to have taken a straightforward extension of single-objective metamodeling approaches. First, every objective and constraint function is metamodeled independently. Thereafter, a standard EMO methodology is applied to

the metamodels, instead of the original objective and constraint functions, to find the non-dominated front. In some studies, the above metamodeling-EMO combination is repeated in a progressive manner so that refinement of the metamodels can occur with iterations. However, with the possibilities of a combined constraint violation function that can be formulated by combining violations of all constraints in a normalized manner [21] and a combined scalarized objective function (weighted-sum, achievement scalarization function or Tchebychev function) [19], different metamodeling frameworks can certainly be explored. While the straightforward approach requires a construction of many metamodels, the suggested metamodels for combined objective and constraint violations will reduce the number of needed metamodels. However, the flip side is that each metamodel of the combined functions is likely to be more complex, having discontinuous, non-differentiable, and multi-modal landscapes. Thus, the success of these advanced metamodeling frameworks is closely tied with the advancements in the metamodeling methods. While these advancements are in progress, in this study, we outline, for the first time, a number of different and interesting metamodeling frameworks [22] for multi-objective optimization, utilizing combined approaches of objectives alone, constraints alone, as well as objectives and constraints together. Our taxonomy includes one framework that requires as many as $(M + J)$ metamodels (where M and J are the number of objectives and constraints, respectively) to another framework that requires only one metamodel.

1.2.3 Adaptive Switching Between Metamodeling Frameworks

Here, our main contribution is twofold. As we mentioned previously regarding the proposed taxonomy of different metamodeling frameworks for multi-objective optimization, there are several ways to build and utilize metamodeling approaches. We argue that it is more efficient

to use different metamodeling frameworks at different stages of the optimization process, we proposed a novel adaptive method for switching among different generative metamodeling frameworks and simultaneous framework in multiple epochs. Statistical tests on multi-objective convergence and diversity-preservation metrics are made at the start of each epoch to determine one of the frameworks which is most suitable at that instant. A switching between metamodeling frameworks is an efficient approach, rather than using one framework throughout, mainly due to the specialties that each metamodeling framework possesses. On a number of multi-objective constrained and unconstrained test problems, the switching methods have produced better results by using a low budget of solution evaluations, compared to the individual metamodeling framework alone.

Additionally, we introduced a trust regions method to achieve a better convergence behavior. We have presented a trust regions concept in order to produce a reliable and focused search by trading-off exploration and exploitation aspects of an optimization algorithm. From another side, the uncertainty of metamodeling effected the accuracy and the efficiency of any approximation mathematical model (surrogated model or metamodel). Therefore, we implemented the trust regions concept for getting more robust solution and reducing the uncertainty as well.

1.3 Dissertation Organization

The rest of the study is structured as follows. In Chapter 2, we present the proposed taxonomy for metamodeling frameworks. In Chapter 3, we present the metamodeling frameworks M1-1, M2-1, M1-2, M3-1, M4-1, M5, and M6. Results and comparison between metamodeling frameworks will be included Chapter 4. While in Chapter 5 we discuss the adaptive

switching between metamodeling frameworks as well as trust regions. We present the future work in Chapter 6. Finally, in Chapter 7, the conclusion of this study is provided.

Chapter 2

A Taxonomy for Metamodeling Frameworks for Evolutionary Multi-Objective Optimization

In this chapter, we provide a review of the most important efforts in a classification of metamodeling literature that relates to our study. Then we describe our proposed taxonomy for metamodeling frameworks in evolutionary multi-objective optimization.

2.1 Introduction

Some researchers have made efforts to classify different metamodeling approaches, but only in the realm of single-objective optimization. Santana et al. [23] has classified surrogates according to the type of model used (e.g., Kriging, radial basis function, and polynomial regression). Jin [24] proposed a classification based on the way single-objective evolutionary algorithms incorporated surrogate models. Shi and Rasheed [25] classified different metamodeling approaches according to direct or indirect fitness replacement methods. Alan and Toscano [26] classified surrogated models depending on the accuracy between metamodel approaches and which approach was best suited for use.

A recent study [27] provided a taxonomy of model-based multi-objective optimization

mainly for unconstrained problems. The taxonomy is based on modeling of independent or aggregation of objectives and how new solutions are created and selected (in-fill or otherwise) in an algorithm. Based on the taxonomy, authors have classified seven existing metamodeling multi-objective methods (including ParEGO [28], SMS-EGO [29], and MOEA/D-EGO [6]). Parallel and batch use of the methods were also highlighted and extended [30]. While these advancements are in progress. In this study, we outline, for the first time, a number of different and interesting metamodeling frameworks for multi-objective optimization utilizing combined approaches of objectives alone, constraints alone, and objectives and constraints together. Our taxonomy includes one method that requires as many as $(M + J)$ metamodels (where M and J are numbers of objectives and constraints) to another framework that requires only one metamodel. The taxonomy proposed here is generic for it to degenerate for single-objective unimodal, multi-modal, and importantly constrained optimization problems.

On a survey of many existing multiple and many-objective metamodeling studies, we have made a classification of them according to our proposed taxonomy. Table 2.1 shows that a majority of the existing studies used M1 framework (in which each objective and constraint function is metamodeled separately) and only a few studies used M2, M3, and M4. Our previous initial proof-of-principle study on M5 framework is the lone study in this category [20]. There does not exist any study implementing our M6 framework which seems to be an interesting and technically challenging proposition. In this study, we propose one such framework; other frameworks are certainly possible.

Table 2.1: Classification of existing metamodeling-based multi-objective optimization studies into six frameworks of this study. Framework M6 is proposed for the first time here.

Frameworks	References	Metamodel
M1-1	[31]	Artificial Neural Network (ANN)
	[32–34]	Radial Basis Function (RBF)
	[35, 36]	Support Vector Machines (SVM)
M1-2	[6, 37–44]	Kriging (KRG)
	[45, 46]	Genetic Programming (GP)
	[47]	KRG+Polynomial Response Surfaces (PRS)
	[48–50]	KRG+RBF
	[51]	KRG+SVM
	[52–56]	RBF
	[57, 58]	SVM
M2-2	[59]	KRG
M3	[60]	KRG+RBF+PRS
	[61]	Moving Least Squares (MLS)
	[62]	KRG+ Polynomial Chaos Expansions (PCE)
	[63]	SVM
M4	[64]	RBF+SVM
	[65–67]	KRG
M5	[20]	KRG

2.2 The Proposed Taxonomy

Multiple and many-objective optimization problems involve a number of (say, M) objective functions ($f_i(\mathbf{x})$) as a function of decision variables x) and a number of (say, J) constraint functions ($g_j(\mathbf{x})$), each as a function of $\mathbf{x}$. For brevity, we do not consider equality constraints in this study, but with certain modifications, they can be handled in the same way as discussed here.

2.2.1 A Taxonomy for Multi-objective Constraint Problems

In this section, we propose a taxonomy of various frameworks for using metamodeling approach in multiple and many-objective optimization algorithms. Our taxonomy finds six different broad frameworks (M1 to M6), as illustrated in Figure 2.1. Our approach is based on the cardinality of metamodels for objectives and constraints.

In the first framework (M1), all objectives and constraints are metamodeled independently, thereby requiring a total of $(M + J)$ metamodels before a multi-objective optimization approach can be applied. This framework is a straightforward extension of the single-objective metamodeling studies, applied to approximate each objective and constraint functions. Once all such metamodels are constructed, an EMO algorithm can use them to find one Pareto-optimal solution at a time (like the generative method used in classical optimization literature [19]) and we call this framework M1-1, or they can be used to find a number of Pareto-optimal solutions simultaneously (like in evolutionary multi-objective optimization (EMO) literature), and we call this method M1-2.

The next metamodeling framework can approximate an overall estimation function of all constraint violations together as one quantity, thereby reducing the overall number of meta-

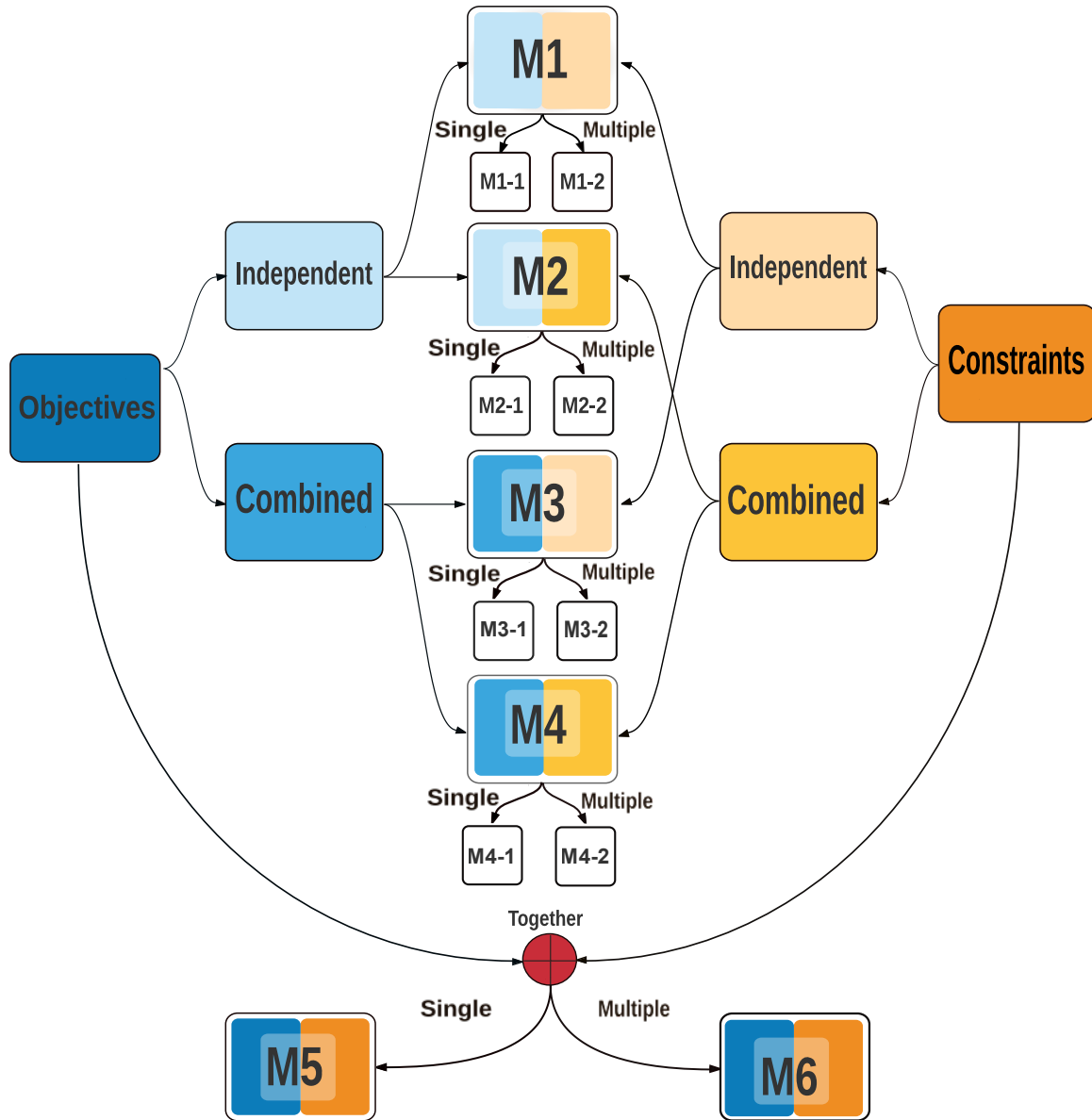


Figure 2.1: The proposed taxonomy of six different metamodeling frameworks for multiple and many-objective optimization.

models to $(M + 1)$. The well-known normalized, bracket-operator based constraint violation functions [11,21] can be used for this purpose. Like in M1, the constructed metamodels can also be used to find one Pareto-optimal solution at a time as a generative approach (we call it M2-1) or simultaneously like in an EMO approach (we call it M2-2).

The next metamodeling framework approximates each constraint function independently, but metamodels a combined objective function involving all M objectives together, similar to ParEGO approach [28]. Both M3 and ParEGO frameworks use parameterized scalarization methods to target one optimal point at a time. While ParEGO uses Tchebyshev function as scalarization, our M3 framework uses the achievement scalarization function (ASF), which is identical to Tchebyshev function under certain conditions. M3 uses a real-valued RGA to optimize the metamodel, while ParEGO uses a steady-state EA where parents are replaced by better offsprings. While M3-1 proposes to find a single Pareto-optimal solution in one run, thereby requiring multiple runs to generate Pareto-optimal solutions, a multi-modal combined landscape, similar to framework M6 described later, but with objective functions alone, can be metamodeled as M3-2.

Then, our fourth classification (M4) requires only two metamodels to be constructed for finding one Pareto-optimal solution, in which one metamodel is for a combined objective function and the second metamodel is made for a combined constraint violation (like in M2 approach). A similar multi-modal approach (M4-2) can also be constructed. The frameworks (M1-1, M2-1, M3-1, and M4-1) are ideal for classical point-based optimization algorithms, each requiring multiple applications to find multiple Pareto-optimal solutions. However, frameworks (M1-2, M2-2, M3-2, and M4-2) are ideal for EMO approaches.

More analysis will reveal that there could be two more frameworks, in which objectives and constraints are somehow combined to have a single overall *selection* function which when

optimized will lead to one or more Pareto-optimal solutions. In M5, the combined selection function has a single optimum coinciding with a specific Pareto-optimal solution, and in M6, the combined selection function is multi-modal and makes multiple Pareto-optimal solutions as its optima. Both M5 and M6 frameworks involve a single metamodel in each iteration, but if K Pareto-optimal solutions are to be found, M5 needs to be applied K times, whereas M6 still involves a single multi-modal metamodel in finding a set of Pareto-optimal solutions. In EMO algorithms, such as in NSGA-II [13], NSGA-III [15], MOEA/D [16], and others, the combined action of the selection operator involving non-domination and niching operations is an ideal way of visualizing the above-mentioned selection function. In this spirit, we believe that M5 and M6 are intricately advantageous for EMO approaches and, although they have not been paid much attention, remain as potential and fertile areas for metamodeling based EMO algorithms.

Thus, it is observed that according to our proposed taxonomy, frameworks M1 to M6 require the maximum possible metamodels ($M + J$) to single metamodel in each iteration of the multi-objective metamodeling algorithms. While M6 requires a minimum number of metamodels, this does not come free and is expected that complexity of the metamodels will increase from M1 to M6. It then becomes an interesting research task to identify a balance between the number of metamodels and the reduced complexity of metamodels for a particular problem-algorithm combination. In this study, we do not study the effect of algorithm per se, but present results of a particular approach on different problems using all six metamodeling frameworks to illustrate each framework's potential in different problems. In an application, all M objectives can be clustered into a smaller number of m clusters ($m < M$) and all J constraints can be clustered into smaller number of j clusters ($j < J$). In such a situation, objectives and constraints within each cluster can be combined and then

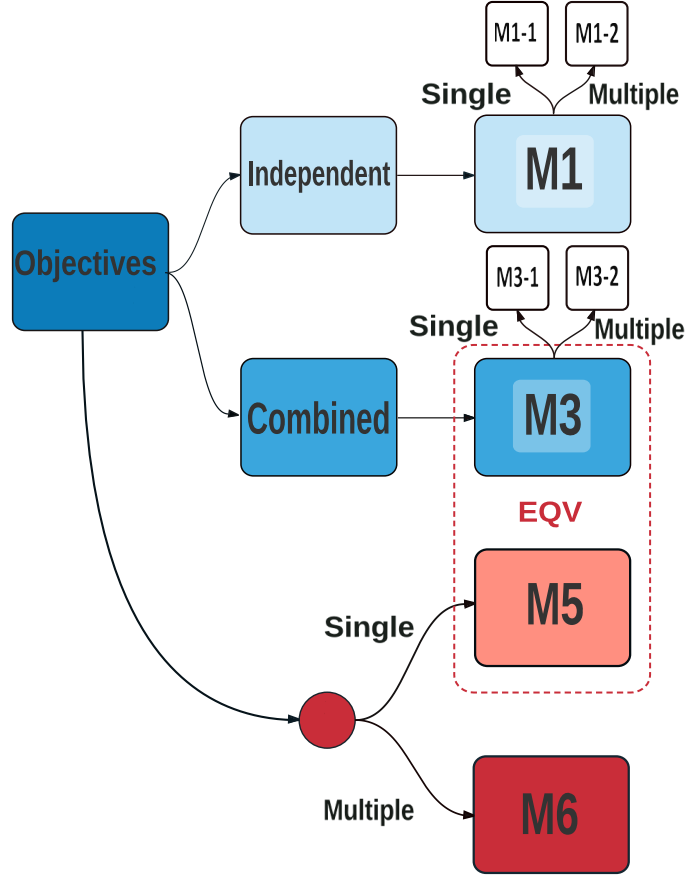


Figure 2.2: Degenerated taxonomy for multi-objective unconstrained optimization.

each combined function can be metamodeled using our proposed taxonomy.

2.2.2 A Taxonomy for Multi-Objective Unconstraint Problems

If the multi-objective optimization problem is unconstrained, frameworks M1 and M2 becomes identical and so are M3 and M4. Interestingly, M3, M4, and M5 also become identical to each other. Figure 2.2 shows the resulting taxonomy of metamodeling methods in this case.

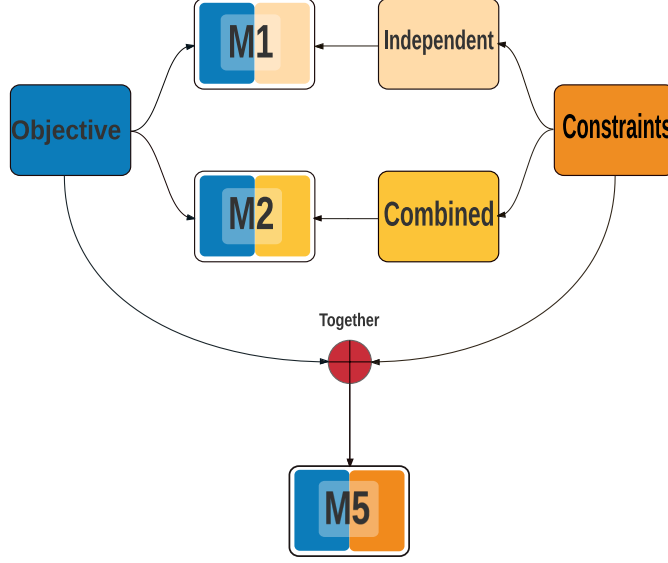


Figure 2.3: Degenerated taxonomy for single-objective optimization for finding a single optimal solution.

2.2.3 A Taxonomy for Single-Objective Constraint Problems

The proposed taxonomy for multi-objective metamodeling frameworks also degenerates to single-objective problems. Figure 2.3 shows the resulting degenerate taxonomy for finding a single optimum in a single-objective problem. In this case, frameworks M1 and M3 are identical and so are M2 and M4. A similar taxonomy can also be derived for finding multiple optima in a single-objective optimization problem, except that M6 framework will now become relevant. Sub-frameworks M1-1 and M1-2 become relevant in determining whether a single optimum at a time or multiple optima simultaneously, respectively, would be found. Similarly, sub-frameworks M2-1 and M2-2 will also be relevant in this case. For brevity, we do not present the respective diagram for single-objective, multi-modal optimization case here.

2.3 Summary

In this chapter, we presented the proposed taxonomy for metamodeling frameworks for evolutionary multi-objective optimization with a literature review that relates to our study. Moreover, we provided a description of each of the six multi-objective frameworks. Additionally, we degenerated the taxonomy to unconstrained multi-objective optimization and also to single-objective optimization problems.

Chapter 3

Metamodeling Frameworks M1-1, M2-1, M1-2, M3-1, M4-1, M5, and M6

In this chapter, we provide further description and representative algorithm of each of the six multi-objective metamodeling frameworks.

3.1 Metamodeling Frameworks M1-1 and M2-1

The metamodeling algorithm for M1-1 and M2-1 starts with an archive of initial population ($\mathcal{A}_0$ of size N_0) created using the Latin hypercube sampling (LHS) method on the entire search space. Each objective function ($f_i(\mathbf{x})$, for $i = 1, \dots, M$) is first normalized to obtain a normalized function $\underline{f}_i(\mathbf{x})$ using high-fidelity evaluation of initial archive members, so that the minimum and maximum values of $\underline{f}_i(\mathbf{x})$ evaluations is zero and one, respectively. Then, metamodels are constructed for each of the M normalized objective functions independently: $(\underline{f}_1(\mathbf{x}), \dots, \underline{f}_M(\mathbf{x})), \forall i \in \{1, 2, \dots, M\}$ using a chosen metamodeling method. For all implementations here, we use the Kriging metamodeling method [10]. While the objective metamodeling approach is identical to both M1-1 and M2-1, the metamodeling of constraint functions is different.

For M1-1, each constraint function ($g_j(\mathbf{x})$, for $j = 1, \dots, J$) is first normalized to obtain a normalized constraint function ($\underline{g}_j(\mathbf{x})$) using standard methods [68], and then metamodeled separately to obtain an approximate function ($\tilde{\underline{g}}_j(\mathbf{x})$) using the same metamodeling method (Kriging method is adopted here) used for metamodeling objective functions. For M2-1, a single aggregated constraint violation function ($\text{ACV}(\mathbf{x})$) is first constructed using the normalized constraint functions, as follows:

$$\text{ACV}(\mathbf{x}) = \begin{cases} \sum_{j=1}^J \underline{g}_j(\mathbf{x}), & \text{if } \mathbf{x} \text{ is feasible,} \\ \sum_{j=1}^J \langle \underline{g}_j(\mathbf{x}) \rangle, & \text{Otherwise,} \end{cases} \quad (3.1)$$

where the bracket operator $\langle \alpha \rangle$ is α if $\alpha > 0$ and zero, otherwise. In M2-1, the constraint violation function is then metamodeled to obtain $\widetilde{\text{ACV}}(\mathbf{x})$. Thus, it is clear that for high-fidelity solutions, $\text{ACV}(\mathbf{x})$ takes a negative value for feasible solutions and a positive value for an infeasible solution.

After all $(M + J)$ or $(M + 1)$ metamodels are constructed for M1-1 or M2-1, respectively, the metamodeled normalized objectives are combined into a single aggregated function and optimized with metamodeled constraints to find a single infill point using a single-objective evolutionary optimization algorithm (real-coded genetic algorithm (RGA) [11]). In τ generations of RGA, the following achievement scalarization function ($\text{ASF}_{12}(\mathbf{x}, \mathbf{z})$) [18] is optimized for every $\mathbf{z}$ vector:

$$\text{ASF}_{12}(\mathbf{x}, \mathbf{z}) = \max_{j=1}^M \left(\tilde{f}_j(\mathbf{x}) - z_j \right), \quad (3.2)$$

where the vector $\mathbf{z}$ is one of the Das and Dennis's [69] approach on the unit simplex on the M -dimensional hyper-space. Thus, for H different $\mathbf{z}$ vectors, H different ASF_{12} functions

are formed and optimized one after the other. The RGA procedure is modified using a trust-region concept, which we describe in Section 5.3. The best solution for each subproblem constitutes one infill point and is sent for a high-fidelity evaluation. The solution is then included in the archive of high-fidelity solutions to obtain $\mathcal{A}_1$. After all H solutions are included in the archive, one epoch of the M1-1 or M2-1 framework optimization problem is completed. In the next epoch, all high-fidelity solutions are used to normalize the objective functions and constraints, and the above process is repeated to obtain $\mathcal{A}_2$. The process is continued until all pre-specified maximum solution evaluations ($SE_{\max}$) is completed. A basic structure of methodologies M1-1 and M2-1 are outlined in Algorithm 1.

Thus, in M1-1, a total of $(M + J)$ metamodels are constructed and H RGA optimization procedures are run to create H infill solutions in each epoch. The only difference in M2-1 is that a total of $(M + 1)$ metamodels are constructed in each epoch.

3.2 Metamodeling Frameworks M1-2

In this framework, each objective and constraint is independently modeled. Thus, any classical generative or any EMO algorithm can be applied on the objective and constraint metamodels, once they are constructed.

In this study, we have used NSGA-II [13] as the EMO algorithm throughout, although any other algorithm could have been used. Furthermore, NSGA-II algorithm is able to find multiple solutions in a single run.

The description of M1-2 framework is given as follows. The metamodeling algorithm starts with an archive of initial population created using the Latin hypercube method on the entire search space. Then, metamodels are constructed for all M objectives ($f_j(\mathbf{x})$, $j =$

Algorithm 1 Metamodeling Frameworks M1-1 and M2-1.

Input : Objectives: $(f_1, \dots, f_M)^T$, normalized constraints: $(\underline{g}_1, \dots, \underline{g}_J)^T$, n (variables), N_0 (initial sample size), $SE_{\max}$ (total high-fidelity evaluations), RGA (real-parameter genetic algorithm), Γ (parameters of RGA), R (reference direction set), α (fraction of samples used for each reference direction), ASF (scalarization function), ACV (constrained violation function)

Output: P_T

```
1  $P \leftarrow \text{LHS}(N_0, n)$  // Latin hypercube sampling
2  $F \leftarrow N(f_m(P)), \forall m \in \{1, \dots, M\}$  // high-fidelity evaluations (Objectives) and
   normalize
3  $G \leftarrow \underline{g}_j(P), \forall j \in \{1, \dots, J\}$  // high-fidelity evaluations (constraints)
4  $eval \leftarrow N_0$  // number of function evaluations
5 while  $eval < SE_{\max}$  do
6   for  $r \in R$  do
7      $P_r \leftarrow$  Choose nearest  $\alpha|P|$  solutions from  $r$ 
8      $\tilde{F} \leftarrow \text{METAMODEL}(f_m(P_r)), \forall m \in \{1, \dots, M\}$ 
9     if M1-1 then
10       $\tilde{G} \leftarrow \text{METAMODEL}(\underline{g}_j(P_r), \forall j \in \{1, \dots, J\})$ 
11    else if M2-1 then
12       $\tilde{G} \leftarrow \text{METAMODEL}(\text{ACV}(G(P_r)))$ 
13     $\mathbf{x}_r \leftarrow \text{RGA}(\text{ASF}_{12}(\hat{F}, r), \hat{G}, \Gamma)$  // returns the best found solution along  $r$ 
14     $F_{\mathbf{x}_r} \leftarrow f_m(\mathbf{x}_r), \forall m \in \{1, \dots, M\}$  // Evaluate objectives of  $\mathbf{x}_r$ 
15     $G_{\mathbf{x}_r} \leftarrow \underline{g}_j(\mathbf{x}_r), \forall j \in \{1, \dots, J\}$  // Evaluate constraints of  $\mathbf{x}_r$ 
16     $P \leftarrow P \cup \{\mathbf{x}_r\}$ 
17     $F \leftarrow N(F \cup F_{\mathbf{x}_r})$ 
18     $G \leftarrow G \cup G_{\mathbf{x}_r}$ 
19     $eval \leftarrow eval + 1$ 
20    if  $eval \geq SE_{\max}$  then
21      Break out of all loops
21 return  $P_T \leftarrow$  Non-dominated solutions of  $P$ 
```

$1, 2, \dots, M$), and each constraint function is metamodeled separately also. Then, NSGA-II procedure is run for τ generations with these metamodels. Each non-dominated solution after τ generations of a NSGA-II run is considered as an in-fill point and is included in the archive. Generations are included in the archive. New metamodels are created again using the archive and the process is repeated until termination.

3.3 Metamodeling Frameworks M3-1 and M4-1

In these two frameworks, we transform the multi-objective optimization problem into a parameterized single-objective optimization problem. We use the achievement scalarization function (ASF) [18] using a set of H reference points $\mathbf{z}^{(h)}$ and a corresponding reference direction $\mathbf{w}$, identical for every $\mathbf{z}$. The reference direction is an equally-angled direction from each objective axis ($\mathbf{w} = (1, 1, \dots, 1)^T / \sqrt{M}$). Reference points $\mathbf{z}^{(h)}$ for $h = 1, 2, \dots, H$ are initialized as equi-spaced points on a unit hyperplane making equal angle to each objective axis. In this study, we have used Das and Dennis' method [69] to create H equi-spaced points on the hyperplane, but any other method or any other biased set of points, if desired, can also be used. Objective values of solutions are normalized ($\bar{f}_i(\mathbf{x})$) using the population-maximum and population-minimum objective values so that reference points on the normalized hyperplane ($z_i^{(h)} \in [0, 1]$ for all i and h) can be compared with the normalized objective values of the population members. The ASF formulation is given below:

$$\text{ASF}(\mathbf{x}) = \max_{i=1}^M \frac{f_i(\mathbf{x}) - z_i^{(h)}}{w_i}. \quad (3.3)$$

In M3 and M4, we construct one metamodel $\text{ASF}(\mathbf{x})$, instead of constructing one meta-

model for each objective function $f_i(\mathbf{x})$ independently. In M3, each constraint function $g_j(x)$ is modeled separately, but in M4, only the overall constraint violation function $\text{CV}(\mathbf{x})$ is metamodeled, as described in Equation 3.1. Since a parameterized scalarization of multiple objectives is used in both M3-1 and M4-1, we use a single-objective evolutionary optimization algorithm (real-coded genetic algorithm (RGA) [11]) for finding in-fill points. RGA uses a penalty parameter-less approach [68] to handle constraints. Like in M1 and M2, both algorithms start with an archive of randomly created solutions created using the Latin Hypercube method. Each archive member is then evaluated exactly, and then suitable metamodels are constructed for a specific objective scalarization parameter values after every τ generations. Constraint functions are metamodeled independently (framework M3) or together as an overall constraint violation (framework M4). The metamodels are then optimized using RGA and the obtained best solution is used as an in-fill point and this solution is included in the archive. Due to similarities of M3-2 and M4-2 with M6, we defer the discussions on these two frameworks until Section 3.5. More details are available in [70].

For frameworks M3-1 and M4-1, since every metamodeling effort results in a formulation that is expected to make one specific Pareto-optimal solution (say, $\mathbf{x}^{(*,h)}$) as the target, there is an important aspect about the sequence of the parameterized formulations which we discuss next. Thus, in total, there are H different scalarizations each focusing on finding a single Pareto-optimal point $\mathbf{x}^{(*,h)}$. Each scalarization may construct a new metamodel at every τ generations in order to progressively approach the respective Pareto-optimal solution. However, the sequence of choosing consecutive scalarization will play an important role in the success of the overall procedure. In one approach (neighborhood sweep approach), the first scalarization targets one extreme Pareto-optimal solution. Once new and improved solutions are found, the next scalarization task can target a neighboring Pareto-optimal

solution to the already found extreme or newer-extreme solution. Hence, having a number of near-optimal solutions from previous scalarizations will allow the overall procedure to be more efficient and implicitly parallel. Also, getting optimal solutions from one extreme part of the Pareto-optimal front may allow a better normalization of objectives, which is an essential part of any EMO algorithm. To find and use all extreme points from the beginning of a run, in another approach (diverse sweep approach), after every scalarization is applied, the next scalarization may consider a new reference point that is maximally away along the reference plane from all the reference points that have already been considered. Both the above approaches have merits of their own and a mixed approach may be better. In this study, we only adopt the neighborhood sweep approach.

For each scalarization, the RGA is started with α proportion of archive points close to the reference line passing through the specific reference point in the objective space. This is because points far away from the focal region of metamodeling do not contribute much to the generated metamodeling function and also fewer points for metamodeling help reduce computational time. Then, the RGA is applied κ times to take care of the inherent stochasticities of the RGA procedure. The parameters $\alpha = 0.7$ and $\kappa = 5$ are observed to perform well experimentally and kept fixed for all problems of this study. Again, every RGA solution is included in the archive to make a new metamodel before a new RGA run is performed. After making one pass of consecutive scalarizations involving all reference points, the process is repeated in reverse order one time to make more refined metamodeling of initial scalarizations. The basic structure of M3 and M4 is outlined in Algorithm 2. For M3-2 and M4-2, RGA must be replaced with a multi-modal RGA, similar to one described in Section 3.5.

Frameworks M1 to M4 are straightforward extensions of single-objective metamodeling

frameworks used in the context of evolutionary algorithms. To take care of a multitude of objectives and constraints, each is categorized the way the objectives and/or constraints are metamodeled separately or in a combined manner.

3.4 Metamodeling Framework M5

In this section, we describe one of our main contributions in this study. Framework M5 proposes a more direct metamodeling approach which not only reduces the cardinality of distinct metamodeling efforts, it is also algorithm specific. A metamodel of the outcome of an algorithm's selection operation is directly constructed here.

The focus of M5 is to find a single Pareto-optimal solution at a time by using a parameterized scalarization of all objective functions. However, instead of constructing metamodels for the scalarized objective function and constraint functions separately, as was done in M3 and M4, here, the combined effect of an EMO's selection operation is metamodeled. For example, while comparing two solutions A and B in an EMO for a particular scalarized problem, say with ASF having a specific $\mathbf{z}$ and $\mathbf{w}$, the operator computes ASF values for both A and B and then the winner is selected using the constraint-domination principle [11]. We can then directly construct a metamodel of the resulting ASF and use as a single-objective optimization problem. In this approach, we formulate the following combined selection function ($S(\mathbf{x})$) by considering all objective functions and all constraint functions together, but the final focus is to make the global optimum of S -function as one of the targeted Pareto-optimal solutions:

$$S(\mathbf{x}) = \begin{cases} \text{ASF}(\mathbf{x}), & \text{if } \mathbf{x} \text{ is feasible,} \\ \text{ASF}_{\max} + \text{CV}(\mathbf{x}), & \text{otherwise.} \end{cases} \quad (3.4)$$

Algorithm 2 Metamodeling Frameworks M3-1 and M4-1.

Input : Objectives: $[f_1, \dots, f_M]^T$, constraints: $[g_1, \dots, g_J]^T$, n (variables), ρ (sample size), $SE_{\max}$ (total high fidelity evaluations), RGA (real-parameter genetic algorithm), Γ (parameters of RGA), R (reference direction set), κ (number of points created for each reference direction), α (fraction of samples used for metamodel), **ASF** (scalarization function), **CV** (constrained violation function)

Output: P_T

```

22  $P \leftarrow \text{LHS}(\rho, n)$  // initialization with Latin Hypercube Sampling
23  $F \leftarrow f_m(P), \forall m \in \{1, \dots, M\}$  // high fidelity evaluations (functions)
24  $C \leftarrow g_j(P), \forall j \in \{1, \dots, J\}$  // high fidelity evaluations (constraints)
25  $eval \leftarrow \rho$  // number of function evaluations
26 while  $eval < SE_{\max}$  do
27   for  $r \in R$  do
28     // for each reference direction  $r$ 
29      $P_r \leftarrow \text{Sort } P \text{ according to distance from } r \text{ and select nearest } \alpha\rho \text{ solutions}$ 
30      $\mathcal{F}_r \leftarrow \text{Create\_Surrogate\_Model}(\mathbf{ASF}(r, P_r))$ 
31     if M3 then
32        $C_r^j \leftarrow \text{Create\_Surrogate\_Model}(C^j(P_r), \forall j \in \{1, \dots, J\})$ 
33     else if M4 then
34        $V_r \leftarrow \mathbf{CV}(C^j(P_r), \forall j \in \{1, \dots, J\})$ 
35        $C_r \leftarrow \text{Create\_Surrogate\_Model}(V_r)$ 
36     for  $i = 1$  to  $\kappa$  do
37        $\mathbf{x}_r \leftarrow \text{RGA}(\mathcal{F}_r, C_r, \Gamma)$  // returns the best found solution
38        $f_{\mathbf{x}_r}^m \leftarrow f_m(\mathbf{x}_r), \forall m \in \{1, \dots, M\}$  // Evaluate objectives of  $\mathbf{x}_r$ 
39        $c_{\mathbf{x}_r}^j \leftarrow g_j(\mathbf{x}_r), \forall j \in \{1, \dots, J\}$  // Evaluate constraints of  $\mathbf{x}_r$ 
40        $P \leftarrow P \cup \{\mathbf{x}_r\}$ 
41        $F \leftarrow F \cup \{[f_{\mathbf{x}_r}^1, \dots, f_{\mathbf{x}_r}^M]\}$ 
42        $C \leftarrow C \cup \{[c_{\mathbf{x}_r}^1, \dots, c_{\mathbf{x}_r}^J]\}$ 
43        $eval \leftarrow eval + 1$ 
44       if  $eval \geq SE_{\max}$  then
45         Break out of all loops
46 return  $P_T \leftarrow P(1 : |R|)$  // One solution closest in objective space to each of  $|R|$ 
      reference lines

```

Here, the parameter $ASF_{\max}$ is the worst ASF function value of all feasible solutions of the archive. After metamodeling the above selection function S , we formulate the expected improvement (EI) function [10] and optimize the EI function using a RGA. The best solution is then used as an in-fill point and a new metamodel of S is created using the new archive.

Clearly, other scalarization approaches, such as weighted-sum function or epsilon-constraint function or a generic Tchebyshev function [19] can also be used. This framework is a generative multi-objective optimization procedure in which one Pareto-optimal point is determined to be found at a time [20]. The algorithm is outlined in Algorithm 3.

3.4.1 The Expected Improvement Procedure

Kriging methodology was also called the Design and Analysis of Computer Experiments (or DACE) stochastic process model [71]. The DACE is one of the efficient tools for dealing with expensive single-objective optimization problems. In this framework, the approximation of a function in terms of a design variable is considered as a sample of the Gaussian stochastic process. Then, the distribution of the function value at any untested point can be estimated using Kriging model. Jones et al [10] proposed a practical approach to determine the location of additional sample points which improves the Kriging model accuracy. This is known as the Efficient Global Optimization (EGO) procedure. On the Kriging model, EGO searches for the location where the expected improvement of the original function is maximized and then reconstructs the surrogate model by adding a new sample point at this location.

The second step of this procedure is based on the fact that Kriging helps in estimating the model uncertainty and stresses on exploring points where we are uncertain. In order to do this, Kriging method treats the value of the function at $\mathbf{x}$ as if it was the realization of a stochastic process $y(\mathbf{x})$, with the mean given by the predictor $\hat{y}(\mathbf{x})$ and variance $s(\mathbf{x})$. The

Algorithm 3 Metamodeling Framework M5.

Input : Objectives: $[f_1, \dots, f_M]^T$, constraints: $[g_1, \dots, g_J]^T$, n (variables), ρ (sample size), $SE_{\max}$ (total high fidelity evaluations), RGA (real-parameter genetic algorithm), Γ (parameters of RGA), R (reference direction set), κ (number of points created for each reference direction), α (fraction of samples used for metamodel), **ASF** (scalarization function), **CV** (constrained violation function)

Output: P_T

```
46  $P \leftarrow \text{LHS}(\rho, n)$  // initialization with Latin Hypercube Sampling
47  $F \leftarrow f_m(P), \forall m \in \{1, \dots, M\}$  // high fidelity evaluations (functions)
48  $C \leftarrow g_j(P), \forall j \in \{1, \dots, J\}$  // high fidelity evaluations (constraints)
49  $eval \leftarrow \rho$  // number of function evaluations
50 while  $eval < SE_{\max}$  do
51   for  $r \in R$  do
52     // for each reference direction  $r$ 
53      $P_r \leftarrow \text{Sort } P \text{ according to distance from } r \text{ and select nearest } \alpha\rho \text{ solutions}$ 
54      $P_r^s \leftarrow \text{Feasible}(P_r)$  // find feasible solutions of  $P_r$ 
55      $P_r^u \leftarrow P_r \setminus P_r^s$  // locate infeasible solutions of  $P_r$ 
56      $\text{Fitness}_r^s \leftarrow \text{ASF}(r, P_r^s)$  // fitness of feasible solutions
57      $V_r^u \leftarrow \text{CV}(C^j(P_r^u), \forall j \in \{1, \dots, J\})$  // constraint violation function
58      $\text{Fitness}_r^u \leftarrow V_r^u + \left( \max_{P_r^s} \text{Fitness}_r^s \right)$  // fitness of infeasible solutions
59      $\mathcal{F}_r \leftarrow \text{Create\_Surrogate\_Model}(\text{Fitness}_r)$  // Surrogate model for constrained
        ASF function model
60     for  $i = 1$  to  $\kappa$  do
61        $\mathbf{x}_r \leftarrow \text{RGA}(\mathcal{F}_r, \mathbf{EI}, \Gamma)$  // returns the best found solution
62        $f_{\mathbf{x}_r}^m \leftarrow f_m(\mathbf{x}_r), \forall m \in \{1, \dots, M\}$  // Evaluate objectives of  $\mathbf{x}_r$ 
63        $c_{\mathbf{x}_r}^j \leftarrow g_j(\mathbf{x}_r), \forall j \in \{1, \dots, J\}$  // Evaluate constraints of  $\mathbf{x}_r$ 
64        $P \leftarrow P \cup \{\mathbf{x}_r\}$ 
65        $F \leftarrow F \cup \{[f_{\mathbf{x}_r}^1, \dots, f_{\mathbf{x}_r}^M]\}$ 
66        $C \leftarrow C \cup \{[c_{\mathbf{x}_r}^1, \dots, c_{\mathbf{x}_r}^J]\}$ 
67        $eval \leftarrow eval + 1$ 
68       if  $eval \geq SE_{\max}$  then
        | Break out of all loops
69 return  $P_T \leftarrow P(1 : |R|)$  // One solution closest in objective space to each of  $|R|$ 
    reference lines
```

expected improvement function is given as follows [10]:

$$E[I(\mathbf{x})] = (y_{best} - \hat{y}(\mathbf{x}))\Phi\left(\frac{y_{best} - \hat{y}(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\phi\left(\frac{y_{best} - \hat{y}(\mathbf{x})}{s(\mathbf{x})}\right) \quad (3.5)$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the normal cumulative distribution function and probability density function, respectively. It is worth noting here that the implementation of Kriging (DACE) is based on universal Kriging, where it is possible to use other regression models as well.

As mentioned above, Kriging and subsequent EGO methodology are standards in the context of single-objective unconstrained problems, but they have not been extended well either to constrained single-objective problems nor to multi-objective optimization problems. A recent study [72] suggested a way to modify the expected improvement function for constrained problems. It selects the best feasible objective function value instead of y_{best} in Equation 3.5. This method has been applied only to single objective optimization problems. The expected improvement function is modified as follows:

$$E_c[I(\mathbf{x})] = E[I(\mathbf{x})]\Pi_{j=1}^J F_j(\mathbf{x}) \quad (3.6)$$

where $F_j(\mathbf{x})$ is given as follows for j -th constraint:

$$F_j(\mathbf{x}) = \begin{cases} 0.5 + 0.5\text{erf}\left(\frac{\bar{g}_j(\mathbf{x})}{s_j(\mathbf{x})}\right), & \text{if } \text{erf}\left(\frac{\bar{g}_j(\mathbf{x})}{s_j(\mathbf{x})}\right) \geq 1, \\ 2 - \text{erf}\left(\frac{\bar{g}_j(\mathbf{x})}{s_j(\mathbf{x})}\right), & \text{if } 0 < \text{erf}\left(\frac{\bar{g}_j(\mathbf{x})}{s_j(\mathbf{x})}\right) < 1, \\ 0, & \text{otherwise,} \end{cases} \quad (3.7)$$

where $\bar{g}_j(\mathbf{x})$ is the normalized version of j -th constraint function, and s_j is the MSE pre-

diction of the j -th constraint function. It is clear that if any constraint is violated, $E_c[I(\mathbf{x})]$ is zero and for near constraint boundary solutions $E_c[I(\mathbf{x})]$ has a large value, thereby emphasizing near-boundary solutions during simulations and for points that are well inside the feasible region $E_c[I(\mathbf{x})] = E[I(\mathbf{x})]$. Here, we do not use this method and instead handle constraint directly through a modeling of the selection function.

3.4.2 Proposed Multi-objective EGO Method

With the brief description of the EGO method for single-objective optimization problems that are discussed above, we are now ready to provide details of our proposed procedure.

The development of a suitable surrogate modeling and its solution are achieved for one targeted Pareto-optimal solution at a time. However, the surrogate building process for multiple Pareto-optimal solutions are not all independent to each other. Rather the whole process is interlinked, as described by the following step-by-step procedure:

Step 1: Create an archive having H random points based on Latin hypercube sampling (LHS) [1] from the entire variable search space. Evaluate each of these solutions using the objective and constraint functions (referred to as ‘high-fidelity’ evaluations).

Step 2: Generate R reference directions on a normalized unit hyperplane in the objective space using Das and Denni’s method [69]. Then, for each reference direction chosen using a `Diversity_Preserver` procedure [73], execute the following steps.

Step 3: Select K points closest to the reference direction based on projected Euclidean distance of all H archive points in the objective space using a `Points_Selector` procedure.

Step 4: Build a local surrogate model (**Surrogate_Model** procedure) using the chosen αH points executed already with high-fidelity evaluations, $0 < \alpha < 1$. According to our empirical results $\alpha = 0.7$.

Step 5: Use the developed surrogate model to find the best possible solution of the model by a specific (**Optimization**) procedure.

Step 6: Perform a high-fidelity evaluation of the newly created solution and add it to the archive.

Step 7: Repeat Steps 3 to 6 p times to add p new solutions to the archive.

Step 8: Move to a new reference line according to the **Diversity_Preserver** procedure and Go to Step 3 until all reference lines are considered or another termination criterion is satisfied.

The above step-by-step procedure is a generic surrogate-based EMO procedure that can be applied with any surrogate modeling technique and optimization algorithm. It also has a few other flexibilities which we describe next.

3.4.3 Diversity_Preserver Procedure

This procedure provides the linking between model building and its solution among several reference lines. Given K reference directions spanning the entire first quadrant of the M -dimensional objective space, this procedure determines the sequence of choosing reference lines. Figure 3.1 illustrates the proposed procedure in which $H = 20$ random points denote the archive and for each of five reference directions, $K = 4$ points are chosen.

This procedure involves fixing the following three entities:

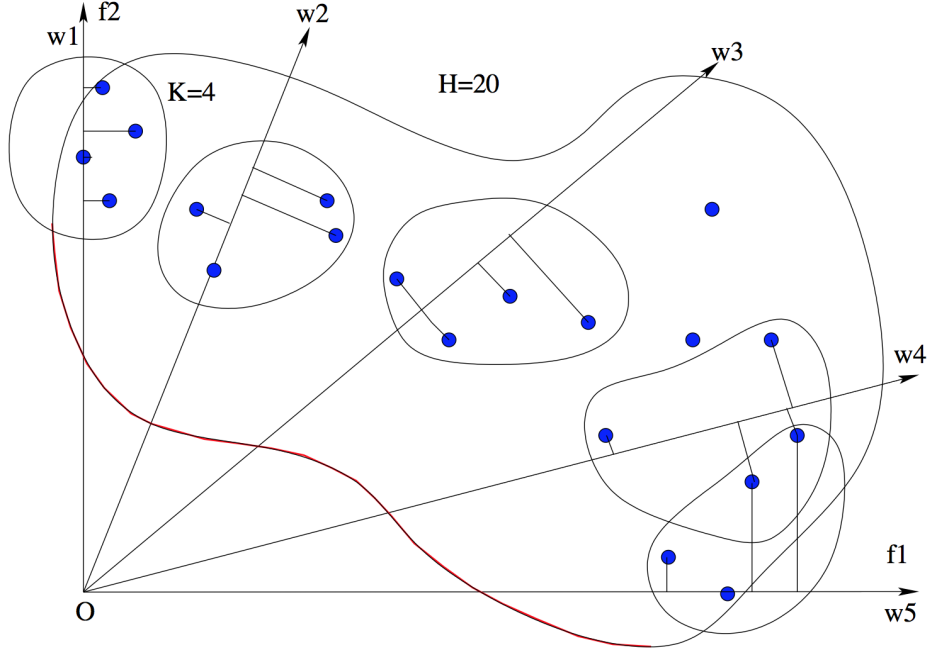


Figure 3.1: The proposed surrogate-based generative EMO.

1. The starting reference direction,
2. The sequence of choosing reference directions, and
3. Repeat pattern of the sequencing.

The surrogate modeling can start anywhere, but either one of the extreme reference directions (lying on one of the objective directions) or in the middle of the objective space (equi-distance from all objective directions) are two unbiased approaches. In our study here, we consider the former approach of choosing one of the objective axis direction as the initial reference direction.

The second entity is probably the most important factor. Once the initial direction is chosen, a model is built and one or more solutions are obtained using the model, and which reference direction to choose next is a relevant question.

In the *neighborhood* approach, we can choose one of the nearest reference directions to

the already chosen directions as the next direction. Since there are one or more models already built around a neighboring reference direction in this approach, inclusion of their best solutions in the neighboring model building process should create a better model. However, the flip side is that each model then has a local perspective and the performance of this approach will depend on the mapping between variable and objective spaces. In the *maximum diversity* approach, a reference direction which is maximally away from all chosen reference directions is selected for building the next surrogate model. In this approach, initial few models are independent of each other, but together they will span the search space well. Later surrogate models tend to have a more global perspective than those in the neighborhood approach. We use both approaches in this study.

Another aspect of the diversity-sweep procedure is whether the above sequencing process needs to be repeated after their first pass on all reference directions. This can be decided adaptively, based on whether a satisfactory set of non-dominated solutions have been found for each reference direction. If it is to be repeated, an exactly the same sequencing or a different sequencing operation can be chosen as well. The repetition does not need all reference directions, and the ones that generated a dominated solution can be repeated. For repetition, it will be a good idea to preserve all surrogate models generated in the first pass of the procedure. In this study, we repeat the first pass in the same sequence one more time.

3.4.4 Optimization Procedure

Once the surrogate model is built, the next step is to optimize the model to find the best possible solution of the model. For this purpose, we use a real-parameter genetic algorithm (RGA) which uses simulated binary crossover and polynomial mutation operator [11]. The population is started with a random population and RGA is run for τ number of generations.

Here, the RGA population can be initialized by the K points used to build the surrogate model. Figure 3.1 illustrates the overall procedure. It can be observed that some of the original archive points may not have been chosen for any of the reference directions, particularly when K is very small compared to H . Also, some points can be chosen for more than one reference directions, particularly when K is comparable to H . Since all H points were already made high-fidelity evaluations, it is better to choose K in a way so that all H evaluated solutions are used for one or more reference directions. Another advantage of the proposed method is that since the points chosen for a surrogate model for a particular reference direction are close to each other, each model will be local to the reference direction and is likely to find the respective Pareto-optimal solution more reliably than if the model was a global spanning the entire search space. Since each surrogate model uses a few points to build it, the proposed algorithm is also likely to be computationally faster than global models.

3.5 Metamodeling Framework M6

Framework M6 culminates into an ambitious metamodeling procedure in which only one metamodel is required to find multiple Pareto-optimal solutions. A little thought will reveal that if by any procedure we are able to construct a multi-modal selection function $M_S(x)$ having H Pareto-optimal solutions as multiple distinct global optimum of it, then we can possibly employ a multi-modal (niching-based [74]), single-objective evolutionary algorithm to find and capture multiple optima in a single simulation run. In this study, we used the Karush-Kuhn-Tucker (KKT) optimality conditions based on a recently developed theoretical performance metric for multi-objective optimization by Deb et al. [75, 76] called the KKT

Proximity Measure, or KKTPM which makes a monotonous increase in KKTPM values for an increase in domination level of solutions. An interesting aspect of KKTPM is that its value is zero for all Pareto-optimal solutions and it takes a positive value as a solution that gets more and more dominated. Such a property motivates us to use KKTPM as a potential multi-modal selection function for the purpose of developing an M6 framework. Also, KKTPM considers all objectives and constraint satisfaction into its computational procedure, thereby making it an ideal M_S -function for our purpose.

First we create a set of sample points from the variable space by using the Latin hypercube approach and then compute their objective values by high fidelity computations. We then compute KKTPM value of each sample point. KKTPM function on the entire search space is then metamodelled using the Kriging procedure. Thereafter, a multi-modal real-coded GA (m-RGA) is employed to search the metamodelled KKTPM function for finding new and multiple multi-modal solutions. In our proposed approach, we set H reference directions like in NSGA-III approaches [15] or in MOEA/D [16]. For M3-2 and M4-2, the above multi-modal RGA can be used with metamodelled combined objective function and single or multiple metamodelled constraint functions, instead of a single combined selection function used in M6. For more details regarding this framework, readers can refer to [22, 70].

3.6 Summary

Based on the above detailed description of six different metamodeling frameworks, we now summarize the number of metamodels needed for each framework in Table 3.1. Recall that M , J , H , and κ are the number of objectives, constraints, desired number of Pareto-optimal solutions, and number of solutions created from a single metamodel, respectively. Clearly,

Table 3.1: Number of metamodels needed for each of the six metamodeling frameworks.

Framework	M1-1	M1-2	M2-1	M2-2	M3	M4	M5	M6
Objective Handling	Independent	Independent	Independent	Independent	Combined	Combined	Combined	Combined
Constrained Handling	Independent	Independent	Combined	Combined	Independent	Combined	Combined	Combined
# of Selection Function (s)	M+J	M+J	M+1	M+1	1+J	2	1	1
# of Metamodel (s)	$H\kappa(M+J)$	$\kappa(M+J)$	$H\kappa(M+1)$	$\kappa(M+1)$	$H\kappa(1+J)$	$H\kappa(2)$	$H\kappa$	κ
# of Solution (s)	One	Many	One	Many	One	One	One	Many

M1-1 requires maximum number of metamodeling constructions ($H\kappa(M+J)$); M2-1 requires $H\kappa(1+J)$; M3 requires $H\kappa(M+1)$; M4 needs $H\kappa(2)$; M5 requires $H\kappa$; and M6 requires κ metamodels. While M1-1 needs the maximum number of metamodeling efforts, M6 requires the least. While the cardinality reduces from M1 to M6, in general, the relative complexity of the resulting landscapes is likely to increase from M1 to M6. For a given set of objective and constraint functions, M1 approximates the original functions as they are. However, M6 attempts to approximate the most complex selection function which is multi-modal and discontinuous at the constraint boundaries.

Chapter 4

Results and Comparison Between Metamodeling Frameworks

In this chapter, we compare the results obtained by all six proposed metamodeling frameworks. We use the same parameter settings for RGA or m-RGA, each of which uses the binary tournament selection operator, simulated binary crossover (SBX), and polynomial mutation, with parameters as follows: Population size = $10n$, where n is a number of variables, number of generations = 100, crossover probability = 0.95, mutation probability = $1/n$, distribution index for SBX operator = 1, and distribution index for polynomial mutation operator = 10. The NSGA-II procedure, wherever used, is also applied with the same parameter values as above. For repeating a metamodeling formulation for each formulation, we have used $\kappa = 5$ and $\tau = 50$ generations. We performed 10 runs of all frameworks on all test problems. As mentioned earlier, for unconstrained problems (ZDT) with no constraints and three-objective problem C2DTLZ2 having a single constraint, the behaviors of both M1 and M2 are identical. In such a case, we only show the results for M1-2. The same situation occurs for M3 and M4 and we only show the results of M3. In order to have a graphical comparison, we show the obtained trade-off solutions for the median IGD run in each case. The inverted generational distance (IGD) [16] is considered as the performance metric for providing a quantitative evaluation of the obtained solutions. The IGD metric

provides a combined information about the convergence and diversity of the obtained solutions. It is worth mentioning here that, in this study, we do not make any effort to choose optimal parameter setting for each algorithm, rather identical parameters are used to provide a representative performance of each framework.

4.1 Two-Objective Unconstrained Problems

First, we apply our proposed frameworks to two-objective unconstrained problems ZDT1, ZDT2, ZDT3, and ZDT6 with ten ($n = 10$) variables, $H = 21$ reference directions, and with a maximum of only $SE_{\max} = 500$ high-fidelity solution evaluations. For each problem, we have used an initial sample size of $\rho = 300$. The obtained non-dominated solutions are shown in Figures 4.1, 4.2, 4.3, and 4.4 for frameworks M1-2, M3, M5, and M6, respectively. As mentioned above, M2-2 becomes identical to M1-2 and M4 becomes identical to M3 for unconstrained problems. For brevity, we also do not show the results from the M1-1 framework.

It is clear from the figures that all four frameworks are able to solve ZDT1 problem fairly well in only 500 high-fidelity SEs, but frameworks M5 and M1-2 perform the best. The obtained points are very close to the respective true Pareto-optimal fronts and have a good distribution of points on the entire front. Most past studies [13] have used tens of thousands of high-fidelity SEs to have a similar performance, while here using the proposed metamodeling frameworks, we are able to find a similar set of points in only 500 high-fidelity SEs. Table 4.1 shows the IGD metric value for all runs performed for all frameworks. While the Table 4.2 shows the generational distance (GD) metric value for all runs performed for all frameworks. The IGD metric is computed using 21 true Pareto-optimal points obtained for

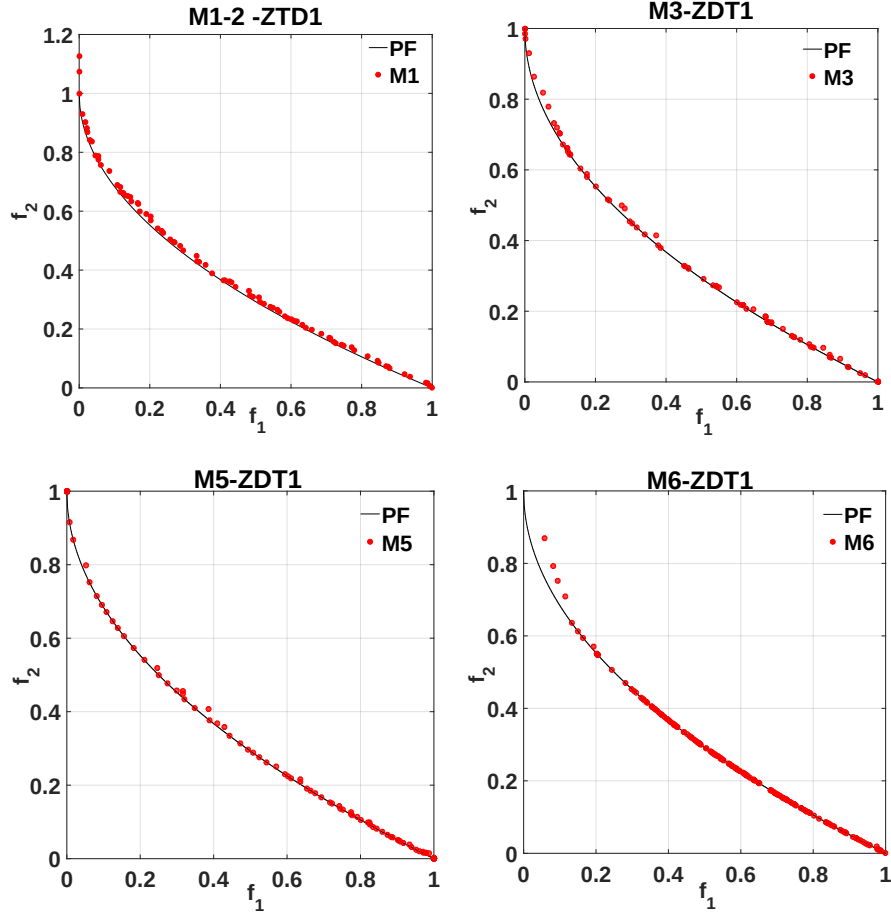


Figure 4.1: Obtained non-dominated solutions for problem ZDT1 using frameworks M1-2, M3, M5 and M6 from left to right.

each reference point, one at a time. Since each framework is expected to find the respective Pareto-optimal point, this IGD computation is able to distinguish a set of non-dominated points from another set depending on their convergence level to the desired set of points. However, to compute the GD metric, we use a large number of Pareto-optimal fronts so as to get a clear idea of the convergence level to the Pareto-optimal front.

In ZDT2 problem, two selection function based meta-modeling frameworks (M5 and M6) perform the best, followed by M1-2 and then M3. It is interesting to note that despite the multi-modal nature of the selection function landscape with M6, it is able to find multiple near Pareto-optimal points with only 500 high-fidelity solution evaluations. The convex or

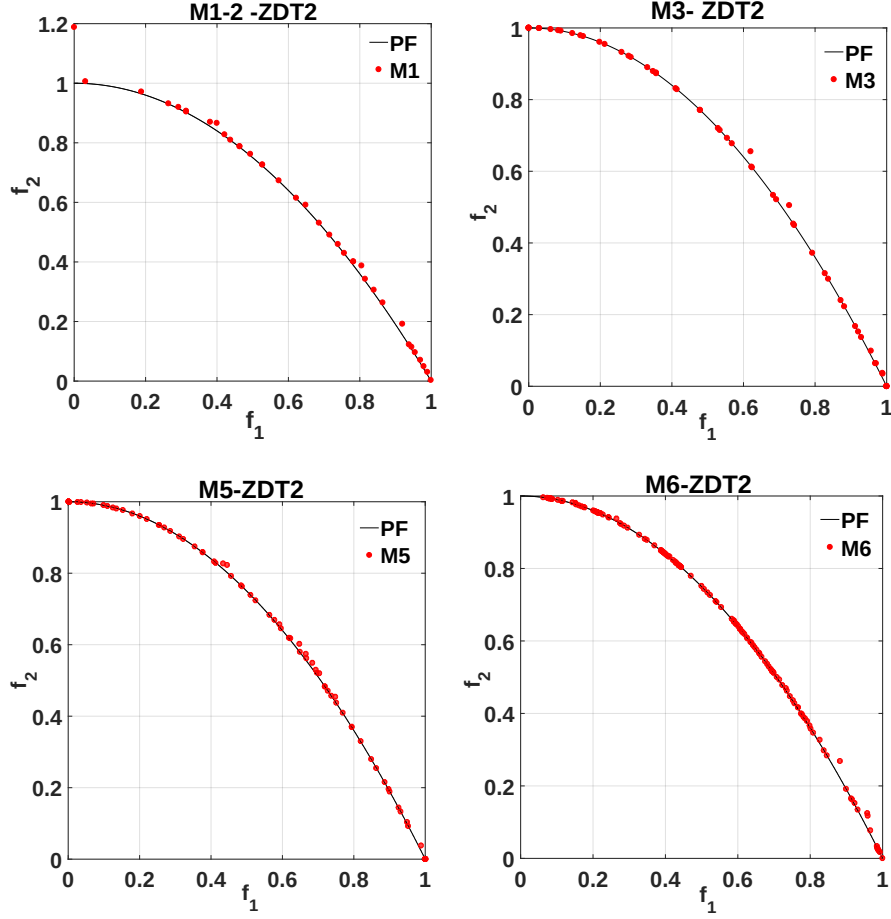


Figure 4.2: Obtained non-dominated solutions for problem ZDT2 using frameworks M1-2, M3, M5 and M6 from left to right.

non-convex nature of the Pareto-optimal front does not seem to matter to all the frameworks.

For ZDT3 problem, which has a disconnected Pareto-optimal front, M1-2 and M3 perform the best, followed by M5 and then M6. The multi-modal KKTPM-based M6 approach is also not able to get quite close to the true Pareto-optimal front with only 500 high-fidelity SEs. This is due to the added complexity the multi-modal KKTPM surface offers to the metamodeling approach when dealing with a disconnected Pareto-optimal front. Although a single metamodel is progressively modeled by adding in-filled points, more high-fidelity SEs are needed in the initial sample to make a better modeling of the entire multi-modal landscape in this problem.

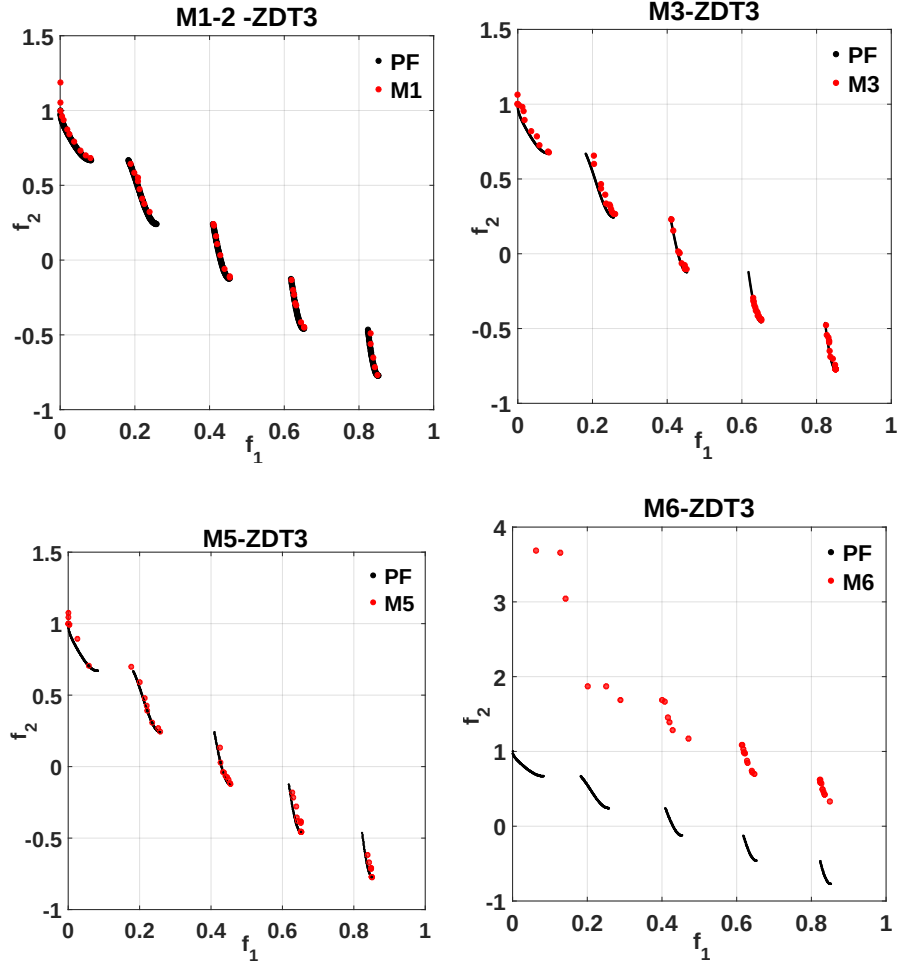


Figure 4.3: Obtained non-dominated solutions for problem ZDT3 using frameworks M1-2, M3, M5 and M6 from left to right.

ZDT6 problem shown in Figure 4.4 raises an interesting aspect common to all metamodelling frameworks. It introduces a bias in the distribution of points along the Pareto-optimal front. Thus, when a set of Latin hypercube based points are created in the variable space, the distribution of points on the objective space is biased towards a particular part (in the large f_1 part of the ZDT6 objective space). Thus, since the entire objective space is not distributed well right from the initial generation, the respective metamodelling frameworks are not able to make a good approximation of the entire objective space. In such problems, a decomposition-based aggregate method should work well. We observe that all other frameworks, except

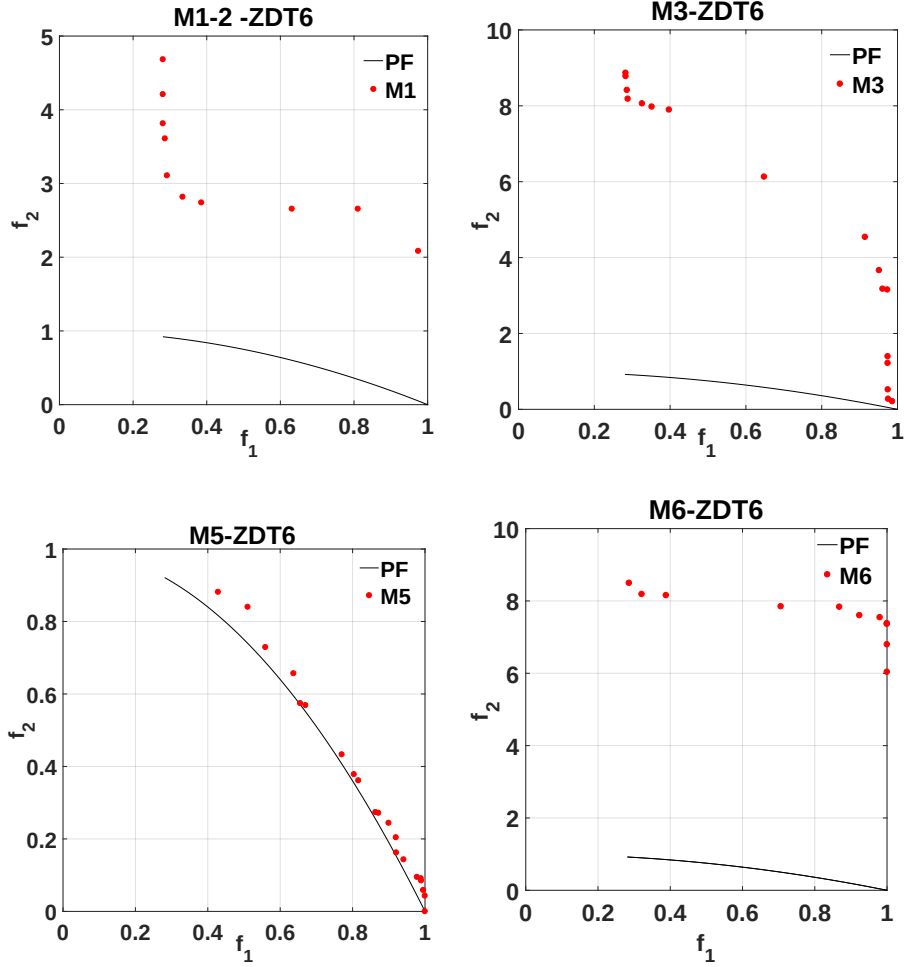


Figure 4.4: Obtained non-dominated solutions for problem ZDT6 using frameworks M1-2, M3, M5 and M6 from left to right.

M5, fail to converge close to the true front in 500 high-fidelity solution evaluations.

Nevertheless, the results of the first three problems indicate that all frameworks have performed well with a fraction of solution evaluations than they are usually used in standard studies [13].

Table 4.1 presents the average and standard deviation of the IGD values for an identical set of reference Pareto-optimal solutions for all four frameworks. It is clear that framework M5 performs the best on the unconstrained two-objective test problems. frameworks having a statistically insignificant performance from the best performing method in each problem

Table 4.1: Computed IGD values for test problems. Best performing framework and other statistically similar frameworks are marked in bold.

Problems	M1-2		M2-2		M3		M4		M5		M6	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
ZDT 1	0.0115	0.0006	-	-	0.0185	0.0046	-	-	0.0103	0.0028	0.0249	0.0153
	$p = 0.12120$		-		$p = 0.00170$		-		-		$p = 0.02570$	
ZDT 2	0.0105	0.0027	-	-	0.0130	0.0013	-	-	0.0078	0.0011	0.0103	0.0051
	$p = 0.00170$		-		$p = 0.00019$		-		-		$p = 0.73370$	
ZDT 3	0.0136	0.0019	-	-	0.0254	0.0154	-	-	0.0246	0.0027	0.7111	0.2054
	-		-		$p = 0.06390$		-		$p = 0.00018$		$p = 0.00018$	
ZDT 6	1.6144	0.1965	-	-	0.3774	0.1099	-	-	0.1208	0.0311	5.9401	0.7896
	$p = 0.00018$		-		$p = 0.00024$		-		-		$p = 0.00018$	
BNH	0.5322	0.2164	0.4605	0.1324	0.3854	0.0724	0.3539	0.1543	0.3727	0.0798	0.4177	0.1852
	$p = 0.02070$		$p = 0.00880$		$p = 0.05290$		-		$p = 0.30530$		$p = 0.27120$	
SRN	0.7452	0.2473	1.0477	0.1765	1.5564	0.1742	1.9072	0.3237	2.8532	0.5795	1.5824	0.2557
	-		$p = 0.00910$		$p = 0.00024$		$p = 0.00016$		$p = 0.00018$		$p = 0.00025$	
TNK	0.0173	0.0059	0.0748	0.0208	0.0107	0.0010	0.0300	0.0075	0.0654	0.0353	0.0388	0.0038
	$p = 0.00033$		$p = 0.00018$		-		$p = 0.00018$		$p = 0.00018$		$p = 0.00018$	
OSY	6.9534	6.3585	31.0684	10.8709	5.8386	1.1182	39.4488	13.2070	23.8865	7.9269	39.7462	8.7765
	$p = 0.47250$		$p = 0.00018$		-		$p = 0.00018$		$p = 0.00018$		$p = 0.00018$	
Welded Beam	0.8428	0.3655	2.0504	0.7854	1.5036	0.5551	2.1265	0.7054	1.0268	0.2396	2.5987	1.1322
	-		$p = 0.00058$		$p = 0.01400$		$p = 0.00044$		$p = 0.16200$		$p = 0.00033$	
DTLZ2	0.0387	0.0045	-	-	0.0634	0.0038	-	-	0.0318	0.0029	0.0345	0.0043
	$p = 0.00077$		-		$p = 0.00018$		-		-		$p = 0.30750$	
DTLZ4	0.1208	0.0380	-	-	0.2614	0.0485	-	-	0.0856	0.0113	0.3828	0.0277
	$p = 0.03760$		-		$p = 0.00018$		-		-		$p = 0.00018$	
DTLZ5	0.0039	0.0005	-	-	0.0176	0.0017	-	-	0.0102	0.0005	0.0057	0.0034
	-		-		$p = 0.00018$		-		$p = 0.00018$		$p = 0.00280$	
C2DTLZ2	0.0485	0.0046	-	-	0.05902	0.0020	-	-	0.0490	0.0017	0.1037	0.0268
	-		-		$p = 0.00025$		-		$p = 0.21210$		$p = 0.00018$	
Car Side	0.2148	0.0252	0.2128	0.0192	0.1992	0.0241	0.2691	0.0425	0.1530	0.0113	0.3915	0.08211
	$p = 0.00044$		$p = 0.00018$		$p = 0.00044$		$p = 0.00018$		-		$p = 0.00018$	
DTLZ2-5D	0.0416	0.0030	-	-	0.0929	0.0029	-	-	0.0840	0.0029	0.2512	0.0202
	-		-		$p = 0.00018$		-		$p = 0.00018$		$p = 0.00018$	
C2DTLZ2-5D	0.1510	0.0266	-	-	0.0720	0.0190	-	-	0.1792	0.0302	0.2311	0.0140
	$p = 0.00018$		-		-		-		$p = 0.00018$		$p = 0.00018$	

are also marked in bold with the respective p -value in Wilcoxon signed-ranked test.

For more investigation and analysis of the proposed metamodeling frameworks, we compared IGD and GD performance metric values of the proposed metamodeling frameworks with NSGA-II for solving the ZDT problems with 500 evaluations. It is clear from Table 4.3 and Table 4.4, that the proposed metamodeling frameworks show superior performance compared with NSGA-II with the same set of high-fidelity evaluations.

Table 4.2: Computed GD values for test problems. Best performing framework and other statistically similar frameworks are marked in bold.

Problems	M1-2		M2-2		M3		M4		M5		M6	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
ZDT 1	0.0132	0.0042	-	-	0.0087	0.0046	-	-	0.0022	0.0004	0.0300	0.0191
	$p = 0.00018$		-	-	$p = 0.00018$		-	-	-		$p = 0.00018$	
ZDT 2	0.0089	0.0037	-	-	0.00195	0.0007	-	-	0.00194	0.0004	0.0195	0.0156
	$p = 0.00018$		-	-	$p = 0.96980$		-	-	-		$p = 0.01400$	
ZDT 3	0.0086	0.0037	-	-	0.0067	0.0051	-	-	0.0222	0.0402	1.1084	0.1907
	$p = 0.08840$		-	-	-		-	-	$p = 0.03760$		$p = 0.00018$	
ZDT 6	2.2308	0.3263	-	-	2.6621	1.3698	-	-	0.4714	0.3616	6.7718	0.28976
	$p = 0.00018$		-	-	$p = 0.00100$		-	-	-		$p = 0.00018$	
BNH	0.1431	0.0382	0.1311	0.0255	0.1182	0.0037	0.1005	0.1172	0.1012	0.0222	0.1049	0.0094
	$p = 0.00056$		$p = 0.00130$		$p = 0.00270$		-		$p = 0.73260$		$p = 0.30570$	
SRN	0.7852	0.2083	0.7860	0.2462	2.8448	0.4905	1.5515	0.1578	1.7792	0.4990	0.9051	0.1479
	-		$p = 0.91590$		$p = 0.00018$		$p = 0.00016$		$p = 0.00044$		$p = 0.10410$	
TNK	0.0124	0.0046	0.0761	0.0265	0.0021	0.0001	0.0133	0.0028	0.0542	0.0315	0.0272	0.0049
	$p = 0.00018$		$p = 0.00018$		-		$p = 0.00018$		$p = 0.00018$		$p = 0.00018$	
OSY	0.7042	0.2461	4.5294	1.7321	1.1246	0.2822	7.9533	4.5279	21.9504	7.8221	31.3670	9.3676
	-		$p = 0.00018$		$p = 0.00580$		$p = 0.00018$		$p = 0.00018$		$p = 0.00018$	
Welded Beam	1.8407	1.4305	5.8158	4.9596	0.2495	0.1214	1.4903	1.1544	0.4935	0.4415	9.1873	10.9327
	$p = 0.00018$		$p = 0.00018$		-		$p = 0.00018$		$p = 0.38400$		$p = 0.00018$	
DTLZ2	0.0171	0.0032	-	-	0.0469	0.0042	-	-	0.0173	0.0008	0.0152	0.0024
	$p = 0.24130$		-	-	$p = 0.00018$		-	-	$p = 0.02830$		-	
DTLZ4	0.0353	0.0049	-	-	0.0438	0.0219	-	-	0.0287	0.0212	0.2555	0.0353
	$p = 0.18590$		-	-	$p = 0.06390$		-	-	-		$p = 0.00018$	
DTLZ5	0.0030	0.0007	-	-	0.0237	0.0035	-	-	0.0120	0.0040	0.0277	0.0345
	-		-	-	$p = 0.00018$		-	-	$p = 0.00910$		$p = 0.00018$	
C2DTLZ2	0.0461	0.0022	-	-	0.0434	0.0036	-	-	0.0294	0.0024	0.0819	0.0252
	$p = 0.00018$		-	-	$p = 0.00018$		-	-	-		$p = 0.00018$	
Car Side	0.8447	0.1059	0.8965	0.0946	0.0394	0.0018	0.0440	0.0022	0.0954	0.0296	0.0573	0.0079
	$p = 0.00018$		$p = 0.00018$		-		$p = 0.00170$		$p = 0.00044$		$p = 0.00018$	
DTLZ2-5D	0.0070	0.0002	-	-	0.0513	0.0190	-	-	0.0517	0.0012	0.3309	0.0346
	-		-	-	$p = 0.00017$		-	-	$p = 0.00017$		$p = 0.00017$	
C2DTLZ2-5D	0.0103	0.0005	-	-	0.0568	0.0031	-	-	0.176	0.0118	0.2327	0.0338
	-		-	-	$p = 0.00018$		-	-	$p = 0.00018$		$p = 0.00018$	

4.2 Two-Objective Constrained Problems

Next, we apply our frameworks to two-objective constrained problems: BNH, SRN, TNK, and OSY [11], For each problem, $H = 21$ reference directions, an initial sample size of $\rho = 400$, and a maximum of $SE_{\max} = 800$ high-fidelity SEs are fixed. The obtained non-dominated solutions are shown in Figures 4.5, 4.6, 4.8, and 4.7 for M1-2, M2-2, M3, M4, M5, and M6 frameworks, respectively.

All six frameworks are able to find a close and well-distributed set of trade-off points to the true Pareto-optimal front (shown by a solid line in each case) for BNH and SRN problems. With only $SE_{\max} = 800$ high-fidelity SEs used in our study, the performance of

Table 4.3: Computed IGD values for ZDT problems. Best performing framework and other statistically similar frameworks are marked in bold.

	M1-2		M3		M5		M6		NSGA-II	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
ZDT 1	0.0115	0.0006	0.0185	0.0046	0.0103	0.0028	0.0249	0.0153	0.3567	0.0935
ZDT 2	0.0105	0.0027	0.013	0.0013	0.0078	0.0011	0.0103	0.0051	0.7063	0.2306
ZDT 3	0.0136	0.0019	0.0254	0.0154	0.0246	0.0027	0.7111	0.2054	0.2801	0.0731
ZDT 6	1.6144	0.1965	0.3774	0.1099	0.1208	0.0311	5.9401	0.7896	5.0010	0.2883

Table 4.4: Computed GD values for ZDT problems. Best performing framework and other statistically similar frameworks are marked in bold.

	M1-2		M3		M5		M6		NSGA-II	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
ZDT1	0.0132	0.0042	0.0087	0.0046	0.0022	0.0004	0.03	0.0191	0.4056	0.1419
ZDT2	0.0089	0.0037	0.00195	0.0007	0.0019	0.0004	0.0195	0.0156	0.5768	0.1483
ZDT3	0.0086	0.0037	0.0067	0.0051	0.0222	0.0402	1.1084	0.1907	0.3488	0.1119
ZDT6	2.2308	0.3263	2.6621	1.3698	0.4714	0.3616	6.7718	0.2897	4.8678	0.3431

these frameworks is noteworthy. However, the problem TNK has provided difficulties to all six frameworks, due to discontinuities in its Pareto-optimal front. Although all frameworks come close to the true Pareto-optimal front, M3 performs the best, but as presented in Table 4.1, frameworks M1-2, M4, and M6 also perform well. In OSY, frameworks M3 and M1-2 perform the best. Although metamodels are constructed progressively, the independent approximation of constraints adopted by M3 and M1-2 produced more accurate results than the use of a combined constraint function in this difficult problem. Another aspect which has become clear from these results is that modeling of a combined constraint violation function (as in M2-2 and M4) is not better than modeling each constraint independently (as in M1-2 and M3). However, an integrated constraint handling using the aggregated ASF approach or the KKTPM approach is better.

Next, a two-objective welded-beam design problem [11] is solved using all six frameworks. Same parameter settings are used as in above two-objective problems except $SE_{\max} = 1,000$ and $\rho = 500$. Figure 4.9 shows the resulting non-dominated solutions.

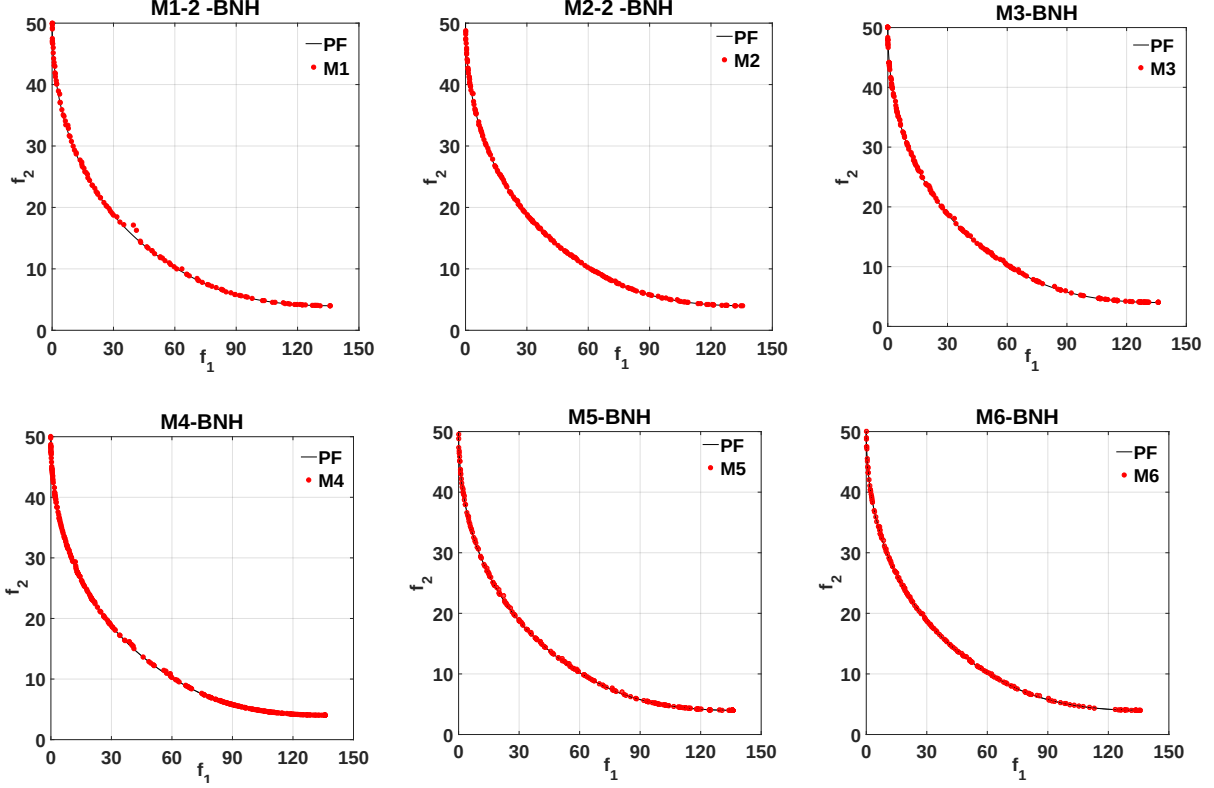


Figure 4.5: Obtained non-dominated solutions for BNH using frameworks M1-2, M2-2, M3, M4, M5 and M6 from left to right and top to bottom.

While M1-2 performs the best, M5 the second-best, all other frameworks find it difficult to find minimum-cost (f_1) solutions, which has also been reported to be challenging in another study [77]. This problem requires more high-fidelity SEs, but the use of a metamodeling based optimization approach with a limited high-fidelity SEs is not expected to find the exact Pareto-optimal solutions on the entire front in real-world problems. Moreover, each framework may require its own parameter set-up for optimal performance – a matter which we plan to pursue in the future.

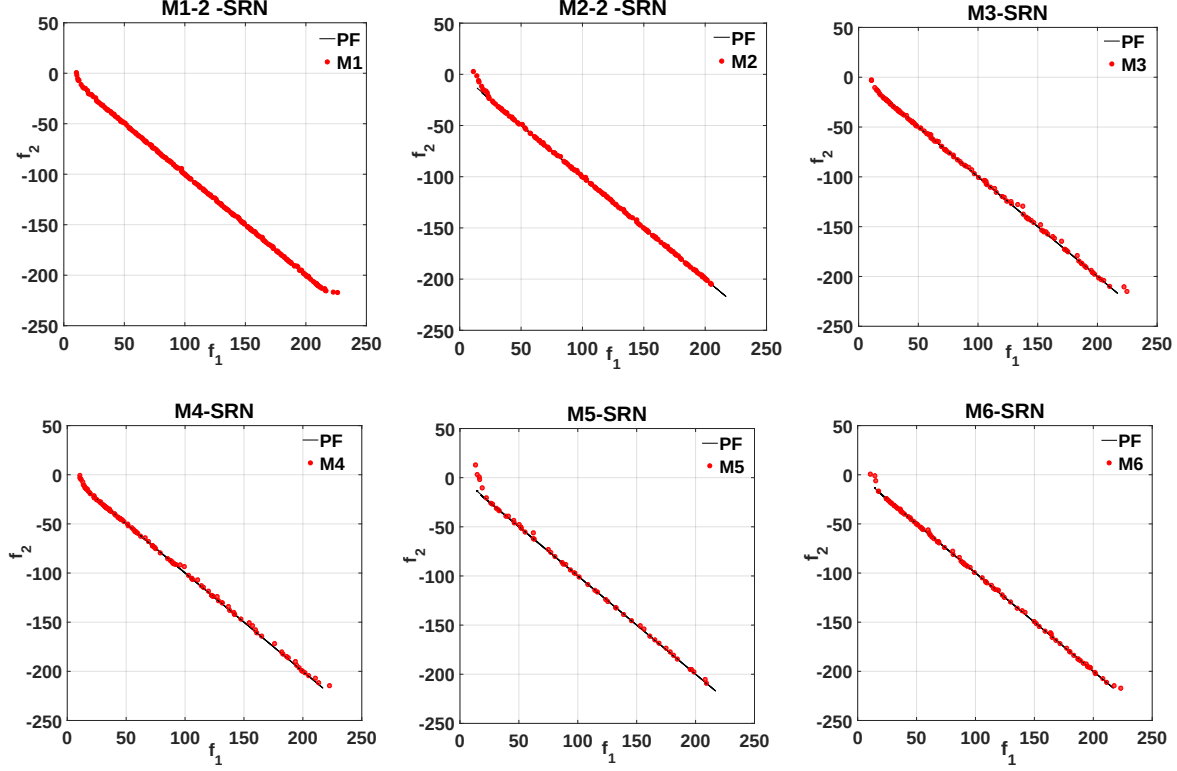


Figure 4.6: Obtained non-dominated solutions for the SRN problem using frameworks M1-2, M2-2, M3, M4, M5 and M6 from left to right and top to bottom.

4.3 Three-Objective Constrained and Unconstrained Problems

Next, we apply all six frameworks to three-objective optimization problems (DTLZ2, DTLZ4, and DTLZ5) and also to a three-objective constrained problem (C2DTLZ2). Each of these problems are considered for seven variables and considered for $H = 91$ reference directions. We fix $SE_{\max} = 1,000$ high-fidelity SEs and $\rho = 500$ for DTLZ2 and DTLZ5, and $SE_{\max} = 2,000$ and $\rho = 700$ for DTLZ4 due to multi-modality in its landscape. For C2DTLZ2, we have used $SE_{\max} = 1,500$ and $\rho = 700$.

The Pareto-optimal surface of DTLZ2, C2DTLZ2, DTLZ4, and DTLZ5 problems and the respective non-dominated solutions (all non-dominated solutions of the high-fidelity so-

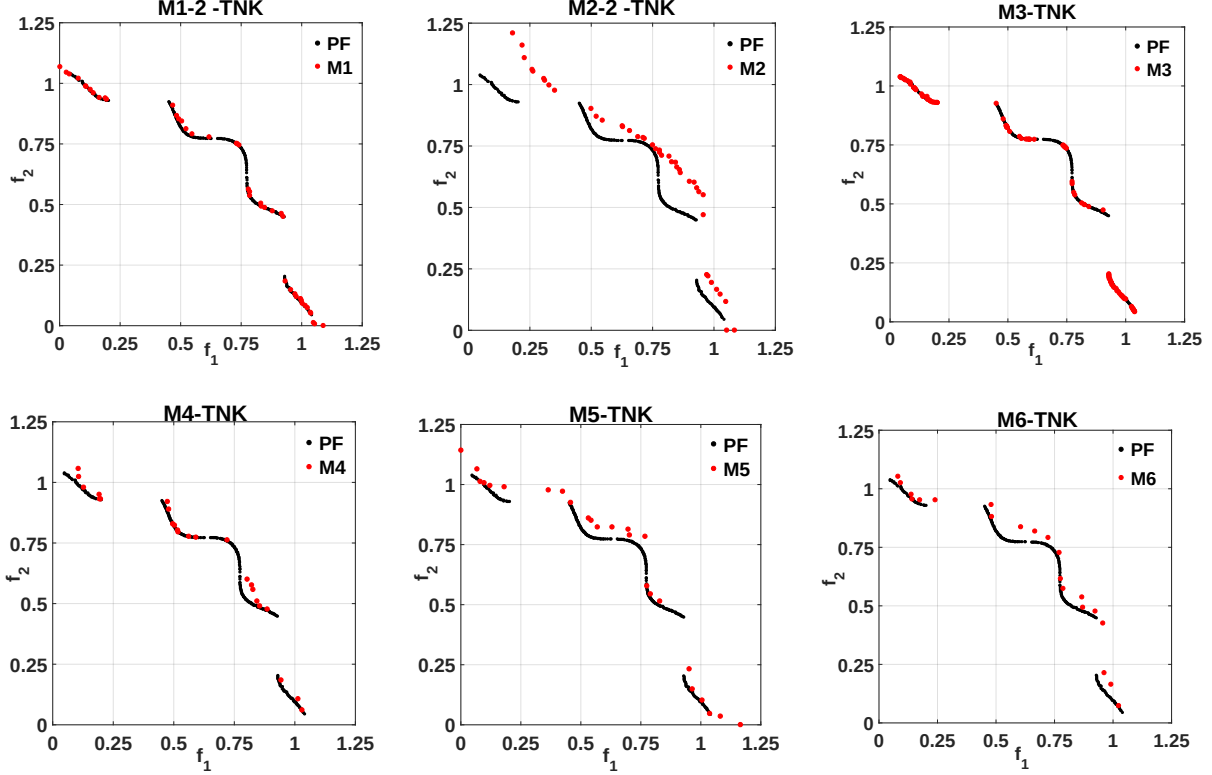


Figure 4.7: Obtained non-dominated solutions for the TNK problem using frameworks M1-2, M2-2, M3, M4, M5 and M6 from left to right and top to bottom.

lutions) are shown in Figures 4.10, 4.13, 4.11, and 4.12 for M1-2, M3, M5, and M6 frameworks, respectively. First, all four frameworks perform well, in general, on the DTLZ2 problem. Frameworks M5 and M6 perform the best on this problem. Then, on problem DTLZ4, framework M5 performs the best. On DTLZ5, all four frameworks performs well, with M1-2 performing the best, followed by M6. Thus, on unconstrained three-objective problems, frameworks M5 and M1-2 perform the best, followed by M6 and then M3.

On C2DTLZ2 problem, frameworks M1-2 and M5 perform the best, followed by M3. As noted before, since this problem has a single constraint, M1-2 and M2-2 frameworks will produce identical results, M3 and M4 will also produce identical results. Interestingly, framework M6 is able to find multiple near Pareto-optimal solutions through the multi-modal RGA approach; however, further developments are needed to understand full potential of

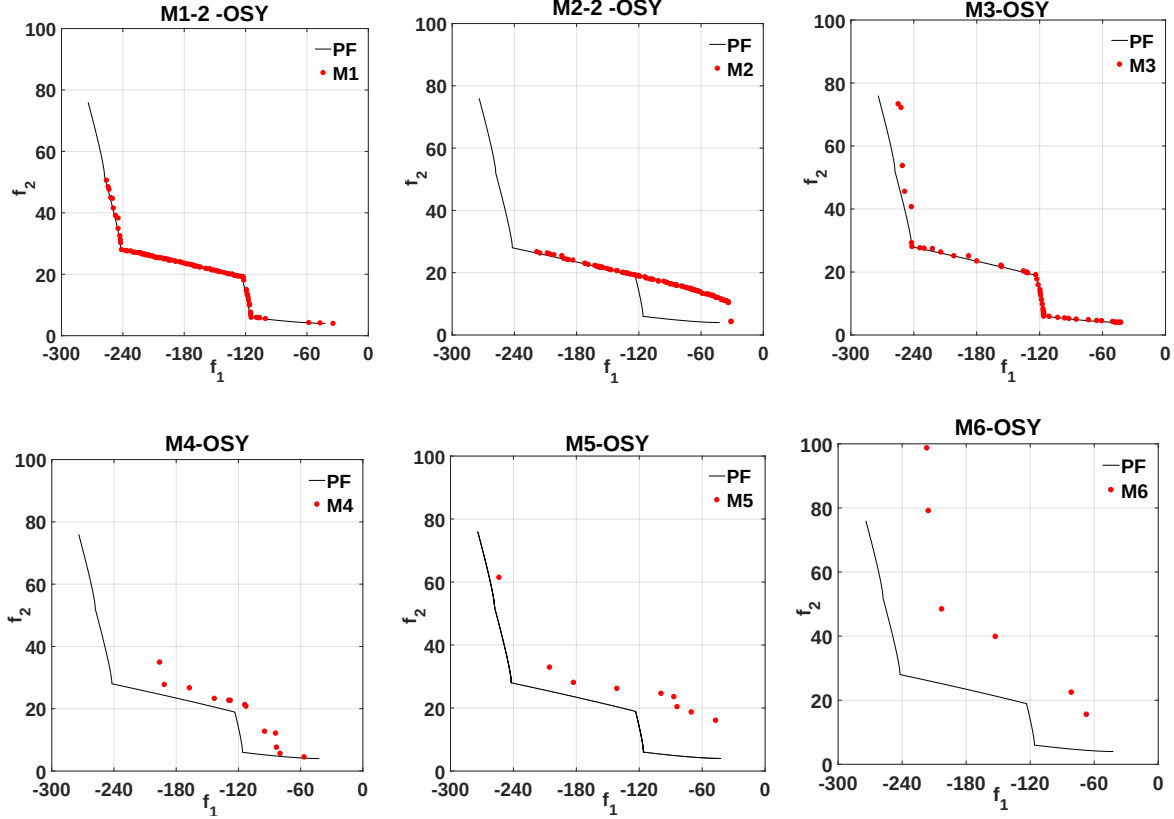


Figure 4.8: Obtained non-dominated solutions for OSY problem using frameworks M1, M2, M3, M4, M5 and M6 from left to right and top to bottom.

M6.

Finally, we apply all six frameworks to a car side-impact problem having three objectives and ten constraints. This problem allows us to apply each of our six frameworks on the same problem. Figure 4.14 presents all obtained non-dominated points by six methods with $SE_{\max} = 2,000$ and with an initial sample size of $\rho = 700$. Framework M5 performs the best on this problem, while M3, M4, and M6 are also able to find a wide distribution and well-converged set of solutions. However, the poor convergence (apparent from large GD values) and distribution of simplistic M1-2 and M2-2 frameworks for a relatively larger number of objectives and constraints is evident from the figure. As the number of objectives is more, M1-2 and M2-2 frameworks demand more metamodels (one for each objective function) to

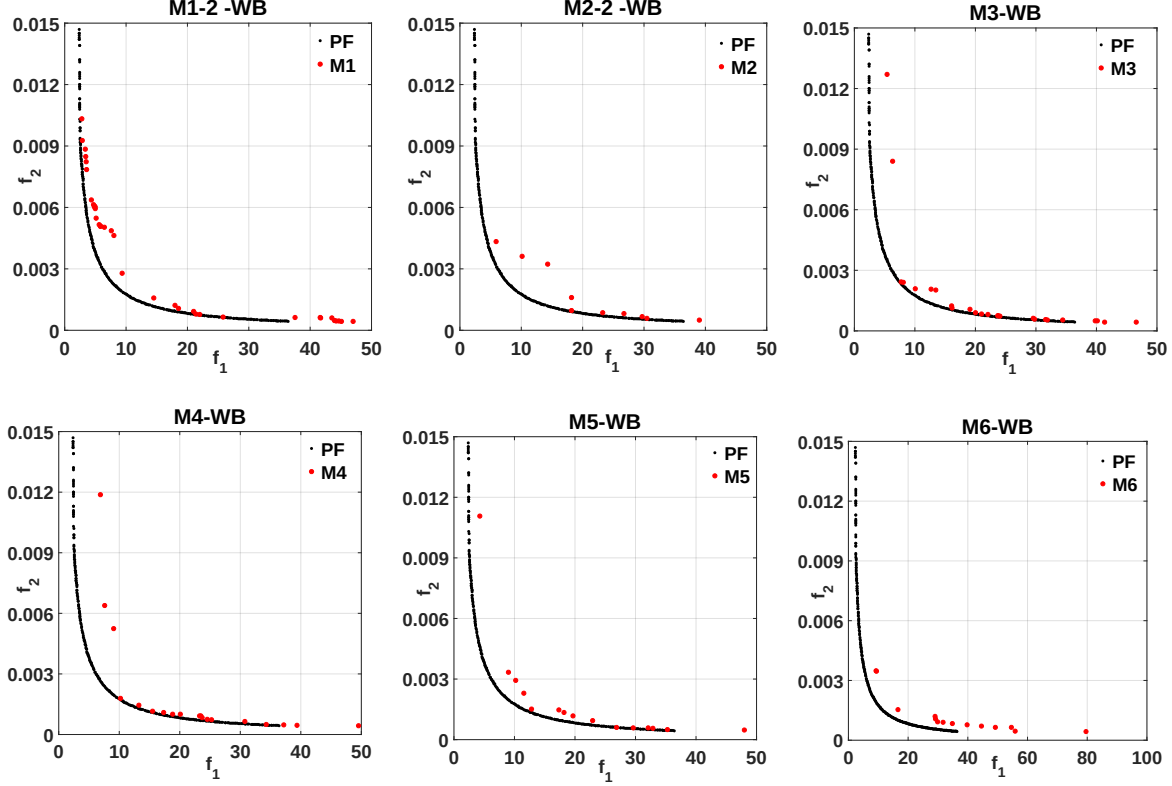


Figure 4.9: Obtained non-dominated solutions for the welded-beam problem using frameworks M1-2, M2-2, M3, M4, M5 and M6 from left to right and top to bottom.

be constructed. The error in each metamodel can be additive and becomes detrimental in properly determining non-domination level and NSGA-III's niching values. Although two to three-objective is not a big jump, it is interesting that M1-2 and M2-2 on this problem are not able to find widely distributed points near the true Pareto-optimal front. The ASF scalarization of objectives in M5 and M3 focuses each metamodel for finding a single Pareto-optimal point, thereby making an excellent performance on these problems with only 1,000 to 1,500 high-fidelity SEs. Another reason for M5's better performance is the use of efficient global optimization (EGO) method [4]. On the more complex problems (DTLZ4 (multi-modal) and C2DTLZ2 (constraints)), framework M6 gets overwhelmed in modeling the multi-modal KKTPM surface having 91 global optima. It is clear that more high-fidelity

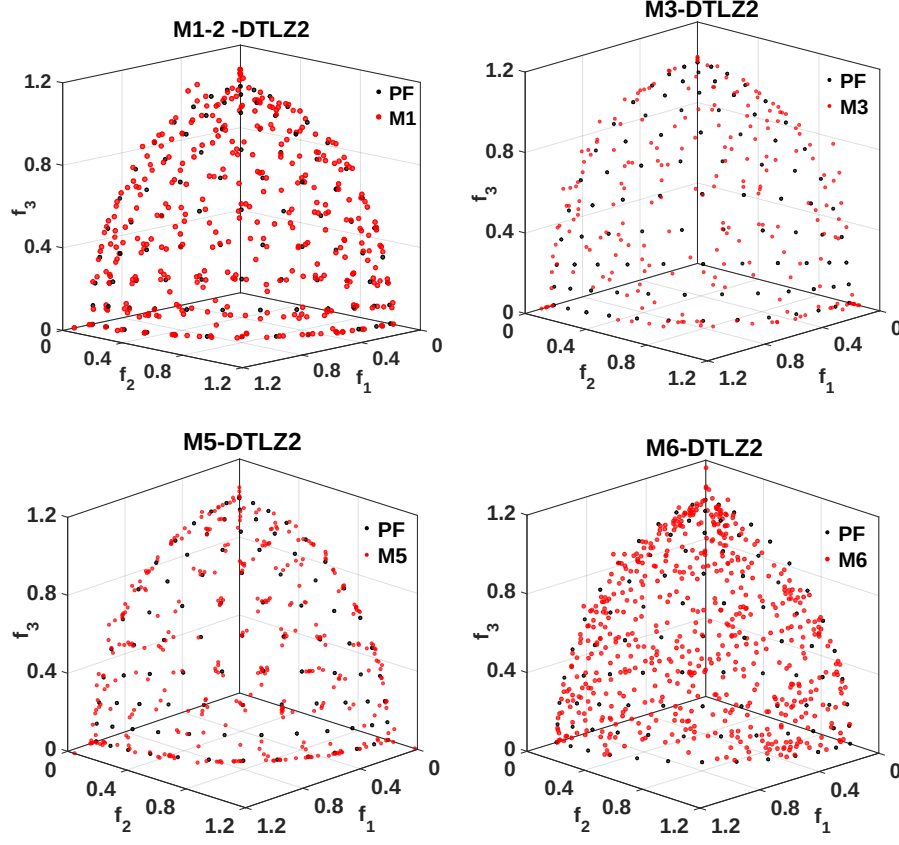


Figure 4.10: Obtained non-dominated solutions for problem DTLZ2 using frameworks M1-2, M3, M5 and M6 from left to right.

SEs are needed to get a better performance by M6 on these more complex problems, but the use of a single metamodel to solve a multi-objective optimization problem as a multi-modal optimization problem is algorithmically novel and challenging, and further studies are needed to fully exploit its potential.

4.4 Five-Objective Constrained and Unconstrained Problems

Finally, we apply all six frameworks to five-objective unconstrained DTLZ2 and to five-objective constrained C2DTLZ2 problem. Each of these problems is considered for seven

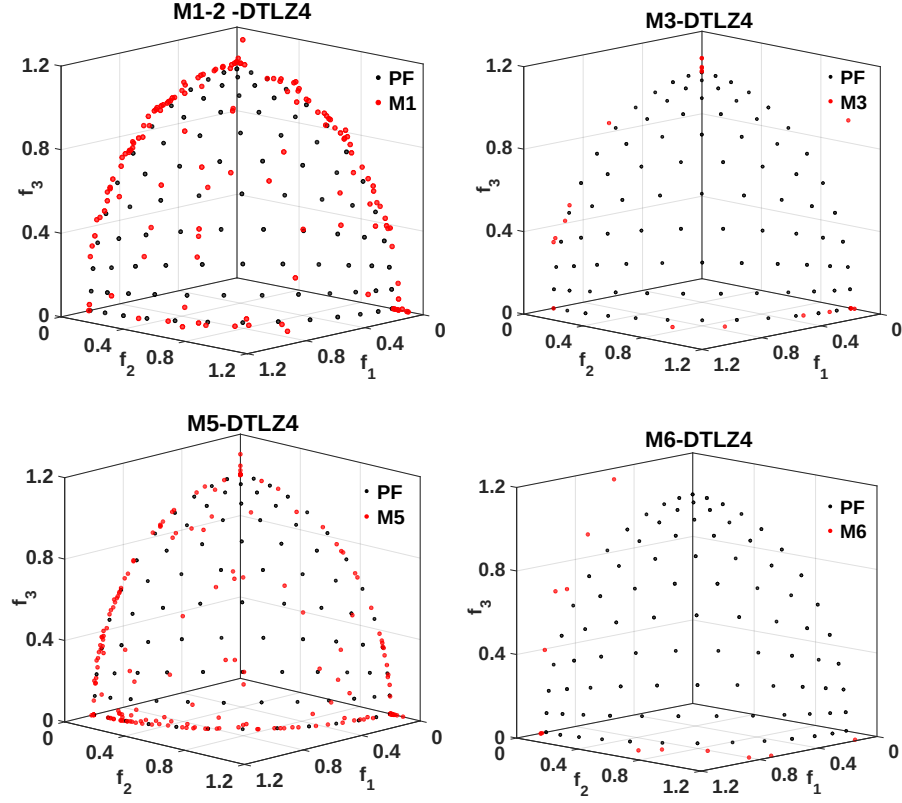


Figure 4.11: Obtained non-dominated solutions for problem DTLZ4 using frameworks M1-2, M3, M5 and M6 from left to right.

variables and considered for $H = 210$ reference directions. We fix $SE_{\max} = 2,500$ high-fidelity SEs and $\rho = 900$ for DTLZ2 and C2DTLZ2, respectively. According to the IGD performance metric value in Table 4.1, M1-2 performs the best on DTLZ2, followed by M5 and then M3. For C2DTLZ2, M3 performs the best, followed by M1-2 and M5. As mentioned earlier, more high-fidelity points are needed for M6 to work better, but these initial results demonstrate the relative performances of proposed six metamodeling frameworks in two to five-objective constrained and unconstrained problems.

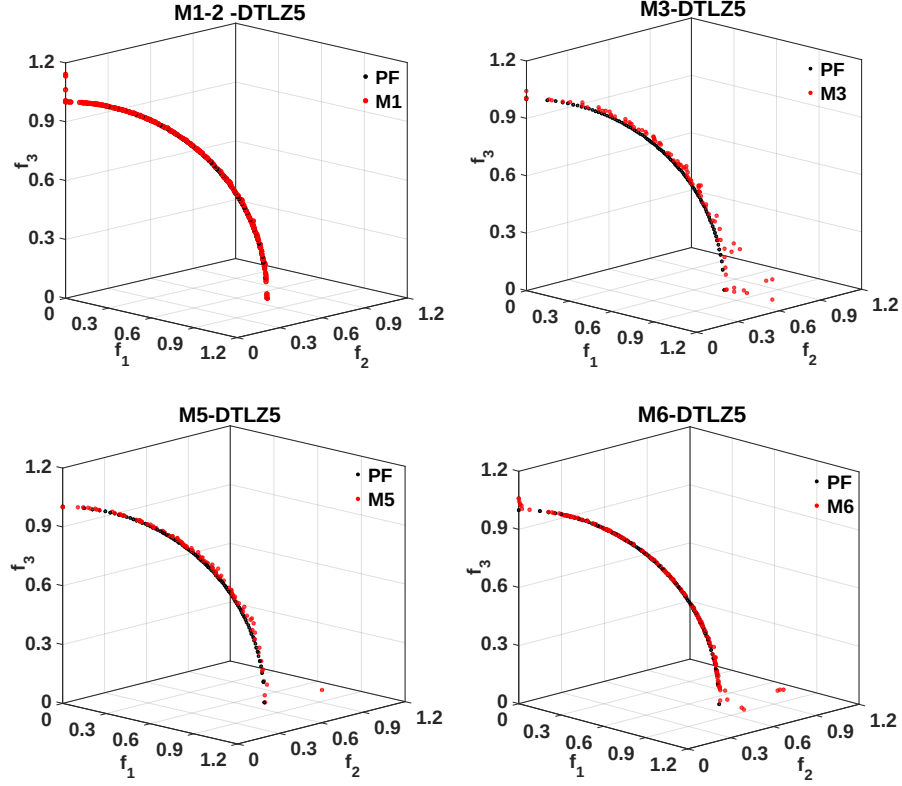


Figure 4.12: Obtained non-dominated solutions for problem DTLZ5 using frameworks M1-2, M3, M5 and M6 from left to right.

4.5 Summary

In this chapter, we have provided the results and comparison among different frameworks of the proposed taxonomy. Important conclusions about the behavior of each of the frameworks have been revealed. It is worth noting that although parameters of all six proposed frameworks have not been fine-tuned for their best performances, our obtained results are able to bring out the generic properties of different possible multi-objective metamodeling frameworks applied to distinct problem classes.

First, it is interesting to note that compared to standard EMO studies in which tens of thousands of solution evaluations are usually used to solve these test problems, here we present results that take only a fraction (a few hundred to a maximum of two thousand)

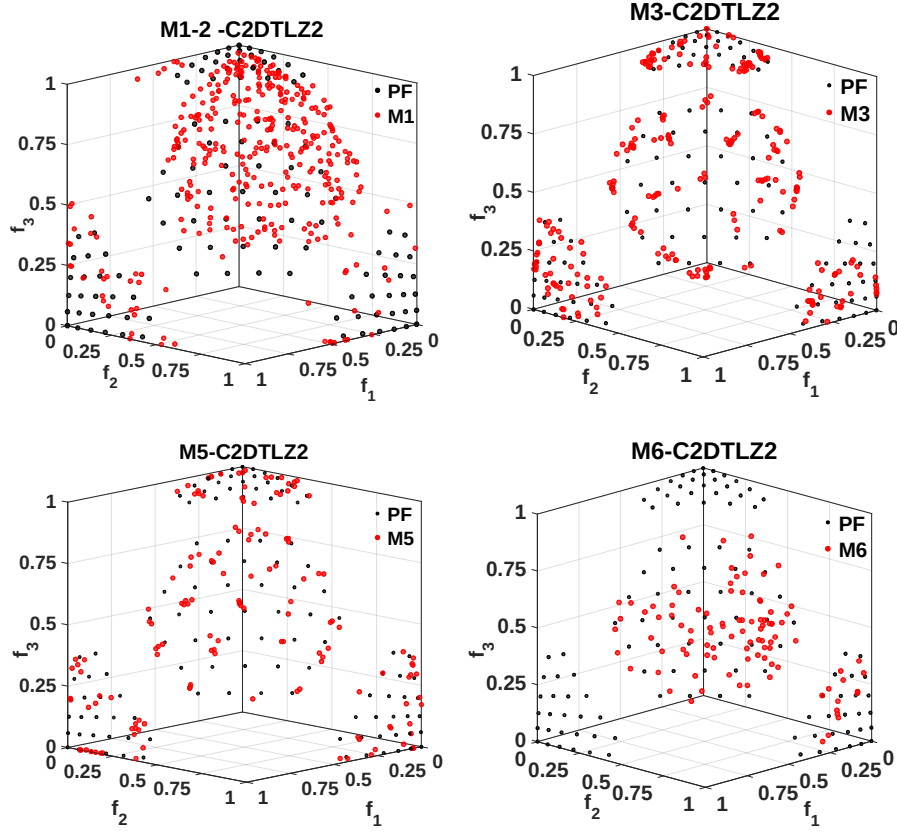


Figure 4.13: Graphical results for problem C2DTLZ2 using frameworks M1-2, M3, M5 and M6 from left to right.

of solution evaluations in each case to find a near Pareto-optimal set of solutions using a metamodeling based EMO approach. This is remarkable and the proposed metamodeling frameworks hold promise to the successful applications of metamodeling methods in evolutionary multiobjective optimization.

Second, there is a trade-off between the number of metamodeling efforts and resulting performance. It is intuitive that as the number of objectives increase, the straightforward implementation of metamodeling frameworks for each objective and constraint function may not be a computationally efficient approach. The flip side is that the resulting integrated functions become complex to the metamodel, thereby requiring more high-fidelity points. Although further studies are needed to make more confident conclusions, we have observed

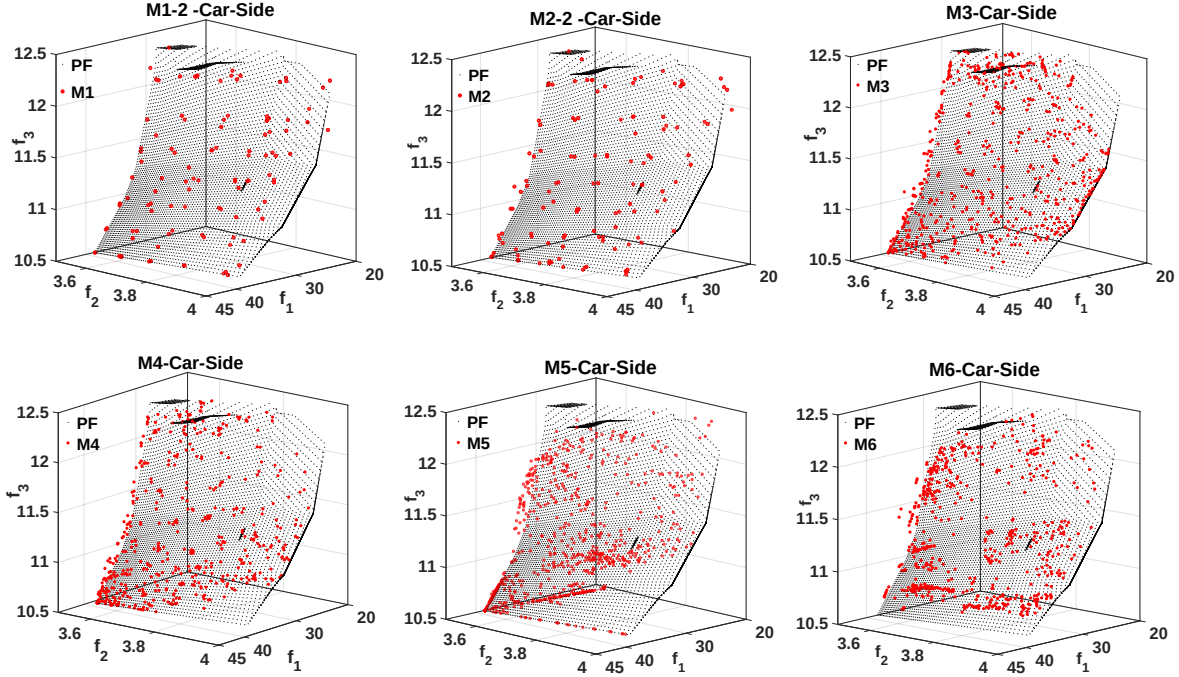


Figure 4.14: Obtained non-dominated solutions for problem car-side impact using frameworks M1-2, M2-2, M3, M4 M5 and M6 from left to right and top to bottom.

that a metamodeling of an aggregate function to find a single Pareto-optimal solution at a time is a better strategy, as evident from the better performance of frameworks M3 and M5. Both these frameworks constitute an implicit parallel approach, as intermediate points obtained for one reference line may lie in the vicinity of the Pareto-optimal solution of another neighboring reference line, thereby helping to create better metamodels for subsequent reference lines. This is one of the key reasons for the superior performance of M5.

Third, it is also clear that a metamodeling of a combined constraint violation function (M2 and M4) is not, in general, a better strategy compared to independent-constraint metamodeling frameworks M1-2 and M3 both in terms of convergence and diversity issues. However, when constraints are integrated with objective functions by a scalarization function (in M5), better performance has been achieved. The superiority of M5 in most problems of this study suggests that metamodeling of an implicit selection function involving all objective and con-

straint functions together either for finding a single solution at a time (or multiple solutions together) as an integrated approach is a better strategy. Framework M6 is interesting, but further investigation is needed with a different parameter setting to completely evaluate its merit.

Chapter 5

Adaptive Switching Between Frameworks and Use of Trust Regions

In this chapter, we describe our proposed adaptive switching method between different frameworks that we proposed previously. Then, we provide a review of the use of trust regions approach with metamodeling for the improvement of the convergence behavior of metamodeling frameworks.

5.1 Introduction

In multi-objective optimization, there are several ways to build and utilize metamodeling approaches as we mentioned in previous chapters. Here we argue that it is more efficient to use different metamodeling approaches at different stages of the optimization process and then propose several switching strategies between the metamodeling methods.

The developments in optimization methods have recently led to an increasing interest in metamodeling efforts. Most metamodeling studies in optimization (single or multiobjective) concentrate on handling unconstrained problems only. In some occasions, bounded variable problems are considered. But, this defeats the whole purpose of using a metamodeling approach in the first place. Metamodeling methods are needed mainly to solve practical problems, which are often computationally expensive. Constraints are also inevitable in

practical problems. Thus, addressing metamodeling methods to solve only unconstrained or bounded variable problems does not solve the issue completely. Also, extending an unconstrained optimization algorithm for constrained optimization is not trivial [78]. Hence, we believe that metamodeling methods in optimization should involve constraint handling to the extent possible.

We propose switching among proposed frameworks adaptively within the optimization procedure for obtaining better results than before. We have also implemented the trust regions concept for getting more robust solutions and reduce the uncertainty as well.

5.2 Literature Review

In this study [22, 79], we performed an extensive literature survey on studies involving metamodeling assisted EMO approaches. Since then, a few more interesting and relevant studies have come to our attention. Here we provide a brief review of them.

Zhao et al. [80] classified the sample data into clusters based on their similarities in the variable space. Then, a local metamodel was built for each cluster of the sample data. A global metamodel is then built using these local metamodels considering their contributions in different regions of the variable space. Due to the construction and optimization of multiple metamodels, one for each cluster, this method belongs to our M-3 framework. The use of a global metamodel by combining all local cluster-wise metamodels qualify this method under the M3-2 framework. No constraint handling method is suggested.

Zhang et al. [6] proposed an MOEA/D-EGO algorithm which metamodeled each objective independently. They constructed multiple expected global optimization (EGO) functions for multiple reference lines of the MOEA/D approach to find a number of trade-off solution in

each optimization task. No constraint handling procedure was suggested. Thus, this method falls under our M1-2 framework.

Chugh et al. [81] proposed a surrogate-assisted adaptive reference vectors guided evolutionary algorithm (K-RVEA) for multi-objective unconstrained optimization problems. Each objective function is metamodeled independently using a Gaussian process approach (Kriging). According to our proposed taxonomy [79], both procedures fall under the M1-2 framework.

Datta et al. [82] proposed a surrogate-assisted evolution strategy for constrained multi-objective optimization (SMES) using Radial Basis Function (RBF) to metamodel each objective and constraint function. Due to this treatment, this method falls under our M1-2 framework. Bhattacharjee et al. [83] used multiple spatially distributed surrogates, in which each surrogate is modeled using multiple methods, such as Radial Basis Function (RBF), Kriging, and Response Surface Methodology (RSM), in different regions of the search space. Both algorithms are build on NSGA-II platform [13]. Since one surrogate model in both the above procedures is constructed for each objective and constraint function separately, they fall under our M1-2 framework.

Rahat et al. [84] proposed an approach which can either construct a scalarized objective from individual metamodels of objective functions or metamodel a scalarized objective function. They provided an user choice for this option. These methods can be classified as our M1-1 and M3-1 frameworks, respectively. Three different scalarizing functions – hypervolume improvement, dominance ranking, and minimum signed distance – are used in their study. Kriging was used as the metamodeling method, but no constraint handling strategy was proposed.

Pan et al. [85] proposed a classification based surrogate-assisted evolutionary algorithm

(CSEA) for solving unconstrained optimization problems by using an artificial neural network (ANN) as a surrogate model. The surrogate model aims to learn the dominance relationship between the candidate solutions and a set of selected reference solutions. This algorithm falls in M3-2 framework, more details of which will be provided in Part-II of this study.

Nghia Le et al. [86] proposed multiple co-objective evolutionary optimization by cross-surrogate assisted memetic augmentation algorithm (CSAMA). In this algorithm, standard quadratic regression or polynomial regression as a surrogate model is employed for construction of each objective function separately. For selecting best solutions, they used NSGA-II algorithm as a global search, Thereafter, the Tchebycheff scalarization function is applied as a local search on aggregated objective function. The CSAMA algorithm is implemented for unconstrained optimization problems only. This algorithm falls under M1-2 (for the global search) and M1-1 (for the local search) frameworks according to our taxonomy.

The above descriptions indicate how our proposed taxonomy [79] includes most of the possible and existing multi-objective metamodeling approaches. We choose three of the above metamodeling-based EMO approaches – MOEA/D-EGO, K-RVEA, and CSEA – as a basis of comparison of our proposed adaptive switching approach.

5.3 Trust Regions

A well-known concept in optimization is to maintain a balance between exploration versus exploitation [87]. Exploration relates to focusing on unknown areas in the design space in order to increase the information gain when a new point is evaluated. On the other hand, exploitation searches only in the areas that seem to be promising. Intuitively, after having explored the search space enough, a gradual switch to more exploitation makes sense. Some

metamodels address this issue by providing an uncertainty measure that can be used in the algorithm [88]. However, we address the exploitation versus exploration trade-off in the algorithm independently from the used metamodel by introducing a trust region concept.

Whenever metamodels are used, some approximation to the true objective and constraint values is introduced. This case an uncertainty to exist in the process, particularly when a stochastic metamodeling approach is adopted. Predictions close to high-fidelity observations are more accurate than predictions far from them. Therefore, we propose a trust region with a radius R_{trust} in order to address the amount of uncertainty allowed within our optimization process. If R_{trust} is infinity, almost any solution is accepted as an in-fill solution, and an in-fill point evaluated well by the metamodel may not fair well on the exact model. Contrarily, if R_{trust} is close to zero, metamodel may be well correlated with the exact model, but the search power is lost. To have a better exploration, we also introduce another radius R_{prox} , within which of an evaluated point no other points can be considered. This is because evaluating points close to existing ones will not change the metamodel in a significant manner. For multi-objective optimization, R_{prox} aids in achieving multiple Pareto-optimal solutions with a variable space diversity from each other. Figure 5.1 shows the use of one such strategy proposed by COIN laboratory researchers recently [89].

Combining both the above concepts, we enforce a feasible search region R_{search} around every exactly evaluated solution having $R_{prox} \leq R_{search} \leq R_{trust}$. Figure 5.1 shows different R_{search} settings during an optimization run and visualizes the gradual movements from exploration to exploitation. We also reduce the two radii after every metamodeling task by constant factors: of 0.75 and 0.1 for $R_{trust}^{new} = 0.75R_{trust}^{old}$ and $R_{prox}^{new} = 0.1R_{prox}^{old}$. A reduction of two radii helps in achieving more trust on exactly evaluation solutions.

For the optimization procedure, two additional constraints are introduced. First, the

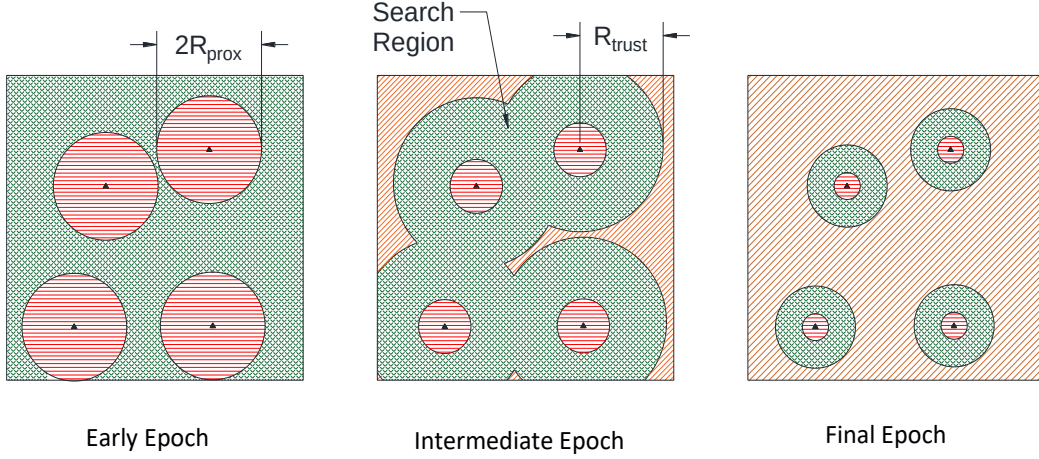


Figure 5.1: Proximal and Trust Region.

trust region R_{trust} which is violated when the minimum distance to any existing point is larger than the R_{trust} . Second, the proximate region that is violated when the distance to any existing point is smaller than R_{prox} . Each non-dominated solutions from NSGA-II generations are included in the archive. Then R_{prox} and R_{trust} are decreased by multiplied constant parameter less than one. After τ generations, new metamodels are created again and the process is repeated until termination.

5.4 Results of Trust Region

In this section, we only show the results of M1-2 framework for comparing the performance of metamodeling framework without and with using trust region. The same parameters setting for NSGA-II that were produced in chapter 4 are used here. For the trust regions method, an initial value of $R_{trust} = 0.75\sqrt{M}$ (M is the number of objectives) is used.

5.4.1 Two-Objective Unconstrained Problems

First, we apply our proposed algorithm to two-objective unconstrained problems ZDT1, ZDT2, ZDT3, and ZDT6 with ten ($n = 10$) variables, $H = 21$ reference directions, and with a maximum of only $SE_{\max} = 500$ high-fidelity solution evaluations. The obtained non-dominated solutions are shown in Figures 5.2, for framework M1-2. The results show that M1-2 with trust regions method is better than without it.

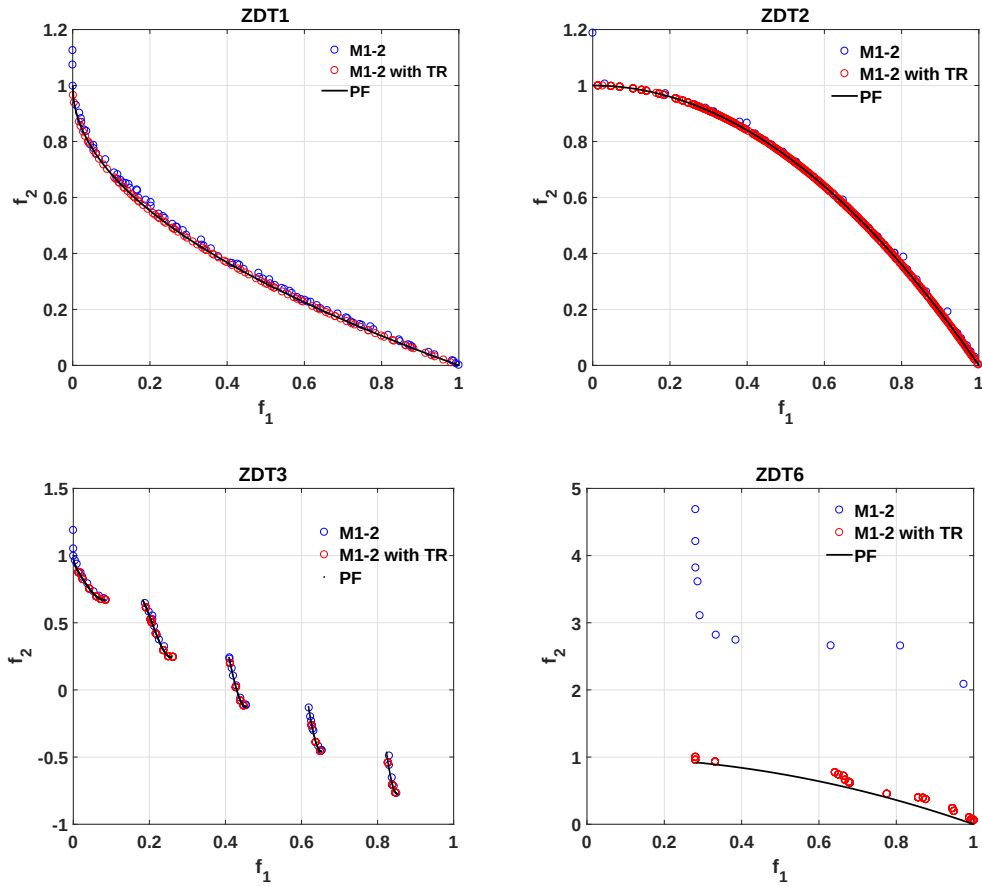


Figure 5.2: Non-dominated solutions for ZDT1, ZDT2, ZDT3, and ZDT6 problems using framework M1-2 from left to right.

5.4.2 Two-Objective Constrained Problems

Next, we apply our algorithm to two-objective constrained problems: TNK, OSY, BNH, and SRN [11], For each problem, $H = 21$ reference directions and a total of 1,000 solution evaluations. The obtained non-dominated solutions are shown in Figures 5.3 for M1-2 framework. Similar conclusions can be made about the use of trust regions method with M1-2.

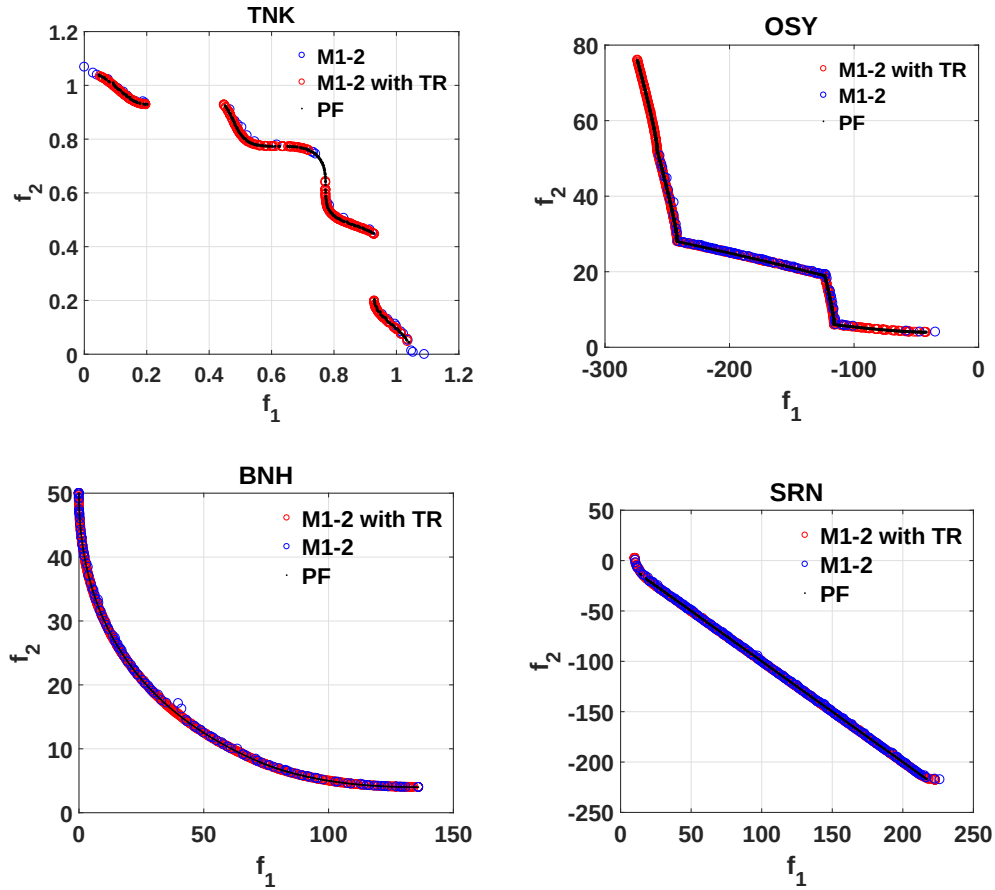


Figure 5.3: Non-dominated solutions for TNK, OSY, BNH, and SRN problems using framework M1-2 from left to right.

Median IGD values are presented in Table 5.1 for M1-2 framework for 10 runs each. The best performance of algorithm are marked in bold. Results in Table 5.1 indicate that M1-2 metamodeling framework perform better with the trust regions method on *all* problems.

Table 5.1: Computed IGD values for test problems. Best performing are marked in bold. W/TR: Without Trust Region.

ZDT1		ZDT2		ZDT3		ZDT6	
W/TR	TR	W/TR	TR	W/TR	TR	W/TR	TR
0.0115	0.0077	0.0105	0.0015	0.136	0.0131	1.6144	0.2726
TNK		OSY		BNH		SRN	
W/TR	TR	W/TR	TR	W/TR	TR	W/TR	TR
0.0173	0.0062	6.9534	0.3303	0.5322	0.1121	0.7452	0.2966

5.5 Adaptive Switching between Frameworks

Each metaomodeling framework in our proposed taxonomy requires to build metamodels for different individual or aggregated objective and constraint functions. Thus, it is expected that each framework may be most suitable for certain function landscapes that produce a smaller approximation error, but that framework may not fair well in other landscapes. During an optimization process, an algorithm usually faces different kinds of landscape complexities from start to finish. Thus, no one framework is expected to perform best during each step of the optimization process. While each framework was applied to different multi-objective optimization problems in our previous studies [22, 79] from start to finish, different problems were found to be solved best by different frameworks. To determine the best performing framework for a problem, a simple-minded approach would be to apply each of the 10 frameworks (or six frameworks of this study) to solve each problem independently using $SE_{\max}$ high-fidelity evaluations, and then determine the specific framework which performs the best using an EMO metric, such as hypervolume or IGD. This will be computationally expensive, requiring 10 (or six) times more than the prescribed $SE_{\max}$. If each framework is allocated only 1/10 (or 1/6) of $SE_{\max}$, they may be insufficient to find comparatively good solutions. A better approach would be use an adaptive switching strategy, in which the most suitable framework is chosen at every step of the optimization process. In this section, we

propose one such adaptive switching strategy.

We call a 'step' during the optimization process for assessing different metamodeling frameworks to choose the best-performing framework as an *epoch*. In each epoch, exactly H new infill solutions are created, thereby consuming H high-fidelity SEs. Clearly, the maximum number of epochs allowable is $E_{\max} = \lceil \frac{SE_{\max} - N_0}{H} \rceil$ with a minor adjustment on the SEs used in the final epoch. At the beginning of each epoch (say, t -th epoch), we have an archive ($\mathcal{A}_t$) of N_t high-fidelity solutions. For the first epoch, these are all N_0 LHS solutions, and in each subsequent epoch, H new infill members are added to the archive. At the start of t -th epoch, each of the six participating frameworks (M1-1, M1-2, M2-1, M3-1, M4-1 and M5) are used to construct its respective metamodels using all N_t archive members. Then, a 10-fold cross-validation method (described in Section 5.5.2) is used with a suitable performance metric (described in Section 5.5.1) to determine the next suitable framework for the next epoch. A flowchart of the proposed adaptive switching based metamodeling strategy is illustrated in Figure 5.4. In the following subsection, we describe a new performance metric used for choosing the selected framework.

5.5.1 Performance Metric for Framework Selection

We propose a *selection error probability* (SEP) metric which is appropriate for an optimization task. SEP is defined as the probability of making an error in correctly predicting the better of two solutions compared against each other using the constructed metamodels. Consider Figure 5.5, which illustrates an minimization task and comparison of three different population members pair-wise. The true function values are shown in solid blue, while the predicted function values are shown in dashed blue. When points x_1 and x_2 are compared based on predicted function, the prediction is correct, but when points x_1 and x_3 are com-

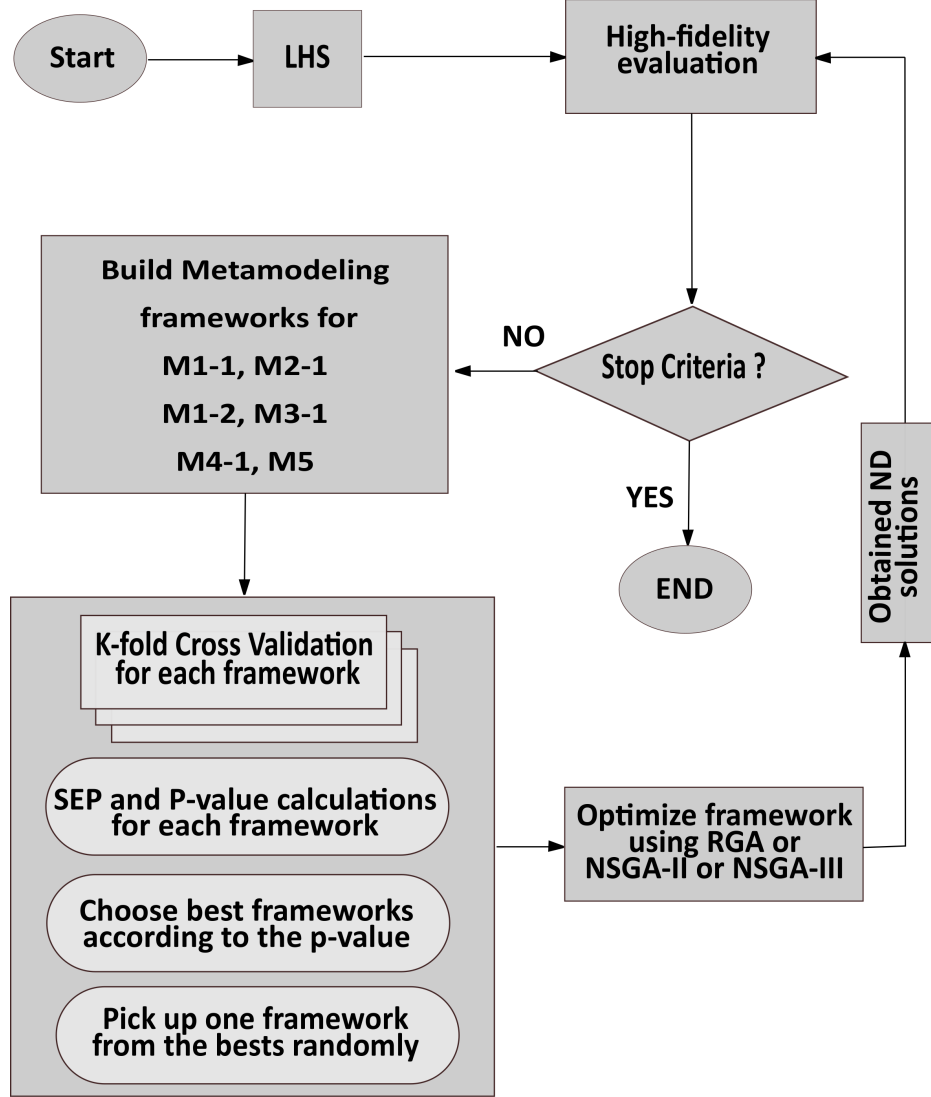


Figure 5.4: Flowchart of the proposed G-ASM method.

pared, the prediction is wrong. Out of three pairwise comparisons, two predictions are correct and one is wrong, thereby making a selection error probability of $1/3$ in this case. We argue that in an optimization procedure, it is the SEP which provides a better selection error than the actual function values. Unfortunately, many optimization methods borrow the mean squared error (MSE) metric, commonly-used in regression and metamodeling studies, but such a metric may not be the most appropriate metric for metamodeling-based approaches in optimization.

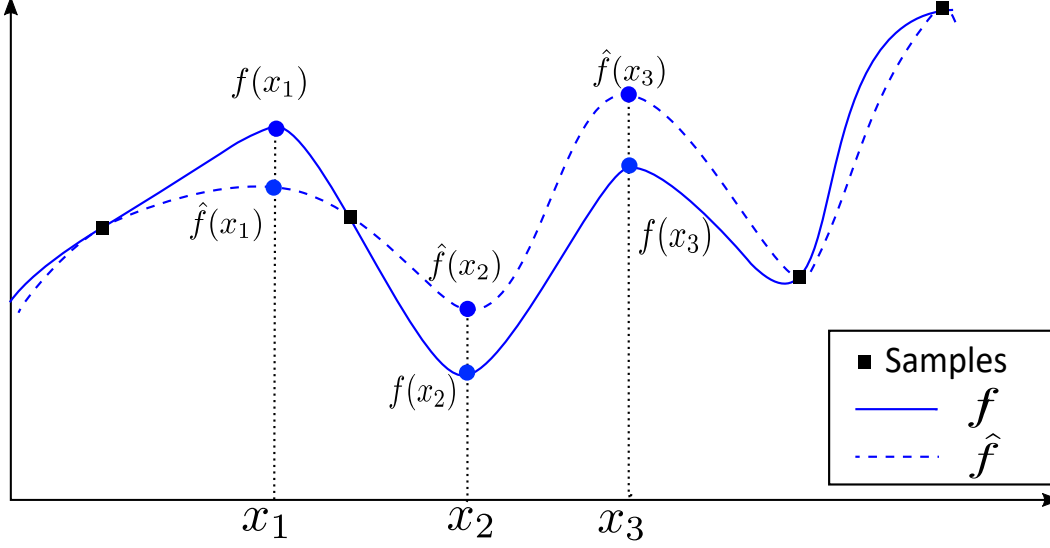


Figure 5.5: Selection Error Probability (SEP) concept is illustrated.

Mathematically, the SEP metric can be defined for n points as follows. For each of $N = \binom{n}{2}$ pairs of points (p and q), evaluate the selection error function ($E(p, q)$), which is one, if there is a mismatch between predicted winner and actual winner of p and q ; zero, otherwise. Then, SEP is calculated as follows:

$$\text{SEP} = \frac{1}{N} \sum_{p=1}^{n-1} \sum_{q=p+1}^n E(p, q). \quad (5.1)$$

The definition of a ‘winner’ can be easily extended to multi-objective and constrained multi-objective optimization by considering the domination [19] and constraint-domination [11] status of two points p and q .

5.5.2 Identifying a Suitable Framework for Next Epoch

A framework with a smaller SEP metric value over a 10-fold cross-validation is considered better. We describe the framework selection procedure in the following. At the end of each epoch, the new infill points are evaluated using high-fidelity evaluations and added to

the archive. Then, 90% of archive points are chosen at random and respective metamodels are constructed according to the framework’s plan described above. Then, the constructed metamodels are used to compare every pair (p and q) of the remaining 10% of archive points. Constrained domination checks are performed to establish the superiority of one point over the other and compared with that on the original problem. Then, the selection error function ($E(p, q)$) and finally the SEP are computed. The above process is repeated 10 times by using different blocks of 90% points and 10 different SEP values are obtained for each framework. Notice that these 10-fold cross-validation procedure does not require any new solution evaluations, as the whole computations are performed based on the archive points, which were already evaluated using high-fidelity computational procedures. Thereafter, the best framework ($\mathcal{M}_b$) is identified based on the median SEP value of frameworks. Finally, the Wilcoxon rank-sum test is performed between $\mathcal{M}_b$ and each other framework. All frameworks within a statistical insignificance (having $p > 0.05$) are identified. Then, a random framework from the statistically insignificant frameworks including $\mathcal{M}_b$ is selected for the next epoch. Since each of these frameworks performs similar to $\mathcal{M}_b$ in a median sense, the choice of a random framework helps to use a diverse metamodeling landscapes for the search from one epoch to another, thereby prohibiting the overall method to not get stuck in similar search behaviors.

5.6 Results

The RGA uses the simulated binary crossover (SBX), and polynomial mutation, with parameters as follows: Number of generations = 100, crossover probability = 0.95, mutation probability = $1/n$, distribution index for SBX operator = 1, and distribution index for poly-

nomial mutation operator = 10. The NSGA-II procedure used in M1-2 is also applied with the same parameter values as above. For one run of RGA or NSGA-II, we use a maximum of $\tau = 50$ generations. We perform 11 runs of all metamodeling frameworks on all test problems. For unconstrained problems (ZDT) with no constraints and three-objective problem C2DTLZ2 having a single constraint, two frameworks M1-1 and M2-1 are identical. In such a case, we only show the results for M1-1. The same situation occurs for M3-1, M4-1, and M5 for unconstrained problems. For We only show the results of M3-1 for such cases. For C2DTLZ2 problems, M5 is different from M3-1 and M4-1. It is worth mentioning here, we do not make any effort to choose optimal parameter setting for each algorithm, rather identical parameters are used to provide a representative performance of each methodology. In Table 5.2, we summarize the other parameter setting for all test problems used in this study.

Table 5.2: Parameter settings used in the study.

Problem	n	M	J	N_0	$SE_{\max}$	#epochs	H
ZDT1	10	2	0	100	500	20	21
ZDT2	10	2	0	100	500	20	21
ZDT3	10	2	0	100	500	20	21
ZDT4	5	2	0	100	1000	43	21
ZDT6	10	2	0	100	500	20	21
OSY	6	2	6	200	800	29	21
TNK	2	2	2	200	800	29	21
SRN	2	2	2	200	800	29	21
BNH	2	2	2	200	800	29	21
WB	4	2	4	300	1000	39	21
DTLZ2	7	3	0	500	1000	6	91
C2DTLZ2	7	3	1	700	1500	9	91
CAR	7	3	10	700	2000	15	91
DTLZ5	7	3	0	500	1000	6	91
DTLZ4	7	3	0	700	2000	15	91
DTLZ7	7	3	0	500	1000	6	91
DTLZ2-5	7	5	0	700	2500	9	210
C2DTLZ2-5	7	5	1	700	2500	9	210

5.6.1 Two-Objective Unconstrained Problems

First, we apply our proposed methodologies to two-objective unconstrained problems ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6 with only $SE_{\max}=500$ or 1,000 high-fidelity evaluations for ZDT4 only. IGD values are presented in Table 5.3. It is clear that except in ZDT4, in other four problems, the generative adaptive switching based metamodeling (G-ASM) method performs either equivalent or better than individual metamodeling frameworks. In ZDT2 and ZDT6, G-ASM performs the best in terms of median performance. Interestingly, M1-1 as a single metamodeling framework performs well in three of the five ZDT problems, but G-ASM performs well in four of the five problems. M1-2 and M3-1 performs well in two of the five problems.

Non-dominated solutions for the median run of the G-ASM are shown in Figure 5.6.

In order to understand the switching mechanism of G-ASM, we calculate the proportion of the times (out of 11 runs) a framework was chosen for the best performing list of frameworks at the end of each epoch. Figure 5.7 marks this proportion in black and white shading.

Corresponding switching plots for the median run are shown in Figure 5.8. Switching among different frameworks is clear from the plot. These plots support G-ASM’s ability to switch between different frameworks to obtain a better overall performance at the end.

5.6.2 Two-Objective Constrained Problems

Next, we apply G-ASM to two-objective constrained problems: BNH, SRN, TNK, OSY [11], and the welded beam design problem (WB). Table 4.1 presents the IGD values of individual frameworks and G-ASM. It can clearly seen that in all five problems, G-ASM performs equally well or better than individual frameworks. No individual framework work well on

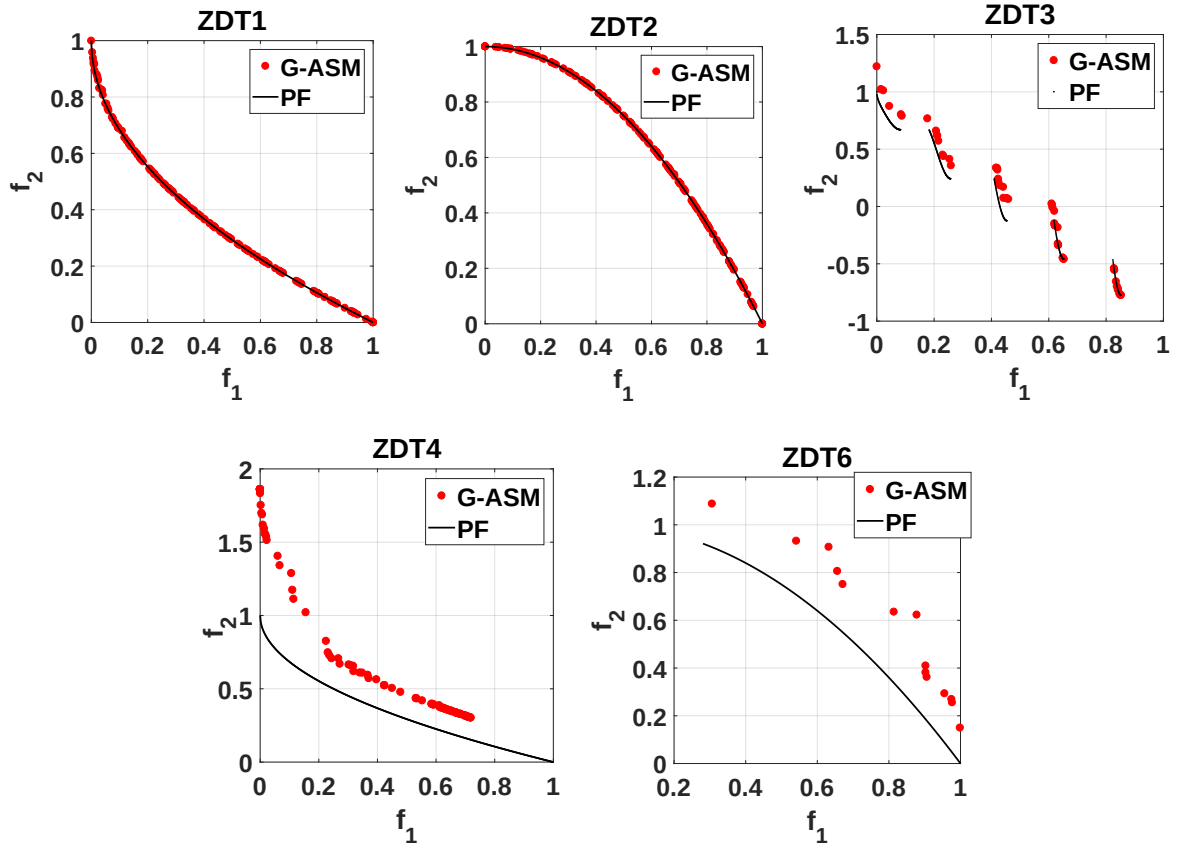


Figure 5.6: Obtained non-dominated solutions of the final archive for the median G-ASM run for ZDT problems.

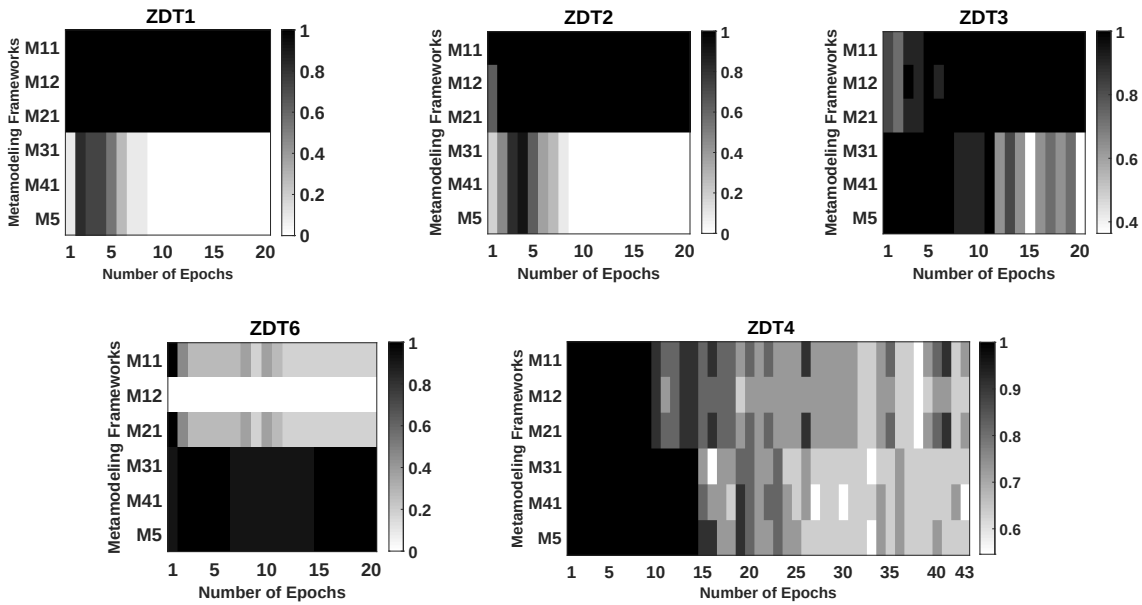


Figure 5.7: Adaptive switching method of six frameworks among average runs for two objectives unconstrained problems.

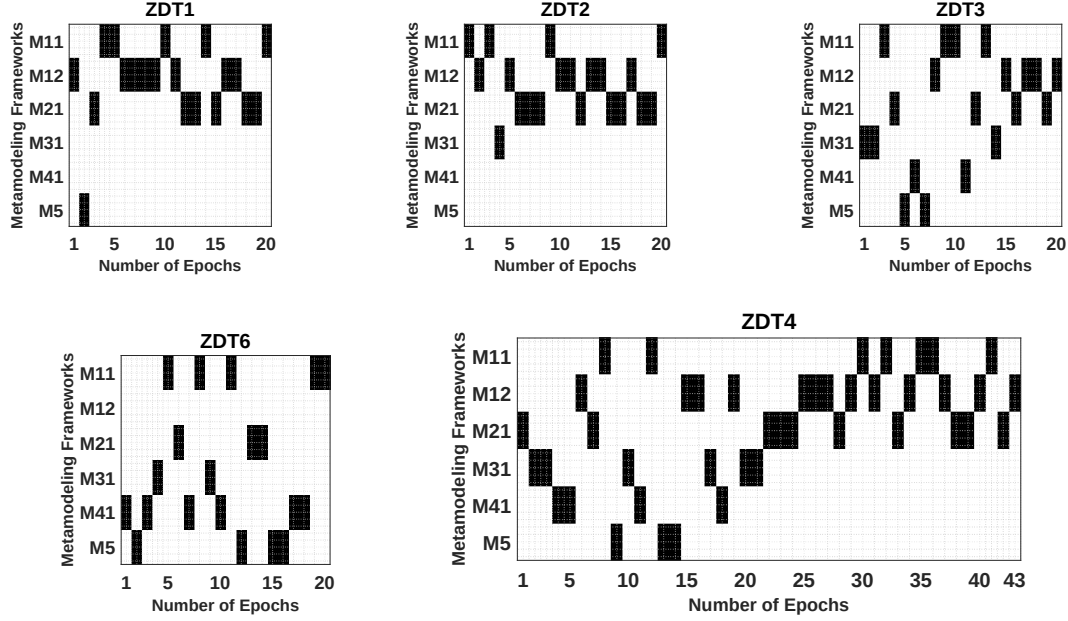


Figure 5.8: Adaptive switching method of six frameworks for median IGD run for two objectives unconstrained problems.

all five constrained problems. Figure 5.9 shows all final non-dominated archive members for the median performing G-ASM run. Such performances with only a total of 800 high-fidelity SEs are remarkable.

The median run switching plots are shown in Figure 5.10. These plots are similar to the ones shown for unconstrained two-objective problems. For example, M1-1 (marked as M11) is always found to be the only best performing framework in all epochs from start to finish for OSY. This is supported by the smallest IGD value of 0.15323 for M1-1 in Table 4.1. By choosing the M1-1 in all epochs by our G-ASM method, it is able to produce statistically similar result on OSY. In WB, G-ASM switches among the six approaches to produce the best IGD value (0.12819), compared to all other individual frameworks. For SRN, BNH and TNK problems, G-ASM finds the frameworks M1-1 and M1-2 mostly useful. A better picture emerges when we plot the switching strategy for a single (median IGD run) in Figure 5.11 for OSY, SRN, BNH, TNK and WB. G-ASM is able to use a switching of frameworks to

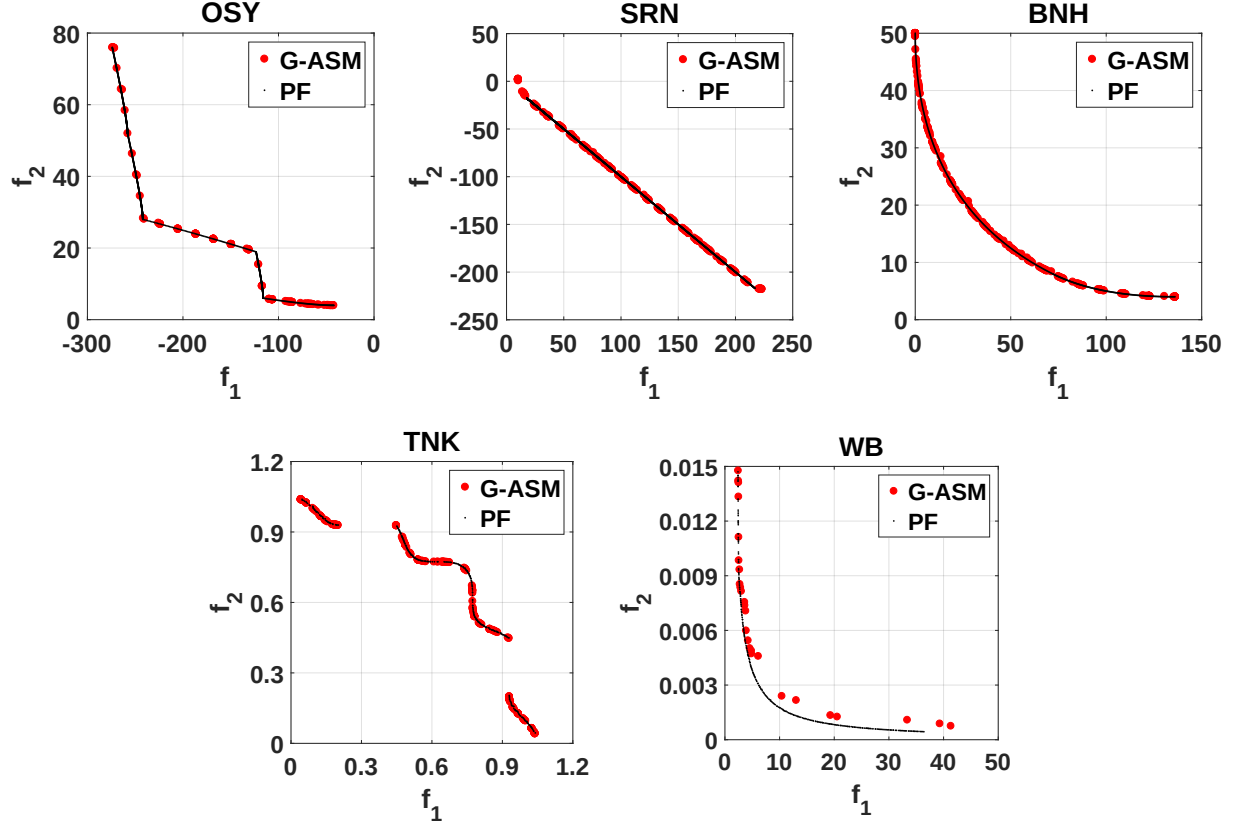


Figure 5.9: Obtained non-dominated solutions of the final archive for the median G-ASM run for two-objective constrained problems.

produce a better performance on these problems overall. These plots make the power of G-ASM method clear.

5.6.3 Three-Objective Constrained and Unconstrained Problems

Next, we apply all six methods to three-objective optimization problems (DTLZ2, DTLZ4, DTLZ5 and DTLZ7) and also to two, three-objective constrained problems (C2DTLZ2 and car side impact problem, CAR). In all six problems, G-ASM performs better or equivalent to individual frameworks. The final non-dominated points obtained for three-objective problems are shown in Figure 5.12.

The average epoch-wise proportion of usage frameworks in the G-ASM method are shown

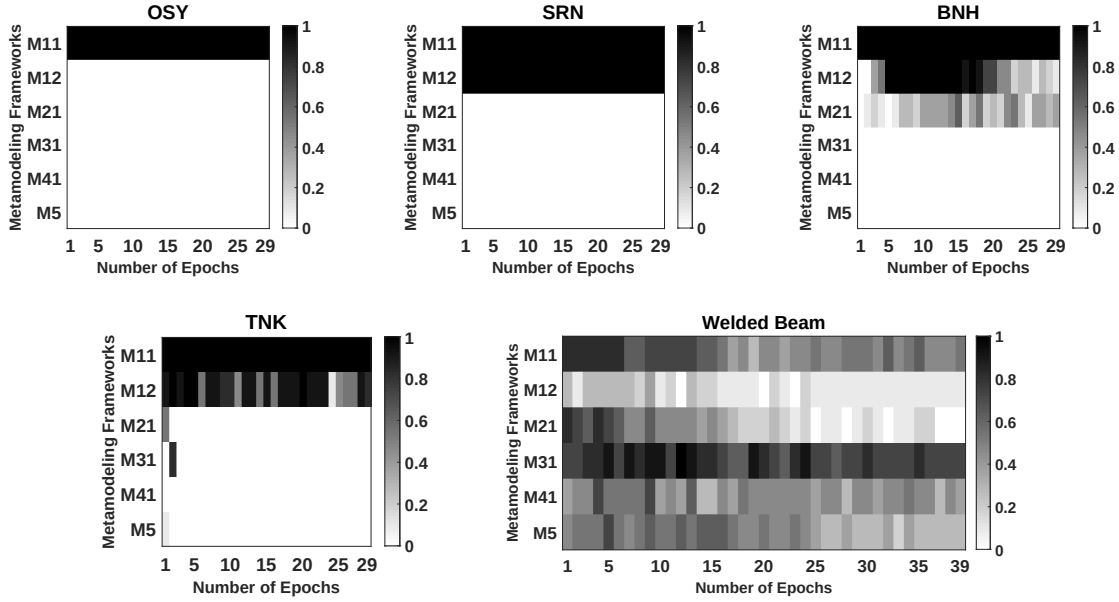


Figure 5.10: Proportion of framework usage in 11 runs for two-objective constrained problems.

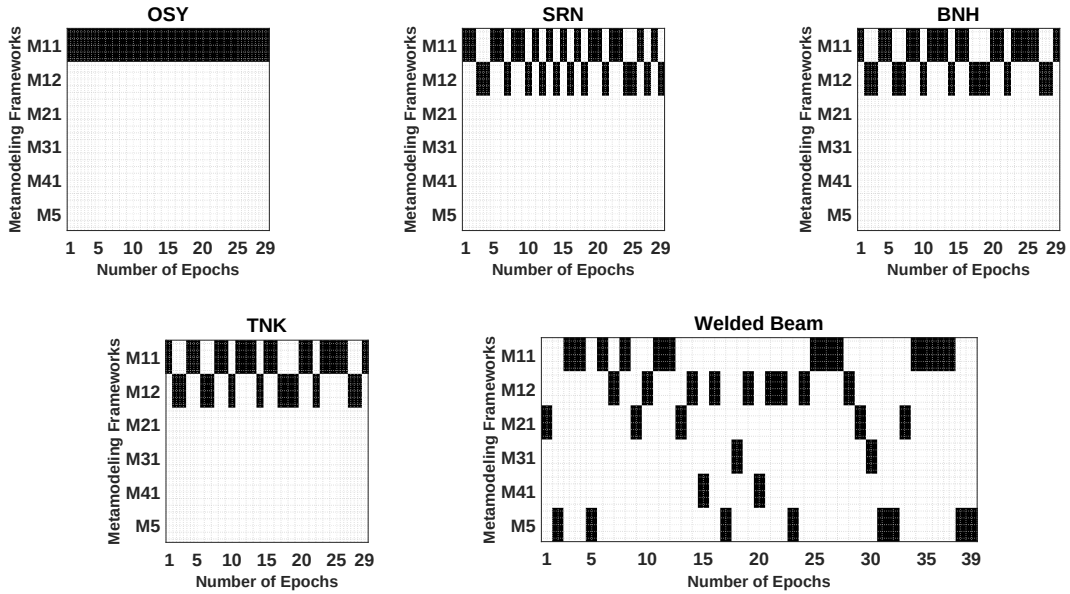


Figure 5.11: Switching of frameworks for the median G-ASM run for SRN, BNH and welded beam problems.

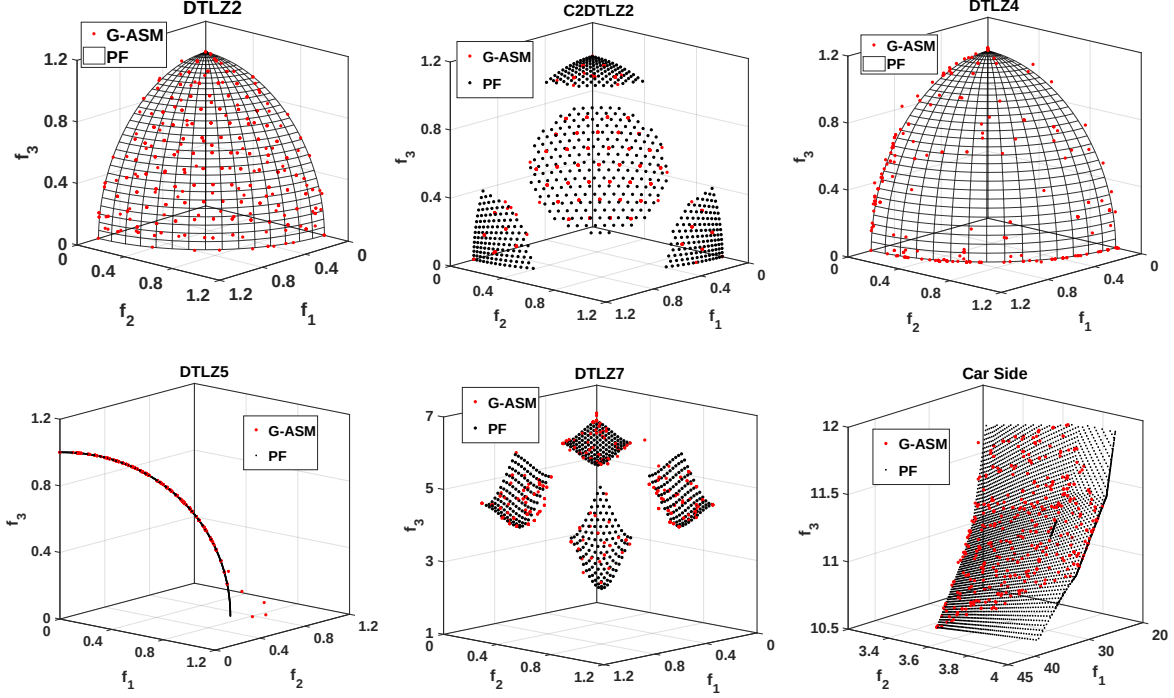


Figure 5.12: Obtained non-dominated solutions for Adaptive switching method of six frameworks for three objectives constrained and unconstrained problems.

in Figure 5.13 plots in all 11 runs. Patterns of their usage and also the non-usage of certain frameworks provide interesting information about the frameworks.

Finally, here we show the switching of frameworks for the median run in Figure 5.14.

5.6.4 Five-Objective Constrained and Unconstrained Problems

Next, we apply metamodeling frameworks to five-objective unconstrained DTLZ2 and constrained C2DTLZ2 problems. For both these problems, M1-2 framework (simultaneous optimization) performs the best and our G-ASM, despite having other frameworks as options, choose M1-2 in most runs to produce similar performing IGD (Figure 5.13 and Figure 5.14 with labels DTLZ2-5 and C2DTLZ2-5). As shown in Table 4.1, none of the other frameworks perform well on these two problems.

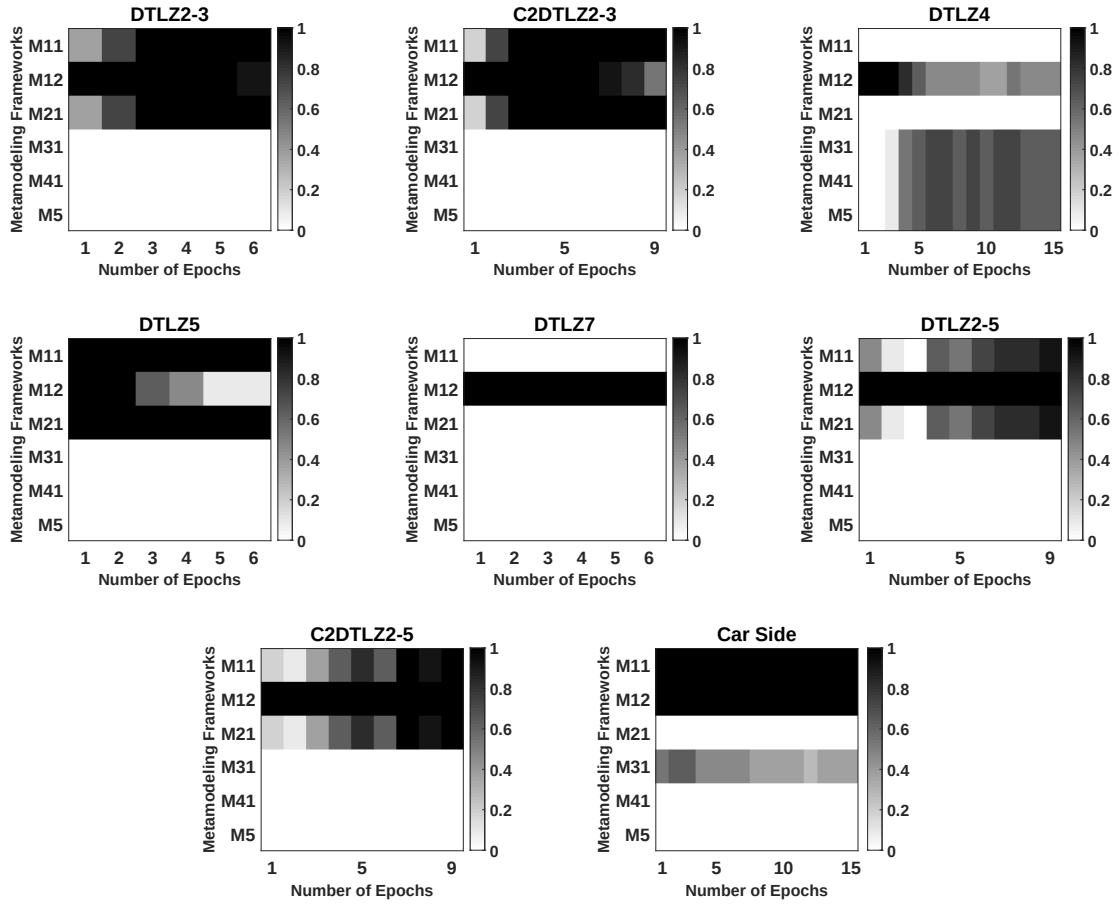


Figure 5.13: Proportion of framework usage in 11 runs for three and five-objective problems.

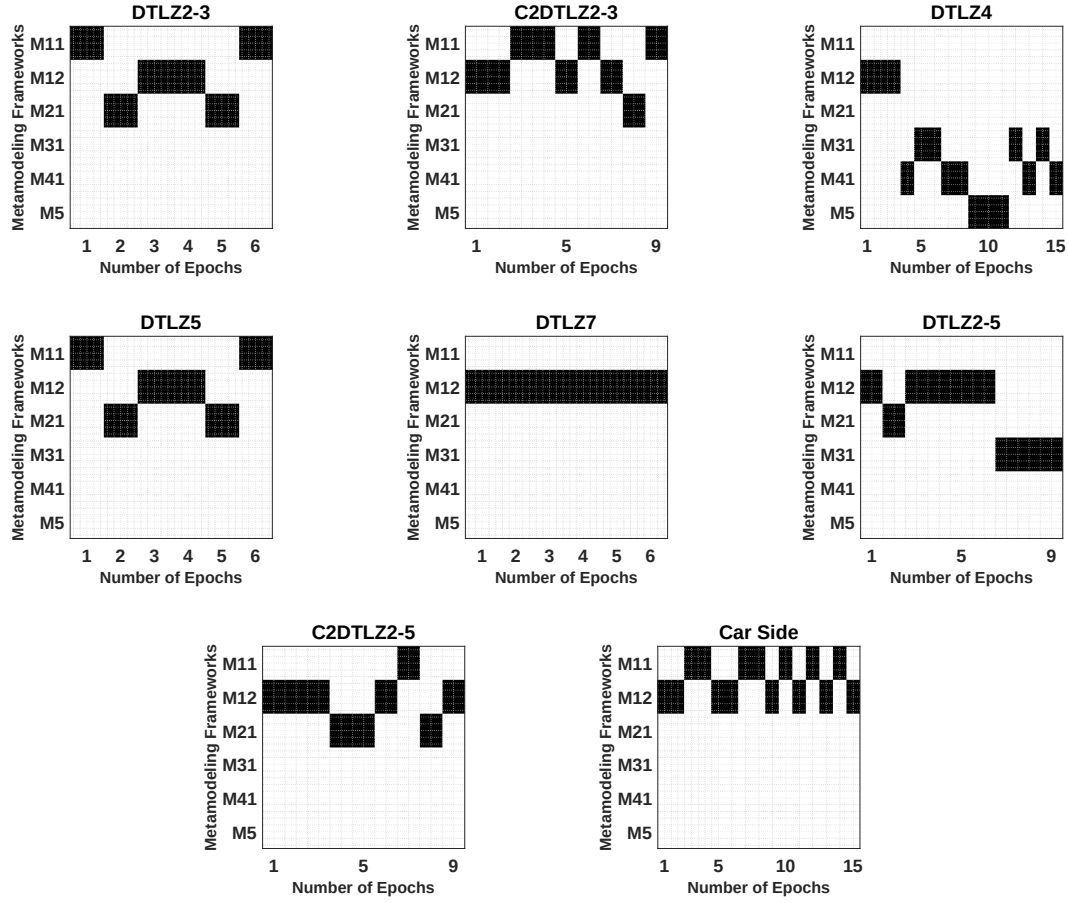


Figure 5.14: Adaptive switching method of six frameworks for median IGD run for three and five objectives problems.

5.6.5 Summary of Results

The rank of each the six frameworks applied alone and our G-ASM method according to Wilcoxon rank-sum test are shown in Table 5.4 for all 18 problems. It is observed that G-ASM method performs best overall, followed by M1-2. Among the generative frameworks alone, M3-1 (metamodeling an aggregate of objectives and each constraint separately) performs the best.

5.6.6 Comparative Study

Finally, we examine the performance of our proposed adaptive switching metamodeling (G-ASM) method by comparing it with three recently-proposed algorithms, namely, MOEA/D-EGO [90], K-RVEA [81], and CSEA [85]. These algorithms are implemented in PlatEMO [91] platform. MOEA/D-EGO and K-RVEA are surrogate-assisted evolutionary multi-objective optimization methods which use Kriging method to approximate individual objective or an aggregation functions. CSEA is a classification based surrogate-assisted evolutionary algorithm. None of these methods suggest a way to handle constraints. Hence, we restrict our comparative study to ZDT and DTLZ problems only. All methods are compared for the same $SE_{\max}$.

Table 5.5 presents the mean IGD values and marks the best performing method in bold and other methods with $p > 0.05$ using the Wilcoxon rank sum test in bold italics. It is clear that K-RVEA performs the best only on DTLZ4 and DTLZ7 problems. Except on DTLZ4, in all other problems G-ASM performs better or equivalent to K-RVEA. MOEA/D-EGO and CSEA do not perform well on any of the nine problems used in this study.

5.7 Summary

In this chapter, we have presented a trust region concept in order to produce a reliable and focused search by trading-off exploration and exploitation aspects of an optimization algorithm. We stated that, in addition, a switching between recently described metamodeling frameworks may be an efficient approach, rather than using one framework throughout, mainly due to the specialties that each metamodeling framework possess. On a number of multi-objective constrained and unconstrained test problems, the switching methods have produced better results by using a low budget of solution evaluations, compared to the individual metamodeling frameworks alone. In this study, we have proposed an adaptive switching strategy which chooses a statistically significant framework at the end of each epoch. In this study, we restrict the frameworks to be chosen from five generative frameworks and one popular simultaneous framework. Results on two to five-objective constrained and unconstrained test and engineering problems, it has been established that the adaptive switching method performs much better than a single framework. Moreover, since the knowledge of which framework would perform the best for a problem is not known a priori, G-ASM method is more appropriate. In order to demonstrate the working of G-ASM method, we have presented an epoch-wise selection of frameworks. In most problems, G-ASM switches between a few frameworks, but in some problems the same framework is used from start to finish. G-ASM has also been found to produce better or equivalent results compared to three newly proposed metamodeling methods on unconstrained problems.

Table 5.3: Computed IGD values for test problems. Best performing framework is marked in bold and other statistically similar frameworks are marked in bold and italic.

Problem	M1-1	M1-2	M2-1	M3-1	M4-1	M5	G-ASM
ZDT1	0.00090	0.00555	-	0.00447	-	-	0.00099
	-	p=8.1e-5	-	p=8.1e-5	-	-	p=0.5114
ZDT2	0.00065	0.00062	-	0.00568	-	-	0.00057
	p=0.2372	p=0.7928	-	p=8.1e-5	-	-	-
ZDT3	0.06778	0.00212	-	0.17123	-	-	0.00456
	p=8.1e-5	-	-	p=8.1e-5	-	-	p=0.421
ZDT4	0.28900	5.43450	-	0.29300	-	-	0.45803
	-	p=8.1e-5	-	p=0.4307	-	-	p=0.0151
ZDT6	0.37058	0.48360	-	0.24192	-	-	0.19038
	p=0.001	p=8.1e-5	-	p=0.0762	-	-	-
OSY	0.15323	0.18806	24.57940	6.26550	18.49200	45.18110	0.16376
	-	p=0.2643	p=8.1e-5	p=8.1e-5	p=8.1e-5	p=8.1e-5	p= 0.4307
TNK	0.01726	0.00082	0.04383	0.01180	0.03332	0.03077	0.00105
	p=8.1e-5	-	p=8.1e-5	p=8.1e-5	p=8.1e-5	p=8.1e-5	p=0.0569
SRN	0.13191	1.00930	4.17160	1.06120	1.20480	1.28450	0.13434
	-	p=8.1e-5	p=8.1e-5	p=8.1e-5	p=8.1e-5	p=8.1e-5	p=0.2370
BNH	0.69434	0.04630	0.74425	0.23728	0.23923	0.23699	0.07765
	p=8.1e-5	-	p=8.1e-5	p=8.1e-5	p=8.1e-5	p=8.1e-5	p=0.652
WB	0.13794	0.23159	0.55529	0.16909	0.88586	0.96166	0.12819
	p=0.6457	p= 0.0660	p=8.1e-5	p=0.8438	p=8.1e-5	p=8.1e-5	-
DTLZ2	0.07870	0.03340	-	0.05377	-	-	0.03921
	p=8.1e-5	-	-	p=8.1e-5	-	-	p=0.1002
C2DTLZ2	0.05130	0.03355	-	0.03493	-	0.12403	0.03401
	p=8.1e-5	-	-	p=0.251	-	p=8.1e-5	p=0.321
CAR	0.43510	0.50119	0.43145	0.39809	0.42223	0.50061	0.39060
	p=0.0041	p=0.0058	p=0.0041	p=0.234	p=0.1149	p=0.0001	-
DTLZ5	0.01960	0.00948	-	0.01352	-	-	0.01123
	p=8.1e-5	-	-	p=8.1e-5	-	-	p=0.0728
DTLZ4	0.05840	0.09024	-	0.20668	-	-	0.08012
	-	p=8.1e-5	-	p=8.1e-5	-	-	p=0.205
DTLZ7	0.89316	0.07664	-	0.87172	-	-	0.07261
	p=8.1e-5	p=0.5994	-	p=8.1e-5	-	-	-
DTLZ2-5	0.21450	0.03981	-	0.14401	-	-	0.04643
	p=8.1e-5	-	-	p=8.1e-5	-	-	p=0.5109
C2DTLZ2-5	0.17341	0.03676	-	0.15388	-	0.29291	0.04891
	p= 8.1e-5	-	-	p=8.1e-5	-	p=8.1e-5	p=0.528

Table 5.4: Ranking of frameworks summarized. A ‘1’ indicates the best performing framework.

Problem	M1-1	M1-2	M2-1	M3-1	M4-1	M5	G-ASM
ZDT1	1	7	1	4	4	4	1
ZDT2	1	1	1	5	5	5	1
ZDT3	3	1	3	5	5	5	1
ZDT4	1	7	1	1	1	1	6
ZDT6	5	7	5	1	1	1	1
OSY	1	1	6	4	5	7	1
TNK	4	1	7	3	6	5	1
SRN	1	2	6	3	4	5	1
BNH	6	1	7	4	5	3	1
WB	1	1	5	1	6	7	1
DTLZ2	6	1	6	3	3	3	1
C2DTLZ2	5	1	7	1	1	7	1
CAR	5	7	4	1	1	6	1
DTLZ5	6	1	6	3	3	3	1
DTLZ4	1	4	1	5	5	5	1
DTLZ7	6	1	6	3	3	3	1
DTLZ2-5	6	1	6	3	3	3	1
C2DTLZ2-5	5	1	5	3	3	6	1
Average	3.56	2.56	4.50	2.94	3.55	4.50	1.27

Table 5.5: Mean IGD comparison among G-ASM, MOEA/D-EGO, K-RVEA, and CSEA algorithms on unconstrained problems.

Problem	MOEA/D-EGO	K-RVEA	CSEA	G-ASM
ZDT1	0.05611	0.07964	0.95330	0.00099
	p=8.1e-5	p=8.1e-5	p=8.1e-5	-
ZDT2	0.04922	0.03395	1.01060	0.00057
	p=8.1e-5	p=8.1e-5	p=8.1e-5	-
ZDT3	0.3038	0.02481	0.94841	0.00456
	p=8.1e-5	p=8.1e-5	p=8.1e-5	-
ZDT4	73.25920	4.33221	12.71601	0.45801
	p=8.1e-5	p=8.1e-5	p=8.1e-5	-
ZDT6	0.51472	0.65462	5.42620	0.19036
	p=8.1e-5	p=8.1e-5	p=8.1e-5	-
DTLZ2	0.33170	0.05481	0.11420	0.03921
	p=8.1e-5	p=8.1e-5	p=8.1e-5	-
DTLZ4	0.64533	0.04490	0.08110	0.08012
	p=8.1e-5	-	p=0.0022	p=0.001
DTLZ5	0.26203	0.01640	0.03081	0.01123
	p=8.1e-5	p=0.0262	p=0.0013	-
DTLZ7	5.33220	0.05310	0.70520	0.07261
	p=8.1e-5	-	p=8.1e-5	p=0.07

Chapter 6

Future Studies

- As an immediate future study beyond this thesis, we plan to investigate other meta-modeling approaches (such as RBF, support vector regression, and response surfaces) using Kriging for all of them simultaneously with adaptive switching among the proposed metamodeling frameworks.
- Many acquisition functions can be used with metamodeling frameworks, such as expected improvement, probability of improvement, hypervolume improvement, and others infill sampling criteria approaches to eventually identify optimized algorithms. To determine the scope for each of the proposed metamodeling frameworks we will solve various types of problems ranging from two to many objectives.
- It will be interesting to investigate the effect of diverse sweep approach that was mentioned in subsection 3.4.3 with metamodeling frameworks M3, M4, and M5 for improving the performance of those frameworks, especially, with problems that have a high-density distribution in the decision space.
- With the development of efficient metamodeling methods, we are now ready to use them to solve industrial and agro-environmental problems which involve multiple conflicting objectives.
- In some problems, objectives and constraints with heterogeneous computational com-

plexity may exist. We plan to extend our methodologies and applications to these problems as well.

Chapter 7

Conclusions

In this study, we have presented a taxonomy for the use of different metamodeling frameworks for multi-objective optimization. The taxonomy also extends to cover single-objective optimization problems with and without constraints; however, handling constraints has been put under the same platform as the handling of objectives - a matter which has been ignored in many past such studies. The study has focused outside the popular strategy of extending metamodeling frameworks for single-objective optimization to multi-objective optimization. The straightforward extension suggests that for every objective function and constraint function, an independent metamodel is needed. Although this is one viable strategy, we have proposed five more potential strategies and classified many existing studies into the proposed six categories. Thereafter, as an initial detailed systematic study, we have implemented a representative algorithm for each of the six proposed frameworks and applied them to solve a number of constrained and unconstrained multi-objective optimization problems. Results are compared against each other and important conclusions about the behavior of each of the frameworks have been revealed. It is worth noting that although parameters of all six proposed frameworks have not been fine-tuned for their best performance, our obtained results are able to bring out the generic properties of different possible multi-objective metamodeling frameworks applied to distinct problem classes.

In this study, we have proposed an adaptive switching strategy which chooses a statistically significant framework at the end of each epoch. Also, we restrict the frameworks to be

chosen from five generative frameworks and one popular simultaneous framework. Results on two to five-objective constrained and unconstrained test and engineering problems, it has been established that the adaptive switching method performs much better than a single framework. Moreover, since the knowledge of which framework would perform the best for a problem is not known a priori, G-ASM method is more appropriate. In order to demonstrate the working of G-ASM method, we have presented an epoch-wise selection of frameworks. In most problems, G-ASM switches between a few frameworks, but in some problems the same framework is used from start to finish.

Finally, G-ASM has been compared against three recently proposed surrogate-assisted, unconstrained, multi-objective optimization methods. On eight or nine problems, G-ASM has been found to perform better than the three existing methods. This study has amply demonstrated the superiority of adaptive switching based metamodeling methods for multi-objective optimization problems. The studies can be extended to include different metamodeling methods at each epoch, such as RBF or other response surface methods. More initial samples can be allocated to complex metamodeling functions, such as for frameworks M5 and M6, so that more accurate metamodels can be obtained.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] A. Forrester, A. Sobester, and A. Keane, *Engineering design via surrogate modelling: A practical guide*. John Wiley & Sons, 2008.
- [2] K. Shimoyama, K. Sato, S. Jeong, and S. Obayashi, “Updating kriging surrogate models based on the hypervolume indicator in multi-objective optimization,” *Journal of Mechanical Design*, vol. 135, no. 9, p. 094503, 2013.
- [3] A. Cassioli and F. Schoen, “Global optimization of expensive black box problems with a known lower bound,” *Journal of Global Optimization*, pp. 177–190, 2013.
- [4] D. R. Jones, “A taxonomy of global optimization methods based on response surfaces,” *Journal of global optimization*, vol. 21, no. 4, pp. 345–383, 2001.
- [5] 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.
- [6] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, “Expensive multiobjective optimization by MOEA/D with gaussian process model,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 456–474, 2010.
- [7] G. Matheron, “Principles of geostatistics,” *Economic Geology*, vol. 58, no. 8, pp. 1246–1266, Dec. 1963.
- [8] D. Krige, “A statistical approach to some basic mine valuation problems on the Witwatersrand,” *Journal of the Chemical, Metallurgical and Mining Engineering Society of South Africa*, pp. 119–139, 1951.
- [9] J. Sacks, W. Welch, T. Mitchell, and H. Wynn, “Design and Analysis of Computer Experiments,” *Statistical Science*, pp. 409–423, 1989.
- [10] D. R. Jones, M. Schonlau, and W. J. Welch, “Efficient global optimization of expensive black-box functions,” *J. of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.
- [11] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001.

- [12] C. Coello, G. Lamont, and D. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-objective Problems*. Springer, New York, 2nd edition, 2007.
- [13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, no. 2, pp. 182–197, April 2002.
- [14] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm,” in *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, Athens, Greece, 2002, pp. 95–100.
- [15] 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. Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [16] 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.
- [17] J. Yaochu, “Surrogate-assisted evolutionary computation: Recent advances and future challenges,” *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.
- [18] A. P. Wierzbicki, “The use of reference objectives in multiobjective optimization,” in *Multiple criteria decision making theory and application*. Springer, 1980, pp. 468–486.
- [19] K. Miettinen, *Nonlinear Multiobjective Optimization*. Kluwer, Boston, 1999.
- [20] R. Hussein and K. Deb, “A generative kriging surrogate model for constrained and unconstrained multi-objective optimization,” in *GECCO’2016*. ACM Press, 2016.
- [21] K. Deb and R. Datta, “A fast and accurate solution of constrained optimization problems using a hybrid bi-objective and penalty function approach,” in *IEEE Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–8.
- [22] K. Deb, R. Hussein, P. Roy, and G. Toscano, “Classifying metamodeling methods for evolutionary multi-objective optimization: First results,” in *Evolutionary Multi-Criterion Optimization EMO 2017*. Springer, 2017.

- [23] L. V. Santana-Quintero, A. A. Montano, and C. A. C. Coello, “A review of techniques for handling expensive functions in evolutionary multi-objective optimization,” in *Computational Intelligence in Expensive Optimization Problems*. Springer, 2010.
- [24] Y. Jin, “A comprehensive survey of fitness approximation in evolutionary computation,” *Soft computing*, 2005.
- [25] L. Shi and K. Rasheed, “A survey of fitness approximation methods applied in evolutionary algorithms,” in *Computational intelligence in expensive optimization problems*. Springer, 2010.
- [26] A. Díaz-Manríquez, G. Toscano, J. H. Barron-Zambrano, and E. Tello-Leal, “A review of surrogate assisted multiobjective evolutionary algorithms,” *Computational Intelligence and Neuroscience*, 2016.
- [27] D. Horn, T. Wagner, D. Biermann, C. Weihs, and B. Bischl, “Model-based multi-objective optimization: taxonomy, multi-point proposal, toolbox and benchmark,” in *EMO*. Springer, 2015, pp. 64–78.
- [28] J. Knowles, “Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems,” *IEEE Transactions on Evolutionary Computation*, pp. 50–66, 2006.
- [29] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, “Multiobjective optimization on a limited budget of evaluations using model-assisted S-metric selection,” in *Parallel Problem Solving from Nature-PPSN X*. Springer, 2008, pp. 784–794.
- [30] B. Bischl, S. Wessing, N. Bauer, K. Friedrichs, and C. Weihs, “Moi-mbo: Multiobjective infill for parallel model-based optimization,” in *Learning and Intelligent Optimization*. Springer, 2014, pp. 173–186.
- [31] P. Rajak, U. Tewary, S. Das, B. Bhattacharya, and N. Chakraborti, “Phases in Zn-coated Fe analyzed through an evolutionary meta-model and multi-objective Genetic Algorithms,” *Computational Materials Science*, pp. 2502–2516, 2011.
- [32] S. Zapotecas Martínez and C. Coello Coello, “MOEA/D Assisted by RBF Networks for Expensive Multi-objective Optimization Problems,” in *Genetic and Evolutionary Computation Conference*. ACM, 2013.
- [33] A. Arias-Montano, C. Coello Coello, and E. Mezura-Montes, “Multi-objective Airfoil Shape Optimization using a Multiple-surrogate Approach,” in *IEEE Congress on Evolutionary Computation*, 2012.

- [34] A. Turco, “MetaHybrid: Combining Metamodels and Gradient-Based Techniques in a Hybrid Multi-Objective Genetic Algorithm,” in *Learning and Intelligent Optimization, 5th International Conference*. Springer., 2011, pp. 293–307.
- [35] S. Zapotecas Martínez and C. A. Coello Coello, “A Multi-Objective Meta-Model Assisted Memetic Algorithm with Non Gradient-Based Local Search,” in *GECCO’2010*. Portland, Oregon, USA: ACM Press, 2010, pp. 537–538.
- [36] S. Zapotecas Martínez and C. Coello Coello, “A Multi-objective Meta-model Assisted Memetic Algorithm with non Gradient-based Local Search,” in *Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2010)*. ACM, 2010.
- [37] N. Namura, K. Shimoyama, and S. Obayashi, “Kriging Surrogate Model Enhanced by Coordinate Transformation of Design Space Based on Eigenvalue Decomposition,” in *8th International Conference, EMO 2015*. Guimarães, Portugal: Springer., 2015, pp. 321–335.
- [38] P. C. Roy and K. Deb, “High dimensional model representation for solving expensive multi-objective optimization problems,” in *CEC’2016*, 2016.
- [39] A. Husain and K.-Y. Kim, “Enhanced multi-objective optimization of a microchannel heat sink through evolutionary algorithm coupled with multiple surrogate models,” *Applied Thermal Engineering*, vol. 30, 2010.
- [40] F. Bittner and I. Hahn, “Kriging-Assisted Multi-Objective Particle Swarm Optimization of permanent magnet synchronous machine for hybrid and electric cars,” in *2013 IEEE International*, 2013.
- [41] J. Beck, D. Friedrich, S. Brandani, and E. S. Fraga, “Multi-objective optimisation using surrogate models for the design of VPSA systems,” *Computers & Chemical Engineering*, 2015.
- [42] R. Carrese, A. Sobester, H. Winarto, and X. Li, “Swarm Heuristic for Identifying Preferred Solutions in Surrogate-Based Multi-Objective Engineering Design,” *AIAA Journal*, pp. 1437–1449, 2011.
- [43] A. Lombardi, D. Ferrari, and L. Santos, “Aircraft Air Inlet Design Optimization via Surrogate-Assisted Evolutionary Computation,” in *8th International Conference, EMO 2015*. Guimarães, Portugal: Springer., 2015, pp. 313–327.

- [44] K. Shimoyama, S. Jeong, and S. Obayashi, “Kriging-Surrogate-Based Optimization Considering Expected Hypervolume Improvement in Non-Constrained Many-Objective Test Problems,” in *CEC’2013*. Cancún, México: IEEE Press, 2013, pp. 658–665.
- [45] J. Sreekanth and B. Datta, “Multi-objective management of saltwater intrusion in coastal aquifers using genetic programming and modular neural network based surrogate models,” *Journal of Hydrology*, pp. 245–256, 2010.
- [46] —, “Coupled simulation-optimization model for coastal aquifer management using genetic programming-based ensemble surrogate models and multiple-realization optimization,” *Water Resources Research*, 2011.
- [47] A. Husain, K.-D. Lee, and K.-Y. Kim, “Enhanced multi-objective optimization of a dimpled channel through evolutionary algorithms and multiple surrogate methods,” *International Journal for Numerical Methods in Fluids*, pp. 742–759, 2011.
- [48] N. Namura, S. Obayashi, and S. Jeong, “Surrogate-Based Multi-Objective Optimization and Data Mining of Vortex Generators on a Transonic Infinite-Wing,” in *CEC’2013*. Cancún, México: IEEE Press, 2013, pp. 2910–2917.
- [49] S. Kanyakam and S. Bureerat, “Comparative Performance of Surrogate-Assisted MOEAs for Geometrical Design of Pin-Fin Heat Sinks,” *Journal of Applied Mathematics*, 2012.
- [50] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, “Generalizing Surrogate-Assisted Evolutionary Computation,” *IEEE Transactions on Evolutionary Computation*, pp. 329–355, 2010.
- [51] D. Verbeeck, F. Maes, K. D. Grave, and H. Blockeel, “Multi-Objective Optimization with Surrogate Trees,” in *GECCO’2013*. New York, USA: ACM Press, 2013, pp. 679–686.
- [52] N. Stander, “An Adaptive Surrogate-Assisted Strategy for Multi-Objective Optimization,” in *World Congress on Structural and Multidisciplinary Optimization*, Orlando, Florida, 2013.
- [53] J. Y. J. Chow and A. C. Regan, “A surrogate-based multiobjective metaheuristic and network degradation simulation model for robust toll pricing,” *Optimization and Engineering*, vol. 15, pp. 137–165, 2014.

- [54] S. A. Kyriacou, V. G. Asouti, and K. C. Giannakoglou, “Efficient PCA-driven EAs and metamodel-assisted EAs, with applications in turbomachinery,” *Engineering Optimization*, vol. 46, pp. 895–911, 2014.
- [55] E. Rigoni and A. Turco, “Metamodels for Fast Multi-Objective optimization: Trading Off Global Exploration and Local Exploitation,” in *SEAL 2010*. Kanpur, India: Springer. Lecture Notes in Computer Science Vol. 6457, 2010, pp. 523–532.
- [56] A. Rosales Pérez, “Surrogate-Assisted Evolutionary Multi-Objective Full Model Selection,” Ph.D. dissertation, Instituto Nacional de Astrofísica, Óptica y Electrónica, Tonantzintla, Puebla, México, January 2016.
- [57] A. Rosales-Pérez, C. A. Coello Coello, J. A. Gonzalez, C. A. Reyes-García, and H. Jair Escalante, “A Hybrid Surrogate-Based Approach for Evolutionary Multi-Objective Optimization,” in *CEC’2013*. Cancún, México: IEEE Press, 2013, pp. 2548–2555.
- [58] I. Loshchilov, M. Schoenauer, and M. Sebag, “A Mono Surrogate for Multiobjective Optimization,” in *GECCO’2010*. Portland, Oregon, USA: ACM Press, 2010, pp. 471–478.
- [59] Y. Zhang, S. Hu, J. Wu, Y. Zhang, and L. Chen, “Multi-objective optimization of double suction centrifugal pump using Kriging metamodels,” *Advances in Engineering Software*, pp. 16–26, 2014.
- [60] M. N. Le, Y. S. Ong, S. Menzel, C.-W. Seah, and B. Sendhoff, “Multi co-objective evolutionary optimization: cross surrogate augmentation for computationally expensive problems,” in *CEC’2012*. Brisbane, Australia: IEEE Press, 2012, pp. 2871–2878.
- [61] R. F. Coelho, J. Lebon, and P. Bouillard, “Hierarchical stochastic metamodels based on moving least squares and polynomial chaos expansion,” *Structural and Multidisciplinary Optimization*, pp. 707–729, 2011.
- [62] R. F. Coelho and P. Bouillard, “Multi-Objective Reliability-Based Optimization with Stochastic Metamodels,” *Evolutionary Computation*, vol. 19, pp. 525–560, 2011.
- [63] S. Zapotecas Martínez and C. A. Coello Coello, “A Memetic Algorithm with Non Gradient-Based Local Search Assisted by a Meta-Model,” in *11th International Conference, Proceedings, Part I*. Kraków, Poland: Springer, 2010, pp. 576–585.
- [64] H. K. Singh, T. Ray, and W. Smith, “Surrogate assisted Simulated Annealing (SASA) for constrained multi-objective optimization,” in *CEC’2010*. Barcelona, Spain: IEEE Press, 2010, pp. 4202–4208.

- [65] J. Martínez-Frutos and D. H. Pérez, “Kriging-based infill sampling criterion for constraint handling in multi-objective optimization,” *Journal of Global Optimization*, pp. 97–115, 2016.
- [66] P. Singh, I. Couckuyt, F. Ferranti, and T. Dhaene, “A Constrained Multi-Objective Surrogate-Based Optimization Algorithm,” in *CEC’2014*. Beijing, China: IEEE Press, 2014, pp. 3081–3087.
- [67] I. Tsoukalas and C. Makropoulos, “Multiobjective optimisation on a budget: Exploring surrogate modelling for robust multi-reservoir rules generation under hydrological uncertainty,” *Environmental Modelling & Software*, 2015.
- [68] K. Deb, “An efficient constraint handling method for genetic algorithms,” *Computer methods in applied mechanics and engineering*, vol. 186, no. 2, pp. 311–338, 2000.
- [69] I. Das and J. E. Dennis, “Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems,” *SIAM J. on Optimization*, vol. 8, no. 3, 1998.
- [70] P. Roy, R. Hussein, and K. Deb, “Metamodeling for multimodal selection functions in evolutionary multi-objective optimization,” in *GECCO’2017*. ACM Press, 2017.
- [71] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard, “DACE—A Matlab Kriging toolbox, version 2.0,” Tech. Rep., 2002.
- [72] C. C. Tutum, K. Deb, and I. Baran, “Constrained efficient global optimization for pultrusion process,” *Materials and manufacturing processes*, vol. 30, no. 4, pp. 538–551, 2015.
- [73] S. Mostaghim and J. Teich, “A new approach on many objective diversity measurement,” in *Practical Approaches to Multi-Objective Optimization*, no. 04461, Dagstuhl, Germany, 2005.
- [74] K. Deb, “An investigation of niche and species formation in genetic function optimization,” in *Proc. 3rd Int. Conf. on Genetic Algorithms*, 1989, pp. 42–50.
- [75] K. Deb, M. Abouhawwash, and J. Dutta, “An optimality theory based proximity measure for evolutionary multi-objective and many-objective optimization,” in *EMO*. Springer, 2015, pp. 18–33.

- [76] K. Deb and M. Abouhawwash, “An optimality theory-based proximity measure for set-based multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 515–528, 2016.
- [77] K. Deb, “Scope of stationary multi-objective evolutionary optimization: a case study on a hydro-thermal power dispatch problem,” *Journal of Global Optimization*, vol. 41, no. 4, pp. 479–515, 2008.
- [78] R. H. Byrd, J. Nocedal, and R. A. Waltz, *Knitro: An Integrated Package for Nonlinear Optimization*. Boston, MA: Springer US, 2006, pp. 35–59.
- [79] K. Deb, R. Hussein, P. C. Roy, and G. Toscano, “A taxonomy for metamodeling frameworks for evolutionary multi-objective optimization,” *IEEE Transactions on Evolutionary Computation*, 2018.
- [80] D. Zhao and D. Xue, “A multi-surrogate approximation method for metamodeling,” *Engineering with Computers*, 2011.
- [81] 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*, vol. 22, no. 1, pp. 129–142, 2018.
- [82] R. Datta and R. G. Regis, “A surrogate-assisted evolution strategy for constrained multi-objective optimization,” *Expert Systems with Applications*, vol. 57, pp. 270–284, 2016.
- [83] K. Bhattacharjee, H. Singh, and T. Ray, “Multi-objective optimization using an evolutionary algorithm embedded with multiple spatially distributed surrogates,” *The American Society of Mechanical Engineers*, vol. 138, no. 9, pp. 135–155, 2016.
- [84] A. A. Rahat, R. M. Everson, and J. E. Fieldsend, “Alternative infill strategies for expensive multi-objective optimisation,” in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2017, pp. 873–880.
- [85] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin, “A classification based surrogate-assisted evolutionary algorithm for expensive many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2018.
- [86] M. N. Le, Y. S. Ong, S. Menzel, C.-W. Seah, and B. Sendhoff, “Multi co-objective evolutionary optimization: Cross surrogate augmentation for computationally expensive

- problems,” in *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012, pp. 1–8.
- [87] M. Karimzadehgan and C. Zhai, “Exploration-exploitation tradeoff in interactive relevance feedback,” in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, ser. CIKM '10. ACM, 2010.
 - [88] P. Giulia and H. N. Szu, “G-star: A new kriging-based trust region method for global optimization,” *Winter Simulation Conference: Simulating Complex Service Systems, WSC 2016*.
 - [89] A. Ahrari, J. Blank, K. Deb, and X. Li, “Niching surrogate-assisted optimization for simulation-based design optimization of cylinder head water jacket,” *unpublished document*, 2018.
 - [90] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, “Expensive Multiobjective Optimization by MOEA/D With Gaussian Process Model,” *IEEE Transactions on Evolutionary Computation*, pp. 456–474, 2010.
 - [91] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, “platemo: A matlab,” *IEEE Computational Intelligence Magazine*.