DLR-IB-AS-BS-2021-251

Global Multi-Objective Optimisation utilising Surrogate Models

Interner Bericht

Autorin
Sihyeong Lim

DLR Deutsches Zentrum
für Luft- und Raumfahrt

Bericht des Instituts für Aerodynamik und Strömungstechnik
Report of the Institute of Aerodynamics and Flow Technology

DLR-IB-AS-BS-2021-251

Global Multi-Objective Optimisation utilizing Surrogate Models

Sihyeong Lim

Herausgeber:
Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Aerodynamik und Strömungstechnik
Lilienthalplatz 7, 38108 Braunschweig

ISSN 1614-7790

Stufe der Zugänglichkeit: 1
Braunschweig, im Dezember 2021

Institutsdirektor:
Prof. Dr.-Ing. habil. C.-C. Rossow

Verfasserin:
Sihyeong Lim

Abteilung: Center of Computer Applications in Aerospace Science and Engineering

Abteilungsleiter:
Prof. Dr. S. Görtz

Der Bericht enthält:

99	Seiten
44	Bilder
7	Tabellen
47	Literaturstellen

Abstract

While global multi-objective optimization problems continue to emerge in aerospace engineering, conventional optimization methods, in particular, evolutionary algorithms such as the Non-dominated Sorting Genetic Algorithm, have shown their capability to solve such problems. However, one distinctive disadvantage of these conventional methods is that they generally require a large number of function evaluations, which makes them incompatible with computationally intensive numerical simulations that are often employed in aerospace design problems.

This thesis substantiates the idea that this limitation can be overcome by using surrogate based optimization, in particular multi-objective Bayesian global optimization that utilizes Kriging as a surrogate model and Expected Hypervolume Improvement as an infill criteria. With this approach, it is possible to obtain the Pareto front with a relatively small computational budget. This is demonstrated through test cases that are conducted by solving analytical optimization problems. The results show that Bayesian optimization is able to reduce the function evaluations by 51 times for the bi-objective problem, and by 91 times for the three-objectives problem compared to genetic algorithms.

Furthermore, its applicability is tested in two aerospace design problems, where function evaluations were performed through Computational Fluid Dynamics (CFD) and Computational Aeroacoustic (CAA) simulations. The proposed optimization method returns Pareto fronts which contain various design trade-offs that result in improved performance in terms of the desired objectives, with a reasonable number of function evaluations. Firstly, in the aerodynamic shape optimization, it is able to obtain the Pareto front, which contains airfoil designs with a combination of reduced drag and reduced pitching moment. Secondly, the aerodynamic-aeroacoustic shape optimization is performed where the Pareto front is obtained for airfoil designs with three objectives: reduced drag, reduced pitching moment and reduced aeroacoustic noise.

This thesis demonstrates the efficiency of the Bayesian global optimization framework by showing how the Pareto front can be obtained at a relatively smaller number of function evaluations compared to some of the conventional multi-objective optimization methods. Moreover, the results obtained from the applied problems verify its capability for practical applications in aerospace design. Hence, the outcomes of this thesis highlight the potential of multi-objective Bayesian global optimization for multidisciplinary design optimization problems in the field of aerospace engineering.

Contents

Acronyms

Abbreviation	**Description**
CAA	Computational Aeroacoustics
CFD	Computational Fluid Dynamics
DOE	Design of Experiments
EHVI	Expected Hypervolume Improvement
EI	Expected Improvement
GA	Genetic Algorithm
GPs	Gaussian Processes
HV	Hypervolume Indicator
HVI	Hypervolume Improvement
LHS	Latin Hypercube Sampling
MDO	Multidisciplinary Design Optimization
NRMSE	Normalized Root Mean Squared Error
NSGA-II	Non-dominated Sorting Genetic Algorithm II
PoI	Probability of Improvement
RANS	Reynolds-Averaged Navier Stokes
RMSE	Root Mean Squared Error
SBO	Surrogate Based Optimization
SMARTy	Surrogate Modeling for Aero-Data Toolbox in Python
WFG	the Walking Fish Group

List of Figures

List of Tables

1 Introduction

In recent developments in engineering design, the importance of multi-objective optimization cannot be overstated. In practice, this is true not only in engineering, but also in a wide range of other fields that involve trade-offs between multiple conflicting objectives, such as economics, finance, and even in general process management. Even within engineering, it is applied as diversely as design optimization and optimal control. Especially, with the recent advancements in computational modeling and numerical tools, the demand for multi-objective optimization is also rapidly increasing. The accuracy of numerical simulation tools has evolved to such a degree that they can mimic physical experiments with nearly insignificant errors, and their speed has become fast enough to produce these reliable results in a matter of hours or even minutes. Therefore, the increase in attempts to find better designs through multi-objective optimization using these advanced simulation tools is natural. Furthermore, multi-objective optimization enables the making of any type of engineering decision through a more quantitative comparison of various options that had previously been made using empirical methods or past experience. In fact, there are various conventional methods through which this can be achieved.

Some of the well known classical multi-objective methods include the weighted sum method as well as the ϵ-Constraint method. In the weighted sum method, a set of objectives are scalarized into a single objective by summing over the objective values weighted by the coefficients. Unfortunately, despite its simplicity, it requires that the Pareto front is convex [43]. Furthermore, it is difficult to set the weighting coefficients without prior knowledge of which combination will lead to the desired region in the objective space. In the ϵ-Constraint method which was first proposed by Haimes et al. in [47], one of the objectives is selected while all other objectives are set to be constraints. While these methods are classified as parametric scalarizing approaches as they convert multi-objectives into a parametric single-objective objective function, one of the biggest disadvantages is that the single optimization needs to be performed multiple times to obtain the full Pareto front, which is the general solution in multi-objective optimization that typically involves multiple promising solutions.

One of the major difficulties of solving multi-objective optimization problems, as shown from the examples of the classical methods, comes from the fact that there is no single best solution, as there is in single-objective optimization problems, which makes multiple runs of optimization inevitable. With the aim of alleviating this difficulty, some of the evolutionary algorithm based multi-objective optimization methods have been developed. In these methods bio-inspired algorithms are used to attempt to find multiple optimal solutions in a single simulation [27]. Of the many evolutionary algorithm based methods that exist, the nondominated sorting genetic algorithm (NSGA-II) is particularly well known for its strategic approach in diversity preserving as well as elite-preservation through evolutionary operators such as mutation and recombination [28]. Although the evolutionary algorithm based methods enabled us to solve multi-objective optimization problems more conveniently compared to the classical methods, there was still a remaining issue with the computational time as it typically requires a large number of function evaluations to perform an optimization. This is not a critical problem for relatively simple problems where the objective function can be evaluated within a matter of seconds to minutes. However, when more computationally intensive numerical simulations are involved to evaluate the objective functions, a large number of function evaluations becomes infeasible, particularly with a fixed time constraint, which is the case for most engineering projects.

This issue motivates the use of surrogate based optimization, where the required number of function evaluations can be reduced by replacing expensive numerical simulations with cheap-to-evaluate surrogate models and treating expensive simulations as black-box functions. Among various methods, the Gaussian processes model along with the optimization criteria that utilizes the Gaussian processes have shown its competence as well as efficiency. The concept of this optimization criteria in single-objective optimization, so-called expected improvement, was first introduced by Mockus in [24] within a framework of surrogate based optimization utilizing Kriging regression. The work has been continued by Jones et. al. [12] where the terminology Efficient Global Optimization was first introduced. The expected improvement is extended to the multi-objective optimization criteria, the expected hypervolume improvement, by Emmerich [34], and it has been demonstrated by numerous applications that it is possible to obtain results with similar accuracy using less function evaluations.

Surrogate based optimization has been actively adapted within the field of aerodynamic optimization as well. While, there is a wide variety of methods that can be used as surrogate models, it is also possible to combine evolutionary algorithms with surrogate models. For instance, in [14], surrogate assisted aerodynamic shape optimization is performed with an evolutionary algorithm such that Support Vector Machines (SVMs) are used as low fidelity surrogate models. Also in [15], the surrogate model, which consists of the proper orthogonal decomposition as well as radial basis functions interpolation, is coupled to an evolutionary algorithm to perform the airfoil design optimization in multi-point transonic conditions. Furthermore, in [38], optimization of a liquid rocket injector design is completed by using an evolutionary multi-objective method utilizing the parametric polynomial regression method as a surrogate model. In particular, surrogate based optimization that uses Gaussian processes (GPs) has been used in Multidisciplinary Design Optimization (MDO), for example in [31], Gaussian processes are used along with the expected hypervolume improvement infill criteria for the partially Reusable Launch Vehicle design.

Fig. 1 briefly illustrates each step involved in a surrogate based multi-objective optimization. Firstly, it starts with the initial sampling where it allocates the sampling points in the design space. After evaluating the corresponding objective functions, a surrogate model can be constructed using the initial sampling data set. Secondly, among these evaluated objective values, the best solutions are selected. Next, using the predefined optimization criteria, e.g. expected hypervolume improvement, the optimal point is searched in the design space assisted by a surrogate model. Lastly, after evaluating this newly found point in objective space, the best solutions are selected again.

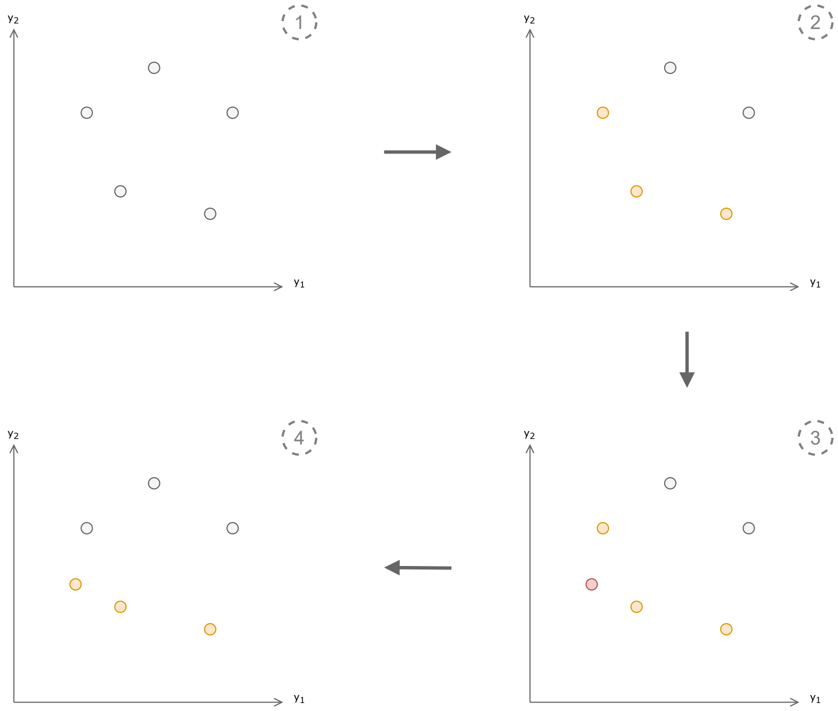


Figure 1: Schematic illustration of the surrogate based multi-objective optimization process

The primary objective of this thesis is to implement, test and analyze the state-of-the-art approaches of the global multi-objective optimization. The focus is particularly on Gaussian processes model as well as the expected hypervolume improvement. Therefore, after theoretical analysis in comparing various existing methods within this framework, selected methods are implemented and rigorously tested using analytical problems. Finally, to validate the applicability of the implemented method for more complicated and practical problems in the field of aerodynamics, optimization is performed for the two applied problems: aerodynamic shape optimization and aerodynamic-aeroacoustic shape optimization.

The thesis is structured as follows. First in Chapter 2, some theoretical insights are given for selecting initial samples, constructing surrogate model as well as a description of the general idea behind infill criteria. In Chapter 3, the multi-objective optimization problem is formally defined and the basic concepts of multi-objective optimization are described. In Chapter 4, detailed descriptions are given for the selected methods along with their practical implementation. In Chapter 5, results obtained for several analytical problems are presented and discussed. Chapter 6 contains the results for the two applied problems. Lastly, the outcomes of the thesis are concluded in Chapter 7.

Throughout the thesis, the terminology surrogate based optimization as well as Bayesian optimization are used. When the term surrogate based optimization is used the focus is generally more on the surrogate based optimization without considering a specific surrogate model. Whereas, Bayesian optimization is used to specifically refer to the surrogate based optimization that uses stochastic processes. In addition, when it is referred to optimization within this thesis, global minimization is meant unless stated otherwise.

2 Surrogate Based Optimization

In surrogate based optimization, the primary objective is to perform optimization more efficiently by replacing expensive numerical simulations with cheaper-to-evaluate surrogate models or meta-models. In order to achieve this, the chosen surrogate model should be able to reproduce these expensive models well enough such that the optimization can be done efficiently using sophisticated criteria. As much as it is important to choose a suitable surrogate model or optimization criteria, it is important to select an effective sampling plan which could have an unexpectedly large influence on the total computational budget. By choosing a suitable sampling plan, numerous unnecessary additional runs of the expensive black-box functions can be avoided. Thus, methods and discussions about sampling methods are included in Section 2.1, followed by a description of the selected surrogate model in Section 2.2. Lastly, the chosen infill criteria is explained briefly showing an single-objective optimization example in Section 2.3.

2.1 Design of Experiments

When treating an expensive simulation as a black-box function, data needs to be provided to build surrogate models that somewhat resemble the actual black-box function. With a limited computational budget, it is crucial to select a minimum yet effective number and location of sampling points. This process of deciding on a minimum yet effective number and location of sampling points is called *design of experiments* (DOE) or initial sampling, and can be done in various ways.

Among various available sampling methods, one of the intuitive methods would be random sampling where a sample is generated as a random realization by a pseudo random generator with some distribution, e.g. $x \sim \mathcal{N}(0, 1)$. The limitation of random sampling, however, is that it may contain clusters and gaps, which are generally undesirable for the design of experiments as they can impede well distributed samples in the design space. To overcome this limitation, stratified sampling is suggested where samples are placed randomly within a prescribed sub-domains [45]. Then, it is possible to determine the feasible sample size using full factorial design as explained in [4]. In this full factorial design, the number of the required simulations N to ensure the adequate coverage of the design space is computed as follows:

$$N = s^k \tag{1}$$

where s denotes the level or the strata and k is the number of parameters. In the most general case, $s = 2$ is used to prevent an explosive growth of total simulations required, which is the disadvantage of using full factorial design for large k and/or s values.

2.1.1 Latin hypercube sampling

Latin hypercube sampling (LHS) is a type of the stratified sampling where each parameter is stratified over $s > 2$ levels such that each level contains the same number of points [4]. The primary goal of LHS is then to partition the sample space such that the design space is well represented through sampling done in these sub-regions. This is shown in Fig. 2 for each 1, 2 and 3 parameters. Using two-level full factorial, the domain of each parameter is partitioned into 2^k hypercubes.

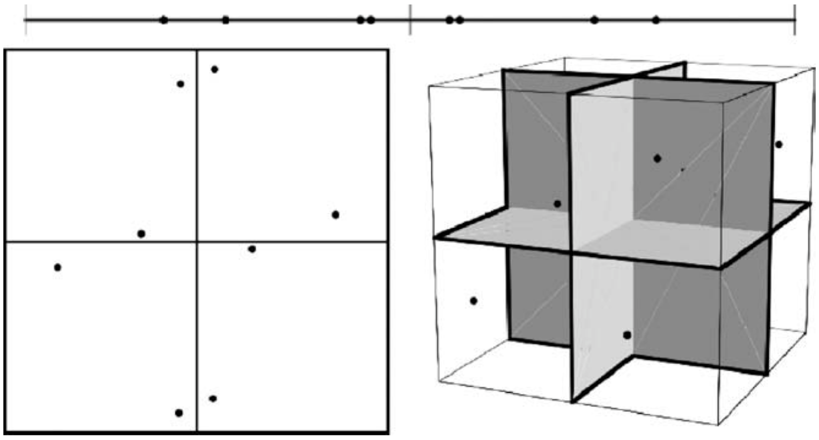


Figure 2: Samples distributed over the hypercubes in Latin hypercube sampling (LHS) in $\mathbb{R}^1$ (top), $\mathbb{R}^2$ (left) and $\mathbb{R}^3$ (right). From [4]

Although, it is ideal to have a full-factorial design, sometimes it can be computationally expensive, e.g. a two-level full factorial would require 1024 simulations for 10 parameters, in which case a fractional factorial design approach is suggested [4].

2.1.2 Sobol sequences based sampling

Sobol sequences based sampling uses quasi-random sequences that enhance the uniformity across the design space [3]. It is also a low-discrepancy sequence which reduces the discrepancy at the theoretically optimal rate as the sequence length becomes larger. This is done by generating low-discrepancy sequences that are somewhat biased towards the unsampled regions to ensure an even distribution while maintaining a certain uniformity [4].

The characteristics of the two different sampling methods are shown in Fig. 3 for $\mathbb{R}^2$. On the top, the LHS sampling results are presented for 20 and 30 sampling points, while the Sobol sampling results are shown on the bottom. Comparing the plots, the distinctive difference between the two methods, randomness, is visible. To elaborate on this, first looking at the two plots created by the LHS sampling, for different numbers of samples it becomes evident that essentially completely new sample points are generated while retaining an equal number of samples per hypercube. The results of the Sobol method, on the contrary, show that it is simply adding samples, in a sequence, as the number of samples increases. Comparing the two plots of the Sobol method, the blue dots on the left plot ($N = 20$) are identical to the blue dots on the right ($N = 30$), which represent the first 20 samples out of the total 30 samples generated. Here, the additional points are marked as orange dots, which are shown to be filling in the rest of the gaps while retaining uniformity.

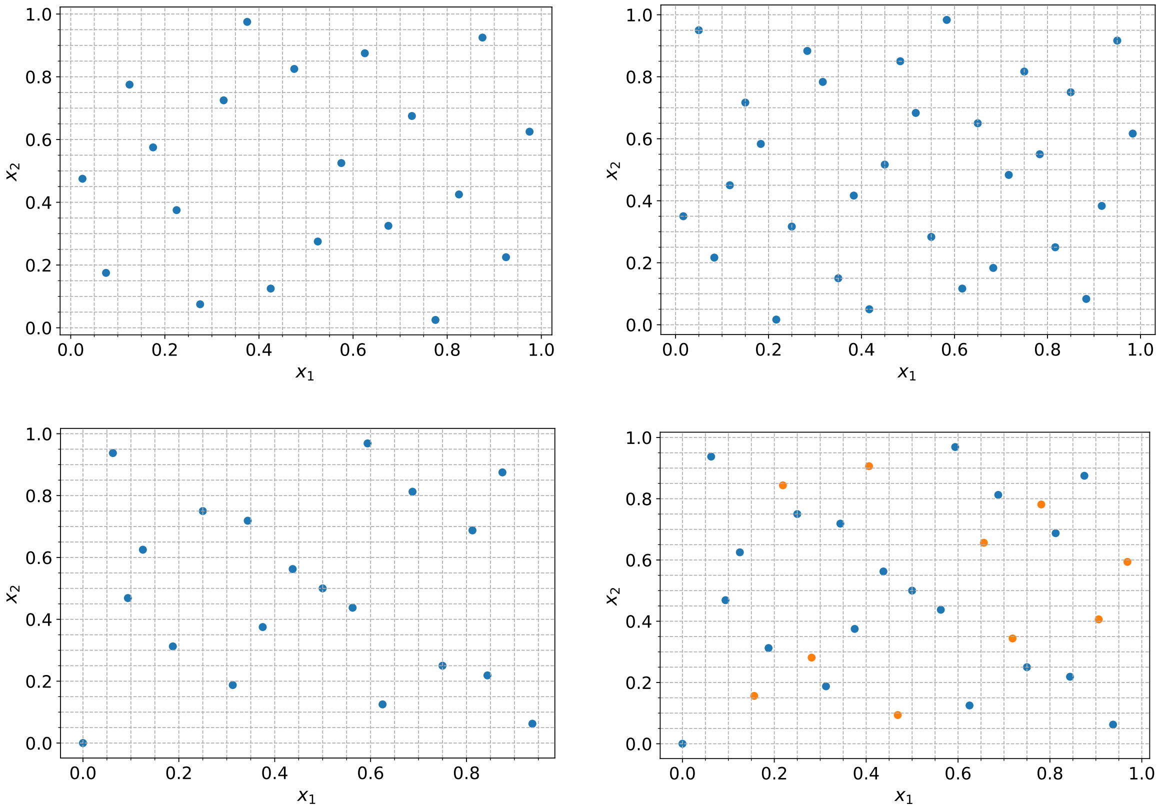


Figure 3: LHS sampling with $N = 20$ (top left) and $N = 30$ (top right). Sobol sampling with $N = 20$ (bottom left) and $N = 30$ (bottom right).

Apart from its ability to ensure uniformity and low-discrepancy, this deterministic feature of the Sobol sampling becomes particularly useful during an optimization process where the accuracy of the initial surrogate model turns out to be insufficient, as the accuracy of the surrogate model can be improved simply by adding the additional points generated by the Sobol sequence [11].

2.2 Surrogate Model

In surrogate based optimization, surrogate models should be able to represent the unknown black-box function to a certain extent. As long as this is satisfied, any prediction method can be used as a surrogate model. Though, one thing to consider when choosing a surrogate model is that the engineering function is almost always continuous such that it is safe to assume that the surrogate model should also be continuous [2]. Particularly, for Kriging, which is used in this thesis work, this is the only assumption. Nonetheless, it is possible to make further assumptions, for instance, based on the known shape of the functions, which can be found in detail in [2].

The simplest yet most intuitive way to construct a continuous surrogate model would then be to employ the regression method. And, some of the most popular methods can be listed as follows [5]:

- Moving least squares
- Regression trees
- Artificial neural networks
- Generalized additive models
- Supported vector regression
- Gaussian Processes (GPs)
- Kernel partial least squares

Particularly in MDO, [5] shows that the methods that use Gaussisan Processes are the most convenient for generalization in complex problems with small data-sets and many variables. Apart from its capability in regression for complex functions, Kriging has gained popularity in surrogate based optimization due to its ability to provide uncertainty quantification, which can be conveniently exploited within the acquisition function for searching for optimal infill points during optimization.

One critical point not to be overlooked is that what we want for a surrogate model is not error-free replication of a black-box function. What is important is that it is accurate enough such that an optimizer can identify promising areas in searching for an optimal point.

2.2.1 Kriging/Gaussian Processes

Kriging is essentially a stochastic process model that relies on Gaussian processes, which can be thought of as a generalization of multivariate Gaussian to infinitely many variables. With Kriging, it is possible to model given multi-dimensional samples as a statistical model by maximum likelihood estimates (MLEs). Furthermore, whenever a new sample is added, it can re-model and provides a most likely prediction as well as an estimated error by following the same procedure.

The mathematical derivations and equations presented in the following discussion are based on [12] and [1].

In order to motivate the use of Kriging from a more fundamental point of view, let's consider the most generic regression form and identify the limitations within it. For a given sampled data at point i, $\boldsymbol{x}^{(i)}$, and its objective value $Y(\boldsymbol{x}^{(i)})$, the simplest and general way of modeling by fitting the data set would be by using a simple regression model of the form,

$$Y(\boldsymbol{x}^{(i)}) = \sum_j w_j h_j(\boldsymbol{x}^{(i)}) + \epsilon \qquad \text{for } \; i = 1, \dots, n \tag{2}$$

here, $h_j(\boldsymbol{x}^{(i)})$ is a linear or nonlinear function, w are the unknown coefficient that need to be estimated and ϵ refers to independent error terms with $\epsilon \sim N(\mathbf{0}, \sigma^2)$. However, the problem of using linear regression is that its assumption of independent errors does not hold when deterministic computer codes are used for modeling, as any error arising from this will be most certainly modeling error, not measurement error or noise, e.g. points $\boldsymbol{x}$ that have not been sampled [12]. Hence, it is more feasible to assume the error terms are a function of $\boldsymbol{x}$ such that $\epsilon(\boldsymbol{x})$. This is how the regression is done with Gaussian Processes (GPs) or Kriging. More precisely, in Kriging, the errors are assumed to be correlated using the Gaussian like function:

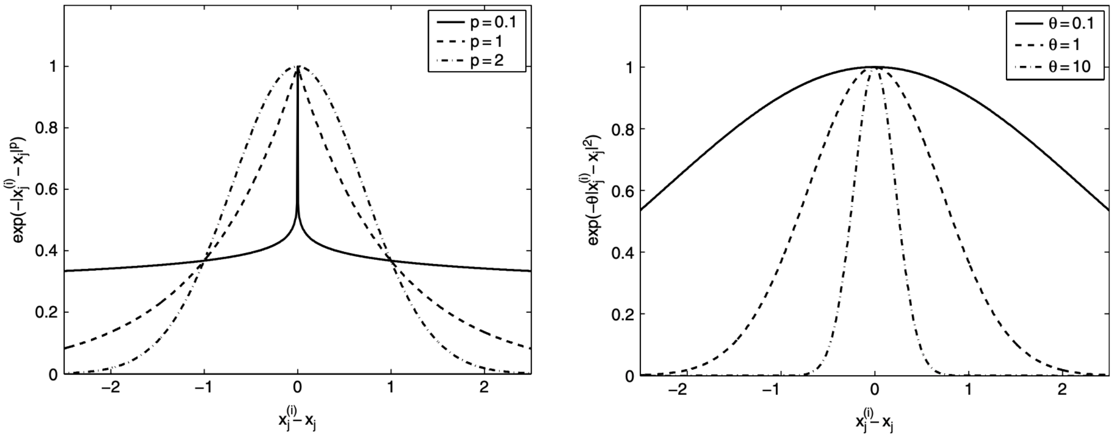
$$\mathrm{Corr}\left[\epsilon(\boldsymbol{x}^{(i)}), \epsilon(\boldsymbol{x}^{(j)})\right] = \exp\left(-\sum_{k=1}^{n} \theta_k |x_k^{(i)} - x_k^{(j)}|^{p_k}\right). \tag{3}$$

While this function expresses the correlation between the errors at $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ by the exponential difference between the two points, the behavior of the hyperparameters θ_k and p_k resemble the ones from the actual Gaussian function except that in Gaussian p is fixed to two. As a result, θ_k behaves similar to the inverse of the variance. Thus, increasing this parameter will result in a narrower correlation function and vice versa as shown in Fig. 4b. From this plot, it can be seen that for increasing θ, the corresponding correlation function distribution becomes narrower, which means that only the points that are nearby, smaller $|x_k^{(i)} - x_k^{(j)}|$, are correlated. The parameter p_k controls the smoothness of the correlation function as shown in Fig. 4a. The plot indicates that as p increases the function becomes smoother and vice versa.

By modeling the correlation of the errors in this way, the stochastic process can be written in a form,

$$Y(\boldsymbol{x}^{(i)}) = \mu + \epsilon(\boldsymbol{x}^{(i)}) \qquad \text{for } \; i = 1, \dots, n \tag{4}$$

where, μ represents the mean of the stochastic process, and $\epsilon(\boldsymbol{x})$ is the random variable with $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2)$ which is correlated as in Equation (3). Then, the parameters θ_k and p_k can be estimated by maximizing the likelihood of the stochastic response $\boldsymbol{Y} = \left(Y(\boldsymbol{x}^{(1)}), \dots, Y(\boldsymbol{x}^{(n)})\right)^\top$. First, let the observed function values be $\boldsymbol{y} = \left(y^{(1)}, \dots, y^{(n)}\right)^\top$, then, the correlation matrix, denoted as $\boldsymbol{\Psi}$, can be constructed using Equation (3),



(a) varying parameter p (b) varying parameter θ

Figure 4: Effect of varying hyperparameters in the correlation function with respect to the distance. From [1]

$$\boldsymbol{\Psi} = \begin{pmatrix} \text{Corr}\left[\epsilon(\boldsymbol{x}^{(1)}), \epsilon(\boldsymbol{x}^{(1)})\right] & \cdots & \text{Corr}\left[\epsilon(\boldsymbol{x}^{(1)}), \epsilon(\boldsymbol{x}^{(n)})\right] \\ \vdots & \ddots & \vdots \\ \text{Corr}\left[\epsilon(\boldsymbol{x}^{(n)}), \epsilon(\boldsymbol{x}^{(1)})\right] & \cdots & \text{Corr}\left[\epsilon(\boldsymbol{x}^{(n)}), \epsilon(\boldsymbol{x}^{(n)})\right] \end{pmatrix} \tag{5}$$

note that the main diagonal of $\boldsymbol{\Psi}$ is always one as the random variable is perfectly correlated to itself, which can be also seen from Fig. 4 where the correlation becomes 1 as $|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i)}| \to 0$. Then, the likelihood of the stochastic responses $\boldsymbol{Y}$ is written as follows:

$$L(\boldsymbol{Y}^{(1)}, \ldots, \boldsymbol{Y}^{(n)} \mid \mu, \sigma) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left[-\frac{\sum(\boldsymbol{Y}^{(i)} - \mu)^2}{2\sigma^2}\right] \tag{6}$$

expressing Equation (6) using the correlation matrix and the sampled data yields,

$$L = \frac{1}{(2\pi\sigma^2)^{n/2}|\boldsymbol{\Psi}|^{1/2}} \exp\left[-\frac{(\boldsymbol{y} - \mathbf{1}\mu)^\top \boldsymbol{\Psi}^{-1}(\boldsymbol{y} - \mathbf{1}\mu)}{2\sigma^2}\right] \tag{7}$$

to make the computation easier, we can take the logarithm to work with log-likelihood,

$$\ln(L) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2}\ln|\boldsymbol{\Psi}| - \frac{(\boldsymbol{y} - \mathbf{1}\mu)^\top \boldsymbol{\Psi}^{-1}(\boldsymbol{y} - \mathbf{1}\mu)}{2\sigma^2} \tag{8}$$

By taking derivative of Equation (8) and setting it to zero, the maximum likelihood estimates (MLEs) of μ and σ^2 are obtained as follows:

$$\widehat{\mu} = \frac{\mathbf{1}^\top \boldsymbol{\Psi}^{-1}\boldsymbol{y}}{\mathbf{1}^\top \boldsymbol{\Psi}^{-1}\mathbf{1}} \tag{9}$$

$$\widehat{\sigma}^2 = \frac{(\boldsymbol{y} - \mathbf{1}\mu)^\top \boldsymbol{\Psi}^{-1}(\boldsymbol{y} - \mathbf{1}\mu)}{n} \tag{10}$$

Lastly, substituting these expressions into Equation (8) yields the concentrated log-likelihood function:

$$\ln(L) \approx -\frac{n}{2}\ln(\widehat{\sigma}^2) - \frac{1}{2}\ln|\boldsymbol{\Psi}| \tag{11}$$

hyperparameters θ_k and p_k can be found by maximizing Equation (11) using a global search method, e.g. a genetic algorithm or simulated annealing.

Furthermore, with this kriging model which is constructed by MLEs, it is possible to do a regression for a new prediction $y^{(+)}$. First, the correlation function for the new data can be augmented into the previous corrleation matrix as follows:

$$\widetilde{\boldsymbol{\Psi}} = \begin{pmatrix} \boldsymbol{\Psi} & \boldsymbol{\psi} \\ \boldsymbol{\psi}^\top & 1 \end{pmatrix} \tag{12}$$

where,

$$\boldsymbol{\psi} = \begin{pmatrix} \mathrm{Corr}\left[\epsilon(\boldsymbol{x}^{(1)}), \epsilon(\boldsymbol{x}^{(+)})\right] \\ \vdots \\ \mathrm{Corr}\left[\epsilon(\boldsymbol{x}^{(n)}), \epsilon(\boldsymbol{x}^{(+)})\right] \end{pmatrix} \tag{13}$$

the log-likelihood of the augmented data with $\tilde{\boldsymbol{y}} = \{\boldsymbol{y}, y^{(+)}\}^\top$ is expressed as follows:

$$\ln(L) = -\frac{n}{2}\ln(2\pi) - -\frac{n}{2}\ln(\widehat{\sigma}^2) - \frac{1}{2}\ln|\widetilde{\boldsymbol{\Psi}}| - \frac{(\tilde{\boldsymbol{y}} - \mathbf{1}\widehat{\mu})^\top \widetilde{\boldsymbol{\Psi}}^{-1}(\tilde{\boldsymbol{y}} - \mathbf{1}\widehat{\mu})}{2\widehat{\sigma}^2} \tag{14}$$

again, this likelihood needs to be maximized to find the hyperparameters. Since, only the last term depends on $y^{(+)}$, this term is used for maximization which then yields,

$$\ln(L) \approx \frac{-\begin{pmatrix} \boldsymbol{y} - \mathbf{1}\widehat{\mu} \\ \widehat{y} - \widehat{\mu} \end{pmatrix}^\top \begin{pmatrix} \boldsymbol{\Psi} & \boldsymbol{\psi} \\ \boldsymbol{\psi}^\top & 1 \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{y} - \mathbf{1}\widehat{\mu} \\ \widehat{y} - \widehat{\mu} \end{pmatrix}}{2\widehat{\sigma}^2} \tag{15}$$

where the inverse of $\widetilde{\boldsymbol{\Psi}}$ can be computed through the partitioned inverse method of Theil:

$$\widetilde{\boldsymbol{\Psi}}^{-1} = \begin{pmatrix} \boldsymbol{\Psi}^{-1} + \boldsymbol{\Psi}^{-1}\boldsymbol{\psi}(1 - \boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})^{-1}\boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1} & -\boldsymbol{\Psi}^{-1}\boldsymbol{\psi}(1 - \boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})^{-1} \\ -(1 - \boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})^{-1}\boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1} & (1 - \boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})^{-1} \end{pmatrix}. \tag{16}$$

Substituting this equation into Equation (15) and rearranging the terms that contain $y^{(+)}$ gives,

$$\ln(L) \approx \left(-\frac{1}{2\widehat{\sigma}^2(1-\boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})}\right)(y^{(+)}-\widehat{\mu})^2 + \left(\frac{\boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}(\boldsymbol{y}-\mathbf{1}\widehat{\mu})}{\widehat{\sigma}^2(1-\boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})}\right)(y^{(+)}-\widehat{\mu}). \tag{17}$$

This can be maximized by differentiating it with respect to $y^{(+)}$ and setting it to zero,

$$\left(-\frac{1}{\widehat{\sigma}^2(1-\boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})}\right)(y^{(+)}-\widehat{\mu}) + \left(\frac{\boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}(\boldsymbol{y}-\mathbf{1}\widehat{\mu})}{\widehat{\sigma}^2(1-\boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})}\right) = 0 \tag{18}$$

finally, rearranging this equation, the maximum likelihood estimate of the new point $y^{(+)}$ is given as follows:

$$y^{(+)}(\boldsymbol{x}) = \widehat{\mu} + \boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}(\boldsymbol{y}-\mathbf{1}\widehat{\mu}) \tag{19}$$

this equation describes the general MLE of, so-called ordinary Kriging, the equation for universal Kriging, where the mean becomes a function of $\boldsymbol{x}$ can be written as follows,

$$y^{(+)}(\boldsymbol{x}) = \widehat{\mu}(\boldsymbol{x}) + \boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}(\boldsymbol{y}-\mathbf{1}\widehat{\mu}) \tag{20}$$

where,

$$\widehat{\mu}(\boldsymbol{x}) = \sum_{i=0}^{m} \mu_i v_i(\boldsymbol{x}) \tag{21}$$

μ_i and v_i each represent the unknown parameters and some known functions. Although this can improve the accuracy by capturing known trends with the additional tuning, there is a risk of introducing inaccuracies if there is no prior knowledge on the trends. Thus, in this thesis work, ordinary Kriging is used extensively.

One of the greatest advantages of using Kriging in surrogate based optimization is that it provides uncertainty quantification on its predictions, which is used for finding subsequent infill sample points during the multi-objective optimization (this will be explained in detail in Chapter 4). The estimated mean squared error (MSE) of Kriging yields [25]:

$$\widehat{s}^2(\boldsymbol{x}) = \sigma^2\left[1-\boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{\psi} + \frac{1-\mathbf{1}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{\psi}}{\mathbf{1}^\top\boldsymbol{\Psi}^{-1}\mathbf{1}}\right] \tag{22}$$

This error diminishes to zero at the sampled points.

2.3 Infill criteria for optimization

Once the initial sampling plan and the surrogate model are constructed, the optimization process proceeds by adding a new point, called infill point, to the surrogate model. In this way, it can perform optimization while improving the model accuracy near the promising regions where the optimization process sees that it is most likely to find the optimal point. This is also sometimes referred to as *adaptive sampling*. The criteria given for selecting new infill points is called *infill criteria*, and the function used as a infill criteria is also known as *acquisition function*. Depending on the purpose of the optimization, for instance, whether it has a single objective or multiple objectives, different infill criteria are used to achieve different goals.

In this section, the infill criteria for the single-objective optimization is explained. Although it is expressed in a much simpler form compared to the infill criteria for multi-objective optimization, which comes from its simplicity in performance metric, it essentially shares some of the common principle features with the infill criteria for multi-objective optimization, such that if one understands principles of the infill criteria for single-objective optimization, extension to multi-objective optimization infill criteria would become much more intuitive. In particular, how uncertainty predictions provided by Kriging play a role in optimization would become clear from this single-objective infill criteria formulation.

2.3.1 Single-objective Optimization

The primary objective of single-objective optimization is to find the objective value y that is a global minimum. Meanwhile, in Section 2.2, it was stated that the advantage of kriging is that it provides uncertainty quantification, which can be used for finding an infill sample point. In fact, the infill criteria for single-objective optimization can be formulated with the uncertainty predictions of kriging such that it enhances both exploitation and exploration of optimization. To be more specific, the infill criteria, so-called *Expected Improvement* (EI) can be formulated. With a kriging model, it is possible to model a stochastic response as $Y(\boldsymbol{x}) \sim \mathcal{N}\left(\widehat{y}(\boldsymbol{x}), \widehat{s}^2(\boldsymbol{x})\right)$, where $\widehat{y}(\boldsymbol{x})$ is a re-expressed notation of $y^{(+)}(\boldsymbol{x})$ from the previous section. In addition, for convenience in terminology, the variance $\widehat{s}^2(\boldsymbol{x})$ will be denoted as $\widehat{\sigma}^2(\boldsymbol{x})$ in this section. Then, the expected value of improvement, defined as $I = y_{min} - Y(\boldsymbol{x})$, at $\boldsymbol{x}$ can be computed as:

$$E[I(\boldsymbol{x})] = (y_{\text{min}} - \widehat{y}(\boldsymbol{x}))\Phi\left(\frac{y_{\text{min}} - \widehat{y}(\boldsymbol{x})}{\widehat{\sigma}(\boldsymbol{x})}\right) + \widehat{\sigma}(\boldsymbol{x})\phi\left(\frac{y_{\text{min}} - \widehat{y}(\boldsymbol{x})}{\widehat{\sigma}(\boldsymbol{x})}\right) \tag{23}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ denote the *cumulative distribution function* (CDF) and the *probability density function* (PDF) respectively. This expected improvement diminishes to zero at the sampled point, i.e. $\widehat{\sigma} = 0$, and a maximum expected improvement infill procedure will eventually find the global optimum [1]. In order to find an infill point, $\boldsymbol{x}$

is found by maximizing Equation (23). Its performance as an infill criteria, i.e. how this equation enhances a balanced exploitation and exploration, can be better understood by examining it. From Equation (23), it is clear that the EI will increase mainly in two different situations. One the one hand, the first term will be maximized at the point where the predicted improvement is higher. Being weighted by the CDF, it will enforce an optimizer to find a point where it is most likely to be the best value, thus, enhancing exploitation. On the other hand, the second term will be maximized for a point where higher error are present within the Gaussian processes, which typically are unsampled regions. Thus, this will make an optimizer to explore more of the not yet discovered design space to make sure that approximately most of the design space has been covered or sampled. In this way, using expected improvement as an infill criteria effectively promotes a balance between exploitation and exploration, which is pivotal in global optimization problems.

3 Multi-objective Optimization

In multi-objective optimization, with the presence of multiple conflicting objectives and their evaluations, it is not very helpful to find just one point among all the possible combinations. For this reason, generally, the goal of the multi-objective optimization is to obtain a well distributed set of optimal solutions that represent different combinations of the objectives. For decades, different approaches have been developed to obtain the optimal solution for multi-objective optimization problems, and in this chapter, some of the basic concepts and ideas behind multi-objective optimization is presented.

Firstly, the formal problem definition of a multi-objective problem of interest is given in 3.1. Fundamental concepts in multi-objective optimization and ideas behind are described in 3.2.

3.1 Problem formulation

Formally, a multi-objective optimization problem is defined as follows,

$$\min_{\boldsymbol{x}\in\Omega} \left(f_1(\boldsymbol{x}), f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})\right) \tag{24}$$

here, $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of *decision variables* subjected to the feasible set $\Omega \subset \mathbb{R}^n$, which is typically defined by the host set $D \subseteq \mathbb{R}^n$ such that $\Omega = \{\boldsymbol{x} \in D\}$ where the host set D is defined by the upper and the lower bounds of the decision variables as $D = [\boldsymbol{x}_L, \boldsymbol{x}_U] \subset \mathbb{R}^n$. Lastly, the functions $f_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, \ldots, m$ represent the *objective functions*, and the *objective space* is the image of the feasible set $\mathcal{F} = \{\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^m \mid \boldsymbol{x} \in \Omega\}$.

3.2 Basics of multi-objective optimization

In this section, some of the basic concepts and the terminology of multi-objective optimization is explained. Although surrogate based optimization takes a slightly different approach compared to the conventional multi-objective optimization methods, it is essentially aiming for solving multi-objective problems. Because of this, many concepts from the conventional multi-objective optimization have been either directly taken or adapted to form the basis of the surrogate based optimization formulation.

In Section 3.2.1, an overview of the concept of Pareto optimality is given along with the definitions of some of the pivotal concepts in multi-objective optimization. Section 3.2.2 discusses performance metrics of general multi-objective optimization along with a brief glimpse of performance metrics in Bayesian optimization.

3.2.1 Pareto optimality

In the following, the basic concepts and definitions of Pareto optimality and multi-object optimization are given based on [7] and [30].

Dominance Dominance in multi-objective optimization refers to the superior-inferior relation when comparing different sets in the objective space.

For given decision vectors $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)} \in \mathbb{R}^n$, let their objective values be $\boldsymbol{f}^{(1)} = f(\boldsymbol{x}^{(1)}), \boldsymbol{f}^{(2)} = f(\boldsymbol{x}^{(2)})$. For the minimization problem, it is said that $\boldsymbol{f}^{(1)}$ dominates $\boldsymbol{f}^{(2)}$ (denoted by $\boldsymbol{f}^{(1)} \prec \boldsymbol{f}^{(2)}$) if and only if:

- $f_i(\boldsymbol{x}^{(1)}) \leq f_i(\boldsymbol{x}^{(2)}) \qquad \forall i \in 1, 2, \cdots, m$
- $f_j(\boldsymbol{x}^{(1)}) < f_j(\boldsymbol{x}^{(2)}) \qquad \exists j \in 1, 2, \cdots, m$

Non-dominance Accordingly, the non-dominance relation can be defined such that decision variables $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^n$ are said to be non-dominated with respect to $\mathcal{X}$, if there is no element $\boldsymbol{x}' \in \mathcal{X}$ such that $\boldsymbol{f}(\boldsymbol{x}') \prec \boldsymbol{f}(\boldsymbol{x})$ holds.

Pareto optimal set The Pareto optimal set refers to the set of design variables that their objective values are non-dominated for all elements in a feasible set, formally defined as follow,

$$\mathcal{P}^* = \{\boldsymbol{x} \in \Omega \mid \nexists \boldsymbol{z} \in \Omega : \boldsymbol{z} \prec \boldsymbol{x}\} \tag{25}$$

Pareto front Finally, the optimal Pareto front is defined as the objective values of the Pareto optimal set. This is essentially the formal indicator of the set of solutions of interest in multi-objective optimization explained previously. The definition is written as,

$$\mathcal{PF}^* = \{\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^m \mid \boldsymbol{x} \in \mathcal{P}^*\} \tag{26}$$

throughout this thesis, the term optimal Pareto front is explicitly used to indicate this mathematical definition. In general, the term Pareto front is used to refer to the non-dominated subset or the approximated Pareto front.

Dominated subspace of a set For a $\mathcal{P} \subset \mathbb{R}^m$, the dominated subspace of $\mathcal{P}$ in $\mathbb{R}^m$ (denoted by $d(\mathcal{P})$) is given as,

$$d(\mathcal{P}) := \{\boldsymbol{y} \in \mathbb{R}^m \mid \exists \boldsymbol{p} \in \mathcal{P} : \boldsymbol{p} \prec \boldsymbol{y}\} \tag{27}$$

Non-dominated space of a set For a $\mathcal{P} \subset \mathbb{R}^m$, let $\boldsymbol{r} \in \mathbb{R}^m$ be dominated by $\boldsymbol{p}$ for all $\boldsymbol{p} \in \mathcal{P}$. Then, the non-dominated space of $\mathcal{P}$ with respect to $\boldsymbol{r}$ (denoted by $nd(\mathcal{P})$) is defined as,

$$nd(\mathcal{P}) := \{\boldsymbol{y} \in \mathbb{R}^m \mid \boldsymbol{y} \prec \boldsymbol{r} \wedge \nexists \boldsymbol{p} \in \mathcal{P} : \boldsymbol{p} \prec \boldsymbol{y}\} \tag{28}$$

3.2.2 Performance metrics

With the formal definition of Pareto optimality, one question that arises is: how do we assess whether the obtained Pareto Front is indeed well represented? This will be dependent on the goal of each problem, but in general, from the perspective of Pareto optimality, the main interest is in obtaining a Pareto front that is as close as possible to the optimal Pareto front while covering as many as possible combinations in the objective space as illustrated in Fig. 5. Though, it is imperative to keep in mind that this optimal Pareto front is almost always not known in advanced - if known, there is no point of performing optimization.

Choosing a performance metric is of particular importance for surrogate based optimization as it becomes the basis of the infill criteria formulation.

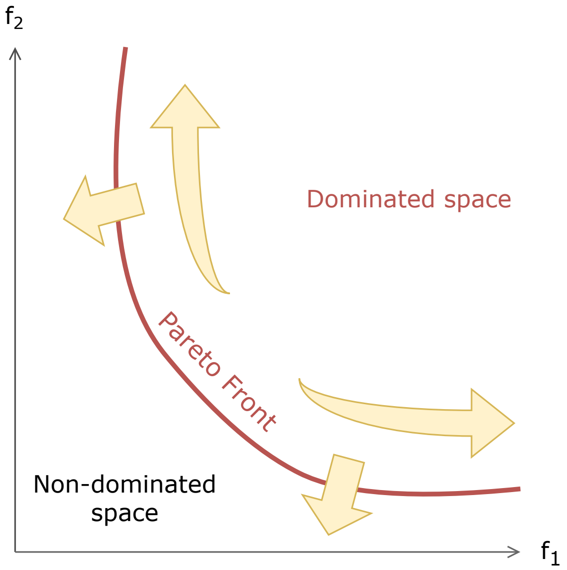


Figure 5: Illustration of the representation of well distributed Pareto Front in the objective space

Since the performance metric can have an impact on the efficiency and accuracy of the optimization process, various metrics have been developed for solving multi-objective optimization problems. For multi-objective Bayesian optimization, *Hypervolume Indicator* (HV) based performance metrics, which was first introduced by [16] (also known as $S - metric$), are most commonly used.

Some of the hypervolume based performance metrics used in Bayesian optimization and are relevant for this work include *Hypervolume Improvement* (HVI) and *Probability of Improvement* (PoI). In the following, formal definitions are given based on [30].

Hypervolume Indicator (HV)

Let $\mathcal{P} = \left\{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \cdots, \boldsymbol{y}^{(k)}\right\} \subset \mathbb{R}^m$ be the approximated Pareto Front. Then, the hypervolume indicator of $\mathcal{P}$ is defined as the m-dimensional Lebesgue measure λ_m on $\mathbb{R}^m$ of the subspace dominated by $\mathcal{P}$ and bounded by the given reference point $\boldsymbol{r}$.

$$HV(\mathcal{P}) = \lambda_m \left(\bigcup_{\boldsymbol{y} \in \mathcal{P}} [\boldsymbol{y}, \boldsymbol{r}] \right) \tag{29}$$

Fig. 6 shows the illustration of the hypervolume indicator defined as the subspace bounded by the given Pareto set $\mathcal{P} = \left\{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \boldsymbol{y}^{(3)}\right\}$ and a reference point $\boldsymbol{r}$ in $\mathbb{R}^2$. Solving an optimization, either minimization or maximization, problem with the hypervolume indicator as a metric is equivalent to maximization of the hypervolume.

One of the most significant properties of the hypervolume indicator is that it is the only strictly monotonic unary indicator known for the Pareto front approximation [18], meaning that for the given sets A and B, $A \prec B$ implies that $I(A) > I(B)$ with I being an unary indicator.

Furthermore, compared to the non-dominated sorting of NSGA-II, the hypervolume indicator metric outperforms the crowding distance metric in a way that solutions around the knee of a curve are often ranked higher within the hypervolume indicator compared to crowding distance, where the outer points, that contribute to the uniform distribution, are often ranked higher [34]. Considering that most of the promising solutions are likely to be found in the proximity of the knee of the curve, it is likely that crowding distance leaves out some of these sets in contrast to the hypervolume indicator. Besides, unlike crowding distance there is no dependency between solution candidates for the hypervolume indicator, which can again impede finding potentially promising solutions.

The reference point $\boldsymbol{r}$ in the hypervolume indicator metric needs to be provided by a user. In general, when the hypervolume itself is used as a direct performance metric of the optimization process, the location of this reference point could alter the process and effect the result of the optimization, e.g. for two sets of Pareto fronts, altering the reference value can change which set of the Pareto front has the larger hypervolume. However, from the perspective of Bayesian optimization this becomes quite trivial, mainly because the hypervolume indicator is not used as a direct performance metric of optimization. Further explanations of this will be given accordingly in the following sections. In the scope of this thesis work, the hypervolume indicator as a direct performance metric is used explicitly to compare two or more computed Pareto fronts.

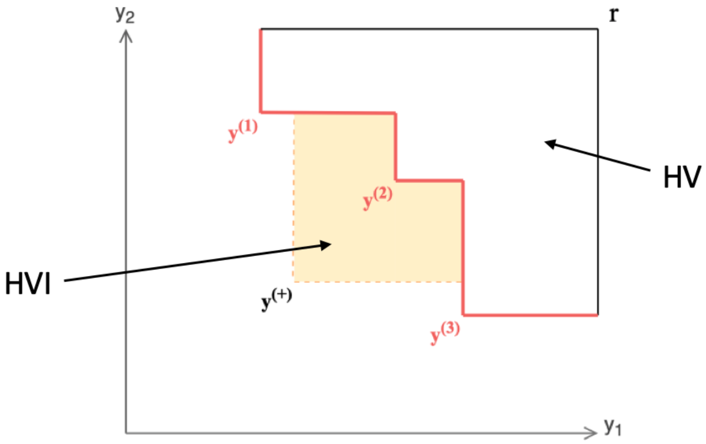


Figure 6: Illustration of the hypervolume indicator (HV) for a given Pareto set $\mathcal{P} = \{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \boldsymbol{y}^{(3)}\}$, and the hypervolume improvement (HVI) for a given vector $\boldsymbol{y}^{(+)}$

Hypervolume Improvement (HVI)

For a given set of vectors of $\mathcal{P} \subset \mathbb{R}^m$, the hypervolume improvement for a given vector $\boldsymbol{y}$ is defined as,

$$HVI(\boldsymbol{y}, \mathcal{P}) = HV(\mathcal{P} \cup \boldsymbol{y}) - HV(\mathcal{P}) \tag{30}$$

from the equation, it can be noticed that the hypervolume improvement is essentially the difference or more precisely the improvement in the hypervolume when a vector $\boldsymbol{y}$ is included in the original set. Interestingly, contrary to what seems so obvious, this property is aptly exploited for the efficient computation of the hypervolume which will be discussed in the next chapter.

From Fig. 6 the visual representation of the hypervolume improvement is shown as colored area. As it can easily be inferred from its definition above, the hypervolume of the set $\mathcal{P} \cup \boldsymbol{y}$ can be expressed as the sum of the original hypervolume and the hypervolume improvement.

As for the hypervolume indicator, the reference point naturally reappears for the hypervolume improvement. While a different reference point could result in a different hypervolume indicator, the hypervolume improvement, is less affected by the reference setting as illustrated in Fig. 7. The two plots have the exact same Pareto fronts, and it can be clearly seen that defining different reference points r and r^* results in

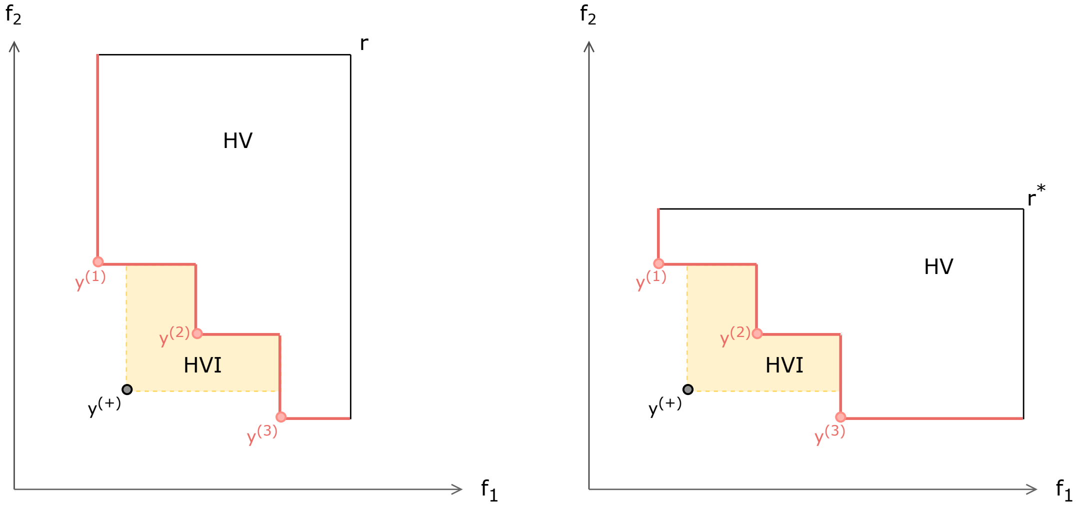


Figure 7: Although the different reference point setting gives the different hypervolume indicator for the same Pareto front, hypervolume improvement is not affected by this change unless the extreme points are dominated by the vector $\boldsymbol{y}^{(+)}$

different hypervolume indicators. However, the hypervolme improvement for the additional vector $y^{(+)}$, e.g. infill point, is not affected by the reference point. Of course, it could change for some extreme cases, e.g. adding a point that dominates the extreme values in the Pareto set. However, the reasoning for setting a reference value for multi-objective optimization Bayesian optimization should be based on the process of optimizing using an infill criteria which will be discussed in detail in the next chapter.

Probability of Improvement

Another performance metric is *Probability of Improvement* (PoI) which indicates the probability that $\boldsymbol{y}$ dominates the Pareto front $\mathcal{P}$. The formal definition is written as:

$$PoI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}) := \int_{\mathbb{R}^m} \mathbb{1}_Y(\boldsymbol{y}) \xi_{\mu,\sigma}(\boldsymbol{y}) d\boldsymbol{y} \tag{31}$$

where ξ represents the independent multivariate Gaussian distribution with mean values $\boldsymbol{\mu} \in \mathbb{R}^m$ and standard deviations $\boldsymbol{\sigma} \in \mathbb{R}^m$, and $\mathbb{1}(\cdot)$ is an indicator function, where $\mathbb{1}_Y(\boldsymbol{y}) = 1$ indicates the improvement for $\boldsymbol{y}$ with respect to $\mathcal{P}$ as,

$$Y = \{\boldsymbol{y} \in \mathbb{R}^m \mid \boldsymbol{y} \prec \boldsymbol{r} \wedge \forall \boldsymbol{p} \in \mathcal{P} : \neg(\boldsymbol{p} \prec \boldsymbol{y})\} \tag{32}$$

Although this performance metric is not extensively used for this thesis, it is closely related to the concept that is used in one of the methods for multi-objective infill criteria formulation which will be discussed in the next chapter.

4 Multi-objective Bayesian Optimization framework

The efficient global optimization framework can be extended to multi-objective optimization by directly adapting the performance indicator discussed in Section 3.2.2 as the infill criteria for the surrogate based optimization process that was explained in Chapter 2. The entire process of the framework is illustrated in Fig. 8 along with the drawings that exemplify objective space of each step. First, starting with the design of experiments, ideally, the samples are well distributed on the objective space. Then, after evaluating this initial sampling plan, corresponding surrogate models can be constructed using Kriging. At this point, it is possible to find the non-dominated subset of from the evaluated objective values, forming an initial Pareto front set. Using Kriging, multi-objective infill criteria, which will be explained in this chapter, can be maximized to find an optimal point from the design space. Although, it is ideal if this point turns out to be non-dominated and become part of the Pareto front in objective space, if not, this point can contribute as an infill point in training surrogate models for the next iteration.

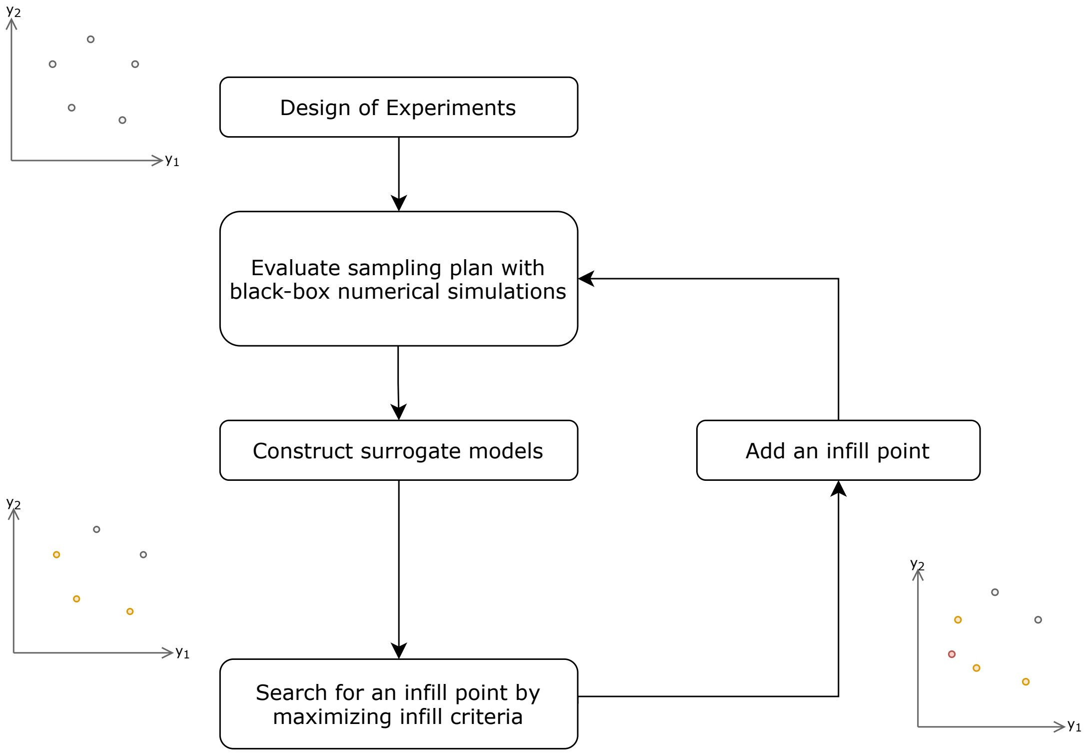


Figure 8: Multi-Objective Bayesian optimization framework

This chapter begins with a description on the idea behind the multi-objective infill criteria formulation that is based on the hypervolume improvement as well as the Kriging model in Section 4.1. Next, two different methods that are used for computing the multi-objective infill criteria are described in 4.2. Finally, in 4.3, the practical implementation of these methods is described.

4.1 Infill criteria

Expected Hypervolume Improvement (EHVI) is, as its name suggests, the expected value of the hypervolume improvement function defined in Section 3.2.2. Its general expression is given in Equation (33) for a given Pareto front set $\mathcal{P}$, statistical quantities $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ and the reference point $\boldsymbol{r}$.

$$EHVI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}.\boldsymbol{r}) = \int_{\mathbb{R}^m} HVI(\mathcal{P}, \boldsymbol{y}, \boldsymbol{r}) \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{y}) d\boldsymbol{y} \tag{33}$$

where ξ denotes the independent joint *probability density function* (PDF) of the standard normal distribution. Its mean values, $\boldsymbol{\mu}$, and the standard deviation values, $\boldsymbol{\sigma}$, of the predictive distributions are obtained from the Gaussian processes by means of maximum likelihood estimation as well as the estimated error in the prediction.

Since the hypervolume improvement function is inserted to compute its expectation, the reference point, remains as the only user defined value. Then, how do we choose this value? or more importantly, does the choice matter?

According to Yang [30], although the reference point $\boldsymbol{r}$ should be chosen such that it is dominated by all elements of a Pareto front approximation set during the optimization process to get the extreme non-dominated points, there is no specific requirement when setting the reference point in practice.

From the perspective of the infill criteria at hand, the reference point should not have a large influence on the optimization process once its larger enough, e.g. larger than the nadir point of a given Pareto front set. This is mainly due to the fact that the hypervolume is not used as a direct metric, instead, the hypervolume improvement is used as a more direct metric. Furthermore, since we are using an infill criteria that represents the expected value for searching in the non-dominated space, this shifts the focus of the optimization process more towards the non-dominated space than the dominated subspace where the hypervolume would belong to. To safely converge to the full optimal Pareto front, however, it is suggested to use a large enough reference value in order to avoid the exclusion of potentially extreme Pareto set points, particularly in the beginning of the optimization process.

Fig. 9 shows the illustration of the expected hypervolume improvement in the objective space $\mathbb{R}^2$. Here, the hypervolume is the light gray area which is the dominated subspace of $\mathcal{P} = \{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \boldsymbol{y}^{(3)}\}$ bounded by the reference $\boldsymbol{r}$. Then, for the bivariate Gaussian distribution with the statistical quantities of $\boldsymbol{\mu} = (2, 1.5)^\top$ and $\boldsymbol{\sigma} = (0.7, 0.6)^\top$, the joint PDF is plotted in 3D. During the optimization process, the optimizer will search for a point from the non-dominated space of a given Pareto set where it maximizes the expected value of the hypervolume improvement using the statistical quantification provided by the Gaussian Processes. While any global search method can be used for

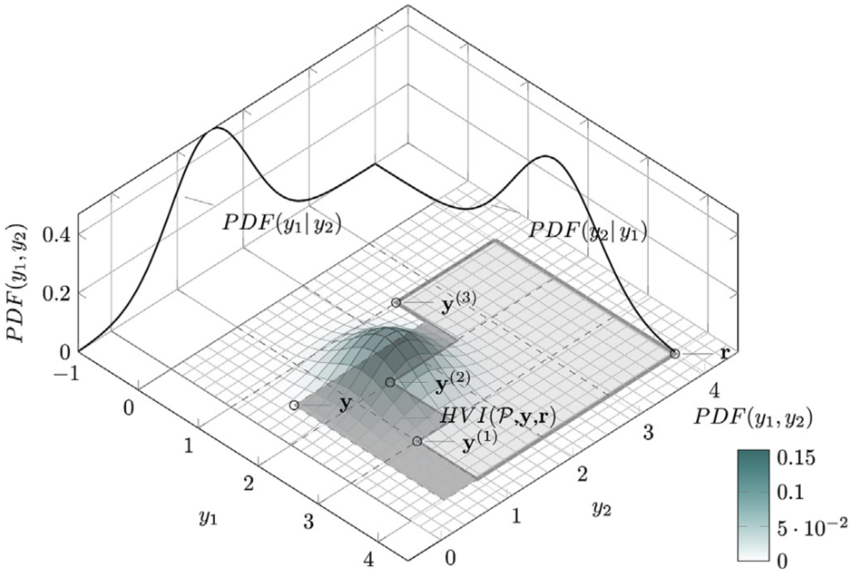


Figure 9: Visualization of the expected hypervolume improvement in $\mathbb{R}^2$. From [29]

this maximization, genetic algorithm or simulated annealing are well-known for their capability [1].

4.2 Method

With the general expression for EHVI in hand, the next step is to actually compute it. Even though Equation (33) suggests that this computation involves multivariate integrals in the objective space $\mathbb{R}^m$, fortunately, it is possible to take advantage of several properties that this equation has. We can go back to Equation (33) for a moment to briefly examine the terms involved. First of all, its integrand expresses the probabilistic hypervolume improvement. Remember from Section 3.2.2 that the main interest of the hypervolume improvement lies, by its definition, in the non-dominated space. This means that there is nothing to integrate over within the dominated subspace which suggests that with the given Pareto front, the domain of the integration can be reduced to the non-dominated space in $\mathbb{R}^m$.

In this thesis, two different approaches are taken to compute the EHVI. The first method is based on the efficient 2D decomposition method by Emmerich et al. [35] with modifications, which are explained in the corresponding chapter. Another method is WFG-based decomposition method by Zhao et al. [20]. While both methods take slightly different approaches in exploiting the geometrical information of the Pareto front to efficiently decompose the integral bounds, the ultimate goal of both methods is to compute the EHVI with the least possible computational effort, i.e. to reduce the number of integration as much as possible. Moreover, both methods compute the exact EHVI without approximations. Some of the key differences between the two methods are as follows.

Firstly, although both of the schemes use decomposition of the objective space to improve efficiency, for the efficient 2D method, the decomposition is done on the non-dominated space, whereas the decomposition is done on the dominated subspace for the WFG-based method.

Secondly, as its name implies, the efficient 2D method is strictly for the bi-objective optimization such that the general equation for EHVI is decomposed in a way it can exclude all the unnecessary computations in $\mathbb{R}^2$. However, the general idea of this method could be extended to $m = 3$ and above by applying suitable decomposition schemes to its general form. On the other hand, the WFG-based is a more generalized method that is capable of computing over $m > 2$ objectives.

Lastly, in terms of computational complexity, the 2D efficient method is known to be asymptotically optimal, i.e. $\mathcal{O}(n \log n)$ [35]. Although, WFG-based method has the worst-case complexity of $\mathcal{O}(m \cdot 2^n)$, it has shown a better average performance, and for $m > 3$, it is considered the state-of-art method by being 10^2 faster than the corresponding existing algorithms [20].

A useful function that enables us to express the EHVI function in a closed form is used throughout the chapter, and is defined as.

Definition 4.1 (Ψ function)**.** Let the function $\phi(s) = \frac{1}{\sqrt{2\pi}} \exp(\frac{-s^2}{2}), s \in \mathbb{R}$ and $\Phi(s) = \frac{1}{2}\left(1 + \text{erf}(\frac{s}{\sqrt{2}})\right)$ be the *probability density function* (PDF) and the *cumulative probability function* (CDF) of the standard normal distribution respectively. Then, the Gaussian distribution with mean μ and variance σ has a general density function of $\phi_{\mu,\sigma}(s) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{1}{2}\left(\frac{s-\mu}{\sigma}\right)^2)$ and the cumulative distribution function of $\Phi_{\mu,\sigma}(s) = \frac{1}{2}\left(1 + \text{erf}(\frac{s-\mu}{\sigma\sqrt{2}})\right)$ for $s \in \mathbb{R}$. Then, the function Ψ is defined as follow:

$$\Psi(a, b, \mu, \sigma) := \int_{-\infty}^{b} (a - z)\frac{1}{\sigma}\phi\left(\frac{z - \mu}{\sigma}\right) dz \tag{34}$$

A useful relation that is employed throughout this chapter which expresses Equation (34) in a closed-form is given by [26]. This relation, which enables the exact computation of Equation (34) is written as:

$$\int_{-\infty}^{b} (a - z)\frac{1}{\sigma}\phi\left(\frac{z - \mu}{\sigma}\right) dz = \sigma\phi\left(\frac{b - \mu}{\sigma}\right) + (a - \mu)\Phi\left(\frac{b - \mu}{\sigma}\right) \tag{35}$$

4.2.1 Efficient 2D method

In the following, the efficient 2D method, based on [30] and [35], is explained in detail. The modification from this basic method is explained accordingly. Some of the terminology are differ from the references to be consistent with the definitions given in the previous section as well as the WFG-based method.

The first step of the efficient 2D method is to partition the non-dominated space as $n+1$ disjoint rectangular stripes $S_1, \ldots, S_{n+1}$ for a given Pareto front set $\mathcal{P} = \{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(n)}\}$ as in Fig. 10, where n refers to the number of elements in a given Pareto front approximation set. The stripes can be formally defined by adding the two sentinels $\boldsymbol{y}^{(\mathbf{0})} = (r_1, -\infty)$ and $\boldsymbol{y}^{(\boldsymbol{n}+\mathbf{1})} = (-\infty, r_2)$ to the set $\mathcal{P}$. From now on, this vector that defines the strip is denoted as $\boldsymbol{y}$. Then, the strip i can be expressed for $\boldsymbol{y} = (y_1, y_2)^\top$:

$$S_i = \left(\begin{pmatrix} y_1^{(i)} \\ -\infty \end{pmatrix}, \begin{pmatrix} y_1^{(i-1)} \\ y_2^{(i)} \end{pmatrix} \right), \quad i = 1, \ldots, n+1 \tag{36}$$

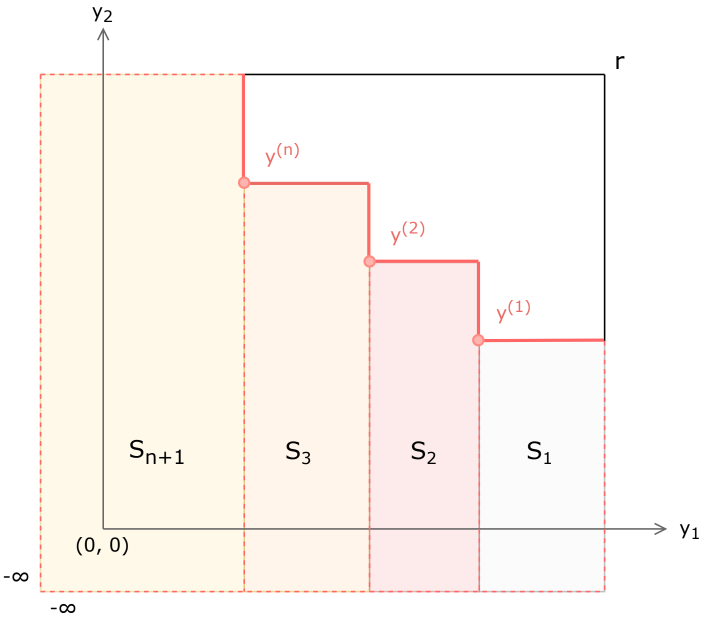


Figure 10: Illustration of the partitioned strips S_i for $i = 1, \ldots, n+1$ defined by the given non-dominated Pareto set $\mathcal{P} = \left\{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \boldsymbol{y}^{(3)}\right\}$. The dashed lines indicate the final integral bounds of the decomposition scheme. Adapted from [35].

The function Δ is defined as:

Definition 4.2 (Δ function)**.** For a given vector of objective function values $\boldsymbol{y} \in \mathcal{R}^m$, $\Delta(\boldsymbol{y}, \mathcal{P}, \boldsymbol{r})$ is defined as the subset of the vectors in $\mathcal{R}^m$ which are exclusively dominated by a vector $\boldsymbol{y}$ and not by elements in $\mathcal{P}$ and that dominate the reference point.

$$\Delta(\boldsymbol{y}, \mathcal{P}, \boldsymbol{r}) = \lambda_m\{\boldsymbol{z} \in \mathbb{R}^m \mid \boldsymbol{y} \prec \boldsymbol{z} \wedge \boldsymbol{z} \prec \boldsymbol{r} \wedge \nexists \boldsymbol{q} \in \mathcal{P} : \boldsymbol{q} \prec \boldsymbol{z}\} \tag{37}$$

For simplicity, this is noted as $\Delta(\boldsymbol{y})$ in the following.

The Δ function of Equation (37) essentially expresses the hypervolume improvement that strictly improves the hypervolume. More precisely, by using the previously defined disjoint stripes, the positive hypervolume improvement in the non-dominated space can be expressed:

$$HVI_+(\mathcal{P}, \boldsymbol{y}, \boldsymbol{r}) = \sum_{i=1}^{n+1} \lambda_m[S_i \cap \Delta(\boldsymbol{y})] \tag{38}$$

With this, EHVI can be re-defined in such a way that it limits the search space to the non-dominated space. This is done by substituting Equation (38) into Equation (33).

$$\begin{aligned} EHVI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}.\boldsymbol{r}) &= \int_{\mathbb{R}^m} HVI_+(\mathcal{P}, \boldsymbol{y}, \boldsymbol{r}) \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{y}) d\boldsymbol{y} \\ &= \int_{y_1=-\infty}^{\infty} \cdots \int_{y_m=-\infty}^{\infty} \sum_{i=1}^{n+1} \lambda_m[S_i \cap \Delta(\boldsymbol{y})] \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{y}) d\boldsymbol{y} \end{aligned} \tag{39}$$

For the bi-objective optimization, this reduces to,

$$EHVI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}.\boldsymbol{r}) = \int_{y_1=-\infty}^{\infty} \int_{y_2=-\infty}^{\infty} \sum_{i=1}^{n+1} \lambda_2[S_i \cap \Delta(\boldsymbol{y})] \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{y}) d\boldsymbol{y} \tag{40}$$

Then, this expression can be simplified further by using the fact that $S_i \cap \Delta(\boldsymbol{y})$ is non-empty if and only if a vector $\boldsymbol{y}$ dominates the non-dominated space with respect to $\boldsymbol{y}_i$ and $\boldsymbol{y}_{i-1}$.

$$EHVI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}.\boldsymbol{r}) = \sum_{i=1}^{n+1} \int_{y_1=-\infty}^{y_1^{(i-1)}} \int_{y_2=-\infty}^{y_2^{(i)}} \lambda_2[S_i \cap \Delta(\boldsymbol{y})] \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{y}) d\boldsymbol{y} \tag{41}$$

The summation is taken out in Equation (41) as the integration is a linear mapping.

Having a closed-form expression of EHVI, the most natural way is to express Equation (41) using Equation (35). To do this, the integral domain of Equation (41) is decomposed to efficiently decompose the integration bounds as,

$$\begin{aligned} EHVI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}.\boldsymbol{r}) = &\sum_{i=1}^{n} \int_{y_1=-\infty}^{y_1^{(i)}} \int_{y_2=-\infty}^{y_2^{(i)}} \lambda_2[S_i \cap \Delta(\boldsymbol{y})] \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{y}) d\boldsymbol{y} \\ &+ \sum_{i=1}^{n} \int_{y_1=y^{(i)}}^{y_1^{(i-1)}} \int_{y_2=-\infty}^{y_2^{(i)}} \lambda_2[S_i \cap \Delta(\boldsymbol{y})] \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{y}) d\boldsymbol{y} \\ &+ \int_{y_1=-\infty}^{y_1^{(n)}} \int_{y_2=-\infty}^{y_2^{(n+1)}} \lambda_2[S_{n+1} \cap \Delta(\boldsymbol{y})] \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{y}) d\boldsymbol{y} \end{aligned} \tag{42}$$

Note that the equation is decomposed in a different way as in the original method to make it compatible with the defined bounds of the strips, which will become more clear after the full derivation of the equation. As a consequence, the computation of the last strip is excluded from the decomposition, and the integral term of the last strip is added as an additional term.

Now, each term of Equation (42) can be written without an integral in a closed form using the Ψ function of Equation (34) and the relation Equation (35).

Evaluating the first term of Equation (42):

$$\begin{aligned}
&\sum_{i=1}^{n} \int_{y_1=-\infty}^{y_1^{(i)}} \int_{y_2=-\infty}^{y_2^{(i)}} \lambda_2[S_i \cap \Delta(\boldsymbol{y})] \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{y}) d\boldsymbol{y} \\
&\qquad = \sum_{i=1}^{n} \int_{y_1=-\infty}^{y_1^{(i)}} (y_1^{(i-1)} - y_1^{(i)}) \cdot \xi_{\sigma_1,\mu_1}(y_1) dy_1 \int_{y_2=-\infty}^{y_2^{(i)}} (y_2^{(i)} - y_2) \cdot \xi_{\mu_2,\sigma_2}(y_2) dy_2 \\
&\qquad = \sum_{i=1}^{n} (y_1^{(i-1)} - y_1^{(i)}) \int_{y_1=-\infty}^{y_1^{(i)}} \xi_{\sigma_1,\mu_1}(y_1) dy_1 \int_{y_2=-\infty}^{y_2^{(i)}} (y_2^{(i)} - y_2) \cdot \xi_{\mu_2,\sigma_2}(y_2) dy_2 \\
&\qquad = \sum_{i=1}^{n} (y_1^{(i-1)} - y_1^{(i)}) \cdot \Phi\left(\frac{y_1^{(i)} - \mu_1}{\sigma_1}\right) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2)
\end{aligned} \tag{43}$$

As it can be seen from the equation, decomposition is done in a way that it exploits that the strips are bounded in the first objective direction by the difference in its objective value which is a constant real value. This can be conveniently employed up to the last strip. However, this does not apply for the last strip where the lower bound of the strip is set to be $-\infty$ in both first and the second objectives. This is also evident from the last form of Equation (43) that it is not solvable with the first objective sentinel value of $-\infty$.

Similarly, evaluating the second term of Equation (42):

$$\begin{aligned}
&\sum_{i=1}^{n} \int_{y_1=y^{(i)}}^{y_1^{(i-1)}} \int_{y_2=-\infty}^{y_2^{(i)}} \lambda_2[S_i \cap \Delta(\boldsymbol{y})] \cdot \xi_{\boldsymbol{\sigma},\boldsymbol{\mu}}(\boldsymbol{y}) d\boldsymbol{y} \\
&\qquad = \sum_{i=1}^{n} \int_{y_1=y^{(i)}}^{y_1^{(i-1)}} (y_1^{(i-1)} - y_1) \cdot \xi_{\sigma_1,\mu_1}(y_1) dy_1 \cdot \int_{y_2=-\infty}^{y_2^{(i)}} (y_2^{(i)} - y_2) \cdot \xi_{\mu_2,\sigma_2}(y_2) dy_2 \\
&\qquad = \sum_{i=1}^{n} \Big(\Psi(y_1^{(i-1)}, y_1^{(i-1)}, \mu_1, \sigma_1) - \Psi(y_1^{(i-1)}, y_1^{(i)}, \mu_1, \sigma_1\Big) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2)
\end{aligned} \tag{44}$$

For the last term, since both of the lower bounds tend to $-\infty$, Ψ function can be used directly as follows:

$$
\begin{aligned}
&\int_{y_1=-\infty}^{y_1^{(n)}} \int_{y_2=-\infty}^{y_2^{(n+1)}} \lambda_2[S_{n+1} \cap \Delta(\boldsymbol{y})] \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{y}) d\boldsymbol{y} \\
&\quad = \int_{y_1=-\infty}^{y_1^{(n)}} (y_1^{(n)} - y_1) \cdot \xi_{\sigma_1,\mu_1}(y_1) dy_1 \cdot \int_{y_2=-\infty}^{y_2^{(n+1)}} (y_2^{(n+1)} - y_2) \cdot \xi_{\mu_2,\sigma_2}(y_2) dy_2 \\
&\quad = \Psi(y_1^{(n)}, y_1^{(n)}, \mu_1, \sigma_1) \cdot \Psi(y_2^{(n+1)}, y_2^{(n+1)}, \mu_2, \sigma_2)
\end{aligned}
\tag{45}
$$

examining the resulting terms of the full Equation (42), it can be seen that the modified term from Equation (45) does not add additional computational complexity onto the original form. It still requires the same number of computation $n+1$ that requests $\mathcal{O}(1)$. Further discussions on computational complexity will be given in the end of the chapter.

Lastly, each term can be fully expressed using the relation given in Equation (35) which is straightforward.

4.2.2 WFG-based decomposition method

The second method uses so called WFG-based decomposition, which is a slightly different approach compared to the efficient 2D method of the previous section. The first step of this method is to re-express the hypervolume improvement in terms of the indicator function. Now, if you recall the expression of the probability of improvement from Section 3.2.2, you can notice that the hypervolume improvement itself can be expressed just by an indicator function if one explicitly gives the conditions for the domination within this function. This has been formally written by Feliot et al. [39]. Following this approach, the hypervolume improvement of $\boldsymbol{y}$ for a given Pareto front $\mathcal{P}$ can be re-written as follows:

$$
HVI(\boldsymbol{y}, \mathcal{P}) = \int_{nd(\mathcal{P})} \mathbb{1}_{\boldsymbol{y} \prec \boldsymbol{y}'} \, d\boldsymbol{y}' \tag{46}
$$

here, $\mathbb{1}(\cdot)$ is the indicator function for domination of $\boldsymbol{y}$. Compared to the probability of improvement, where the indicator function was multiplied to the PDF and integrated to obtain a probability, it is now integrated over the non-dominated space for the given Pareto front. The difference here is that the information about the non-dominated space is given as an integral bounds instead of embedding it into the indicator function as in the probability of improvement. Then the derivation for the EHVI equation continues as:

$$\begin{aligned} EHVI(\boldsymbol{y}, \mathcal{P}) &= \mathbb{E}_{\boldsymbol{y}}\left[\int_{nd(\mathcal{P})} \mathbb{1}_{\boldsymbol{y} \prec \boldsymbol{y'}}\, d\boldsymbol{y'}\right] \\ &= \int_{nd(\mathcal{P})} \mathbb{E}_{\boldsymbol{y}}(\mathbb{1}_{\boldsymbol{y} \prec \boldsymbol{y'}})\, d\boldsymbol{y'} \\ &= \int_{nd(\mathcal{P})} P(\boldsymbol{y} \prec \boldsymbol{y'})\, d\boldsymbol{y'} \end{aligned} \tag{47}$$

One of the distinctive aspects of using this modified equation is that it eases the decomposition of the integral domain of EHVI computation by transferring the problem of computing the EHVI in the non-dominated space into computing the integral of the cumulative probability $P(\boldsymbol{y} \prec \boldsymbol{y'})$ over the entire objective space. In this way, it is possible to select either in the non-dominated space or the dominated space (hypervolume in this case) for the decomposition. That is, the integral can be either computed directly within the non-dominated space or it can be computed by subtracting the integration of the dominated space from the entire domain. Then, the question is: why would one prefer to compute the integral over the dominated space than the non-dominated space? One rather practical reason is that if we compute the over the dominated space, we can utilize many of the already existing efficient algorithms for computing the hypervolume indicator.

In fact, two different algorithms, the grid-based algorithm that decomposes the domain into m-dimensional disjoint boxes by Hupkens et al. [22] and the WFG-based algorithm that decomposes the dominated space - hypervolume - by While et al. [32] were compared by Zhao et al. in [20]. According to Zhao et al., the grid-based algorithm, despite the fact that it is intuitive and easy to implement, is undesirable as it contains a high number of integrals which increases computational cost and has certain requirements in grid structure.

Similar to the efficient 2D method, the integrand of Equation (47) can also be expressed as:

$$P(\boldsymbol{y} \prec \boldsymbol{y'}) = \prod_{j=1}^{m} \Phi\left(\frac{y'_j - \mu_j}{\sigma_j}\right) \tag{48}$$

where $\Phi(\cdot)$ denotes the cumulative distribution function.

One of the main features of the WFG-based algorithm is the decomposition of the dominated space in high-dimensional "boxes". Formally defining this box to be $\mathcal{B}(\boldsymbol{u}, \boldsymbol{l})$ that spans $(l_1, u_1] \times (l_2, u_2] \times \cdots \times (l_m, u_m]$, the full integral can be discretized for grid-based boxes as in Equation (49). Here, $\delta(\mathcal{B}(\boldsymbol{u}, \boldsymbol{l}))$ indicates the evaluation of Equation (47) in each box bounded by the upper bounds $\boldsymbol{u} = (u_1, u_2, \ldots, u_m)$ and the lower bounds $\boldsymbol{l} = (l_1, l_2, \ldots, l_m)$.

$$\delta(\mathcal{B}(\boldsymbol{u},\boldsymbol{l})) = \int_{\mathcal{B}(\boldsymbol{u},\boldsymbol{l})} \prod_{j=1}^{m} \Phi\left(\frac{y'_j - \mu_j}{\sigma_j}\right) d\boldsymbol{y}' = \prod_{j=1}^{m} \int_{l_j}^{u_j} \Phi\left(\frac{y'_j - \mu_j}{\sigma_j}\right) dy'_j \tag{49}$$

using the Ψ function from Equation (34) and the relation from Equation (35), Equation (49) can be expressed in a closed-form as:

$$\delta(\mathcal{B}(\boldsymbol{u},\boldsymbol{l})) = \prod_{j=1}^{m} \left[\Psi(u_j, u_j, \mu_j, \sigma_j) - \Psi(l_j, l_j, \mu_j, \sigma_j)\right] \tag{50}$$

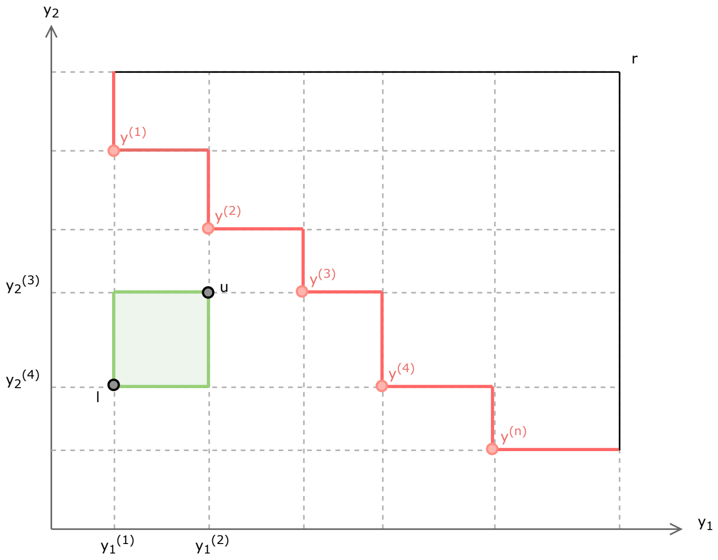


Figure 11: Dashed lines indicate the decomposition of the objective space according to the non-dominated set $\mathcal{P}$. The green box in the non-dominated space represents $\mathcal{B}(\boldsymbol{u},\boldsymbol{l})$. Adapted from [20].

With this capability of decomposing the feasible domain in boxes, any efficient algorithm can be applied to evaluate Equation (50). In this thesis, the WFG algorithm by [32] is adapted, which is known to be the state-of-the-art algorithm for the exact calculation of the hypervolume [42].

The WFG algorithm is developed originally as an efficient algorithm for the hypervolume computation. In the original algorithm, the main idea is to compute the hypervolume by summation of hypervolume improvements ("exclusive hypervolumes") as shown in Equation (51), which was first introduced by [36].

$$HV(\mathcal{P}) = \sum_{i=1}^{n} HVI(\boldsymbol{y}_i, \{\boldsymbol{y}_{i+1}, \dots, \boldsymbol{y}_n\}) \tag{51}$$

In this equation, for the given non-dominated set $\mathcal{P} = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_n\}$, its hypervolume is computed as a sum of the hypervolumes of each element $\boldsymbol{y}_i$ with respect to a set $S = \{\boldsymbol{y}_{i+1}, \dots, \boldsymbol{y}_n\}$ for $i = 1, \dots, n$.

Then, to make computation more efficient, two ways were introduced by [32]. Firstly, the concept of so-called limit set is introduced to identify and include only contributing points into the calcuation. In addition, recall from Section 3.2.2 that the hypervolume improvement is expressed as the difference of the hypervolume. Combining these two concepts, the hypervolume improvement in Equation (51) can be computed as the difference in the hypervolume of $\boldsymbol{y}$ and and the hypervolume of a limit set, denoted as S' as follows:

$$HVI(\boldsymbol{y}, S) = HV(\{\boldsymbol{y}\}) - HV(S') \tag{52}$$

here, the limit set is defined as $S' = \{\text{limit}(\boldsymbol{s}, \boldsymbol{y}) \mid \boldsymbol{s} \in S\}$, where

$$\text{limit}(\boldsymbol{s}, \boldsymbol{y}) = (\max(s_1, y_1), \ldots \max(s_m, y_m)) \tag{53}$$

by filtering the original set S to its limiting subset, it essentially simplifies the computation and reduces computational cost by reducing the number of integrals that need to be evaluated. The computation can be simplified further by filtering out the dominated subset of S'. According to While et al. [32], during this process the set looses about 50% to 80% of its elements, which highlights the efficiency in applying this filtering procedure. Then, the computation of Equation (52) can be done recursively by Equation (51). Consequently, this method computes the hypervolume by sequencially summing over the difference in the hypervolume.

Extending this algorithm to compute the EHVI is straightforward. First, Equation (50) is computed over the dominated space following exactly the same principle for computing the hypervolume, and then it can be subtracted from the integral of the entire domain bounded by $\boldsymbol{r}$:

$$EHVI(\boldsymbol{y}, \mathcal{P}) = \delta(\mathcal{B}(\boldsymbol{r}, -\infty)) - \delta(d(\mathcal{P})) \tag{54}$$

where the second term of Equation (54) is defined as:

$$\delta(d(\mathcal{P})) = \int_{d(\mathcal{P})} \prod_{j=1}^{m} \Phi\left(\frac{y'_j - \mu_j}{\sigma_j}\right) d\boldsymbol{y}' \tag{55}$$

Afterwards, Equation (55) can be computed following the exact procedure that was described for computing the hypervolume. Applying the same principles,

$$\delta(d(\mathcal{P})) = \sum_{i=1}^{n} (\delta(d(\boldsymbol{y}_i)) - \delta(d(S'))) \tag{56}$$

The second term inside Equation (56) can be computed recursively. This recursive computation is illustrated in Fig. 12, where the dominated subspace is defined by the Pareto front $\mathcal{P} = \{\boldsymbol{y}_{(1)}, \dots, \boldsymbol{y}_{(n)}\}$ and a reference point $\boldsymbol{r}$. Here, for $i = 2$, the colored area refers to the domain for $\delta(d(\{\boldsymbol{y}^{(2)}\}))$. From the original limit set, $S' = \{\boldsymbol{y}^{(3)}, \boldsymbol{y}^{(4)}, \boldsymbol{y}^{(n)}\}$, the dominated elements, $\boldsymbol{y'}^{(4)}$ and $\boldsymbol{y'}^{(n)}$, are filtered, which means that the recursive algorithm does not even have to be applied further for this case as the exact same computation can be defined by one contributing point.

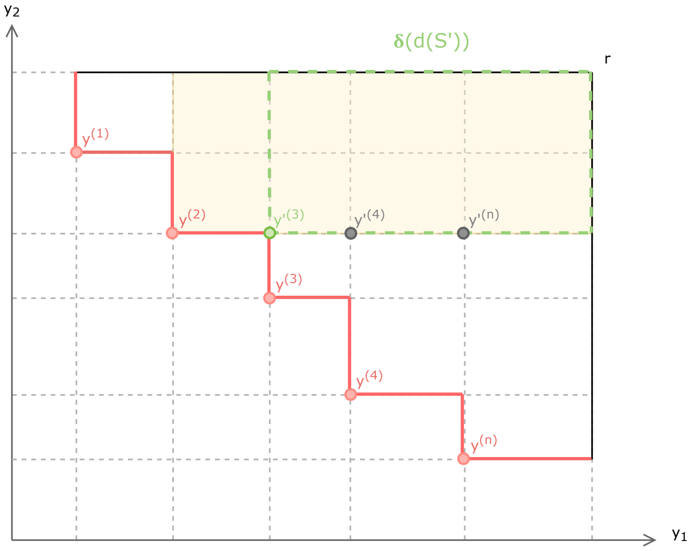


Figure 12: Illustration of an iteration in computing $\delta(d(\mathcal{P}))$ recursively. Colored area in the dominated subspace refers to the domain for $\delta(d(\{\boldsymbol{y}^{(2)}\}))$ where the dashed lines indicate the integral bounds for $\delta(d(S'))$. The elements in the dominated subset of S', $\boldsymbol{y'}^{(4)}$ and $\boldsymbol{y'}^{(n)}$, have been filtered out.

Finally, substituting Equation (56) into Equation (54) gives,

$$EHVI(\boldsymbol{y}, \mathcal{P}) = \delta(\mathcal{B}(\boldsymbol{r}, -\infty)) - \sum_{i=1}^{n} (\delta(d(\boldsymbol{y}_i)) - \delta(d(S'))) \tag{57}$$

The first term, although it is correctly defined in terms of its actual bounds, it can be somewhat misleading as its lower bound tends to $-\infty$ and that Equation (50) does not hold directly. Furthermore, Equation (57) shows that applying the WFG-based recursive algorithm altered the computation by converting the integration problem on the complex domain into a series of summation and subtraction. As discussed previously, the key point of decomposition schemes for EHVI is to reduce the number of computation as much as possible.

Throughout the derivation of the WFG-based recursive algorithm, it has been shown how the majority of the computations could be excluded by defining a limit set and applying filtering procedure. Hence, it is not unpredictable that this algorithm outperforms compared to other algorithms in terms of efficiency, especially in high dimensional problems where $m > 3$.

4.3 Implementation

Having discussed the theoretical background of the methods used for the multi-objective Bayesian optimization framework, in this section, practical implementation details of the framework are explained, which is coded in Python. Throughout the program utilizes various modules that are available within DLR's in-house toolbox SMARTy, namely the initial DOE sampling, surrogate model construction as well as the optimization of the acquisition function.

In Section 4.3.1, the structures of the some of the main algorithms are presented and explained. The implementation is discussed in terms of computational complexity and efficiency in Section 4.3.2. Furthermore, additional adjustments for practicality are described in Section 4.3.3. Finally, computational budget and the convergence criteria are discussed in Section 4.3.4.

4.3.1 Algorithms

Algorithm 1 Multi-Objective Surrogate Based Optimization

```
Input: $m$ Objective functions $f(\boldsymbol{x})$, bounds, size of initial DOE $n$, computational budget
Output: Pareto front $\mathcal{P}$
$\boldsymbol{x} \leftarrow$ Compute initial DOE using LHS or Sobol $^*$
$\boldsymbol{y} \leftarrow$ Evaluate initial DOE by calling objective functions $f(\boldsymbol{x})$
$S \leftarrow ((\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n))$
while computational budget $> 0$ do
    Train $m$ Gaussian Processes (GPs) models with $S$ $^*$
    $\mathcal{P} \leftarrow$ ParetoFilter($\boldsymbol{y}$)                ▷ Sort non-dominated subset of $\boldsymbol{y}$
    if Acquisition Function is efficient 2D EHVI then
        $S_i \leftarrow$ Set up the partition bounds
    end if
    $\boldsymbol{x}_{\text{infill}} \leftarrow$ Use an optimizer$^*$ to find an infill point by a chosen EHVI algorithm
    $\boldsymbol{y} \leftarrow$ Evaluate an infill point by calling objective functions $f(\boldsymbol{x}_{\text{infill}})$
    Update $S \leftarrow S \cup \{(\boldsymbol{x}_{\text{infill}}, f(\boldsymbol{x}_{\text{infill}})\}$
end while
$\mathcal{P} \leftarrow$ ParetoFilter($\boldsymbol{y}$)
return $\mathcal{P}$                ▷ (*) : DLR's SMARTy modules
```

Algorithm 1 shows the process of the complete multi-objective surrogate based optimization framework. The program starts with constructing the initial sampling plan (DoE) unless provided by a user. The initial sampling is done by either Latin Hypercube Sampling or Sobol which are both available in SMARTy. Then, the initial sampling plan is evaluated by calling the objective functions provided. Provided optimization bounds are normalized such that the decision variables have a scale of $(\boldsymbol{x}_u, \boldsymbol{x}_l) = [0, 1]^n$ throughout the optimization. According to [1], this scaling of decision vectors can prevent potential scaling issues in optimization.

Then, it enters the main optimization or the infill loop with the given computational budget. The computational budget and the termination criteria of the program will be discussed in detail. The first task in the optimization loop is to build the surrogate model, specifically, Gaussian processes models are constructed using the module in SMARTy with the initial sampling plan. Next, the first set of the Pareto front is computed with the function that filters out the dominated subset from the given set. A user can choose either the efficient 2D EHVI or the WFG-based EHVI algorithm for the acquisition function. An additional pre-processing is done to define the partition bounds in case the efficient 2D EHVI is used. Then, the infill optimization can be processed with a single-objective optimizer. In this thesis, the differential evolution method from the SMARTy module is selected. For this, the corresponding EHVI function along with the scaled bounds and estimations from the Kriging model are used. Then, an optimizer finds a promising point by maximizing given EHVI function. When the optimizer returns the optimal design point $\boldsymbol{x} \in \mathbb{R}^n$, the corresponding objective value is computed, which is then appended to the data set for training the surrogate models. This process is repeated until the computational budget is exceeded. In the end, it returns the computed Pareto front.

Algorithm 2 Efficient 2D EHVI algorithm

```
Input: $\boldsymbol{x}$, $m$ Gaussian Processes (GPs), partitioned bounds set $\boldsymbol{y}$
Output: Expected Hypervolume Improvement (EHVI)
$\boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{\sigma}(\boldsymbol{x}) \leftarrow$ obtain from the GPs
EHVI $\leftarrow 0$ initialize
for all elements in $\boldsymbol{y} - 1$ do
    a $\leftarrow \Big(\Psi(y_1^{(i-1)}, y_1^{(i-1)}, \mu_1, \sigma_1) - \Psi(y_1^{(i-1)}, y_1^{(i)}, \mu_1, \sigma_1))\Big) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2)$
    b $\leftarrow (y_1^{(i-1)} - y_1^{(i)}) \cdot \Phi\left(\frac{y_1^{(i)} - \mu_1}{\sigma_1}\right) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2)$
    EHVI $\leftarrow$ adds $a$ and $b$
end for
EHVI $\leftarrow$ adds the last fraction of the integral
return EHVI
```

Algorithm 2 shows the procedure for computing the EHVI using the efficient 2D method. Apart from the modifications that were explained in the previous section, the implementation of this method is straightforward. As it was shown during the derivation of the equations for this method, all unnecessary computations are eliminated such that the final form of the equation is expressed specifically for 2D. In fact, this was evident since the beginning of the formulation when the strips S_i were being defined in terms of the integral bounds in $\mathbb{R}^m$.

The algorithm for the WFG-based EHVI computation is shown in Algorithm 3. Unlike the 2D method, this algorithm takes any $m > 2$. To point out a few of the differences in this algorithm compared to the 2D method, first, it explicitly requires the reference point for the computation. As explained in the previous section, the main computational effort is within the hypervolume of the dominated subspace which is computed through the function DOMINATEDEHVI. It can be seen that this function is called recursively when the number of elements in the input $\mathcal{F}$ exceeds 1. For other cases, it can be computed directly using function DELTA. Finally, this is subtracted from $\delta(\mathcal{B}(\boldsymbol{r}, -\infty))$ to get the EHVI value.

Algorithm 3 WFG-based EHVI algorithm

```
Input: $\boldsymbol{x}$, $m$ Gaussian Processes (GPs), Pareto front $\mathcal{P}$, reference point $\boldsymbol{r}$
Output: Expected hypervolume improvement (EHVI)
$\boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{\sigma}(\boldsymbol{x}) \leftarrow$ obtain from the GPs
$\delta(d(\mathcal{P})) \leftarrow$ DOMINATEDEHVI($\mathcal{P}$)
$\delta(\mathcal{B}(\boldsymbol{r}, -\infty)) \leftarrow$ product of $\Psi(r_j, \mu_j, \sigma_j)$
return EHVI $\leftarrow \delta(\mathcal{B}(\boldsymbol{r}, -\infty)) - \delta(d(\mathcal{P}))$
function DOMINATEDEHVI($\mathcal{F}$)
    for all elements in $\mathcal{F}$ do
        if number of elements in $\mathcal{F}$ > 1 then
            for $j$ in all elements do
                $\delta(d(\mathcal{F})) \leftarrow$ sum over PORTIONEHVI($\mathcal{F}$,$j$)
            end for
        else
            $\delta(d(\mathcal{F})) \leftarrow$ compute DELTA($\mathcal{F}$) directly
        end if
    end for
    return $\delta(d(\mathcal{F}))$
end function
function PORTIONEHVI($\mathcal{M}$, $j$)
    $r \leftarrow$ Compute DELTA($j$ th element of $\mathcal{M}$) $-$ DOMINATEDEHVI($\mathcal{M}'$)
                                        ▷ $\mathcal{M}'$ refers to the dominated filtered limit subset of $\mathcal{M}$
    return $r$
end function
function DELTA($\boldsymbol{q}$)
    return product of $\Psi(r_j, \mu_j, \sigma_j) - \Psi(\boldsymbol{q}, \mu_j, \sigma_j)$ in each objective dimension
end function
```

4.3.2 Computational complexity and efficiency

Computational complexity and efficiency of computing the EHVI needs to be addressed as apart from the evaluation of the black functions, this is the computationally most expensive part of the code. This is computationally expensive as it is called many times during the optimization of the EHVI while searching for an infill point.

According to [35], the efficient 2D algorithm is computed with an asymptotically optimal computational complexity of $\mathcal{O}(n \log n)$, where n denotes the size of the non-dominated set. On the other hand, the WFG-based algorithm has a complexity that is proportional to the number of decomposed boxes. Although, this method has a worst case computational complexity of $\mathcal{O}(m \cdot 2^n)$ when the number of decomposition is $2^n - 1$, it has been demonstrated that the average complexity is much smaller for most real-world simulations [20].

Numerical experiments are conducted by Zhao et al. [20] to compare the computational complexity of the different methods for computing the exact EHVI. For this a randomly generated non-dominated set is created and used for the EHVI computation. From the results for $m = 3$, it has been shown that the WFG-based method outperforms most of the other decomposition methods, e.g. the grid-based method. While, the WFG-based method is shown to have the lowest computational complexity for $m > 3$, for $m \to n = 10$, the computational complexity is bounded by the worst case complexity $\mathcal{O}(2^n)$.

In order to increase the efficiency, sorting of the non-dominated set is suggested by [32] and [35]. For the efficient 2D approach, it is possible to achieve a linear time complexity $\mathcal{O}(n)$ by adding the sorting process for the non-dominated set before computing the EHVI, which means that the computational complexity is bounded from below by $\mathcal{O}(n)$ [35]. For the case of the WFG-based algorithm, it is possible to increase its efficiency by exploiting the features of the functions in the algorithm. From the WFG-based algorithm in Algorithm 3, it shows that the more elements of the limit subset $\mathcal{M}$ are dominated, and thus, filtered, the number of computation within DOMINATEDEHVI can be limited and reduced, which results in faster algorithm. This can be implemented also by sorting the Pareto front set such that they are monotonically increasing in one of the objectives. Although there are more complicated sorting systems that produce even more dominated points, sorting in a single objective is cost efficient yet effective method [32]. This sorting procedure has been implemented within this thesis by sorting the non-dominated set by its 1st objective before optimizing the EHVI. It is expected to increase efficiency for both, the efficient 2D and the WFG-based methods.

Reducing computational complexity and increasing the efficiency of the algorithm is surely one of the particular interests when computing the exact EHVI in the surrogate based solving multi-objective optimization problems, as it affects the running time of the algorithms, especially during the optimization of the EHVI. However, from a more practical point of view, does it still play a significant role in terms of running time? In applications, particularly for aerospace applications, evaluation of one objective value from the black box functions can take one hour to several hours, e.g. CFD, CAA or FEM simulations. Therefore, considering the total running time, increasing the efficiency of the EHVI algorithm has barely any effect. What is more effective in terms of the total running time would be to reduce the number of function evaluations.

4.3.3 Penalty for invalid objective values

The methods and algorithms described up until now work under the assumption that the returned objective values from calling the black-box functions are valid in a sense that the numerical simulations involved have been converged. However, this is not always the case. For instance, in CFD simulations, problems can arise during mesh generation that result in invalid CFD solutions. Sometimes, it is even possible that the numerical simulations crash and do not return anything. In order to treat this issue, penalty values have been introduced to handle all the above-mentioned situations where the returned values of the black-box functions are considered invalid. Thereby, it is possible to train the surrogate models for the valid and promising areas as well as the invalid areas.

When the black-box functions are called, if the evaluated objective values are invalid, e.g. in terms of accuracy etc., it needs to return None type in stead of a float. Then, after constructing the initial design of experiments, the code checks whether all the evaluated objective values are valid; if not, it replaces the invalid results with the penalty value. The penalty value is defined as the worst value in that objective multiplied by the penalty level given by a user.

$$\text{penalty value} = \max(\boldsymbol{y}_i) \times \text{penalty level} \qquad \text{for } i = 1, \dots, m \tag{58}$$

Therefore, during optimization, an optimizer will perceive the area around this penalized point to be infeasible and avoid this region.

Unfortunately, since this is essentially altering the Gaussian processes in an artificial way, although it does help an optimizer to avoid potentially invalid regions in the decision space, it can also hinder an optimizer from approaching promising regions where global minima could possibly be found. Therefore, for each multi-objective optimization problem, careful examination needs to be done to check the effect of the penalty values and penalty level on the solution.

4.3.4 Computational budget and termination criteria

As it is seen from Algorithm 1, the main optimization loop continues until the given computational budget is exceeded or the convergence criteria is met. The most general way to define a computational budget in multi-objective optimization would be in terms of the number of function evaluations, particularly when computationally expensive black-box functions are involved. Thus, the basic computational budget is set to be a prescribed number of infill points, i.e. additional number of the function evaluations after the completion of the initial design of experiments.

Of course, it is possible to implement the performance metrics discussed in the previous chapter as a termination criteria. This is not a problem if one has an idea of what the optimal Pareto Front should be, e.g. shape and whether it should be discrete or continuous etc. For most applications, however, this is not known in advance. Once the analysis is done such that the optimal Pareto front can be estimated to a certain extent, the general performance metrics for the multi-objective optimization could be implemented as a termination criteria, e.g. hypervolume improvement. In this way, for newly encountered multi-objective optimization problems, this is another trade-off that needs to be considered.

One simple and effective way to check the convergence, which is also implemented in this thesis, is to set the minimum number for the non-dominated set. Setting this criteria also needs some knowledge of the optimal Pareto front in advance, as the total execution time could become unexpectedly large when a high number is given for the required minimum number of non-dominated set. For analytical problems, this approach can almost always help, as it can be more difficult to estimate the total number of function evaluations required. For practical applications, this convergence criteria is recommended only if there is no given computational budget in terms of total execution time. In such cases it is still recommended to set a reasonable maximum infill sample budget.

The difficulties involved in setting the convergence criteria of multi-objective optimization mainly come from the fact that there is no absolute metric that can be used without any prior knowledge on the Pareto front. For instance, in single-objective optimization, since the solution is purely quantitative, more direct metrics can be used for ensuring local convergence, e.g. improvement in the objective values or Euclidean distance between successive infill points. Furthermore, for ensuring the exploration in single-objective optimization, Forrester et al. [1] suggested validation techniques, e.g. MSE of successive surrogates. However, for multi-objective optimization, applying these criteria become inefficient as it could result in spending available computational budget to improve the quality of the surrogate around regions that are far from the Pareto front or even invalid. The higher the dimension of the problem, avoiding this waste of budget could become even more challenging.

5 Analytical Problems

Even though the total computational time can be substantially saved by using surrogate models and performing surrogate based optimization, it would still be difficult to conduct as many numerical simulations using the expensive simulations, especially with the time constraints. Therefore, numerical experiments have been completed using analytical functions, first to validate the method that has been implemented, and second to investigate certain aspects of the settings in multi-objective Bayesian optimization, and lastly to compare the results obtained with other existing multi-objective optimization methods, particularly with evolution algorithms and genetic algorithms.

In this way, any information acquired from this analysis would not only be meaningful on its own, but it could also provide useful prior knowledge for solving applied problems that are difficult to solve as the optimal Pareto front is not known. Still, this does not entail that the analytical problems are any less challenging, because the functions involved are designed to be particularly challenging in order to test the various aspects of the multi-objective optimizer.

Considering that in many practical applications, available computational resources are limited by expensive evaluation of black-box functions, the total number of function evaluations in the following tests is kept relatively small in order to create a similar circumstance.

For this, few of the well-known test functions have been selected as test cases. The first two cases are the bi-objective problems that have different number of design variables as well as different shape of the optimal Pareto front, and the last case has 3 objectives. First test case has been carried out with Poloni's test function in Section 5.1. Secondly, a test function with a concave optimal Pareto front, namely the Fonseca-Fleming function, is solved, and its results are presented in Section 5.2. Lastly, the 3D Viennet function is solved, and the relevant results are given and discussed in Section 5.3.

5.1 Poloni's Test Function

The Poloni's test function comprises of two objective functions and two decision variables with a given bound. It has some of the typical properties of many engineering design problems such that it has multi-modal behavior as well as sufficient smoothness [9].

5.1.1 Problem formulation

The Poloni's two-objective test functions and the bounds for the decision variables are given as follows:

$$\text{minimize} \quad \begin{cases} f_1(\boldsymbol{x}) = 1 + (A_1 - B_1(x_1, x_2))^2 + (A_2 - B_2(x_1, x_2))^2 \\ f_2(\boldsymbol{x}) = (x_1 + 3)^2 + (x_2 + 1)^2 \end{cases} \tag{59}$$

$$\begin{aligned} \text{where} \quad & x_i \in [-\pi, \pi] \qquad \text{for } i = 1, 2 \\ & A_1 = 0.5\sin(1) - 2\cos(1) + \sin(2) - 1.5\cos(2) \\ & A_2 = 1.5\sin(1) - \cos(1) + 2\sin(2) - 0.5\cos(2) \\ & B_1(\boldsymbol{x}) = 0.5\sin(x_1) - 2\cos(x_1) + \sin(x_2) - 1.5\cos(x_2) \\ & B_2(\boldsymbol{x}) = 1.5\sin(x_1) - \cos(x_1) + 2\sin(x_2) - 0.5\cos(x_2) \end{aligned}$$

5.1.2 Results

The result of the multi-objective optimization is shown in Fig. 13. The initial sampling is completed by Sobol method with a total of 60 initial sample points. Then the infill budget is set to 40, resulting in the total 100 function evaluations. From the plot, it can be seen that the computed Pareto front aligns well with the known optimal Pareto front. The small gaps in between the solution at $f_1 > 2$ will be filled accordingly with the increased number of function evaluations. Along with its Pareto front, the dominated sampled points are also plotted which are used for Gaussian processes.

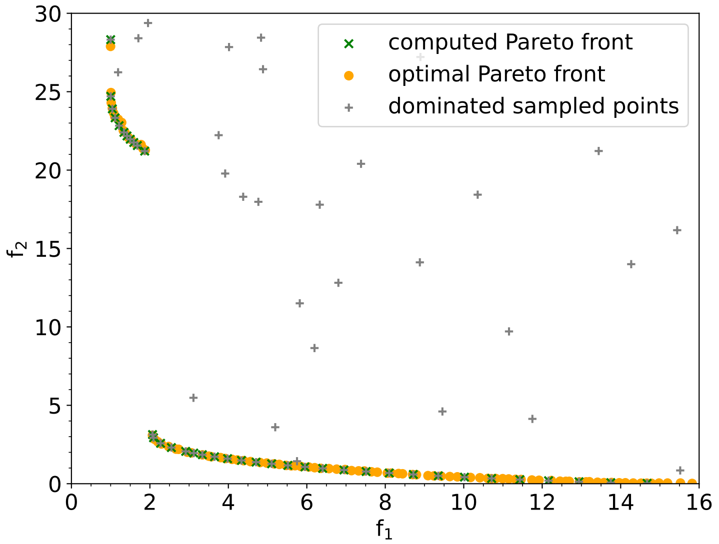


Figure 13: Result of the computed Pareto front with 100 function evaluations

Since Poloni's two-objective function is a popular test case for multi-objective optimization, results obtained using various multi-objective optimization techniques have been reported in many literature. In [37], it is reported that the Pareto front was obtained after 600 function evaluations using a variant of differential evolution method. In [9], the optimization is performed using multi-objective genetic algorithm with 50 individuals evolved for 50 generations, resulting in 2500 function evaluations. This demonstrates that the method taken for this thesis, i.e. surrogate based optimization with EHVI, has a capability of solving the multi-objective optimization in a more efficient way than some of the conventional approaches.

5.1.3 Analysis of the effect of the initial DOE

In order to investigate how the size of the initial DOE affects the accuracy of the initial surrogate model as well as the final Pareto front, optimization is performed for 3 different cases where every case has a different size of the initial DOE and the number of infill points. For fair comparison, the total computational budget, i.e. the number of function evaluation, is kept constant and set to $N = 60$. Therefore, another significance of the following numerical experiments is to investigate the efficient distribution of the computational budget between the initial DOE and the infills to a certain extent. While the LHS method is used for the initial DOE sampling, the efficient 2D method is taken as an acquisition function. And, the reference point is set to $(70, 70)$. The first optimization case is performed with 5 initial DOE samples and 55 infill points. The resulting Pareto front is shown in Fig. 14, where along with the computed Pareto front and the optimal Pareto front known, the initial Pareto front, which refers to the Pareto front of the initial DOE samples, is given.

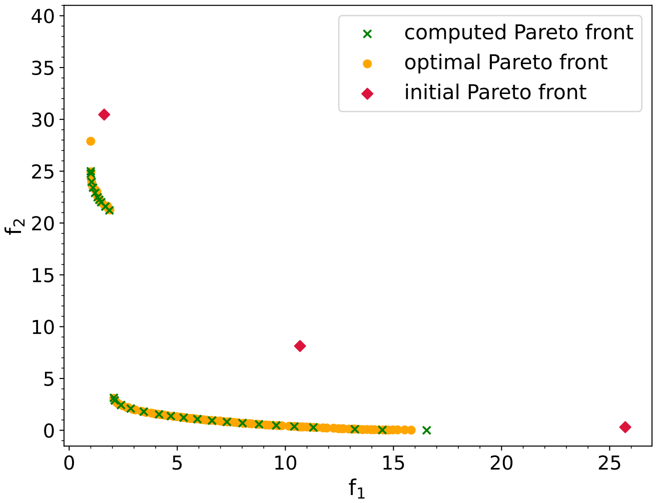


Figure 14: Result of the optimization with the 5 initial DOE and 55 infills

From the plot, it can be seen that although the initial Pareto front is off from the computed or the optimal Pareto front, this is still nicely aligning with them, which implies that the selected initial DOE sampling method was able to cover the objective space. Then, after 55 iterations, it was able to capture the whole trend of the optimal Pareto front.

The second case is performed with 13 initial DOE and 47 infill points, and the resulting Pareto front is given in Fig. 15. Compared to the previous result, although there is no obvious discrepancy within the computed Pareto front, improvement in the initial Pareto front is clearly visible in a sense that the discrepancy between the initial Pareto front and the optimal Pareto front has been reduced for this case. This implies that it is possible to obtain the better initial Pareto front with the increased number of initial DOE.

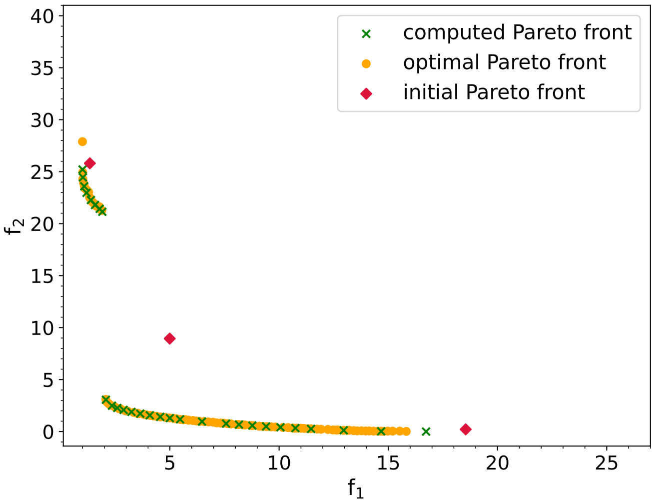


Figure 15: Result of the optimization with the 13 initial DOE and 47 infills

The last case is performed with 20 initial DOE and 40 infills, and the resulting Pareto front is shown in Fig. 16. Here, it was able to nicely obtain the full Pareto front. One distinctive difference, compared to the previous cases, is that it has much larger and widely distributed set of the initial Pareto front. In a way, this is quite obvious and expected, as the increased number of the initial sample should provide a higher chance of better coverage in the objective space. However, strictly comparing the discrepancy, i.e. the distance to the optimal Pareto front, this result does not seem to perform any better compared to the second case, for instance, the three initial Pareto front points from Fig. 15 are much closer to the optimal Pareto front than they are for this case. From the perspective of optimization using the 2D acquisition function, this implies that the results obtained in the previous case are likely to be premature progress due to a lack of samples to construct an accurate enough Gaussian processes. Aside from its distribution of the initial Pareto front, what the reduced initial sampling size would effect is the accuracy of the initial surrogate model. And the key point here is that the initial surrogate model does not need to be an error-free replication of the actual objective function, instead, it needs to be accurate enough for an optimizer to be able to identify the promising regions where the potential solutions reside.

The same optimization cases were performed using the Sobol sampling method as well, and the results can be found in Fig. 17. This results show that discrepancies in the

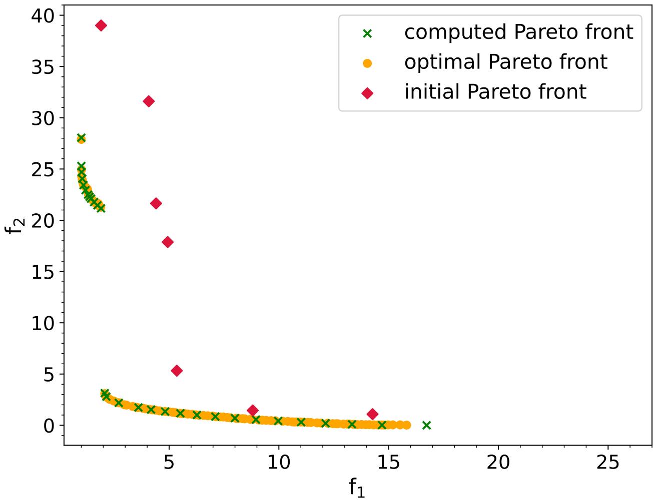


Figure 16: Result of the optimization with the 20 initial DOE and 40 infills

initial Pareto front for three cases are not as drastic as the ones obtained with the LHS. Instead, it shows a consistent trend of the gradual convergence of the initial Pareto front towards the optimal Pareto front, which suggests the possibility that the Gaussian processes have been less affected by the size of the initial DOE that are generated using the Sobol sequence.

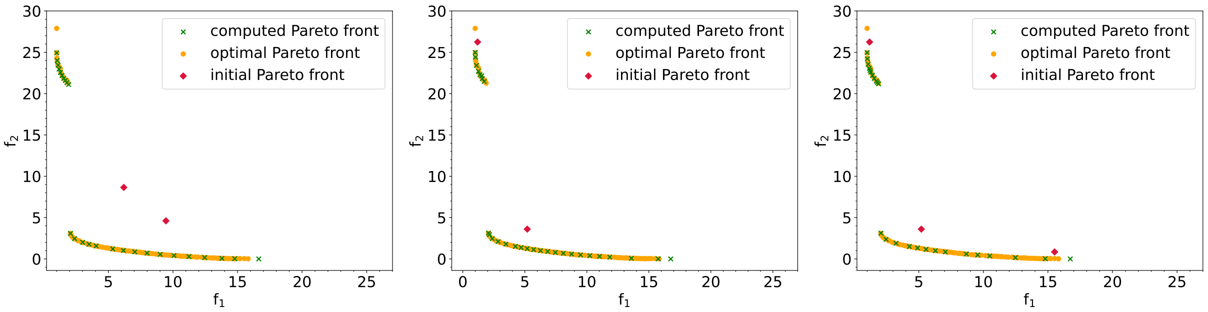


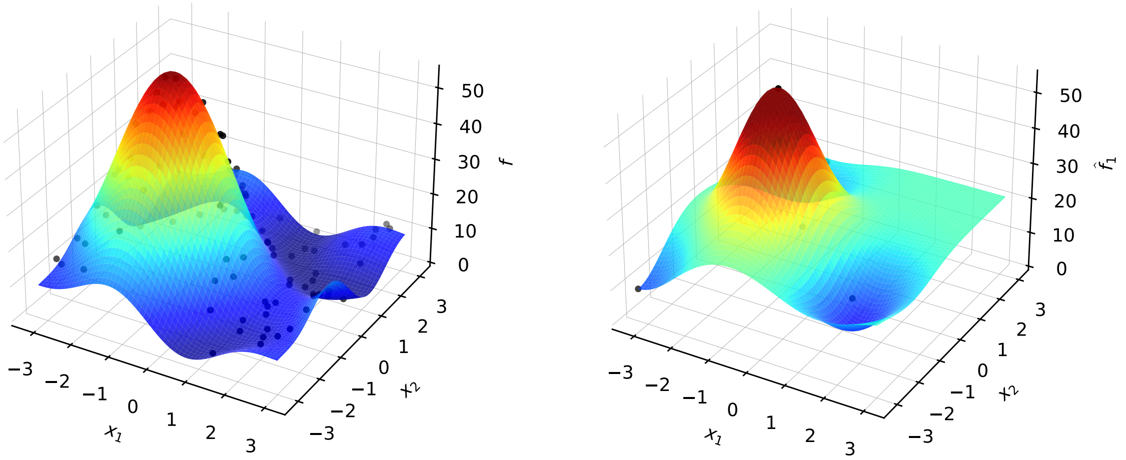
Figure 17: Results obtained using Sobol sampling using 5 (left), 13 (middle) and 20 (right) initial DOE

In order to further investigate the effect of the initial sampling size on the accuracy of the initial surrogate model resulting from Gaussian processes, the initial surrogate models of five different number of initial DOE were analyzed, which are 5, 8, 13, 16 and 20. Then, the corresponding errors were computed based on the 100 randomly generated samples with a uniform distribution, and the initial sampling is completed by the Sobol method. One rather an obvious prediction one can make is that with the growing number of the initial samples used for constructing Gaussian processes, the resulting surrogate model should become more accurate. And, this can be seen clearly in Fig. 18 and Fig. 19 each of which shows the initial surrogate model $\widehat{f_1}$ and $\widehat{f_2}$ constructed with a different size of the initial DOE. From both figures, the first plot of each figure Fig. 18a and Fig. 19a are the plots of the actual objective functions, f_1 and f_2 which are treated as black-box functions, and the black dots plotted over the surface indicate the location of the samples that were used for the error computation. The dots plotted over the surface of all the other plots in the two figures indicate the location of the sample points of the initial DOE which were used to construct the initial surrogate model.

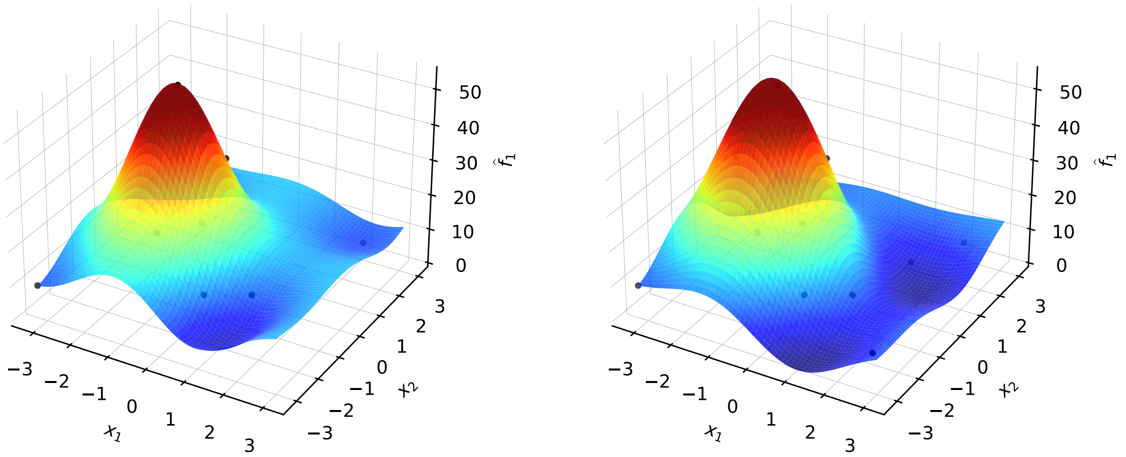
First examining Fig. 18, there is a trend of $\widehat{f_1}$ becoming more accurate with the increasing size of the initial DOE, denoted as N. This is particularly apparent in the first three cases: the flat regions where the values are much higher than the objective function seems to be lowered just by adding three more samples to the initial sampling, and at the same time, the surrogate model starts to capture the regions of the local minimum. Then, from $N = 13$, it starts to identify more accurate landscape around these local minimum. By $N = 16$, the surrogate model has identified more or less the entire local minimum including the regions around $(x_1, x_2) = [0, 2]$ which has not been identified by the one with $N = 13$. Finally with $N = 20$, it was able to identify the regions with local minimum as well as more accurate landscapes around these regions. Before going into a deeper analysis, a similar trend can be observed for the results of the second objective function in Fig. 19, here, the Gaussian processes were able to properly capture a monotonically increasing landscape function due to simplicity as it only has one optimum region. Thus, for this function, the Gaussian processes were able to identify its optimum region already with $N = 5$, and it was able to capture the full curvature as soon as it had $N = 8$.

This result, further explains the obtained Pareto front results for $N = 5$, 13 and 20. First of all, premature convergence that occurred for the smaller N, could be due to the inaccuracy of its initial surrogate model. Of course, there are other factors that might potentially have affected the obtained result, e.g. error estimations by Gaussian processes, besides that a well balanced search between exploration and exploitation is generally expected using the EHVI. However, it is possible that its lack of knowledge of the design space led to deterioration in exploration part compared to the case where it started an optimization with a better knowledge of the design space. This suggests that enough budget should be allocated for the initial DOE sampling in order to ensure that the balance between exploration and exploitation is achieved within the limited number of iterations.

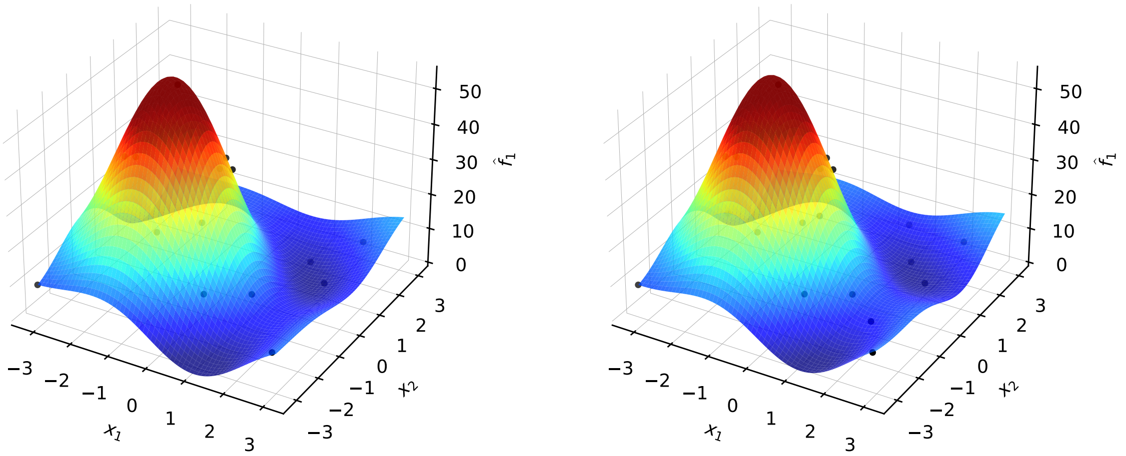
For the direct comparison to the three optimization cases done with the LHS sampling, the initial sampling size was increased further to $N = 30$. From the result in Fig. 20 (left), the similar result is shown as the previous result such that it not only has more elements in the initial Pareto front, but also its distribution better aligns with the optimal Pareto front. Eventually, it was able to better capture the full Pareto front just as the result obtained using the LHS sampling. Furthermore, the total number of function evaluations was increased to 100. And, its result, in Fig. 20 (right), is shown to be



(a) objective function f_1 (b) $N = 5$

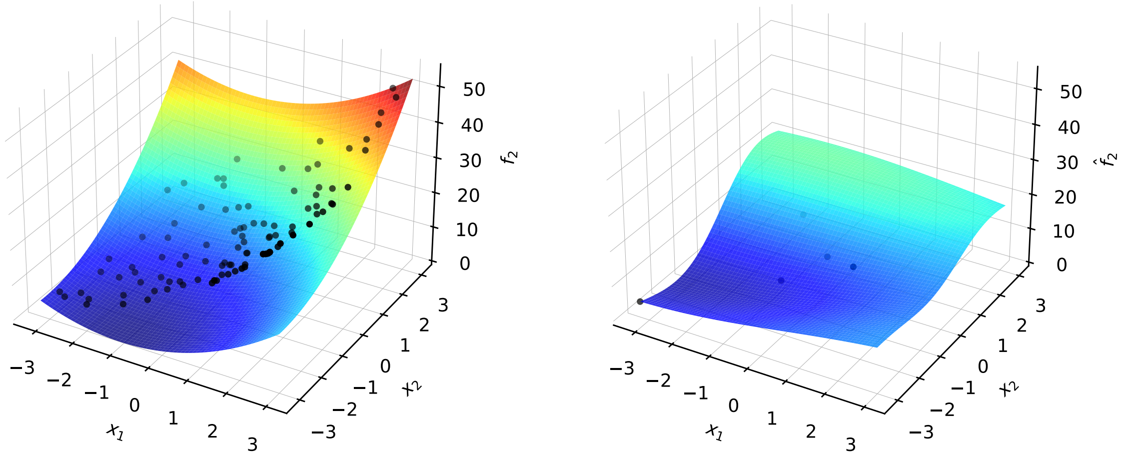


(c) $N = 8$ (d) $N = 13$



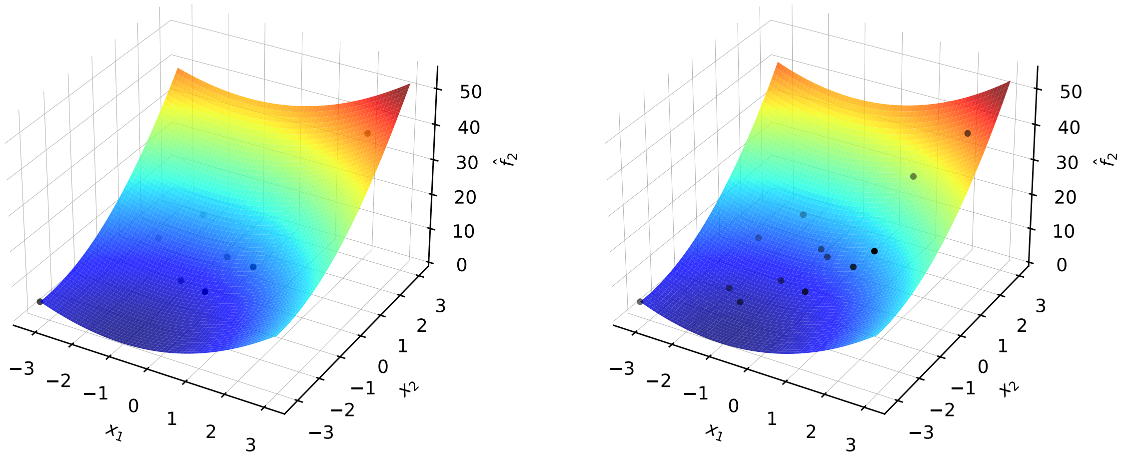
(e) $N = 16$ (f) $N = 20$

Figure 18: objective function f_1 and the Gaussian processes models $\widehat{f_1}$ with different size of the initial DOE (N)



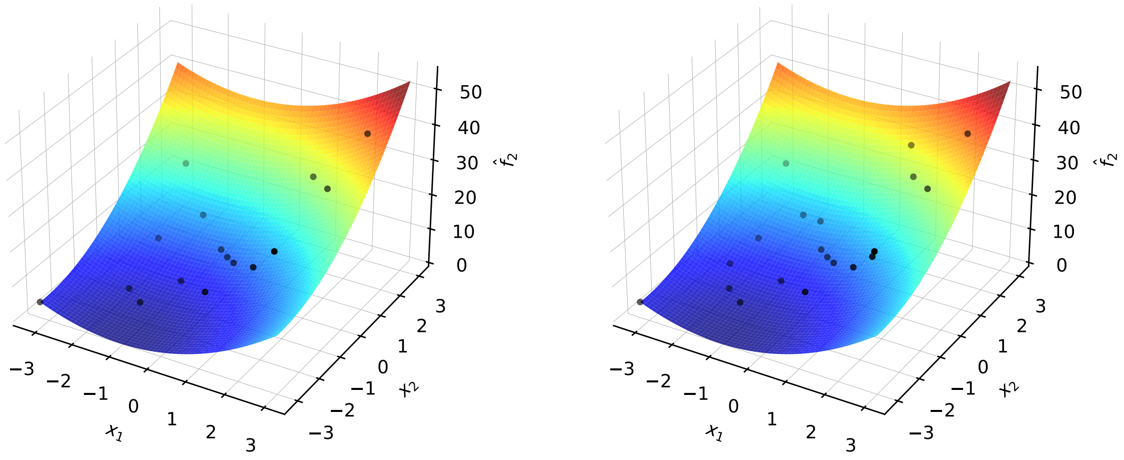
(a) objective function f_2

(b) $N = 5$



(c) $N = 8$

(d) $N = 13$



(e) $N = 16$

(f) $N = 20$

Figure 19: objective function f_2 and the Gaussian processes models $\widehat{f_2}$ with different size of the initial DOE (N)

improved further, and started to fill in the intermediate Pareto front points by exploitation.

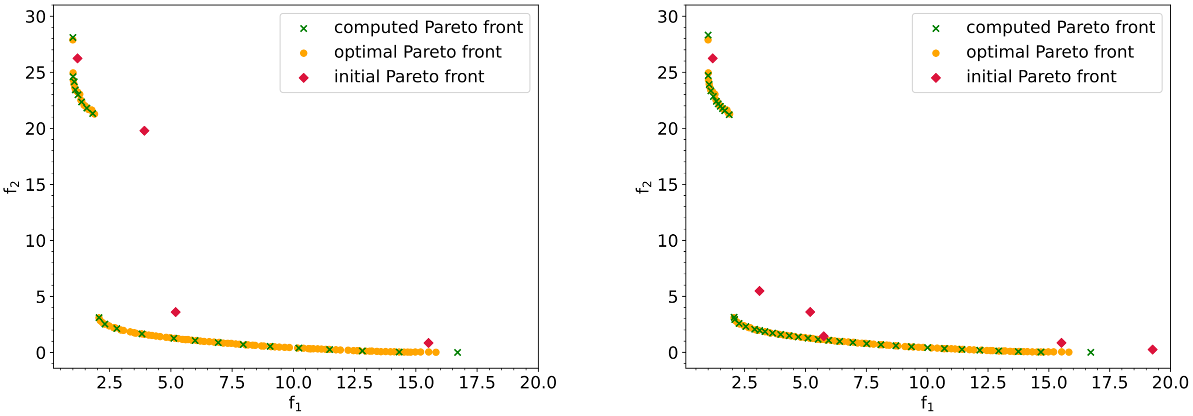


Figure 20: Obtained Pareto front with increased number of initial DOE (left) and increased total number of function evaluations (right)

Another perspective is that eventually, all three cases were able to obtain the trend of the Pareto front that align well with the optimal Pareto front. This not only proves the effectiveness of EHVI in exploration as well as exploitation, but also confirms that the surrogate model does not need to be an error-free replicate of the actual black-box function, in order to perform optimization. What is more important is a proper allocation of computational budget between the initial sampling and the infills such that the better Pareto front, in terms of its distribution as well as values, can be obtained within smaller number of total function evaluations. Thus, going back to the results for a moment, the initial surrogate model obtained with $N = 20$, or even $N = 16$ would be considered adequate, and it would be unnecessary to further increase the size of the initial DOE, unless it improves the Pareto front in the area of interest, e.g. extreme points, or reduces the total computational budget. Nevertheless, how the function evaluation budget is distributed between initial DOE and infill points depends on the ultimate objective of each optimization problem. For example, if obtaining a full, continuous Pareto front is the main interest, the higher number of function evaluations would be absolutely necessary.

The exact values of the computed errors are presented in Table 2 and Table 3 for $\widehat{f_1}$ and $\widehat{f_2}$ respectively. These results are consistent with its graphical results such that a larger size of the initial DOE results in more accurate surrogate model, and just as the previous results, it is particularly evident for the first three cases where the error reduces drastically in theses first three cases. Therefore, this also shows that the efficient choice of the size of the initial sampling could be $N = 16$ or 20.

Furthermore, it can be seen that the errors are higher for $\widehat{f_1}$ in general compared to $\widehat{f_2}$, which is consistent with the graphical results where due to its simplicity. The initial

Table 2: Errors in the surrogate models $\widehat{f_1}$ with different size of the initial DOE

Size of the initial DOE	5	8	13	16	20
RMSE	10.2	6.235	2.61	1.76	1.403
NRMSE	0.198	0.122	0.0468	0.03068	0.0237
max. prediction error	20.02	13.7	7.57	5.22	4.308
normalized max. prediction error	0.388	0.268	0.136	0.09097	0.0729
surrogate model range	51.603	51.07	55.7	57.3	59.063

surrogate model $\widehat{f_2}$ was able to mimic the function f_2 already with the small number of the initial DOE samples.

Table 3: Errors in the surrogate models $\widehat{f_2}$ with different size of the initial DOE

Size of the initial DOE	5	8	13	16	20
RMSE	7.495	0.176	0.002	0.00239	0.00135
NRMSE	0.3603	0.003807	4.265e-05	5.103e-05	3.04e-05
max. prediction error	25.27	0.936	0.00616	0.01	0.0044
normalized max. prediction error	1.214	0.02021	1.32e-04	2.16e-4	9.85e-05
surrogate model range	20.8	46.3	46.8	46.8	44.5

5.2 Fonseca-Fleming function

The second analytical test case is completed by Fonseca-Fleming function which was introduced by Fonseca et al. [8]. The optimal Pareto front of this function is concave, which is particularly difficult to solve using some of the conventional methods, e.g. weighting method. Thus, in this section, the capability of finding the concave Pareto front is tested.

5.2.1 Problem formulation

The problem is formally written as follows:

$$\text{minimize} \quad \begin{cases} f_1(\boldsymbol{x}) = 1 - \exp\left[-\sum_{i=1}^{3}\left(x_i - \frac{1}{\sqrt{3}}\right)^2\right] \\ f_2(\boldsymbol{x}) = 1 - \exp\left[-\sum_{i=1}^{3}\left(x_i + \frac{1}{\sqrt{3}}\right)^2\right] \end{cases} \tag{60}$$

$$\text{where} \qquad x_i \in [-\pi, \pi] \qquad \text{for } i = 1, \ldots, 3$$

5.2.2 Results

The result shown in Fig. 21 is obtained after 70 function evaluations, with the initial sampling size and the number of infills being 20 and 40, respectively. The initial DOE sampling is completed with the LHS, and the acquisition function used is WFG. The result shows that the Bayesian optimization was able to detect the full trend of the optimal Pareto front within 70 function evaluations.

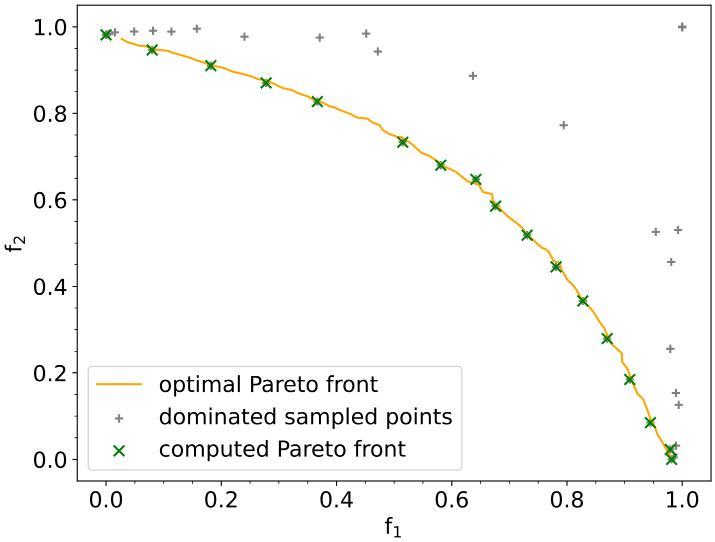


Figure 21: Result of optimization

The obtained Pareto front has been compared to the Pareto front computed by genetic algorithm (GA), a variant of NSGA-II, in Fig. 22. Here, the genetic algorithm required the population size of 60 and 60 generations in order to obtain the Pareto front that has comparable quality to the one obtained by the Bayesian optimization with 70 function evaluations, which again confirms the efficiency of Bayesian optimization in terms of computational budget.

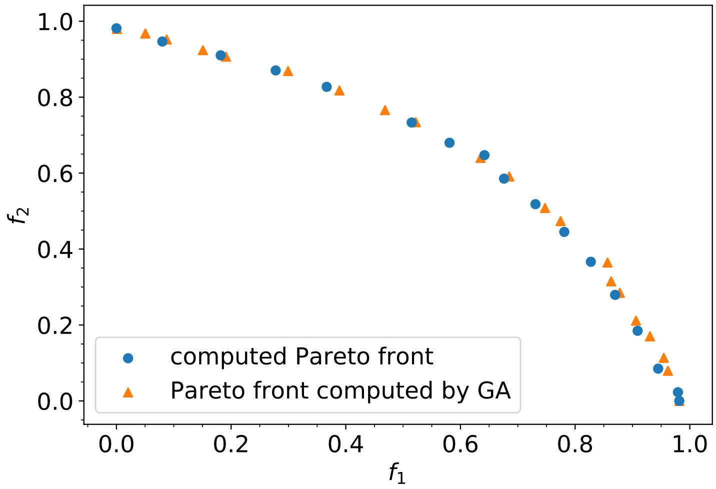


Figure 22: Comparison in the Pareto front computed by 70 function evaluations and the Pareto front obtained by genetic algorithm (GA) with 3600 function evaluations

5.3 Viennet function

The Viennet function proposed by [44] consists of three objective functions with two decision variables. Some of the challenges that are present in this test function involve high-dimensional objective space, discontinuous Pareto optimal set and several local minima in objective functions [19].

5.3.1 Problem formulation

The 3D Viennet function is formally written as follows:

$$\text{minimize} \quad \begin{cases} f_1(\boldsymbol{x}) = 0.5\,(x_1^2 + x_2^2) + \sin\,(x_1^2 + x_2^2) \\ f_2(\boldsymbol{x}) = \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15 \\ f_3(\boldsymbol{x}) = \frac{1}{x_1^2 + y_2^2 + 1} - 1.1\exp\left(-(x_1^2 + x_2^2)\right) \end{cases} \tag{61}$$

$$\text{where} \qquad x_i \in [-3, 3] \qquad \text{for } i = 1, 2$$

5.3.2 Results

The Viennet function was solved with the settings as summarized in Table 4.

The resulting Pareto front has been plotted as shown in Fig. 23. For validation, the optimization has been performed by genetic algorithm with 5000 individuals that re-

Table 4: Optimization settings for solving the 3D Viennet function

Size of the initial DOE	Number of Infills	Total number of function evaluations	Initial sampling method	Acquisition function
60	50	110	LHS	WFG

quired overall 685000 function evaluations. And, the resulting Pareto front is plotted along with the Pareto front computed by the Bayesian optimization framework.

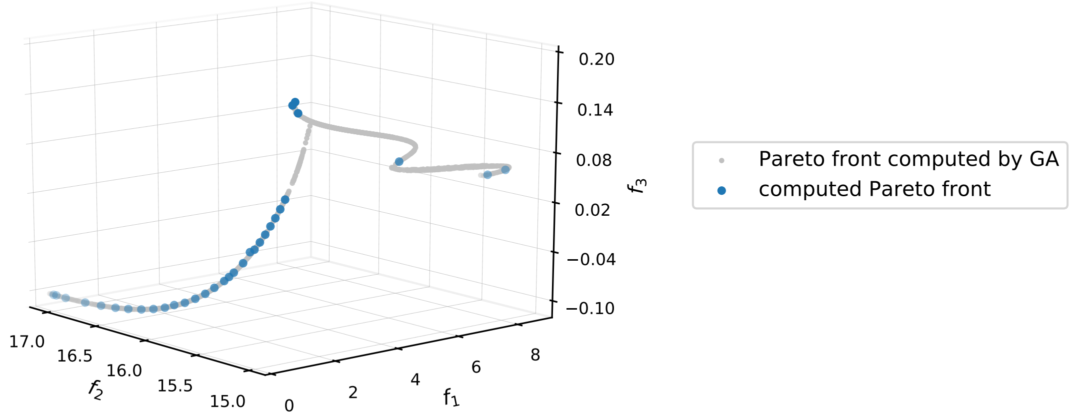


Figure 23: The computed Pareto front by Bayesian optimization plotted over the Pareto front obtained by genetic algorithm (GA)

From Fig. 23, it can be seen that in general, the computed Pareto front is consistent with the Pareto front of a genetic algorithm. Furthermore, although it did not fully capture the Pareto front compared to the one obtained by the genetic algorithm, it was still able to capture the trend of the most of the Pareto front.

The regions where it struggled to find the full Pareto front set were those with highly non-linear shape of the optimal Pareto front, to be precise, where $f_1 > 4.5$. In contrast, it was able to find a rather dense set of Pareto front subset near $f_1 < 4$, and the possible scenarios are as follows. First of all, the Pareto front being very dense in one area implies that there were no other regions that were likely to have the higher EHVI value, assuming that the optimizer of an acquisition function was able to find the optimal value. Then, this results in two remaining possibilities that first, according to the EHVI criteria, the points around the sparse region were not likely to contribute enough to the hypervolume compared to the points near the dense region. Another possibility is that from the design space point of view, the optimizer did "explore" enough with enough sampling around this region such that the estimated errors in this region are low, which unfortunately were not the optimal points, and thus were dominated. In the end, the EHVI is a criteria that balances between these possibilities, so the combination of the two above-mentioned scenarios are also possible.

For a more realistic and feasible comparison, the Viennet problem was solved once again by a genetic algorithm with a reduced number of the total function evaluations.

Here, the number of function evaluations resulted in 10000 from 100 generations with a population size of 100. And, the corresponding results are shown in Fig. 24.

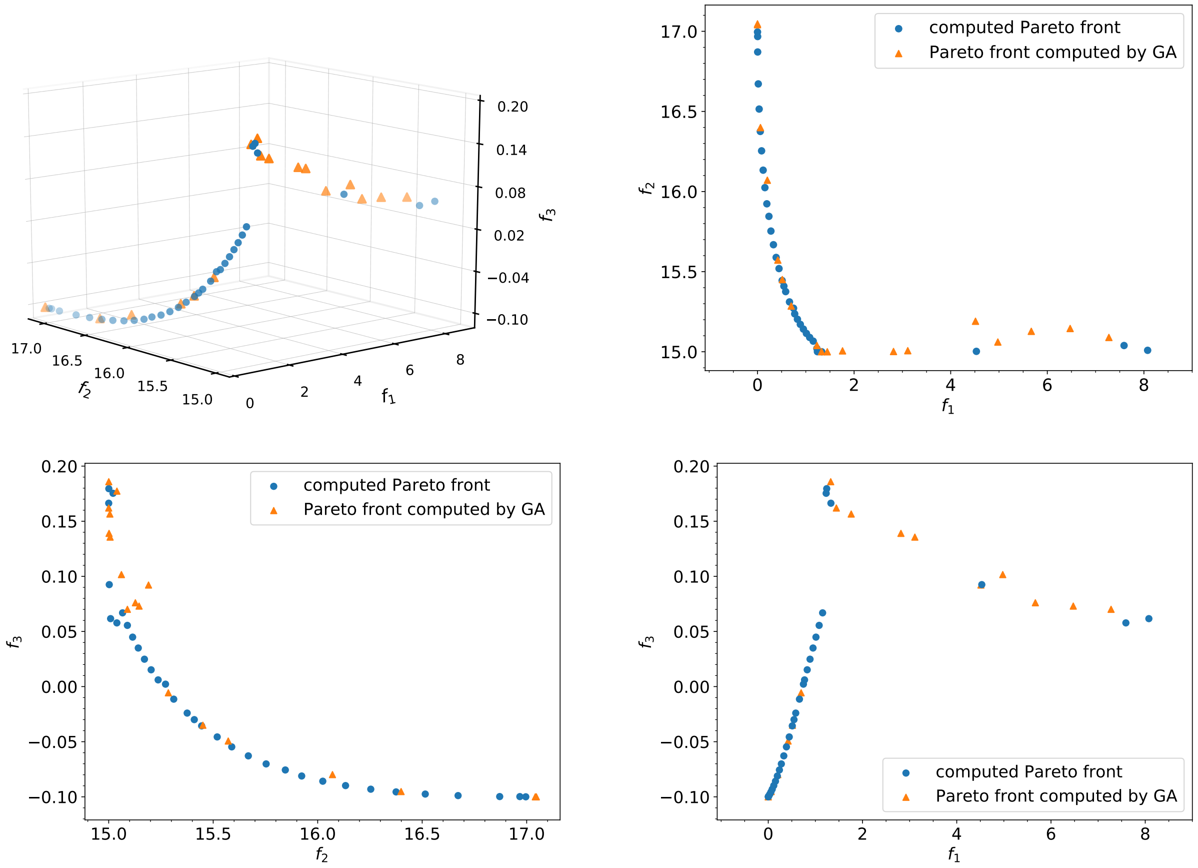


Figure 24: Pareto front plotted in 3D (top left), and against each axis

First, on the top left of Fig. 24, the full 3D plot is shown, where the computed Pareto front is denoted by circles and the Pareto front obtained from genetic algorithm is shown as triangles. Then, the projections on each axis are plotted on the top right, f_1 versus f_2, and the bottom left, f_2 versus f_3, lastly at the bottom right, f_1 versus f_3.

Again, what is noticeable is the non-linear regions, where the Pareto front of GA seem to have provided much dense solutions. Here, it seems that the Pareto front obtained by the GA has captured the non-linearity to a certain extent. Arguably, this does not necessarily mean that the Pareto front of GA has a better quality or that the subset of the Pareto front of GA in these areas is outperforming that of the Pareto front of Bayesian optimization, which can be shown by examining each plot more closely. For instance, tracking one element of Pareto front that at $f_1 \approx 4.5$ and $f_3 \approx 0.09$, will call this point A, and let one element of Pareto front of GA at $f_1 \approx 5$ and $f_3 \approx 0.1$ be point B. Now, tracking these two points on all three planes by fixing the given f_1 and f_3 coordinates, it can be easily seen that the point B is dominated by point A.

A possible reason for this discrepancy between the two Pareto fronts in this region could be the difference in the Pareto sorting system. Considering that the genetic algorithm used here is a variant of NSGA-II, throughout the sorting processes, the higher ranks have been awarded to the solutions that contribute to uniformity. Therefore, even though it does not progress as fast as a whole Pareto front, it is capable of keeping the entire trend of the Pareto front.

On the other hand, for the EHVI, which is the hypervolume based criteria, full distribution does not mean anything, instead what is more important is to find a single, if not many, point that can significantly contribute to the hypervolume improvement, and if the expected amount of improvement in hypervolume is not sufficient, the EHVI would rather make an optimizer to exploit or explore other areas where it finds the higher expected improvement. And this is, of course, under the assumption that the estimated error of Gaussian processes is also not high enough in this region, which seems to be the case.

Nevertheless, another important point that should never be overlooked is the substantial difference in the total number of function evaluations. The fact that the Pareto front obtained by the GA required the total function evaluations that are nearly 100 times higher than that of the surrogate based optimization most certainly underscores its efficiency as well as efficacy.

6 Applied Problems

In this chapter, two applied problems involving aerodynamics and aeroacoustics are solved. The goal of the analyses involved in this chapter is therefore to validate the applicability of the implemented Bayesian optimization method towards more complicated and applied problems.

6.1 Formulation

The main objective of the airfoil shape optimization is to find promising airfoil shapes that results in a combination of reduced drag, reduced absolute value of pitching moment, and reduced noise. While reducing the drag has an effect of reduction in fuel consumption, decrease in absolute value of pitching moment is expected to reduce the trim drag. The problem is subdivided into the two separate problems, where in the first problem, the focus is onto the aerodynamic optimization, where the aerodynamic shape optimization has been carried out to obtain airfoil designs with mixed combination of lower drag and lower pitching moment for a fixed value of lift coefficient. Next, the aerodynamic-aeroacoustic shape optimization has been completed to obtain the Pareto front that contains airfoil designs with the mixed combination of lower drag, lower pitching moment as well as lower noise. For both of the problems, the lift coefficient is fixed to the target $C_{l,target}$, which is typically based on the weight estimation and the wing loading, was set to be 0.792.

The main difference of these applied problems from the analytical problems is that these are the actual black-box functions, meaning there is no prior information on the shape of the Pareto front, feasible optimization settings or the computational budget allocation. In analytical problems, this was not a critical issue as the time required by a single function evaluation was relatively low, on the order of fraction of a second. Now, we are dealing with the problems where one function evaluation would take about 5 to 10 minutes up to several hours, and therefore, because of the given time constraint, it is infeasible to test every possible aspect of the EHVI for these applied problems: this is particularly true for the second applied problem, the aerodynamic-aeroacoustic shape optimization, where one function evaluation takes approximately 3 to 4 hours. Therefore, more in-depth analysis has been completed, mainly in aerodynamic shape optimization, where one function evaluation takes much less time, around 8 minutes. In this way, it is also possible to utilize some of the information obtained from the first problem to more efficiently perform the more time-constrained optimization case. Therefore, in aerodynamic optimization, several tests were performed to verify the effects of optimization settings on the computed Pareto front. Here, the primary objectives is to observe the effect of the different optimization settings and identify the suitable settings for the aerodynamic-aeroacoustic problem. However, this analysis is by no means conclusive due to the given time constraint.

Then, based on the results obtained from the aerodynamic optimization, aeroacoustic-aerodynamic optimization has been completed to find the Pareto front that contains different airfoil designs with lower drag, lower pitching moment as well as the lower noise generation.

For both of the problems, a fixed penalty value has been applied to invalid results, i.e. cases when the mesh generation fails due to unreasonable geometry, or the resulting C_l value of the flow simulation does not meet the target C_l value. Thus, whenever the resulting C_l of the flow simulation is below the target $C_{l,target}$ with a margin of 0.05, the corresponding objective optimization is considered to be invalid, which is then subsequently penalized by the set penalty value.

6.1.1 Airfoil geometry parametrization

In the airfoil shape optimization, one essential requirement is to choose the feasible geometry parametrization method such that any change in airfoil shape can be well represented by means of few selected design parameters, besides, it should be possible to control the shape change to a certain extent with the given geometric parameters. Throughout the analysis presented in this chapter, the class function/shape function transformation (CST) geometry representation method is used for parametrizing the airfoil geometry. In this method, various classes of geometries of an airfoil, e.g. round nose, are represented by the analytic well-behaved functions [6]. The CST parametrization is briefly explained in the following through the fundamental CST functions, while a more detailed description is available in [6].

Here, the upper and lower surfaces of a cambered airfoil is defined by Bernstein polynomials of order n. Then, the set of component shape functions are defined by the terms involved in the Bernstein polynomial as follows:

$$S_i(\psi) = K_i \psi^i (1-\psi)^{n-i} \tag{62}$$

where,

$$K_i = \frac{n!}{i!(n-i)!} \tag{63}$$

here, $\psi = x/c$. Then, the class function for the airfoil is defined as follows:

$$C_{n2}^{n1}(\psi) = \psi^{n1}(1-\psi)^{n2} \tag{64}$$

$n1$ and $n2$ are the coefficients that are determined by the specific type of an airfoil, where, for a typical NACA type airfoil $n1 = 0.5$ and $n2 = 1.0$. Then, the upper and the lower surfaces are defined by the following equations,

$$\zeta_{\mathrm{u/l}} = C_{n2}^{n1}(\psi)S(\psi) + \psi\Delta\xi_{u/l} \tag{65}$$

where $\zeta = z/c$, and $\Delta\xi_{u/l}$ represent the trailing edge thickness ratios for the upper and the lower surface respectively. $S(\psi)$ for unknown coefficients $A_{i,u/l}$ is expressed as follows:

$$S(\psi) = \sum_{i=1}^{n} A_{i,u/l}S_i(\psi) \tag{66}$$

6.2 Aerodynamic Shape Optimization

For the aerodynamic shape optimization, the optimization objectives were the drag coefficient C_d and the absolute value of pitching moment coefficient $|C_{m_y}|$ while keeping the constant lift coefficient at $C_{l,target}$. As a baseline airfoil, the RAE2822 airfoil as seen in Fig. 25, is used, where its geometry is represented by a CST parametrization with 10 coefficients.

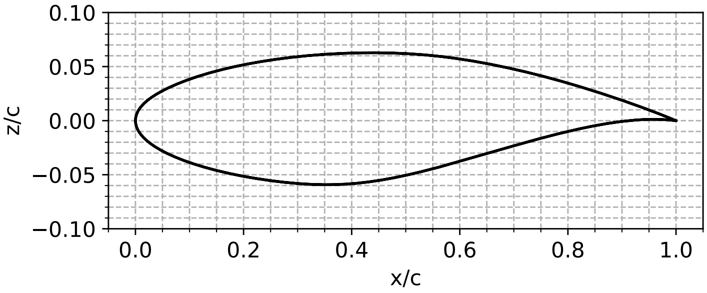


Figure 25: Geometry of the RAE2822 airfoil

Since one of the most critical aspects of surrogate based optimization is to reach as close as possible to the optimal Pareto front with a given computational budget, it is important to test and verify the effect of different settings on the resulting Pareto front. Therefore, in this analysis, several tests were performed to investigate the effect of the different optimization settings which includes: the optimization bounds of the decision variables, computational budget allocation between the initial DOE and the infills, the penalty level, the sampling method for the initial DOE and lastly, the different acquisition function. For all the tests performed, all other variables that are not being tested were set constant.

The optimization problem is then written formally as follows:

$$\min_{\boldsymbol{x}\in\Omega} \left(C_d(\boldsymbol{x}), |C_{m_y}(\boldsymbol{x})|\right) \tag{67}$$

where $\Omega = \{\boldsymbol{x} \in D\}$ for $D = [\boldsymbol{x}_L, \boldsymbol{x}_U] \subset \mathbb{R}^{10}$. Since the feasible bounds differ for each test case, it is explicitly given for each optimization case. For evaluating the objectives, DLR's in-house flow solver TAU is used with a computational grid of 28572 nodes as shown in Fig. 26.

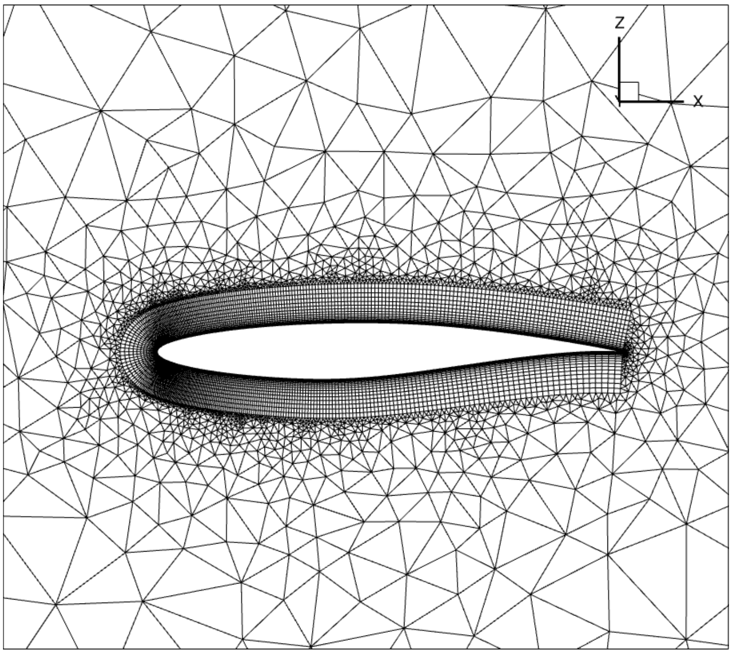


Figure 26: CFD mesh of the RAE2822 airfoil

6.2.1 Effect of the optimization bounds

The optimization bounds in aerodynamic simulation that come from the CST variables should be set wide enough to allow a meaningful shape deformation but restricted enough to not generate too many invalid geometries for CFD simulations. To test the effect of different bound settings, five test cases were performed, each with from the widest bound $(0.0, 1.0)$, $(0.3, 0.9)$, $(0.4, 1.0)$, $(0.3, 0.8)$ and $(0.4, 0.7)$, where 0 and 1 indicate a variation with respect to original CST coefficients of -0.3 and +0.3, respectively. Note that these bounds were applied for all 10 parameters of each case. For the number of the initial design of experiments, without any prior knowledge on the problem, 100 initial samples were used, which was suggested by Loeppky et al. in [23] to use ten times the problem dimensionality for Gaussian processes. Then for all cases, the total function evaluation was 500, with 400 infill iterations. The WFG is used as an acquisition function, and for the initial sampling, the Sobol method is used. Finally, for invalid results, penalty values have been applied with the penalty level of 2. The resulting Pareto fronts are shown in Fig. 27.

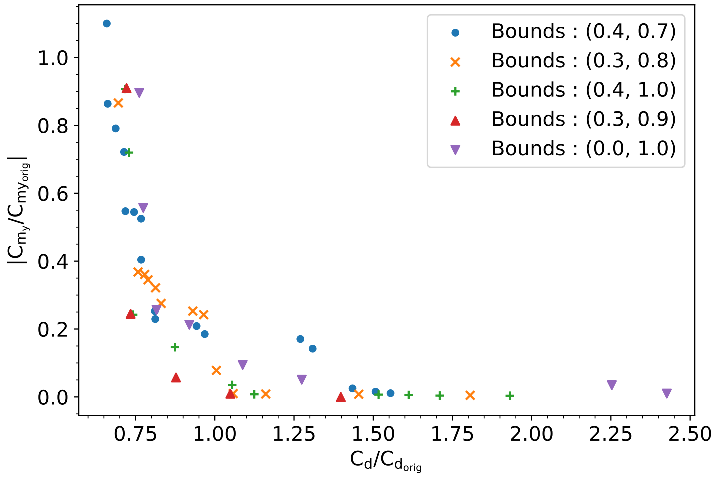


Figure 27: Pareto fronts obtained for the different bound settings

In the figure, the x-axis is the computed C_d of the modified airfoil scaled by the baseline airfoil $C_{d_{orig}}$, and similarly for the y-axis, the obtained $|C_{m_y}|$ was scaled by the baseline airfoil $|C_{m_{y_{\mathrm{orig}}}}|$. From the result, it can be seen that, in general, all the Pareto fronts reside nearby. Some of the noticeable points are that first, compared to the other bounds, the bound $(0.4, 0.7)$ did not make enough progress through the region $C_d/C_{d_{orig}} \approx 1.25$. On the other hand, it made much more progress compared to the other cases in minimum drag region, i.e. $C_d/C_{d_{orig}} < 0.75$. Particularly, around $C_d/C_{d_{orig}} < 0.75$ and $|C_{m_y}|/|C_{m_{y_{\mathrm{orig}}}}| > 0.7$, this was the only case that was able to obtain a more or less full distribution that dominates the solutions from the other cases.

One possible reason could be that this region in the objective space could have been restricted by the upper values of the bounds in the design space, which made it more difficult for the one with the lower upper bound to sample and progress in this region. For instance, for the case with the bounds $(0.4, 1.0)$ had no problem in progressing this region, thus considering that the upper bound of this case, $(0.4, 0.7)$, was the most restricted, it is possible that this limited upper bound has resulted in restricted progress in certain area compared to the other cases.

Similarly, the case with the bound $(0.0, 1.0)$, although it did made more progress than the case with the bounds $(0.4, 1.0)$, it did not progress as much as the other three cases in this region. Though, the issue for this case is slightly different. For this case, it had the widest range in the bound, meaning that it had the least restricted exploration in design space, thus it did not have a problem in identifying and progressing in this area to a certain extent also. However, the least restricted exploration also means that it is very much likely to have identified other promising regions compared to more restricted bounds. Thus, for this case, it would require much more time to observe the apparent progress in the Pareto front as a whole as it would not rapidly identify and exploit few of the promising regions as other cases. However, having widest bounds in the design variables mean that it can actually examine through the most various potential designs, and then provide the optimum solution. The biggest problem for having a wider bound, however, is that it can produce too many, unexpected invalid samples. For this specific case, it produced the total 155 invalid samples meaning that nearly one third of the total samples were used to mark the infeasible or invalid regions in the design space instead of the actual optimization, which further explains the slowest convergence of this case. Thus, having to explore more wider design space as well as producing more invalid samples mean that this widest bound $(0.0, 1.0)$ is infeasible for the optimization with given time constraint. How the optimizer spends more budget on exploration when the bounds are wider can be seen in Fig. 28. Here, it is evident that the wider bound leads to infill locations that are far more distributed than more restricted bound, where the infill evaluations are more clustered near the computed Pareto front.

The restriction also occurs for the bound $(0.3, 0.8)$ to a certain extent such that it seems to have started to exploit the regions around $0.75 < C_d/C_{d_{orig}} < 1$. Then, among the two remaining cases, it can be seen that the case with the bounds $(0.3, 0.9)$ has resulted in the most progressed Pareto front with an adequate distribution. Firstly, its solutions near the knee of the curve of its own Pareto front are the knee of the curve of the entire distribution of all Pareto fronts. Secondly, even though it does not provide the full distribution along the C_d axis, its solution at $C_d/C_{d_{orig}} \approx 1.4$ actually dominates the rest of the Pareto fronts form the other cases in $C_d/C_{d_{orig}} > 1.4$ by being a solution with the minimum value of $|C_{m_y}|/|C_{m_{y_{\text{orig}}}}|$. Although, its performance slightly reduces for the lowest C_d, it is still adequate considering the other nearby solutions except for the unusual solution from $(0.4, 0.7)$, and in general, it provides the best solutions in compromising region as well as the lowest C_{m_y}. Lastly, it produced 38 invalid samples which is large enough to mark the infeasible regions in the design space, but small enough to make a progress in convergence of the Pareto front.

From the perspective of the airfoil shape optimization, the solutions at $C_d > 1.50$ would not be very interesting even if the solution of the bound $(0.3, 0.9)$ was assumed to be excluded. This is mainly because along this axis and the distribution of the Pareto front, there is only a very small amount of reduction in C_{m_y}, while the trade-off for the C_d increases much more abruptly. Generally, the solutions at this region is also expected to contribute less to the hypervolume, which could be the reason why most of the cases have been produced more dense set of solutions near the knee point.

6.2.2 Analysis on the computational budget allocation between the initial DOE and the infills

Having a larger set of initial DOE samples means that the optimization can be performed with a more accurate initial surrogate model such that it does not have to spend excessive infill budget to explore the design space. However, this means that

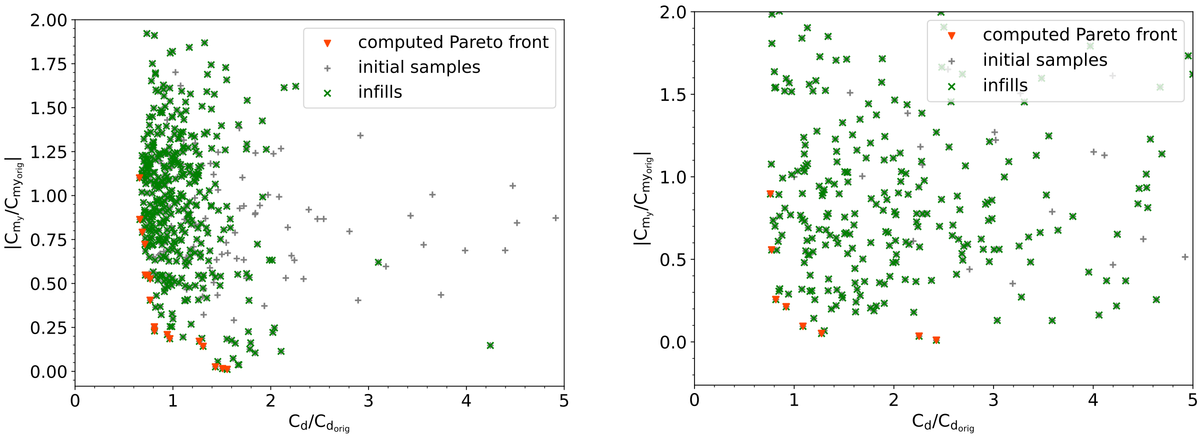


Figure 28: Initial samples, infills and the Pareto front for the bound $(0.4, 0.7)$ (left) and $(0.0, 1.0)$ (right). The design space exploration is clearly shown to be done much more for the wider bound.

the number of infill iterations needs to be reduced for a fixed amount of the total computational budget, which could lead to its reduced performance in exploitation. On the other hand, having a larger infill budget means that the optimization would leverage a less accurate, or even a largely erroneous, initial surrogate model. Thus, with a given amount of the total computational budget, it is critical to find the balance between the number of the initial DOE samples and the infill iterations that can achieve the most optimal Pareto front.

Thus, in the following analysis, the effect of the computational budget allocation between the initial DOE sampling and the infills has been investigated. For this, a total of four optimization cases were performed, where each case has a number of the initial DOE samples of 100, 400, 800 and 1024. The total number of function evaluations was fixed to be 1300. Therefore, each case had the infill budget of 1200, 900, 500 and 276. Again, in order to minimize the effect of the other variables, other parameters were set to be the safest or the most restricted, if possible. The optimization bound was fixed to $(0.4, 0.7)$ as this is found to be the safest bound which returned the least amount of invalid samples. Furthermore, the Sobol method was used to generate the initial sampling, the penalty level was set as 2, and the WFG-based EHVI is used as an acquisition function. The resulting Pareto fronts are shown in Fig. 29.

The result shows the similar behavior of the case with the initial DOE of 100 to the previous result of the bound $(0.4, 0.7)$. Considering that the only difference for this case compared to the previous case of the bound $(0.4, 0.7)$ is the number of additional infills, this result confirms that even after 700 more infill iterations, it was impossible to progress through the region, $C_d/C_{d_{orig}} \approx 1.2$. However, for all other three cases with the higher initial samples, it was possible to progress through this region. This confirms that its upper bound is not the limitation that prevents it from progressing,

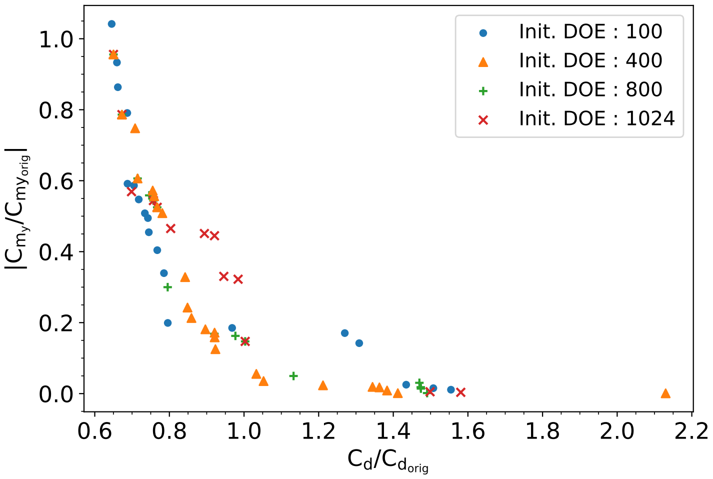


Figure 29: Pareto fronts obtained for the different computational budget allocation

but the potential issue was more on its lack of exploration in the design space, and similar to the previous case, this resulted in its immense exploitation at the lowest C_d region. For the case with the increased number of initial DOE of 400, it seems to have produced a better distributed Pareto front with particularly dense solutions around the knee points. The next case with 800 initial DOE samples shows that it did not progress to convergence as much as the previous case with 400 initial samples. Again, this becomes even worse for the increased initial samples of 1024 as it has the clearly visible region where it struggled to progress through near $0.8 < C_d/C_{d_{orig}} < 1.0$. The progress in the Pareto front for the four cases can be investigated further by examining the change in the hypervolume indicator through the infill iteration as shown in Fig. 30. First of all from the right plot, from the converged values of the hypervolume indicator, the difference between the case with 100 initial samples and the other cases are distinctive. Furthermore, the region where the initial samples of 1024 struggled is also visible, by being the second lowest, it also implies that this region was smaller compared to the region where the 100 initial sample case struggled. Then, the remaining two cases have converged to the Pareto front that resulted in the higher hypervolumes, with the initial samples of 400 being the highest. Thus, this result in the hypervolume indicator is consistent with the previous result of its Pareto front in Fig. 29.

Apart from the converged values of the hypervolume indicator for the different cases, what is noticeable is its progress in increasing the HV through the infill iterations. First of all, comparing the two cases with the initial samples of 800 and 1024, it can be seen that despite the fact that the two cases started with the different number of the initial samples, it started with more or less the same HV, and this resulted in the better final value for the case of 800 initial samples as it had more budget for infill iterations. Another interesting point is that the case with 400 initial samples has nearly reached its final HV value as soon as its 600th function evaluation, which has higher value than

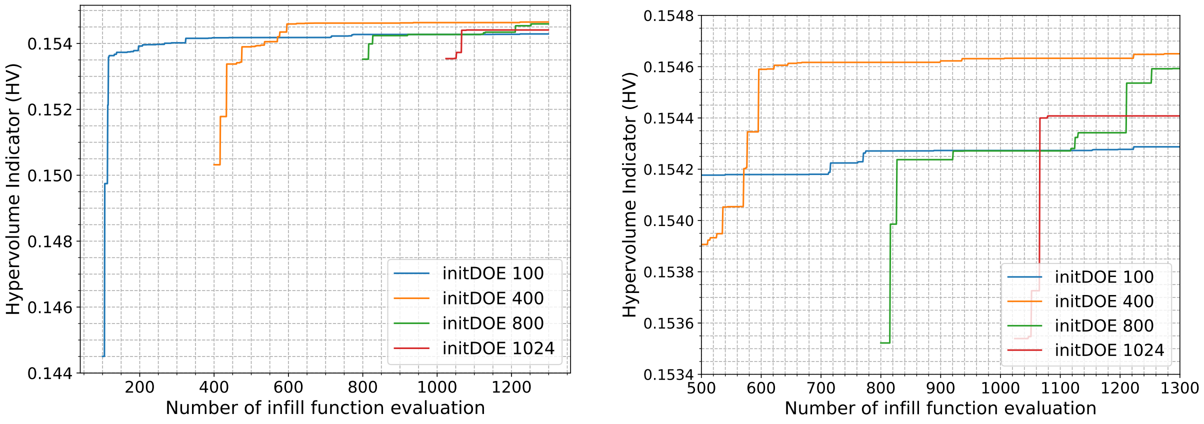


Figure 30: Change in hypervolume indicator (HV) for increasing number of infill iterations

the converged values of all other cases. This suggests that the most efficient way to reach the highest possible hypervolume for given cases is to allocate 400 evaluations of the initial sample with the additional infills of around 600. Thus, in the following analyses, initial samples of 400 and the infills of 800 were used which include some margin to ensure the convergence.

For the hypervolume indicator computation, the reference point was set to $(0.4, 0.4)$ for the pre-scaled data. Usually, the hypervolume computation is affected by the reference point, particularly when comparing the different Pareto fronts that are asymmetric, e.g. one Pareto front set is more skewed towards one objective while the other is in the opposite situation. However, from the converged Pareto fronts in Fig. 29, the Pareto front of all three cases are more or less well aligned except for the two visible gaps. One way to check this skewness of the Pareto fronts would be to draw vertical lines of extreme Pareto front sets along each objective. In this way, it is possible to check if there exists extreme sets that cause abnormally larger hypervolume. Once these extreme abnormality is checked, the difference in the hypervolume indicator consequently indicates the difference along its distribution. Since no such abnormality exists in the results from Fig. 29, the selected reference point used for computing the result in Fig. 30 can be justified. Further examination on the reference value to compare the Pareto fronts in "fairer" manner would require further analysis for given shape of Pareto fronts, e.g. [21] suggests analysis to find the better reference point for Evolution Algorithm (EA).

Furthermore, in order to quantify and compare the errors involved in the initial surrogate model for the different number of initial samples, the errors involved in the initial surrogate models that are constructed with 100, 200, 300, 400 and 800 initial DOE samples have been computed. Firstly, the initial surrogate model errors for C_d is shown in Table 5. For this error computation, all the invalid samples, which are set to penalty values in optimization, are excluded. For this reason, one could argue that the computed errors are not fully representative of the actual surrogate models, however, from the perspective of surrogate based optimization, accuracy near the infeasible regions is not of interest. Instead, what would be more valuable is accuracy near the promising optimum region. For instance, from Table 5, where the RMSE gradually decreases for the increased number of initial DOE samples, the RMSEs for the initial samples of 400 and 800 would be increased when the penalized samples are included. However, this penalized samples are actually helping an optimizer to be aware of infeasible regions. Thus, strictly from an optimization point of view, computed errors are indeed representative of the actual surrogate models for all five cases. Again, another noticeable point from Table 5 is the reduction in RMSE for adding additional 100 DOE samples to the initial 100 samples, which could be an indication to use at least 200 samples unless limited by the available computational budget.

Table 5: Errors in the initial surrogate models for C_d with different number of the initial DOE samples

Number of initial DOE samples	100	200	300	400	800
RMSE	0.0221	0.0175	0.0159	0.01307	0.0144
NRMSE	0.1421	0.1127	0.1025	0.0842	0.0926
max. prediction error	0.0661	0.065	0.0582	0.04905	0.06704
normalized max. prediction error	0.4257	0.4188	0.375	0.316	0.432
surrogate model range	0.155				

Next, the initial surrogate model errors for C_{m_y} can be found in Table 6. Similarly, for this result, penalized values are excluded from computation. Also, a similar trend can be observed such that the RMSE gradually reduces for an increased number of the initial DOE.

Table 6: Errors in the initial surrogate models for C_{m_y} with different number of the initial DOE samples

Number of initial DOE samples	100	200	300	400	800
RMSE	0.0118	0.009056	0.00835	0.00775	0.007044
NRMSE	0.111	0.0853	0.0787	0.0729	0.0663
max. prediction error	0.0376	0.0359	0.0327	0.0309	0.0362
normalized max. prediction error	0.354	0.338	0.3077	0.291	0.341
surrogate model range	0.106				

6.2.3 Effect of different penalty values

Having completed the analysis on the reasonable bounds as well as the computational budget allocation, further investigation has been completed to observe the effect of different penalty values on the final Pareto front. The higher penalty values would act as the stronger markers for an optimizer to avoid infeasible areas in the design space. However, this could induce certain unexpected artificial features in the underlying surrogate model that can effect the resulting Pareto front. The smaller penalty value, on the other hand, would limit this problem, but its values might not act as strong as the larger value to mark the infeasible regions.

To assess this aspect, two optimization test cases were performed with penalty level of 2 and 3 respectively. Based on the previous analyses, a total of 800 function evaluations were done with 400 initial DOE samples and 400 infills. Since it needs to produce invalid samples to test the effect of the penalty value, the optimization bounds were set to $(0.3, 0.9)$. For the initial sampling, the Sobol method is used, and the WFG-based EHVI is used as an acquisition function Then, the resulting Pareto fronts are presented in Fig. 31. From the figure, first of all, it can be observed that there is no distinctive gap between the two results where one case clearly outperforms the other. However, the differences can be observed mainly in two regions. Firstly, around knee-points $0.7 < C_d/C_{d_{orig}} < 1.4$, the result from the penalty level 2 is shown to have been exploiting much more, and outperformed the result from the penalty level 3 in general in this region, meaning that the solutions produced by the first case is dominating most of the solutions produced by the other case. In the regions along the minimum values of each axis, in contrast, the solution produced by the penalty level 3 is outperforming, and it has a more uniformly distributed Pareto front. However, considering the fact that the points near the knee-point are contributing the most towards the hypervolume improvement as observed from the previous analysis, the fact that it has shown less performance in knee-point area is critical.

6.2.4 Effect of different initial sampling methods

In order to keep efficiency as high as possible in surrogate optimization, it is important to observe the effect of different initial sampling methods as much as determining the number of initial samples. Thus, the test was performed to see the difference between, the Sobol and the LHS methods, in resulting Pareto fronts. For the optimization bounds, $(0.4, 0.7)$ was used as it is not necessary to generate invalid samples for this case, and, the WFG method was used along with the penalty level of 2. Again, it completed the total 800 function evaluations with 400 initial samples and 400 infill iterations, and, the resulting Pareto fronts of the two cases can be seen in Fig. 32. The result shows that, first of all, each case outperformed the other in different regions, for instance, the Sobol result is outperforming in the region, $C_d/C_{d_{orig}} > 0.8$, where the

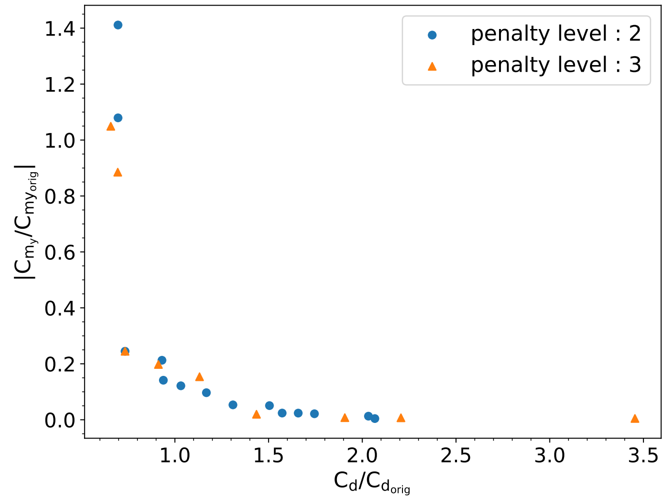


Figure 31: Result of the computed Pareto fronts for the different penalty level settings

clear gap is visible between the two Pareto fronts. Meanwhile, the LHS method seems to have produced marginally better solutions near $0.4 < |C_{m_y}|/|C_{m_{y_{\text{orig}}}}| < 0.55$. Inspecting the gaps between the two Pareto fronts, the Sobol method has been shown to have created a better result. This aspect is verified by the the exact computation of the hypervolume for the pre-scaled reference point of $(0.4, 0.4)$: the result of the Sobol method has the HV of ≈ 0.1546, whereas that of the LHS has the HV of ≈ 0.1532. Nonetheless, considering that the LHS involves randomness, the result might change if more test runs were performed. Therefore, further analysis would be required to investigate the effect of each sampling method more precisely.

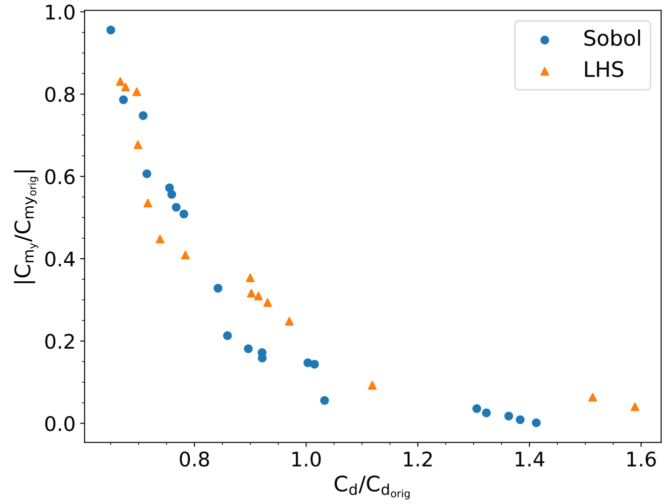


Figure 32: Result of the computed Pareto fronts for different initial sampling methods

6.2.5 Effect of the acquisition function

Although the 2D efficient EHVI and the WFG-based EHVI aims to the same goal of the exact computation of the EHVI as seen from Section 4.2, the two methods are taking a different approach, which resulted in different final expressions as well as algorithms. Therefore, another numerical test was conducted to verify the effect of the chosen acquisition function on the resulting Pareto front. Similar to the previous tests, all the other parameters were set constant. Here, 400 initial samples and 400 infills were allocated, and, the optimization bound was set to $(0.4, 0.7)$. For the initial sampling, the Sobol method is used with the penalty level was set to two.

The result in Fig. 33 shows that the two solutions are aligning in general, except for the area near $C_d \approx 0.8$ and $C_{m_y} \approx 0.2$ where the solution of the efficient 2D method dominates the whole subset of the solutions of the WFG method in that region. However, in the areas near $C_d \approx 1.0$ and $C_d \approx 1.4$, the solutions of the WFG method dominate.

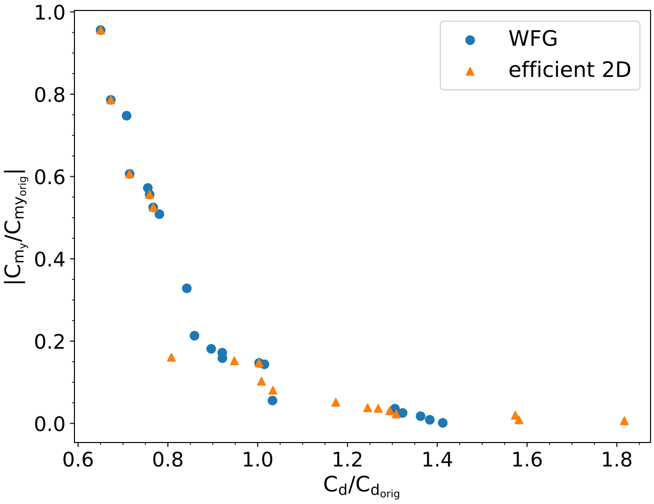


Figure 33: Pareto fronts obtained from the different acquisition function

For the two results, quantitative comparison is not very helpful as the difference in the hypervolume indicator of the two solutions is less than 0.0001, which makes it difficult to compare the quality of the two Pareto fronts. Qualitatively comparing the two results, the efficient 2D solution would be appealing considering that its Pareto front was able to progress through the knee point, which is often a major interest as it provides a good compromise between the two objectives. On the other hand, the WFG result could be more appealing if the interested area is in $C_d \approx 1.0$ or $C_d \approx 1.4$. Therefore, the precise assessment on the quality of the two Pareto fronts are difficult such that the further analysis would be required along with the comparison in the difference in the obtained designs of each Pareto front. Nevertheless, no drastic discrepancy is observed in the Pareto fronts obtained by the two different acquisition functions.

6.2.6 Airfoil design results

From the computed Pareto fronts in Section 6.2.1, few of the resulted airfoil designs are presented. The first design point selected is the compromising point where the there is reasonable reduction for both the drag and the pitching moment. The corresponding solution from the obtained Pareto front is indicated in Fig. 34 along with the modified airfoil shape and its c_p distribution. Here, for the angle of attack of 3.83 degree and C_l of 0.792, the drag (C_d) was reduced by 26.6% compared to the baseline airfoil, and the pitching moment coefficient ($|C_{m_y}|$) was reduced by 75.5%.

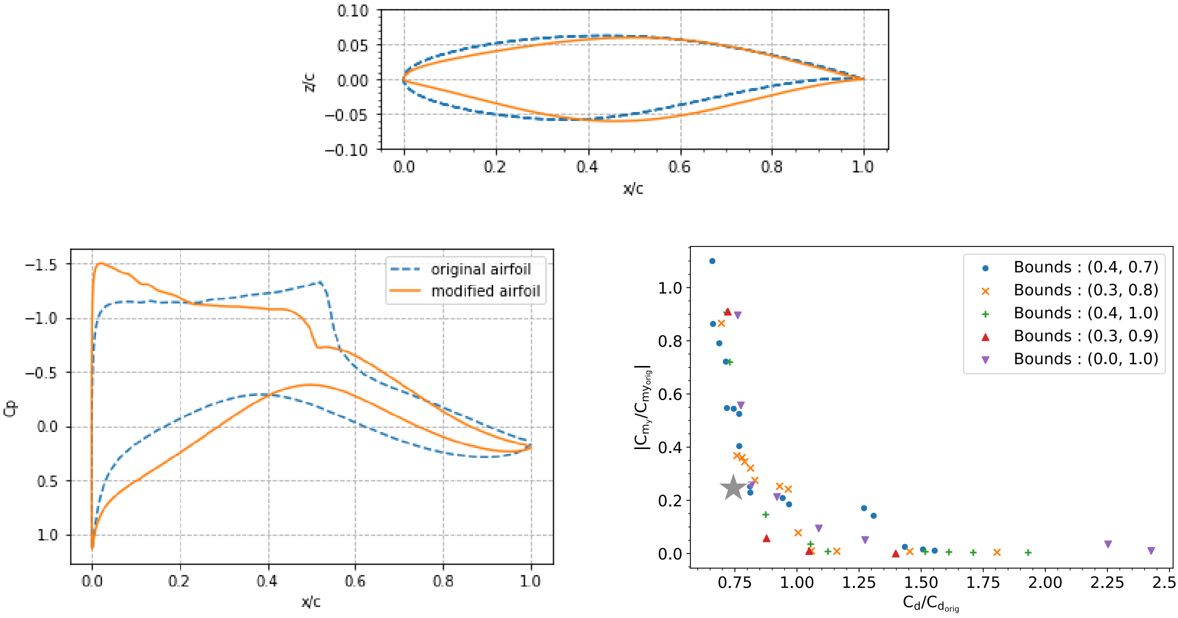


Figure 34: Change in airfoil shape (top) and C_p distribution (bottom left) for the compromising solution, and the location of the compromising solution from the Pareto front is marked with star (bottom right)

Next, the solution from the Pareto front which yielded the minimum drag coefficient C_d was evaluated, where its location from the Pareto front is marked in Fig. 35.

This solution resulted in the reduction in the drag coefficient by 34.1% compared to the baseline airfoil, and increase in the absolute value of the pitching moment coefficient by 10 %. The lift coefficient was again set to 0.792 and the angle of attack was 2.73. The resulting airfoil shape as well as its C_p distribution are compared to the single-objective surrogate based optimization result of [10], where the resulting reduction in the drag coefficient was by 34.9%. Considering that in multi-objective optimization case, the two objectives are being minimized simultaneously, difference in the drag coefficient reduction compared to the reference is reasonable.

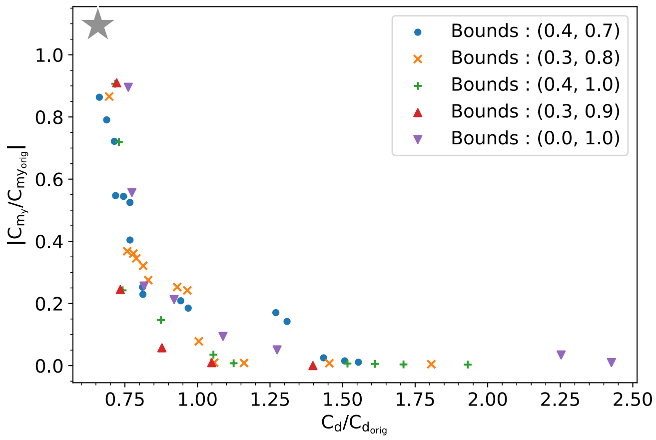
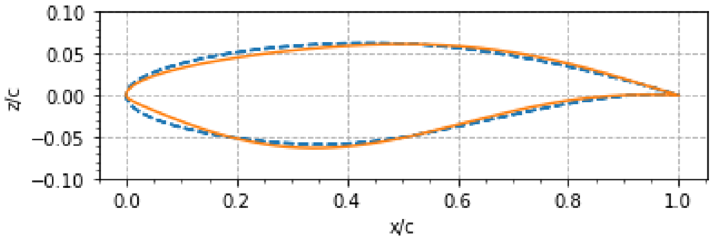
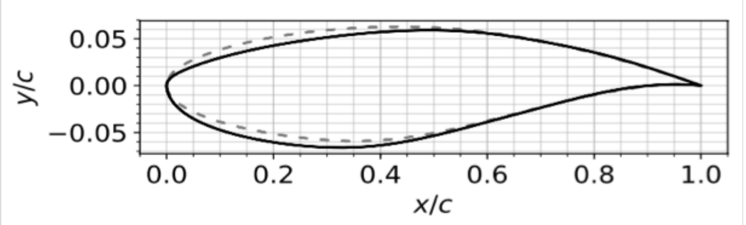


Figure 35: Location of the minimum drag coefficient solution from the Pareto front



(a) Airfoil shape for the Pareto front solution with the minimum drag

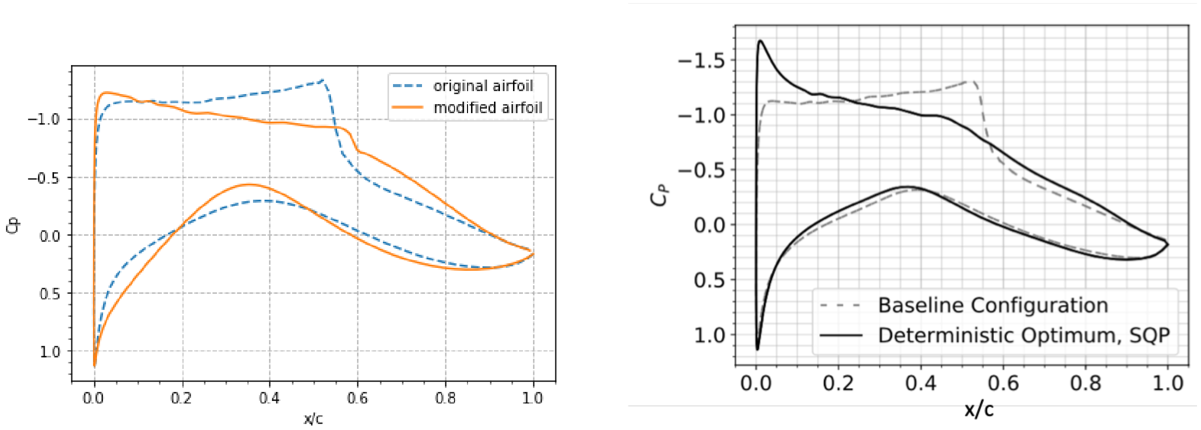


(b) Optimized airfoil design obtained by single-objective optimization. From [10]

Figure 36: Modifications in airfoil shape (RAE2822) for the minimum drag

First, comparing the resulting airfoil shape change in Fig. 36, the two airfoils have different geometries, which could possibly be due to the different approaches in airfoil shape parametrization.

Fig. 37 shows the resulting change in the C_p distribution, here, both of the results shows a similar trend such that the size of the normal shock wave of the baseline configuration has been reduced.



(a) Result obtained from the Pareto front

(b) Result obtained by single-objective optimization. From [10]

Figure 37: Changes in C_p distribution for the minimum drag

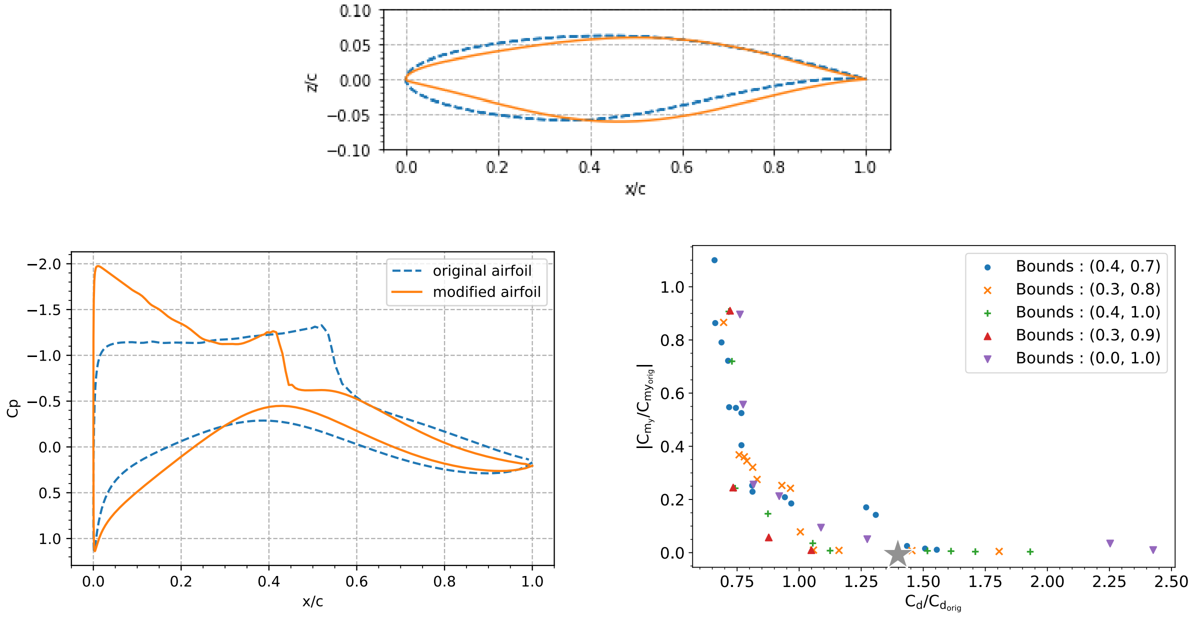


Figure 38: Change in airfoil shape (top) and C_p distribution (bottom left) for the minimum $|C_{m_y}|$, and the location of this solution from the Pareto front is marked with star (bottom right)

Lastly, the Pareto front solution that yielded the minimum $|C_{m_y}|$ value was evaluated as seen in Fig. 38. Here, for C_l of 0.792 and the angle of attack of 4.32, the drag coefficient is increased by 39.8%, while the absolute value of the pitching moment coefficient is decreased by 99.9%.

6.3 Aerodynamic-Aeroacoustic Shape Optimization

In the following, method and result of the aerodynamic-aeroacoustic airfoil shape optimization is presented.

6.3.1 Problem statement and description

The primary objective of this optimization is to obtain the Pareto front, which contains the promising design solutions for an airfoil that has a combination of minimum drag C_d, minimum absolute value of the pitching moment C_{m_y} and the minimum aeroacoustic noise for a fixed lift coefficient $C_{l,target}$. Thus, the problem is formulated as follows:

$$\min_{\boldsymbol{x}\in\Omega} \left(C_d(\boldsymbol{x}), |C_{m_y}(\boldsymbol{x})|, NoiseLevel(\boldsymbol{x})\right) \tag{68}$$

where $\Omega = \{\boldsymbol{x} \in D\}$ for $D = [\boldsymbol{x}_L, \boldsymbol{x}_U] \subset \mathbb{R}^6$.

The objective functions were evaluated by means of a black-box framework that couples the DLR's CFD-RANS solver TAU and the DLR's Computational Aeroacoustic (CAA) code PIANO.

The approximate execution time of each simulation was 1 to 1.5 hours and 2 to 2.5 respectively. However, in order for the CAA simulation to compute the mean flow vorticity that partially describes the dominated vortex sound source, which is responsible for a direct sound source for the sound generation in free turbulence, it requires the mean flow variables from the steady RANS simulation [41].

This means that the two simulations essentially need to be carried out sequentially, which makes the total function evaluation time of around 3 to 4 hours. As a baseline airfoil, the NACA 63416 airfoil in Fig. 39 is used with 6 CST parameters that describe the geometry of an airfoil, which results in 6 decision variables for the optimization. With a given time constraint of this thesis work as well as the total function evaluations, obtaining the most optimal Pareto front with the minimum possible total function evaluations is therefore the most critical point.

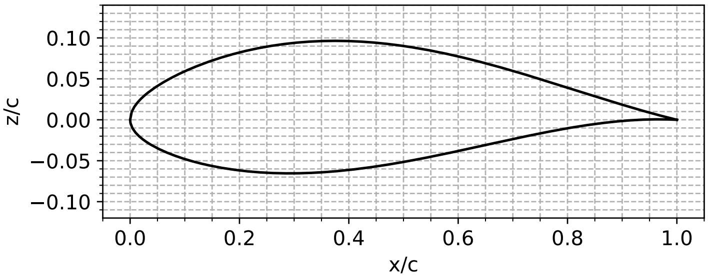


Figure 39: Geometry of the NACA 63416 airfoil

6.3.2 Optimization parameters

The first step of the aerodynamic-aeroacoustic optimization is to determine the optimization bounds and the computational budget allocation, which has been reasonably estimated by using the results obtained from Section 6.2.2. In this section, the suggested initial samples and the infill evaluation were 400 and 400 respectively. Since the decision variables have been reduced to six, the corresponding number of the function evaluation were also adjusted using full factorial design. Given that the aerodynamic analysis used total 10 design variables, the strata of the design space can be computed using Equation (1). This gives the strata of $s \approx 1.82$ for the 400 initial samples with 10 design variables. Assuming the new design space needs the similar coverage, i.e. the strata of $s \approx 1.82$, the required number of simulation for the aerodynamic-aeroacoustic optimization case is $N \approx 36.41$. Thus, applying this for both of the initial samples as well as the infill, the required number of initial samples and the infill iteration are 37 and 37. For the final values of the initial samples and the number of infill evaluations, these values were rounded to 40 each to give some margins.

For the optimization bounds, the analysis from Section 6.2.1 was taken. From this analysis, it was shown that the non-dimensionalized bounds of $(0.3, 0.9)$ have resulted in the most progressed Pareto front compared to the other bounds.

6.3.3 Results

The resulting Pareto front obtained from the aerodynamic-aeroacoustic shape optimization is shown in Fig. 40 as 3D. From the figure, the Pareto front is seen to be the foremost scattered points. Compared to the result from the 3D analytical problem, the main difference that can be noticed here is that there is no one particular region in the objective space that was immensely exploited or at least not as much as it did for the 3D Viennet function. This could imply that the Pareto front as a whole has been progressed with a more or less balanced exploration as well as exploitation.

In order to observe the obtained result more closely in Fig. 41, the same result is projected on each 2D plane, where each objective is scaled relative to the baseline configuration. First of all, the result shows that from the computed Pareto front which contains 11 solutions, 7 solutions have been found during the infill iterations while remaining 4 solutions were found directly from the initial DOE samples. In addition, comparing the set of the initial DOE samples to the infill points, the infill points are more tilted towards the lower x-axis as well as y-axis, which means that the optimizer has been identified the correct direction to proceed, thus was exploiting more on this region. Although there is no distinctive area of exploitation as it was in some of the the previous results, from the c_d and c_{m_y} plot, it shows that there are four consecutive points near

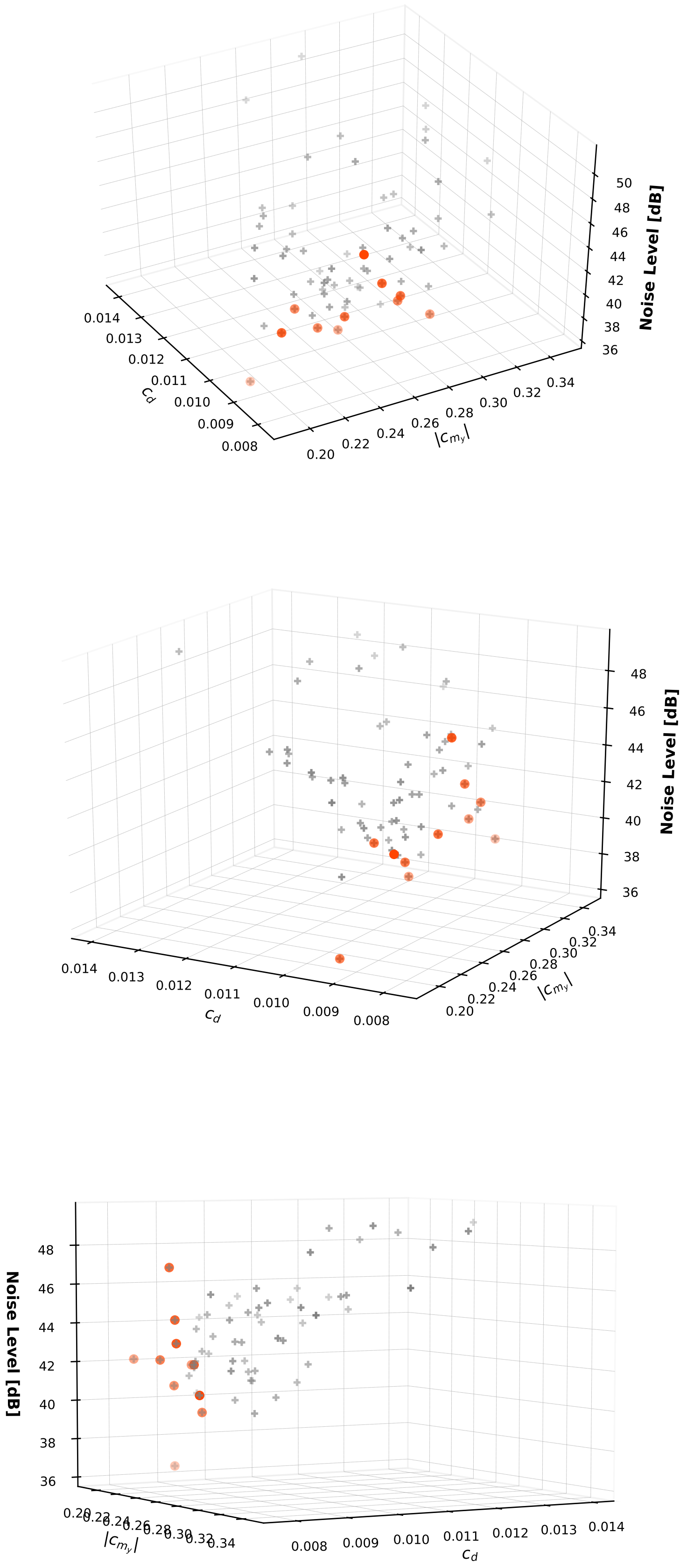


Figure 40: Result of the Pareto front obtained from the aerodynamic-aeroacoustic shape optimization, the Pareto front is marked with red circles, and the gray plus signs indicate the dominated samples

$c_d \approx 1.0$, which could suggest that the optimizer might have had a favor in specific region to a certain extent.

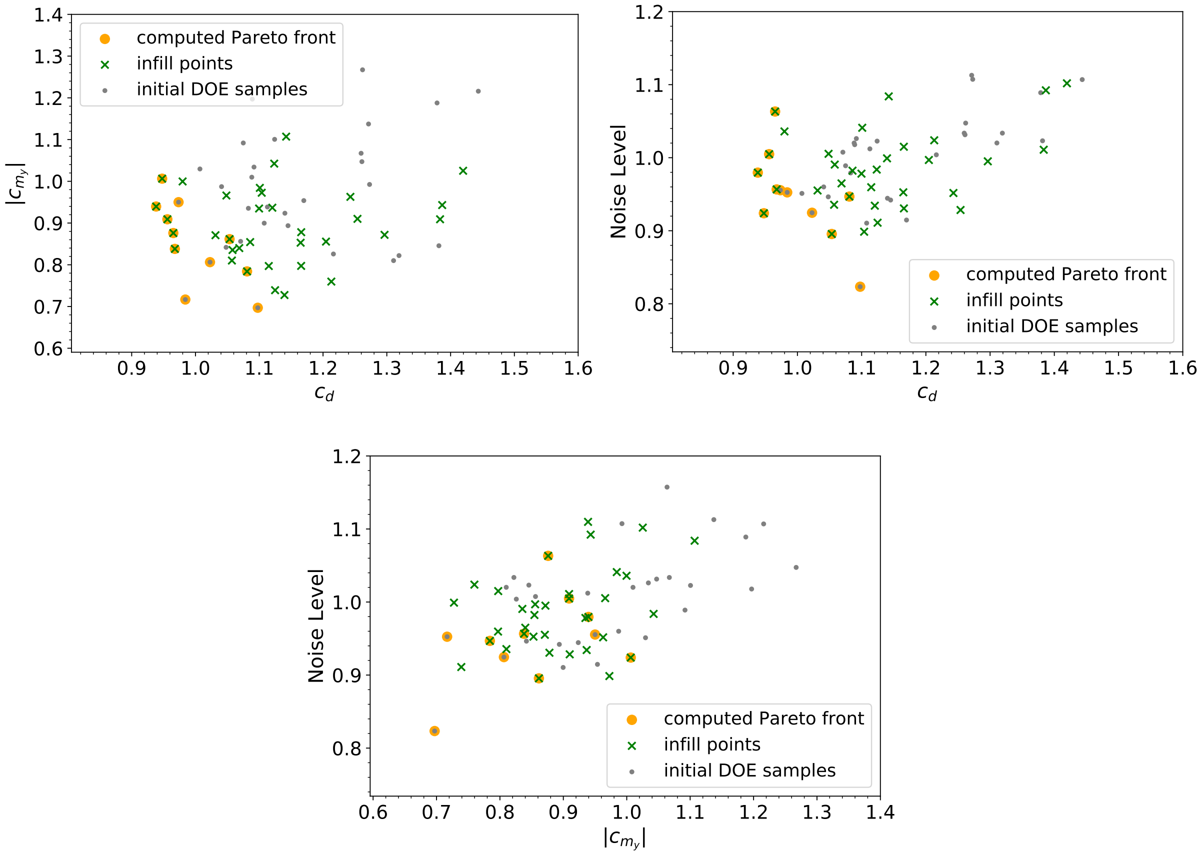


Figure 41: 2D projection of the 3D Pareto front

Without knowing the optimal Pareto front, it is impossible to check whether the computed Pareto front has been fully converged. Based on the iteration count, it shows that the solution with the minimum c_d value has been found during the last iteration, which could imply that it is possibly has not reached the full convergence. On the other hand, this point, which can be seen from the first plot in Fig. 41, is in the region where the optimizer seemingly has been started to exploit. Hence, without further evaluations with increased number of the infill budget, it is difficult to judge whether the computed Pareto front is optimal or not. Even after additional evaluations with the increased infill budget, it could still be difficult to verify its full convergence as seen from the previous results in the aerodynamic optimization, where seemingly converged distribution of the Pareto front could be the sign of premature exploitation and convergence due to some other factors such as lack of exploration due to not enough size of the initial design of experiments samples.

Nevertheless, with the given amount of the computational budget, it was possible to obtain some of the promising airfoil designs from the computed Pareto front. Out of all solutions present in the Pareto front, there were four designs with a reduction in all three objectives compared to the baseline airfoil. Results are summarized for these four designs in Table 7.

Table 7: Four designs from the Pareto front that resulted in reduction in all three objectives

Case	ΔC_d	$\Delta \lvert C_{m_y} \rvert$	Noise level (change in %)
1	- 1.59 %	- 28.33 %	- 4.75 %
2	- 2.65 %	- 4.99 %	- 4.45 %
3	- 3.23 %	- 16.2 %	- 4.34 %
4	- 6.17 %	- 6.04 %	- 2.04 %

Next, few of the selected Pareto front solutions with the lowest noise level as well as a compromising objectives are evaluated in more detail.

Fig. 42 shows the airfoil shape deformation and corresponding change in the C_p distribution as well as its location from the Pareto front. This case is the one with the minimum noise level of 36.44 [dB], which resulted in the reduction of 17.66 $\%$ compared to the baseline airfoil. Unfortunately, but as expected, this has been paid off by increase in the drag which resulted in 9.77 $\%$ increase compared to the baseline airfoil. Still, it achieved reduction of 30.3 $\%$ in the absolute pitching moment.

Fig. 43 shows the corresponding results of the case that resulted in one of the lowest noise levels (second lowest) of 39.6 [dB], with the noise reduction of 10.45%. Here, the drag was also reduced by 5.35 %, though it was compensated by increase in the pitching moment by 30.29 %.

Lastly, one of the compromising solution from the Pareto front can be closely examined, as this is often an interesting solution in multi-objective optimization. Fig. 44 shows one of the compromising solution between the three objectives. This design achieved reduction in acoustic noise level by 7.61 % and reduction in the drag coefficient by 5.22 %. Unfortunately but as expected, these reductions were compensated by increase in the pitching moment coefficient $|c_{m_y}|$ by 0.63 %, which is, in fact, relatively small. In this way, without penalizing one objective too much, it can obtain a relatively adequate design that has been improved in most of the objectives by reasonable amount.

As it can be seen from all the various design solutions that the Pareto front provides, the final design choice depends on the particular interest of each problem. For instance, if the desired goal is to achieve reduction in all three objectives, one of the designs from the four possible solutions is possible as shown. Otherwise, if the main interest is the optimization of a particular objective, this choice can also be made with the drawback of some increase in the other objectives.

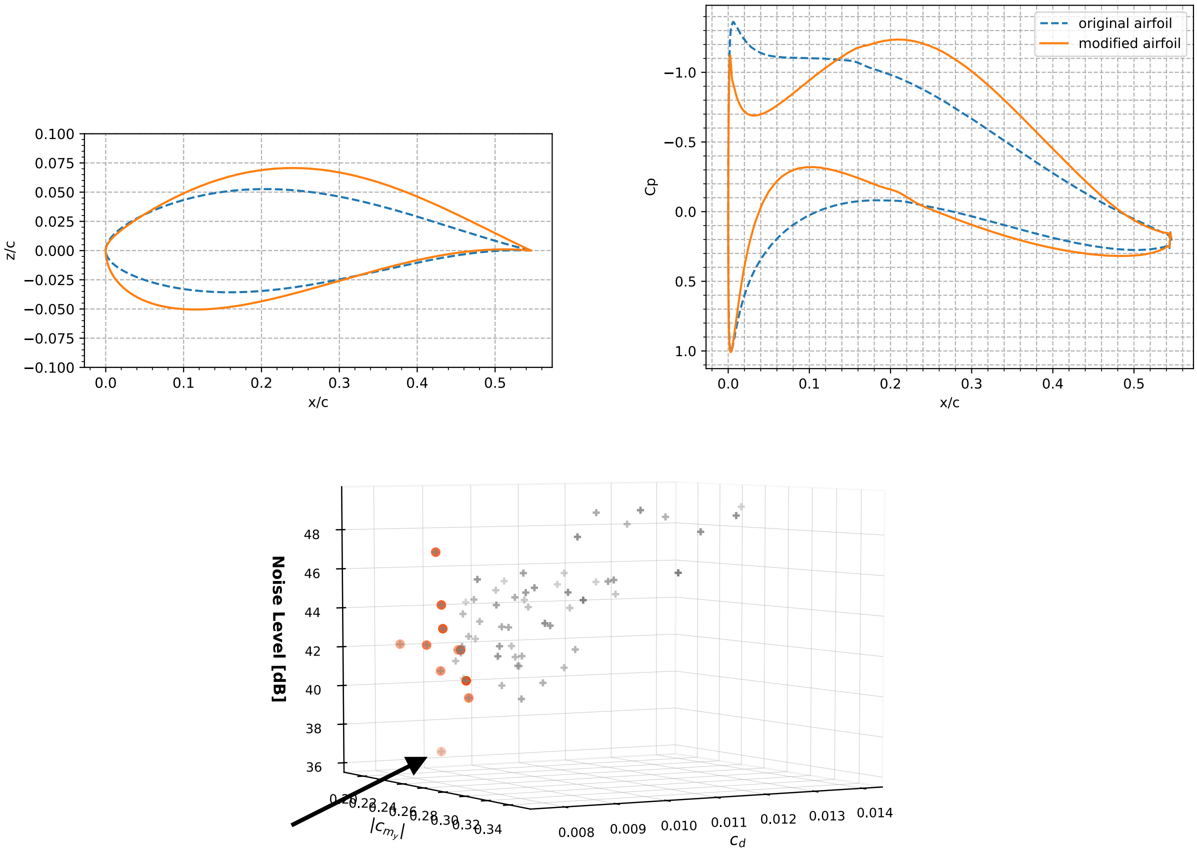


Figure 42: Airfoil design with the lowest noise level. Airfoil shape (top left), c_p distribution (top right) and the location of the solution point within the Pareto front (bottom).

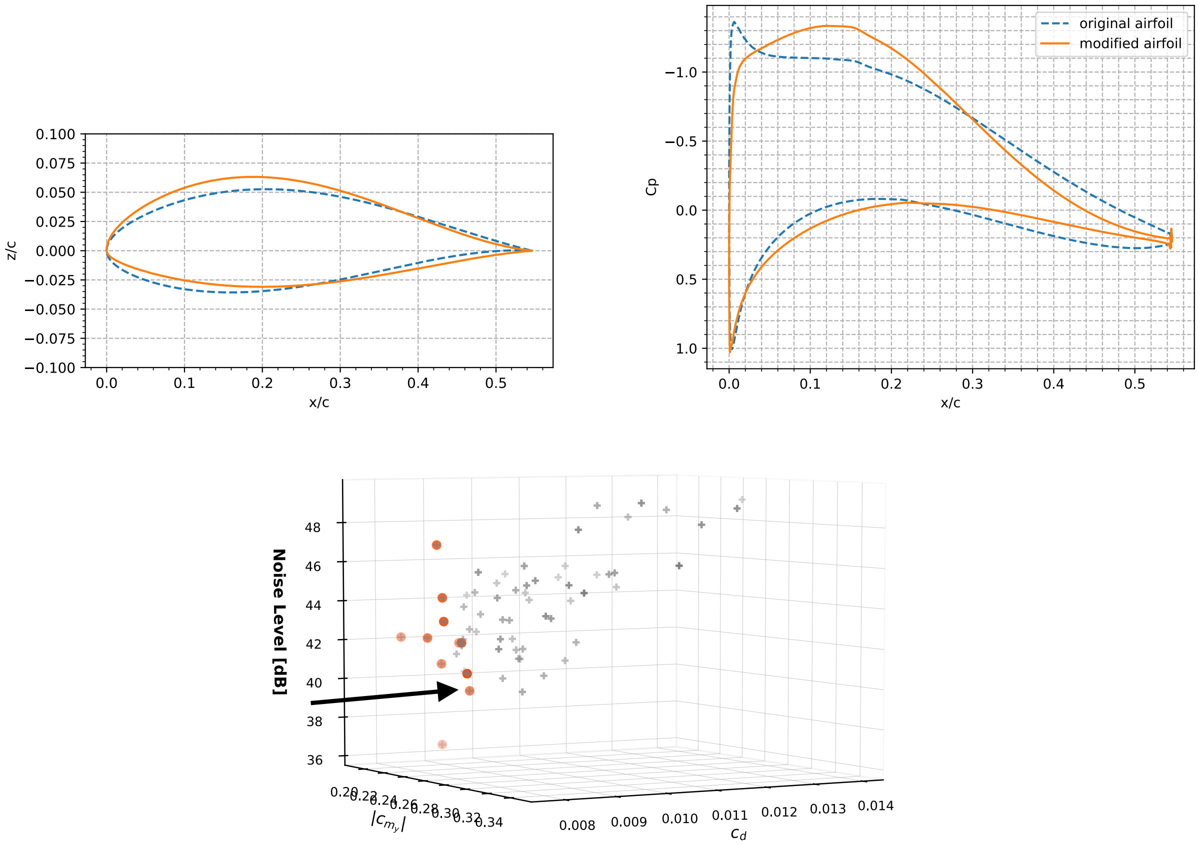


Figure 43: Airfoil design with the second lowest noise level. Airfoil shape (top left), c_p distribution (top right) and the location of the solution point within the Pareto front (bottom).

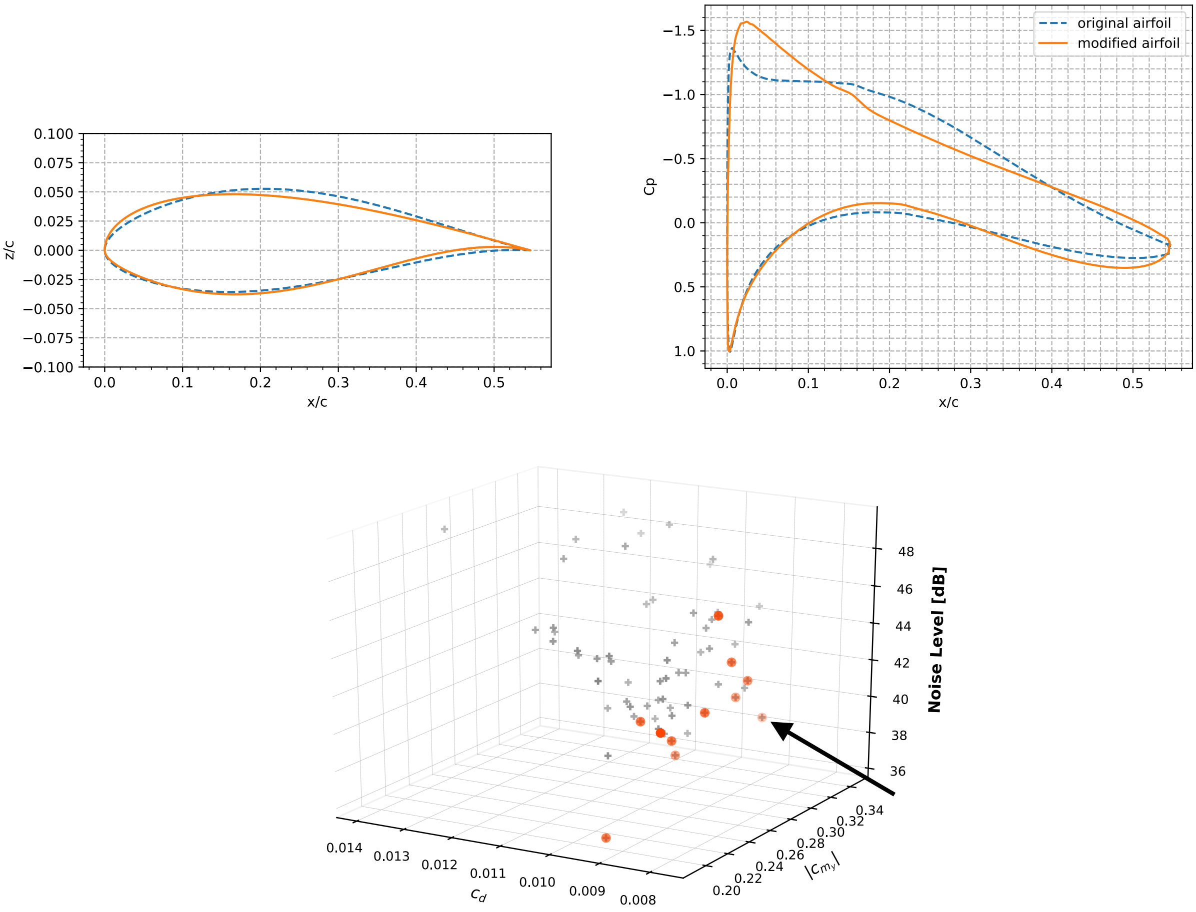


Figure 44: Airfoil design with a compromise in three objectives. Airfoil shape (top left), c_p distribution (top right) and the location of the solution point within the Pareto front (bottom)

7 Conclusion and Outlook

Throughout the thesis, the efficiency as well as the efficacy of multi-objective surrogate based optimization have been shown by obtaining Pareto fronts with relatively small computational budget. This has been particularly demonstrated through analytical problems, where the result and the number of function evaluations were compared to some of the conventional methods such as genetic algorithms. For the case of the Poloni's test function, it needed 6 to 25 times less function evaluations compared to the evolutionary based algorithms. For the Fonseca-Fleming function, it required 51 times less function evaluations to generate the Pareto front of comparable quality. Lastly for the 3D Viennet function, it used about 90 times less function evaluations. Given that the computational time for evaluating a function in applied problems ranged from 10 minutes to 4 hours, obtaining comparable quality Pareto fronts with conventional multi-objective methods can take up to 1200 days, which is simply infeasible for most of the applications with a limited time budget. Furthermore, its applicability to more complex problems with actual black-box functions has been validated through applied problems. Here, it was able to obtain the Pareto front, which contains the various design trade-offs that resulted in improved performance in terms of the desired objectives.

For the design of experiments, mainly two methods were used, namely Sobol and LHS. Although neither one of them clearly outperformed the other, for applied problems, the Sobol method provided convenience of being a deterministic method such that an increase in the number of initial samples was straightforward. Besides, the random feature of the LHS sometimes resulted in somewhat irregular Pareto fronts, which is inevitable when a stochastic approach is involved. Therefore, if one prefers the LHS method, several consecutive optimization runs are recommended to filter out these potential variations.

As a surrogate model, Kriging has been used throughout the work, and its capability of being able to mimic black box functions with only a few number of samples was demonstrated during the initial surrogate error analysis for the Poloni's test function, where the errors of the model were reduced remarkably with just around ten initial samples. Furthermore, owing to Kriging's ability to estimate uncertainty quantification, it was able to use an efficient infill criterion such as the EHVI.

The expected hypervolume improvement is shown to be an efficient infill criteria that works particularly well with Gaussian process models. However, its potential weakness in providing a full distribution has been shown compared to the non-dominated sorting of a genetic algorithm during the 3D Viennet function analysis, where it struggled to capture the full Pareto front in regions where the Pareto front has a highly non-linear shape. This was possibly due to the fact that having a uniformly distributed Pareto front does not add any advantage in terms of hypervolume improvement, whereas uniformity in the Pareto front is considered during the ranking process of the NSGA-II, which ensures a certain uniformity within the distribution. Hence, if the resulting Pareto front consists of two regions, a dense region which clearly shows that the optimizer has been exploited certain areas only compared to other regions, and a spares region with much less populated solutions, it would be suggested to solve the corresponding optimization problems by other means. To further improve its efficiency, its gradient in objective space could be utilized. For instance, [29] shows that by computing the gradients of the EHVI, gradient ascent methods can be also used. In addition, this study suggests that this gradient information can also be used as a convergence criteria. Similarly, [46] shows improved performance of the gradient assisted optimization using EHVI.

The reference point is given as a user input for computing the expected hypervolume improvement. Although this is not expected to effect the results once it is fully dominated, to safely converge to the full Pareto set including the extreme points, a large enough value was suggested to use. For more rapid convergence of the Pareto front, it should be not too large to restrict the search space. Nonetheless, further analysis and tests could be performed to clarify its effect on the optimization process. One suggestion would be to use the reference point as an adaptive value such that it readjusts its value depending on the current Pareto front. In this way, it would be possible to enhance convergence while keeping the rate of convergence at its optimum.

To improve applicability, penalty values were introduced such that the computational budget used for evaluating objective functions that resulted in invalid results, e.g. target C_l was not met, are not wasted. Instead, the surrogate model as well as the optimizer learn the infeasible or invalid regions within the design space. Although in-depth analysis is not carried out due to the time constraint, it enabled an optimizer to progress without wasting its resources and showed its capability to obtain the Pareto front which contained the promising solutions as it was seen in the two applied problems. An improvement can be made for penalty values such that its value can be changed in an adaptive way. For instance, the current implementation takes the penalty value based on the initial design of experiments, which means that in case of poor initial sampling, some of the critically worse values might not be included in this initial sampling, rather it could be found and added to the sampling during the optimization process. Thus, if the penalty value re-adapts itself, any infill value can be also taken into account. Additionally, further analysis could be completed to fine-tune the penalty levels.

During the aerodynamic optimization, it has been seen that larger bounds of the design variables allows better exploration of the design space such that it can introduce more variety in the design options, however, this resulted in slower convergence compared to the smaller bounds. One the one hand, in design problems, it is important to ensure certain variety within the solutions. On the other hand, for general optimization problems where the focus is more on rapid convergence, bounds should be more restricted. In the end, a good balance between the two should be found through rigorous analysis to ensure both, the variety of the designs as well as the convergence speed. Regarding the setting of the bounds, further studies could be extended to investigate the effect of bounds for each design parameter.

Within the scope of this master thesis, unconstrained approaches were developed and used. However, for further improvement, it would be possible to introduce penalty terms within the infill criteria similar to the constraint method for the single-objective optimization as introduced in [1]. Alternatively, it would be possible to construct a Lagrangian and evaluate the KKT conditions as suggested by [13].

Although it was shown through the thesis that Bayesian optimization is indeed an efficient method that can provide solutions with a fairly small number of function evaluations, further improvement could be made in a direction to increase the convergence rate of the Pareto front. However, as it was shown, when the convergence rate is improved, this usually implies the premature convergence due to lack of its exploration in design space. And, this issue will only become more pronounced with the growing number of the objectives. Looking at this problem from a different perspective, the current method is limited in that the information is propagating only from the reference point towards the non-dominated space. Therefore, the availability of any information from the non-dominated space could help treating this issue.

One promising way to further improve on this aspect would be to implement and utilize prior information given during the optimization. The only difference would then be that certain decisions need to be made in advance. In engineering problems, it is expected that there are always preferences that are known to a certain extent. For instance, during the aerodynamic shape optimization, it was stated that the full Pareto front might not always be of interest if the reduction along the pitching moment is small while its increase in drag is much larger. If this is indeed a not very interesting area, this means that the optimizer is essentially wasting computational budget for searching in this region.

Some of advanced versions of the expected hypervolume improvement have been developed to perform optimization under prescribed preferences. In [33], a new acquisition function based on expected improvement has been introduced to perform an optimization under preferences in the objectives. Here, its hypervolume calculation is weighted by the posterior probability of satisfying the preferences from a gradient Gaussian process model to either award or penalize given preference conditions. With this method, they showed that enhancement in optimization search towards a certain objective is possible, however, due to the increased complexity of the new acquisition function, it needed to be evaluated through Monte-Carlo simulation, which means that the exact computation of the acquisition function is no longer possible.

A similar technique has been applied by [40], where the hypervolume is defined by a user-defined continuous measure instead of the Lebesgue measure which was proposed by [17]. By encoding the preferences within weight functions of the acquisition function, this method can populate interesting regions of the Pareto front within much less iteration. However, similar to the previous method, sequential Monte-Carlo had to be applied to deal with the computation of the modified acquisition function.

The last aerodynamic-aeroacoustic problem was a great showcase of how important it is to obtain the converged Pareto front at a given limit of iteration where further iterations are not possible due to time constraints. Despite the fact that complex black-box functions are employed, with 80 total function evaluations, it was able to obtain the Pareto front with improved designs. Based on the outcomes of the analytical problem analysis, this could have taken a potentially infeasible amount of time to do with conventional multi-objective optimization approaches. Hence, this not only emphasizes the efficiency of Bayesian optimization, but also shows the potential for further development of this approach.

References

[1] A. Forrester, A. Sobester and A. Keane. *Engineering design via surrogate modelling: a practical guide*. Wiley, July 2008. ISBN 9780470770795.

[2] A. I. J. Forrester and A. J. Keane. *Recent advances in surrogate-based optimization*. *Progress in Aerospace Sciences*, 45(1):50–79, 2009. ISSN 0376-0421.

[3] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto and S. Tarantola. *Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index*. *Computer Physics Communications*, 181(2):259–270, 2010. ISSN 0010-4655.

[4] A. Saltelli,M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, S. Tarantola. *Experimental Designs*, chapter 2, pages 53–107. John Wiley Sons, Ltd, 2008. ISBN 9780470725184.

[5] A. Serna and C. Bucher. Advanced Surrogate Models for Multidisciplinary Design Optimization. In 6^{th} *Weimar Optimization and Stochastic Days*, 2009.

[6] B. M. Kulfan. *Universal Parametric Geometry Representation Method*. *Journal of Aircraft*, 45(1):142–158, 2008.

[7] C. A. Coello. Evolutionary multi-objective optimization: Basic concepts and some applications in pattern recognition. In *Pattern Recognition*, pages 22–33, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-21587-2.

[8] C. M. Fonseca and P. J. Fleming. *An Overview of Evolutionary Algorithms in Multiobjective Optimization*. *Evolutionary Computation*, 3(1):1–16, 03 1995. ISSN 1063-6560. doi: 10.1162/evco.1995.3.1.1. URL `https://doi.org/10.1162/evco.1995.3.1.1`.

[9] C. Poloni, A. Giurgevich, L. Onesti and V. Pediroda. *Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design problem in fluid dynamics*. *Computer Methods in Applied Mechanics and Engineering*, 186:403–420, 06 2000. doi: 10.1016/S0045-7825(99)00394-1.

[10] C. Sabater, and S. Görtz. *Gradient-Based Aerodynamic Robust Optimization Using the Adjoint Method and Gaussian Processes*, pages 211–226. Springer International Publishing, Cham, 2021. ISBN 978-3-030-57422-2. doi: 10.1007/978-3-030-57422-2_14.

[11] C. Sabater, P. Bekemeyer and S. Görtz. *Efficient Bilevel Surrogate Approach for Optimization Under Uncertainty of Shock Control Bumps*. *AIAA Journal*, 58(12):5228–5242, 2020.

[12] D. Jones, M. Schonlau and W. Welch. *Efficient Global Optimization of Expensive Black-Box Functions*. *Journal of Global Optimization*, 13:455–492, 12 1998. doi: 10.1023/A:1008306431147.

[13] D. Khatamsaz, L. Peddareddygari, S. Friedman and D. Allaire. *Bayesian Optimization of Multiobjective Functions Using Multiple Information Sources*. *AIAA Journal*, 59, 03 2021.

[14] E. Andrés-Pérez, L. Carro-Calvo, S. Salcedo-Sanz and M. J. Martin-Burgos. *Aerodynamic Shape Design by Evolutionary Optimization and Support Vector Machines*, pages 1–24. Springer International Publishing, Cham, 2016. ISBN 978-3-319-21506-8. doi: 10.1007/978-3-319-21506-8_1. URL `https://doi.org/10.1007/978-3-319-21506-8_1`.

[15] E. Iuliano. *Adaptive Sampling Strategies for Surrogate-Based Aerodynamic Optimization*, pages 25–46. Springer International Publishing, Cham, 2016. ISBN 978-3-319-21506-8. doi: 10.1007/978-3-319-21506-8_2. URL `https://doi.org/10.1007/978-3-319-21506-8_2`.

[16] E. Zitzler and L. Thiele. *Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach*. *IEEE Trans. Evol. Comput.*, 3:257–271, 1999.

[17] E. Zitzler, D. Brockhoff and L. Thiele. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *Evolutionary Multi-Criterion Optimization*, pages 862–876, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[18] E. Zitzler, J. D. Knowles and L. Thiele. Quality assessment of pareto set approximations. In *Multiobjective Optimization*, 2008.

[19] F. Freschi and M. Repetto. Multiobjective optimization by a modified artificial immune system algorithm. In P. J. Bentley C. Jacob, M. L. Pilat and J. I. Timmis, editors, *Artificial Immune Systems*, pages 248–261, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31875-0.

[20] G. Zhao, R. Arroyave and X. Qian. Fast exact computation of expected hypervolume improvement, 2019.

[21] H. Ishibuchi, R. Imada, Y. Setoguchi and Y. Nojima. Reference point specification in hypervolume calculation for fair comparison and efficient search. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’17, pages 585–592, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349208. doi: 10.1145/3071178.3071264.

[22] I. Hupkens, A. Deutz, K. Yang, and M. Emmerich. Faster exact algorithms for computing expected hypervolume improvement. In *Evolutionary Multi-Criterion Optimization*, pages 65–79, Cham, 2015. Springer International Publishing.

[23] J. L. Loeppky, J. Sacks and W. J. Welch. *Choosing the Sample Size of a Computer Experiment: A Practical Guide*. *Technometrics*, 51(4):366–376, 2009.

[24] J. Močkus. On bayesian methods for seeking the extremum. In G. I. Marchuk, editor, *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, pages 400–404, Berlin, Heidelberg, 1975. Springer Berlin Heidelberg. ISBN 978-3-540-37497-8.

[25] J. Sacks, W. J. Welch, T. J. Mitchell and H. P. Wynn. *Design and Analysis of Computer Experiments*. *Statistical Science*, 4(4):409–423, 1989. ISSN 08834237.

[26] J.K. Patel and C.B. Read. *Handbook of the Normal Distribution, Second Edition*. Statistics: A Series of Textbooks and Monographs. Taylor & Francis, 1996. ISBN 9780824793425.

[27] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley amp; Sons, Inc., USA, 2001. ISBN 047187339X.

[28] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197, 2002. doi: 10.1109/4235.996017.

[29] K. Yang, M. Emmerich, A. Deutz and T. Bäck. *Multi-Objective Bayesian Global Optimization using expected hypervolume improvement gradient*. *Swarm and Evolutionary Computation*, 44:945–956, 2019.

[30] K. Yang, M. Emmerich, A. H. Deutz and T. Bäck. *Efficient Computation of Expected Hypervolume Improvement Using Box Decomposition Algorithms*. *Journal of Global Optimization*, 75(1):3–34, 2019. doi: 10.1007/s10898-019-00798-7.

[31] L. Brevault, M. Balesdent and A. Hebbal. *Multi-Objective Multidisciplinary Design Optimization Approach for Partially Reusable Launch Vehicle Design*. *Journal of Spacecraft and Rockets*, 57(2):373–390, 2020. doi: 10.2514/1.A34601.

[32] L. While, L. Bradstreet and L. Barone. *A Fast Way of Calculating Exact Hypervolumes*. *IEEE Transactions on Evolutionary Computation*, 16(1):86–95, 2012. doi: 10.1109/TEVC.2010.2077298.

[33] M. Abdolshah, A. Shilton, S. Rana, S. Gupta and S. Venkatesh. Multi-objective bayesian optimisation with preferences over objectives. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/a7b7e4b27722574c611fe91476a50238-Paper.pdf`.

[34] M. Emmerich. *Single- and multi-objective evolutionary design optimization assisted by gaussian random field metamodels*. PhD thesis, 12 2005.

[35] M. Emmerich, K. Yang, A. Deutz, H. Wang and C. Fonseca. *A Multicriteria Generalization of Bayesian Global Optimization*, volume 107. 11 2015. ISBN 978-3-319-29973-0. doi: 10.1007/978-3-319-29975-4_12.

[36] M. Fleischer. The measure of pareto optima. applications to multi-objective metaheuristics. In *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 519–533. Springer, 2003.

[37] M. J. W. Riley, T. Peachey, D. Abramson and K. W. Jenkins. *Multi-objective engineering shape optimization using differential evolution interfaced to the Nimrod/O tool*. 10:012189, jun 2010. doi: 10.1088/1757-899x/10/1/012189. URL `https://doi.org/10.1088/1757-899x/10/1/012189`.

[38] N. V. Queipo, R. T. Haftka, W. Shyy and Tushar Goel and Rajkumar Vaidyanathan and P. Kevin Tucker. Surrogate-based Analysis and Optimization. *Progress in Aerospace Sciences*, 41:1–28, 2005.

[39] P. Feliot, J. Bect and E. Vazquez. A Bayesian approach to constrained single- and multi-objective optimization. *Journal of Global Optimization*, 67(1):97–133, 01 2017. doi: 10.1007/s10898-016-0427-3.

[40] P. Feliot, J. Bect and E. Vazquez. User preferences in Bayesian multi-objective optimization: the expected weighted hypervolume improvement criterion. In Giuseppe Nicosia, Panos Pardalos, Giovanni Giuffrida, Renato Umeton, and Vincenzo Sciacca, editors, *Machine Learning, Optimization, and Data Science. 4th International Conference (LOD 2018)*, volume Lecture Notes in Computer Science, vol 11331 of *Machine Learning, Optimization, and Data Science. 4th International Conference, LOD 2018, Volterra, Italy, September 13–16, 2018, Revised Selected Papers*, pages 533–544, Volterra, Italy, 09 2018. Springer, Cham.

[41] R. Ewert, C. Appel, J. Dierke and M. Herr. *RANS/CAA Based Prediction of NACA 0012 Broadband Trailing Edge Noise and Experimental Validation*.

[42] R. Lacour, K. Klamroth and C. M. Fonseca. *A Box Decomposition Algorithm to Compute the Hypervolume Indicator*. *Computers Operations Research*, 79:347–360, 07 2017.

[43] R. Marler and J. Arora. *The weighted sum method for multi-objective optimization: New insights*. *Structural and Multidisciplinary Optimization*, 41:853–862, 06 2010.

[44] R. Viennet, C. Fonteix and I. Marc. New multicriteria optimization method based on the use of a diploid genetic algorithm: Example of an industrial problem. In *Artificial Evolution European Conference*, pages 120–127. Springer-Verlag, 1995.

[45] S. Burhenne, D. Jacob and G. P. Henze. Sampling based on sobol′ sequences for monte carlo techniques applied to building simulations. pages 1816–1823, 01 2011.

[46] S. Daulton, M. Balandat and E. Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. In *34th Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, 2020.

[47] Y. Haimes, L. Lasdon and D. Wismer. *On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization*. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(3):296–297, 1971. doi: 10.1109/TSMC.1971.4308298.

DLR-IB-AS-BS-2021-251

Global Multi-Objective Optimisation utilizing Surrogate Models

Sihyeong Lim

<u>Verteiler:</u>

Institutsbibliothek	1 Exemplar
Verfasser/Co-Autoren	4 Exemplare
Institutsleitung	1 Exemplar
Abteilungsleiter	1 Exemplar
Deutsche Bibliothek in Frankfurt/Main	2 Exemplare
Niedersächsische Landesbibliothek Hannover	1 Exemplar
Techn. Informationsbibliothek Hannover	1 Exemplar
Zentralbibliothek BS	1 Exemplare
Zentralarchiv GÖ	1 Exemplar