



Universiteit  
Leiden  
The Netherlands

## Multi-objective Bayesian global optimization for continuous problems and applications

Yang, K.

### Citation

Yang, K. (2017, December 6). *Multi-objective Bayesian global optimization for continuous problems and applications*. Retrieved from <https://hdl.handle.net/1887/57791>

Version: Not Applicable (or Unknown)

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/57791>

**Note:** To cite this publication please use the final published version (if applicable).

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/57791> holds various files of this Leiden University dissertation

**Author:** Yang, Kaifeng

**Title:** Multi-objective Bayesian global optimization for continuous problems and applications

**Date:** 2017-12-06

# Multi-Objective Bayesian Global Optimization for Continuous Problems and Applications

Kaifeng Yang



# Multi-Objective Bayesian Global Optimization for Continuous Problems and Applications

**Proefschrift**

ter verkrijging van  
de graad van Doctor aan de Universiteit Leiden,  
op gezag van Rector Magnificus prof.mr. C.J.J.M. Stolker,  
volgens besluit van het College voor Promoties  
te verdedigen op woensdag 6 december 2017  
klokke 11.15 uur

door

**Kaifeng Yang**

geboren te Shaanxi, China  
in 1987

## Promotiecommissie

Promotor: Prof. Dr. T.H.W. Bäck  
Co-promotor: Dr. M.T.M. Emmerich  
Overige leden: Prof. Dr. A. Plaat  
Prof. Dr. H. Hoos  
Prof. Dr. T. Dhaene (Ghent University, Belgium)  
Prof. Dr. Z. Cai (Central South University, China)  
Dr. I. Yevseyeva (De Montfort University, UK)

Copyright © 2017 Kaifeng Yang All Rights Reserved

ISBN: 978-94-6299-801-8

Het onderzoek beschreven in dit proefschrift is uitgevoerd aan het Leiden Institute of Advanced Computer Science (LIACS, Universiteit Leiden).

This research is financially supported by the China Scholarship Council (CSC), Grant No. 201306370037.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Research Questions . . . . .	2
1.3	Dissertation Outline . . . . .	4
<b>2</b>	<b>Continuous Multi-objective Optimization</b>	<b>7</b>
2.1	Multi-objective Optimization . . . . .	8
2.2	Terminologies . . . . .	9
2.3	Infill Criteria . . . . .	10
2.4	Evolutionary Multi-objective Optimization Algorithms . . . . .	14
2.4.1	NSGA-II . . . . .	14
2.4.2	SMS-EMOA . . . . .	14
2.4.3	Example . . . . .	15
2.5	Multi-objective Bayesian Global Optimization . . . . .	19
2.5.1	Kriging . . . . .	20
2.5.2	Structure of MOBGO . . . . .	21
2.5.3	Example . . . . .	23
2.6	Summary . . . . .	25
<b>3</b>	<b>Efficient EHVI Calculation</b>	<b>26</b>
3.1	EHVI Definition . . . . .	27
3.2	State-of-the-art . . . . .	29
3.3	Non-dominated Space Partitioning Algorithm . . . . .	30
3.3.1	Low Dimensional case . . . . .	30
3.3.2	High Dimensional case . . . . .	36
3.4	Computing the integrals . . . . .	38
3.4.1	2-D EHVI . . . . .	39
3.4.2	3-D EHVI . . . . .	40
3.4.3	High Dimensional Case . . . . .	42

## CONTENTS

---

3.5	Other Related Criterion . . . . .	46
3.6	Empirical Experiments . . . . .	47
3.6.1	Speed Comparison . . . . .	47
3.6.2	Benchmark Performance . . . . .	51
3.7	Summary . . . . .	52
<b>4</b>	<b>TEHVI Calculation</b>	<b>55</b>
4.1	Motivations . . . . .	56
4.2	TEHVI Definition . . . . .	57
4.2.1	Formula Derivation . . . . .	58
4.2.2	Computational Speed Test . . . . .	60
4.3	Experimental Setup . . . . .	62
4.4	Empirical Results . . . . .	63
4.5	Summary . . . . .	65
<b>5</b>	<b>Preference-Based Multi-Objective Optimization</b>	<b>67</b>
5.1	Background . . . . .	68
5.2	Algorithms . . . . .	69
5.2.1	TEHVI-EGO for Preference-based Multi-Objective Opti- mization . . . . .	69
5.2.2	Preferred region with EAs . . . . .	70
5.3	Empirical Experiments . . . . .	75
5.3.1	TEHVI assisted EGO . . . . .	75
5.3.2	Preferred region based on EAs . . . . .	79
5.4	Summary . . . . .	83
<b>6</b>	<b>EHVI Gradient Calculation</b>	<b>86</b>
6.1	Motivations . . . . .	87
6.2	Expected Hypervolume Improvement Gradient (EHVIG) . . . . .	87
6.3	Gradient Ascent Algorithm . . . . .	90
6.4	Stopping Criterion – EHVIG . . . . .	91
6.5	Experimental Results . . . . .	92
6.6	Summary . . . . .	96
<b>7</b>	<b>Applications</b>	<b>97</b>
7.1	Backgrounds . . . . .	97
7.2	Problem Definition . . . . .	99
7.2.1	PID Parameter Tuning . . . . .	99
7.2.2	Robust PID Tuning . . . . .	100

## CONTENTS

---

7.2.3	Bio-gas Plant Optimization . . . . .	101
7.3	Experimental Settings . . . . .	104
7.4	Experimental Results . . . . .	105
7.4.1	PID Parameter Tuning . . . . .	105
7.4.2	Robust PID Tuning . . . . .	108
7.4.3	Bio-plant Optimization . . . . .	111
7.5	Summary . . . . .	116
<b>8</b>	<b>Conclusions and Outlooks</b>	<b>117</b>
8.1	Conclusions . . . . .	118
8.2	Outlooks . . . . .	119
	<b>Appendix</b>	<b>121</b>
A.1	Symbols . . . . .	121
A.2	Abbreviations . . . . .	123
A.3	EHVIG Formula Derivation . . . . .	125
A.4	2-D EHVI Formula (Minimization Case) . . . . .	127
	<b>Bibliography</b>	<b>129</b>
	<b>English Summary</b>	<b>142</b>
	<b>Nederlandse Samenvatting</b>	<b>144</b>
	<b>Curriculum Vitae</b>	<b>146</b>

## CONTENTS

---

# Chapter 1

## Introduction

### 1.1 Background

Optimization is a process of searching for the best solution from a set of available solutions. Single-objective optimization and multi-objective optimization are two main branches. They are distinguished with respect to the number of considered objective functions. Specifically, single-objective optimization considers one objective function, while multi-objective optimization (MOO) involves more than one objective function simultaneously.<sup>1</sup> An easy way to solve a multi-objective optimization problem is to convert it into single-objective optimization, by forming a weighted sum of all objective functions. This method is simple but not effective when the objectives are conflict. This is the reason why researchers are interested in treating each objective separately and using the Pareto front concept for optimization.

Based on evolutionary algorithms, many evolutionary multi-objective optimization methodologies have been proposed over the past several decades in order to find an efficient approximation of the Pareto front. However, evolutionary multi-objective optimization (EMO) is inefficient when dealing with expensive function evaluation problems, because EMO usually needs more than ten thousand function evaluations and such a large number of function evaluations are unrealistic to be applied in many practical applications.

---

<sup>1</sup>In some papers, multi-objective optimization means the number of the objective functions is 2 or 3, and many-objective optimization is used to indicate more than 3 objective functions. Typically, optimization problems are defined by means of one or more objective function(s).

## 1. INTRODUCTION

---

A common remedy to this problem is Multi-objective Bayesian global optimization (MOBGO), which partially replaces exact objective function evaluations by using predictions from the so-called Kriging or Gaussian process models. These surrogate models provide a predictive distribution, consisting of a mean value and a standard deviation of each objective function.<sup>2</sup> The scheme of a MOBGO involves sequentially updating the surrogate models by a predicted optimum and its corresponding objective function value. An optimizer is utilized in MOBGO in order to search for a predicted optimum, which is the next point for evaluation, according to the so-called infill criterion.

In practical applications, MOBGO is actually not widely utilized. This is because MOBGO requires a lot of execution time, even though it only requires far fewer evaluations of the objective functions. Three main aspects limit the efficiency of MOBGO:

1. Updating the surrogate models is very expensive;
2. Computational complexity of an infill criterion is very high;
3. The optimizer for searching an optimal solution from the surrogate models is not sufficiently effective.

To improve the efficiency and effectiveness of MOBGO, many researchers have attempted to overcome the drawbacks mentioned above. In order to update the surrogate models more efficiently, some researchers reduced the sampling data by using a clustering method. In this dissertation, the central research questions are surrounding the second and the third aspects to improve the performance of MOBGO.

## 1.2 Research Questions

**The first research question of this work is how to improve the efficiency of exact EHVI calculation.** Since an infill criterion plays an important selecting role in MOBGO and values the performance of the Pareto front approximation, it is essential to find an effective and efficient infill criterion in MOBGO. Some common infill criteria are *Hypervolume* (HV), *Hypervolume Improvement* (HVI), *Probability of Improvement* (PoI), *Expected Hypervolume Improvement* (EHVI). Compared to other criteria, EHVI takes the predictive mean value and

---

<sup>2</sup>The hypothesis of MOBGO is that objective functions are independent.

standard deviation into account and can balance exploitation<sup>1</sup> and exploration<sup>1</sup> well, which are two important aspects in optimization. Nevertheless, EHVI is mainly utilized in scientific research but seldom applied in real applications, which is caused by its high computational complexity. In order to solve this problem, an efficient EHVI calculation algorithm is proposed in this dissertation.

**The second research question of this work is how to improve the effectiveness of MOBGO, by taking a-priori knowledge of the objective functions into consideration.** The definition of EHVI is based on the concept of a normal distribution, with the assumption that an objective function value is a real number, from minus infinity to infinity. The EHVI assumes an unbounded objective space even if it is often known a-priori that the objective function values are within a prescribed range. In some cases, a-priori knowledge of the range of an objective function value is already available. For instance, in a PID<sup>1</sup> parameter tuning problem, the rising time is always a positive value. In these cases, it is assumed that surrogate-model based algorithms could converge to true Pareto front faster, if the range of the objective functions could be applied during the optimization. To take advantage of such a-priori knowledge, a new criterion called *Truncated Expected Hypervolume Improvement* (TEHVI) is proposed in this dissertation.

**The third research question of this work is how to solve the preference-based Pareto front problems.** In a practical application, what interests a decision maker (DM) is not the entire Pareto front approximation set, but how to find more solutions which can best match his/her preferences. TEHVI is capable of solving the preference-based Pareto front problems by setting the domain of the truncated normal distributions according to a decision maker's preference. Inspired by the concept of TEHVI, *Truncated Hypervolume* is also applied to solve this problem.

**The last research question of this work is how to improve the efficiency of the optimizer in MOBGO.** An optimizer searches for the optimal solution according to an infill criterion, which is based on the predictions of the surrogate models. Theoretically, any single-objective optimization algorithm can be applied as the optimizer in MOBGO. Usually, some state-of-the-art single-objective evolutionary algorithms are chosen for the optimizer, such as genetic algorithm (GA) and covariance matrix adaptation evolution strategy (CMA-ES). However,

---

<sup>1</sup>Exploitation means to search a limited but promising region in the search space.

<sup>1</sup>Exploration means to search a much larger region of the search space, with the hope of finding other promising solutions.

<sup>1</sup>PID controller is short for the proportional-integral-derivative controller.

## 1. INTRODUCTION

---

EAs requires a large number of function evaluations of the infill criterion, in order to find the optimal solution and update the surrogate models. Because of this reason, MOBGO still requires much more execution time than evolutionary multi-objective optimization algorithms (EMOAs), even though MOBGO needs far fewer evaluations than EMOAs. To improve the efficiency of MOBGO, a new criterion called *Expected Hypervolume Improvement Gradient* (EHVIG) is utilized in an optimizer.

### 1.3 Dissertation Outline

The outline of this dissertation is described in this section. Each chapter of this dissertation is based on at least one publication of the author. The following provides a brief outline of each chapter.

- Chapter 2 lays out the definition of the multi-objective optimization problem and the terminologies used in this dissertation. A brief introduction of evolutionary multi-objective algorithms (EMOAs) is provided. These EMOAs are illustrated by a real application problem. Moreover, a brief introduction of Bayesian Global Optimization is also described, including a brief example to illustrate how MOBGO works. Parts of the definitions are previously published in [1, 2, 3]:

Hupkens, I., Deutz, A., **Yang, K.**, Emmerich, M. (2015). Faster Exact Algorithms for Computing Expected Hypervolume Improvement. In: Gaspar-Cunha A., Henggeler Antunes C., Coello C. (Eds.), Evolutionary Multi-Criterion Optimization. EMO 2015. Lecture Notes in Computer Science, vol 9019. Springer, pp. 65-79, Cham.

Emerich, M., **Yang, K.**, Deutz, A., Wang, H., Fonseca, M. (2016). Multicriteria generalization of Bayesian global optimization. In: Pardalos, P., Zhigljavsky, A., Žilinskas, J. (Eds.), Advances in Stochastic and Global Optimization. Springer, pp. 223-236.

**Yang, K.**, Emmerich, M.T.M., Li, R., Wang, J., Bäck, T. (2014). Power Distribution Network Reconfiguration by Evolutionary Integer Programming. In: Bartz-Beielstein T., Branke J., Filipič B., Smith J. (Eds.), Parallel Problem Solving from Nature–PPSN XIII. PPSN 2014. Lecture Notes in Computer Science, vol 8672, pp. 11-23. Springer, Cham.

- Chapter 3 defines what is *Expected Hypervolume Improvement* (EHVI) and how to calculate EHVI efficiently. The computational complexity of EHVI is analyzed, together with a comparison between the performance of EHVI and other infill criteria. Parts of this chapter are published in the following articles [4, 5]:

**Yang, K.**, Emmerich, M., Deutz, A., Fonseca, C.M. (2017). Computing 3-D Expected Hypervolume Improvement and Related Integrals in Asymptotically Optimal Time. In: Trautmann H. et al. (Eds.), *Evolutionary Multi-Criterion Optimization. EMO 2017. Lecture Notes in Computer Science*, vol 10173. Springer, pp. 685-700, Cham.

**Yang, K.**, Deutz, A., Fonseca, C.M., Bäck, T., Emmerich, M. (2017). Efficient exact computation of expected hypervolume improvement in Bayesian global optimization. *Journal of Global Optimization*, Submitted.

- Chapter 4 describes the definition and the exact calculation method of *Truncated Expected Hypervolume Improvement* (TEHVI). TEHVI is derived from the definition of EHVI, which utilizes the a-priori knowledge of the objective functions, in order to improve the efficiency of MOBGO, by means of the conditional distribution. Part of this chapter is published in the following article [6]:

**Yang, K.**, Deutz, A., Yang, Z., Bäck, T., Emmerich, M. (2016). Truncated expected hypervolume improvement: Exact computation and application. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 4350-4357, IEEE.

- Chapter 5 introduces preference-based multi-objective optimization. This chapter aims at finding a more fine-grained resolution of a preferred region, instead of exploring the whole set of Pareto front solutions. Two methods are applied in this chapter: TEHVI assisted by Bayesian global optimization and *Truncated Hypervolume* assisted by EAs. The works are previously published in [7, 8]:

**Yang, K.**, Li, L., Deutz, A., Bäck, T., Emmerich, M. (2016). Preference-based multiobjective optimization using truncated expected hypervolume improvement. In: 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), pp. 276-281, IEEE.

Wang, Y., Li, L., **Yang, K.**, Emmerich, M. (2017). A new approach to

## 1. INTRODUCTION

---

target region based multiobjective evolutionary algorithms. In: 2017 IEEE Congress on Evolutionary Computation (CEC), pp. 1757-1764, IEEE.

- Chapter 6 proposes a new infill criterion, namely, the gradient of EHVI (EHVIG) and utilizes EHVIG in multi-objective optimization by two approaches: one is applying EHVIG in gradient ascent algorithm and the other is regarding EHVIG as a stopping criterion in evolutionary algorithms to find the globally optimal solution. This work is mainly in the following paper [9]:

**Yang, K.**, Emmerich, M., Bäck, T., Deutz, A. (2017). Multi-objective Bayesian global optimization using expected hypervolume improvement gradient. Swarm and Evolutionary Computation, Submitted.

- Chapter 7 mainly concerns real-world applications of multi-objective optimization, particularly in the fields of bio-gas plant and PID parameter tuning. These works have been previously published in [6, 10]:

**Yang, K.**, Gaida, D., Bäck, T., Emmerich, M. (2015). Expected hypervolume improvement algorithm for PID controller tuning and the multiobjective dynamical control of a biogas plant. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 1934-1942, IEEE.

**Yang, K.**, Deutz, A., Yang, Z., Bäck, T., Emmerich, M. (2016). Truncated expected hypervolume improvement: Exact computation and application. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 4350-4357, IEEE.

- Chapter 8 summarizes the contribution of this dissertation and provides some suggestions for future work.
- Besides the publications mentioned above, other publications of the author is [11]:

Yang, Z., Wang, H., **Yang, K.**, Bäck, T., Emmerich, M. (2016). SMS-EMOA with multiple dynamic reference points. In: 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), pp. 282-288, IEEE.

## Chapter 2

# Continuous Multi-objective Optimization

Evolutionary algorithms (EAs) and Bayesian global optimization (BGO) are two major branches in the field of continuous optimization algorithms. Both of them share a similar structure: (1) initialization, (2) evaluation of current solutions, (3) adjustment of the current solutions for the aim of seeking an improvement in the next loop, and (4) repetition of the evaluation and adjustment loop. The difference lies in the adjustment mechanism. For EAs, it is accomplished by evolutionary operators, such as recombination and mutation. For BGO, it is achieved by learning from the past evaluations and updating a surrogate model.

To start the journey, a good preparation is always required. This chapter serves to lay out the terminologies and the groundwork of the studies in this dissertation. The structure of this chapter is structured as follows: Section 2.1 provides the definition of multi-objective optimization; Section 2.2 defines some fundamental terminologies in the field of multi-objective optimization; Section 2.3 provides the definitions of some common infill criteria; Section 2.4 describes two state-of-the-art evolutionary multi-objective optimization algorithms, namely SMS-EMOA and NSGA-II, which are utilized to solve a power distribution network reconfiguration problem in this chapter; Section 2.5 introduces multi-objective Bayesian global optimization, together with Kriging and a simple example.

## 2. CONTINUOUS MULTI-OBJECTIVE OPTIMIZATION

---

### 2.1 Multi-objective Optimization

Multi-objective optimization is a generalization of single-objective optimization. It can be generalized by means of selecting the best combination of parameters in order to optimize the multiple performances simultaneously. The basic idea of MOO is that it optimizes the performances depending on these parameters, possibly subject to some restrictions on the allowed parameter ranges. The performances of the problem which needs to be optimized are called *objective functions or fitness functions* and they depend on the combination of parameters; the parameters are called *decision variables or the decision vector*; the range of the decision vectors is known as *search space*; the restrictions on allowed parameters are called *constraints*; an allowed decision vector is called a *feasible decision vector*. A multi-objective optimization (MOO) problem is an optimization problem that involves multiple objective functions and it can be formulated as:

$$\begin{aligned} & \mathbf{max}(y_1(\mathbf{x}), y_2(\mathbf{x}), \dots, y_d(\mathbf{x})) \\ & \text{subject to} \quad \mathbf{x} \in \mathcal{X} \subseteq \mathbb{S} \end{aligned} \tag{1-1}$$

where the integer  $d$  is the number of objective functions,  $\mathcal{X}$  is the feasible set of decision vectors,  $y_i$   $i = 1, \dots, d$  are the objective functions, and  $\mathbb{S}$  is the search space of decision vectors  $\mathbf{x}$  in  $m$  dimensional space.

Multi-objective optimization consists of two main branches of algorithmic solution approaches. The first approach is called *weighted sum method*. It converts a multi-objective optimization problem into a single-objective optimization problem by multiplying each objective function with a corresponding weighting factor and summing them up. The *weighted sum method* is simple and easy to be implemented. However, its weakness is obvious. On the one hand, it is very difficult to depict the thoroughly complete Pareto front set [12]. On the other hand, the solution obtained by using the weighted sum method does not necessarily reflect the preferences, when we want to represent the preferences of a decision maker by weights [13].

The second approach treats each objective function separately and utilizes the concept of a Pareto front as the fundamental concept to optimize each objective function by using different mechanisms (non-dominated ranking, infill criteria based on Pareto front, etc.). This research focuses on the second approach.

## 2.2 Terminologies

This section mainly introduces the concepts and terminologies of a **Pareto front** based on the objective space<sup>1</sup>. A **Pareto front** set is based on the dominance concept. The dominance is defined as follows:

**Definition 2.1 (Dominance [14])** *Given two decision vectors  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in \mathcal{X}$  and their corresponding objective values  $\mathbf{y}^{(1)} = \mathbf{y}(\mathbf{x}^{(1)})$ ,  $\mathbf{y}^{(2)} = \mathbf{y}(\mathbf{x}^{(2)})$ , it is said that  $\mathbf{y}^{(1)}$  dominates  $\mathbf{y}^{(2)}$ , being represented by  $\mathbf{y}^{(1)} \prec \mathbf{y}^{(2)}$ , iff  $\forall i \in \{1, 2, \dots, d\} : y_i(\mathbf{x}^{(1)}) \leq y_i(\mathbf{x}^{(2)})$  and  $\exists j \in \{1, 2, \dots, d\} : y_j(\mathbf{x}^{(1)}) < y_j(\mathbf{x}^{(2)})$ .*

Dominance is a fundamental concept in multi-objective optimization, and it provides an explicit relation between two solutions. In some cases, it can be used to decide which solution is better than the other. However, more interests from the perspective of research are put on the non-dominated solutions in MOO, because a point in a non-dominated space means a potential improvement of the objective function values.

**Definition 2.2 (Non-dominance [2])** *Given a decision vector set  $\mathbf{X} \subset \mathbb{S}$ , and the image of the vector set is  $\mathbf{Y} = \{\mathbf{y}(\mathbf{x}) | \mathbf{x} \in \mathbf{X}\}$ , the non-dominated subset of  $\mathbf{Y}$  is defined as:*

$$nd(\mathbf{Y}) := \{\mathbf{y} \in \mathbf{Y} | \nexists \mathbf{z} \in \mathbf{Y} : \mathbf{z} \prec \mathbf{y}\} \quad (2-2)$$

A vector  $\mathbf{y} \in nd(\mathbf{Y})$  in the objective space is called a non-dominated point. A non-dominated set means that there is no solution better or equally good in all components of the objective space. However, there could be solutions that are, at least, better in some component(s) with sacrificing the performance in the other component(s). The goal of MOO is trying to find all non-dominated solutions in a whole feasible search space, which is called **Pareto front**, and defined as:

**Definition 2.3 (Pareto front [14])** *For a feasible decision set  $\mathcal{X} \subset \mathbb{S}$ , the image of it is  $\mathcal{Y} = \{\mathbf{y}(\mathbf{x}) | \mathbf{x} \in \mathcal{X}\}$ , the Pareto front set  $\mathcal{P}^*$  is defined as:*

$$\begin{aligned} \mathcal{P}^* &:= \{\mathbf{y} \in \mathcal{Y} | \nexists \mathbf{z} \in \mathcal{Y} : \mathbf{z} \prec \mathbf{y}\} \\ &= nd(\mathcal{Y}) \end{aligned} \quad (2-3)$$

---

<sup>1</sup>In some papers, dominance is represented in a search space, instead of an objective space.

## 2. CONTINUOUS MULTI-OBJECTIVE OPTIMIZATION

---

In a Pareto front, each solution is a non-dominated point in  $\mathbf{Y}$ . Then, the problem of MOO is converted to how to find a Pareto front set  $\mathcal{P}^*$ . However, a Pareto front  $\mathcal{P}^*$  is difficult to be obtained, especially for a high-dimensional and continuous black-box problem. This is because usually only a finite number of non-dominated points can be obtained. Commonly, a Pareto front approximation set, which contains only a subset of a Pareto front set, is used to be optimized in MOO. Then, the final question will become how to define a Pareto front approximation set  $\mathcal{P}$  which can approximate the Pareto front set  $\mathcal{P}^*$  best. A Pareto front approximation is defined as:

**Definition 2.4 (Pareto front approximation)** *Given a Pareto front set  $\mathcal{P}^*$ , a Pareto front approximation set  $\mathcal{P}$  is any set of mutually non-dominated points and it is defined as:*

$$\mathcal{P} := \{\mathbf{y} \in \mathcal{Y}' \subseteq \mathcal{Y} \mid \nexists \mathbf{z} \in \mathcal{Y}' : \mathbf{z} \prec \mathbf{y}\} \quad (2-4)$$

**Example 2.1** *Figure 2.1 illustrates the concept of Pareto Dominance, Pareto front and Pareto front approximation in 2-D case. Suppose the image of the decision space  $\mathcal{X} \subset \mathbb{S}$  is  $\mathcal{Y}$ , then  $\mathcal{Y}$  can be expressed by dots in Figure 2.1. The Pareto front  $\mathcal{P}^*$  of  $\mathcal{Y}$  is the non-dominated set of  $\mathcal{Y}$  and it is represented by solid black curves. A Pareto front approximation  $\mathcal{P}$ , represented by the solid gray dots surrounded by dashed curves, is dominated by  $\mathcal{P}^*$ . The other gray dots are the dominated points, which are dominated by  $\mathcal{P}$ .*

### 2.3 Infill Criteria<sup>2</sup>

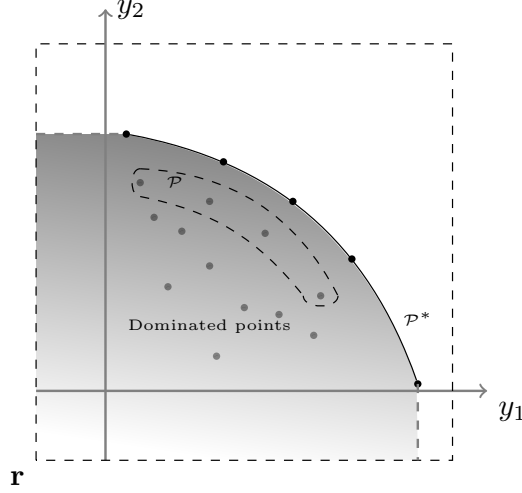
Given two Pareto front approximation sets, how to evaluate and compare the quality between the two Pareto front approximation sets? This section introduces some basic infill criteria, which will be used in later chapters, in MOO.

**Hypervolume Indicator:** The *Hypervolume Indicator*, proposed by Zitzler and Thiele [15], measures the size of the dominated subspace bounded from below<sup>3</sup>

---

<sup>2</sup>This section only considers maximization problems.

<sup>3</sup>The original definition was for minimization problems and the reference point bounds the set from above.



**Figure 2.1:** Example of 2-D Pareto front and Pareto front approximation.

by a reference point  $\mathbf{r}$ . The hypervolume indicates the performance of a Pareto-front approximation set  $\mathcal{P} \subset (\mathbb{R}^d)^n$ , where  $n$  stands for the number of the points in  $\mathcal{P}$ , and the maximization of HV can lead to a Pareto-front approximation set that is close to the true Pareto front. In 2-D and 3-D cases, the hypervolume indicator can be computed in time  $\Theta(n \log n)$  [16]. In more than 3 dimensions, the algorithm proposed by Chan [17] achieves  $O(n^{\frac{d}{3}} \text{polylog } n)$  time complexity. The hypervolume indicator is defined as:

**Definition 2.5 (Hypervolume Indicator)** *Given a finite Pareto front approximation set, say  $\mathcal{P} = \{\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n)}\} \subset \mathbb{R}^d$ , the Hypervolume Indicator (HV) of  $\mathcal{P}$  is defined as the  $d$ -dimensional Lebesgue measure of the subspace dominated by  $\mathcal{P}$  and bounded below by a reference point  $\mathbf{r}$ :*

$$HV(\mathcal{P}) = \lambda_d(\cup_{\mathbf{y} \in \mathcal{P}} [\mathbf{r}, \mathbf{y}]) \quad (3-5)$$

with  $\lambda_d$  being the Lebesgue measure on  $\mathbb{R}^d$ .

The reference point needs to be provided by the user, and it should, if possible, be chosen in such a way that it is dominated by all elements of the Pareto-front approximation sets  $\mathcal{P}$  that might occur during the optimization process.

**Hypervolume Improvement** *Hypervolume Improvement (HVI) is also called Improvement of Hypervolume in [18]. The basic idea of HVI is the HV change of a Pareto front approximation set  $\mathcal{P}$  before and after adding an evaluated point  $\mathbf{y}$  in it. The definition of Hypervolume Improvement is:*

## 2. CONTINUOUS MULTI-OBJECTIVE OPTIMIZATION

---

**Definition 2.6 (Hypervolume Improvement)** *Given a finite collection of vectors  $\mathcal{P} \subset \mathbb{R}^d$ , the Hypervolume Improvement (HVI) of a vector  $\mathbf{y} \in \mathbb{R}^d$  is defined as:*

$$HVI(\mathbf{y}, \mathcal{P}) = HV(\mathcal{P} \cup \{\mathbf{y}\}) - HV(\mathcal{P}) \quad (3-6)$$

*In case we want to emphasize the reference point  $\mathbf{r}$ , the notation  $HVI(\mathbf{y}, \mathcal{P}, \mathbf{r})$  will be used to denote the Hypervolume Improvement. Note that  $HVI(\mathbf{y}, \mathcal{P}) = 0$ , in case  $\mathbf{y} \in \mathcal{P}$ .*

**Hypervolume Contribution** Another HV based criterion is *Hypervolume Contribution* (HVC). It is applied as a selection criterion in SMS-EMOA [19]. The most efficient algorithm to calculate HVC (one time) currently holds a time complexity  $\Theta(n \log n)$  for  $d = 2, 3$  as proposed by Emmerich and Fonseca in [20]. The basic idea behind HVI and HVC is the same, that is, to calculate the difference of the hypervolume between two Pareto front approximation sets. The *Hypervolume Contribution* is defined as:

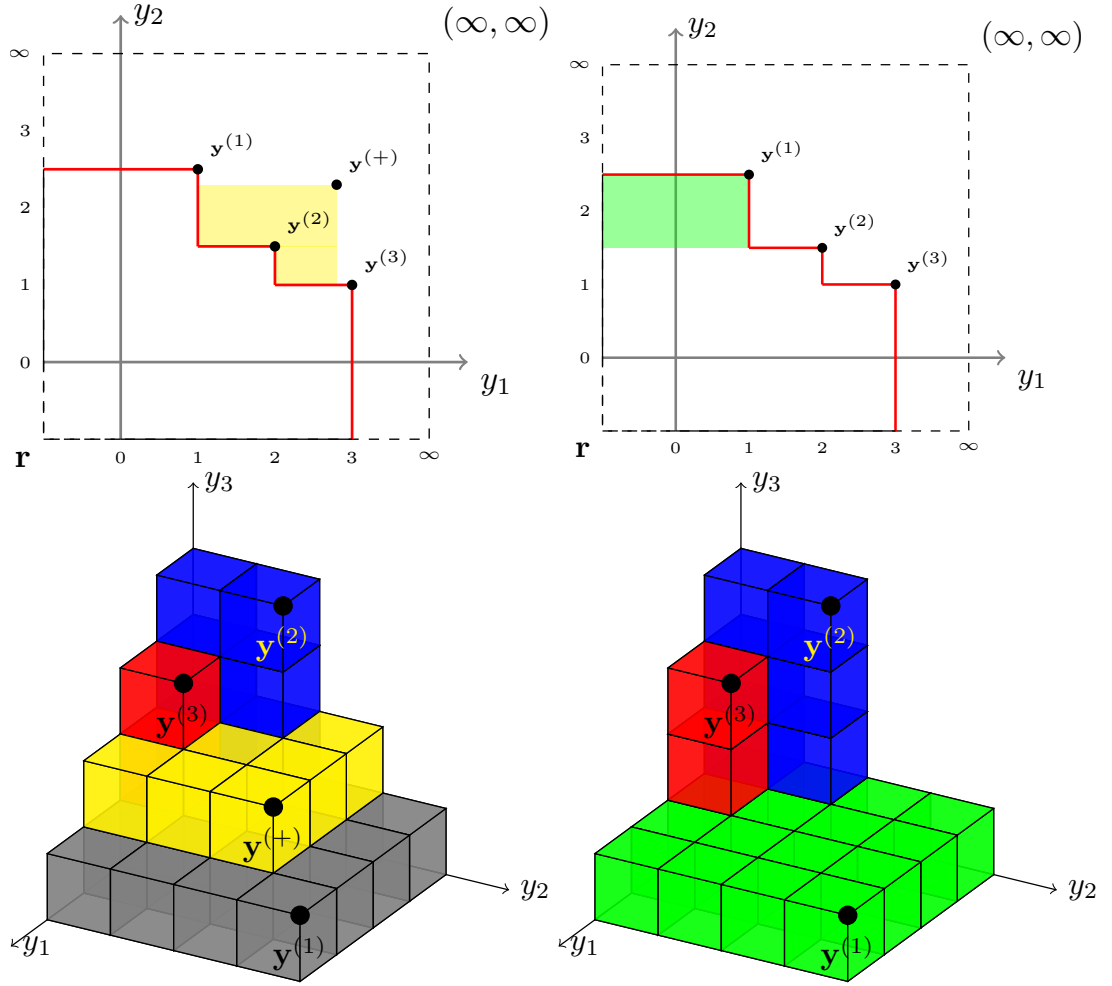
**Definition 2.7 (Hypervolume Contribution)** *Given a finite collection of vectors  $\mathcal{P} \subset \mathbb{R}^d$ , the Hypervolume Contribution (HVC) of a vector  $\mathbf{y} \in \mathbb{R}^d$  is defined as:*

$$HVC(\mathbf{y}, \mathcal{P}) = HV(\mathcal{P}) - HV(\mathcal{P} \setminus \{\mathbf{y}\}) \quad (3-7)$$

*In case we want to emphasize the reference point  $\mathbf{r}$ , the notation  $HVC(\mathbf{y}, \mathcal{P}, \mathbf{r})$  will be used to denote the Hypervolume Contribution.*

**Example 2.2** *Figure 2.2 illustrates the concept of the Hypervolume Improvement and the Hypervolume Contribution. For the 2-D case, suppose a Pareto front approximation set is  $\mathcal{P}$ , which is composed by  $\mathbf{y}^{(1)} = (1, 2.5)^T$ ,  $\mathbf{y}^{(2)} = (2, 1.5)^T$  and  $\mathbf{y}^{(3)} = (3, 1)^T$ . When a new point  $\mathbf{y}^{(+)} = (2.8, 2.3)^T$  is added, the Hypervolume Improvement  $HVI(\mathcal{P}, \mathbf{y}^{(+)})$  is the yellow area. The Hypervolume Contribution of  $\mathbf{y}^{(1)}$  is the green area. The right figures illustrate the Hypervolume Improvement and the Hypervolume Contribution for the 3-D case. A Pareto front approximation is  $\mathcal{P} = (\mathbf{y}^{(1)} = (4, 4, 1)^T, \mathbf{y}^{(2)} = (1, 2, 4)^T, \mathbf{y}^{(3)} = (2, 1, 3)^T)$ . The Hypervolume Improvement of  $\mathbf{y}^{(+)} = (3, 3, 2)^T$  relative to  $\mathcal{P}$  is given by the joint volume covered by the yellow slices. The Hypervolume Contribution of  $\mathbf{y}^{(1)}$  is given by the joint volume covered by the green slices.*

Other infill criteria, for instance, *Expected Hypervolume Improvement*, *Probability of Improvement* and *Truncated Hypervolume Improvement* will be introduced in later chapters.



**Figure 2.2:** The left column illustrates *Hypervolume Improvement* for 2-D and 3-D cases. The right column illustrates *Hypervolume Contribution* for 2-D and 3-D cases. The yellow areas stand for the Hypervolume improvement of  $y^{(+)}$ , and green areas represent the Hypervolume contribution of  $y^{(1)}$ .

### 2.4 Evolutionary Multi-objective Optimization Algorithms

Evolutionary algorithms are population based metaheuristics optimization algorithms and they are inspired by biological paradigm of natural selection, recombination and mutation [21, 22]. The main advantage of evolutionary algorithms, when applied to solve multi-objective optimization problems, is the fact that they typically generate sets of solutions, allowing computation of an approximation of the entire Pareto front.

#### 2.4.1 NSGA-II

NSGA-II is an improved version of NSGA (Nondominated Sorting Genetic Algorithm) [23], and it is a classical multi-objective algorithm proposed by Deb et al. [24, 25]. Being the most commonly applied evolutionary algorithm in the field of multi-objective optimization, NSGA-II serves as a reference algorithm in this dissertation. NSGA-II implements elitist mechanism,<sup>2</sup> where all the non-dominated solutions discovered are preserved from the beginning of the initial population. The selection mechanism of NSGA-II considers two factors: non-dominated rank of an individual in the population and its crowding distance (the average distance between two points on either side of this point along each of the objectives) for two objectives optimization. The priority between these two factors is the non-dominated rank. If two solutions are in the same non-dominated rank, the one that resides in the less crowded region is chosen. The basic structure of NSGA-II is shown in Algorithm 1.

#### 2.4.2 SMS-EMOA

A popular algorithm that uses the hypervolume indicator as a selection criterion is SMS-EMOA [27]. In its two-dimensional instantiation, it can be viewed as a steady state variant of the NSGA-II algorithm [25] that replaces the crowding distance by hypervolume contributions, and thereby generates a sequence of approximation sets that grow according to the hypervolume indicator. The basic structure of SMS-EMOA is illustrated in Algorithm 2.

---

<sup>2</sup>This was claimed by the authors, but it is wrong and explained in Emmerich et al. [26].

## 2.4 Evolutionary Multi-objective Optimization Algorithms

---



---

### Algorithm 1: NSGA-II

---

**Input:** Crossover rate  $p_c$ , mutation rate  $p_m$ , population size  $\mu$ , offspring size  $\lambda$ , objective functions  $\mathbf{y}$

**Output:** Pareto front approximation  $\mathcal{P}$

- 1: Evaluate an initial set of  $\mu$  points and store these in  $\mathcal{P} = ((\mathbf{x}^{(1)}, \mathbf{y}^{(1)} = \mathbf{y}(\mathbf{x}^{(1)})), \dots, (\mathbf{x}^{(\mu)}, \mathbf{y}^{(\mu)} = \mathbf{y}(\mathbf{x}^{(\mu)})))$ ;
- 2: **while** *termination criterion not satisfied* **do**
- 3:     Select parents  $\mathbf{X}'$  from  $\mathcal{P}$ :  $\mathbf{X}' = \text{Selection}(\mathcal{P})$  ;
- 4:     Crossover for  $\mathbf{X}'$ :  $\mathbf{X}' = \text{Crossover}(\mathbf{X}', p_c)$  ;
- 5:     Mutation for  $\mathbf{X}'$ :  $\mathbf{X}' = \text{Mutation}(\mathbf{X}', p_m)$  ;
- 6:     Evaluate the offspring  $\mathbf{X}'$  and store these in  $\mathcal{P}' = ((\mathbf{X}'^{(1)}, \mathbf{y}^{(1)} = \mathbf{y}(\mathbf{X}'^{(1)})), \dots, (\mathbf{X}'^{(\lambda)}, \mathbf{y}^{(\lambda)} = \mathbf{y}(\mathbf{X}'^{(\lambda)})))$ ;
- 7:     Fast non-dominated sorting for  $\mathcal{P} \cup \mathcal{P}'$  :  
         $F = \text{FastNonDominatedSorting}(\mathcal{P} \cup \mathcal{P}')$  ;
- 8:     Update  $\mathcal{P}$  by selecting best  $\mu$  individuals from  $\mathcal{P} \cup \mathcal{P}'$  ;
- 9: **Return**  $\mathcal{P}$

---

SMS-EMOA shows a slightly better performance on standard benchmarks than other commonly applied multi-objective optimization algorithms such as NSGA-II [27]. Therefore, SMS-EMOA was chosen as another reference algorithm in this research. We will compare its performance with that of multi-objective Bayesian global optimization defined in Chapters 3, 4, 5 and 8.

### 2.4.3 Example<sup>1</sup>

The utilization of evolutionary multi-objective algorithm is illustrated through a power distribution network reconfiguration problem (DNRP), which is served as a preliminary study in this dissertation. The network reconfiguration problem in a power distribution system aims at finding the best configuration of a radial network by changing the status of the switches in a power network system. There are two types of switches: normally closed switches and normally open switches. See Figure 2.3, for an example of a power distribution network configuration, the

---

<sup>1</sup>This example is a discrete optimization problem, and all the other parts of this dissertation consider only continuous optimization problems.

## 2. CONTINUOUS MULTI-OBJECTIVE OPTIMIZATION

---



---

### Algorithm 2: SMS-EMOA

---

**Input:** Objective functions  $\mathbf{y}$ , population size  $\mu$

**Output:** Pareto front approximation  $\mathcal{P}$

- 1: Evaluate an initial set of  $\mu$  points and store these in  
 $\mathcal{P} = ((\mathbf{x}^{(1)}, \mathbf{y}^{(1)} = \mathbf{y}(\mathbf{x}^{(1)})), \dots, (\mathbf{x}^{(\mu)}, \mathbf{y}^{(\mu)} = \mathbf{y}(\mathbf{x}^{(\mu)})))$ ;
  - 2: **while** *termination criterion not satisfied* **do**
  - 3:     Generate a new solution  $\mathbf{x}_{new}$  using PMX recombination and/or  
        polynomial mutation operator (cf. [25]) on (some) solutions of  $\mathcal{P}$  ;
  - 4:     Add point  $\mathbf{x}_{new}$  to  $\mathcal{P}$  ;
  - 5:     Compute dominance rank of each solution in  $\mathcal{P}$  by means of  
        non-dominated sorting ;
  - 6:     Determine  $R_{max}$  as the worst ranked layer of  $\mathcal{P}$  and remove it from the  
        solution with smallest hypervolume contribution ;
  - 7: **Return**  $\mathcal{P}$
- 

119 bus system [28], where the solid black lines represent normally closed switches, while dashed red lines represent normally open switches. Network reconfiguration is the process of changing the topology of the power network by operating these switches for the purpose of minimization of the power loss. Since each switch has two conditions, a system which has  $N$  nodes should contain  $2^{N-1}$  possible switch configurations. In order to ensure that all the customers can get electricity and no short circuit exists in the system, there are two constraints for network reconfiguration: no cycles (the radial structure of the network must be maintained in each new structure) and no islands (all the loads must be served).

The objective functions are the minimization of power loss and the maximization of the network's reliability—i.e. minimization of voltage deviation in this section. The objective function for the **minimization of power loss** can be described as [29]:

$$\min f_{loss} = \sum_{i=1}^b k_i R_i \frac{P_i^2 + Q_i^2}{V_i^2} = \sum_{i=1}^b k_i R_i |I_i|^2 \quad (4-8)$$

subject to:

$$V_i^{min} \leq V_i \leq V_i^{max} \quad (4-9)$$

$$I_i \leq I_i^{max}, i = 1, \dots, b \quad (4-10)$$

Here  $b$  is the number of branches and for each branch  $i \in \{1, \dots, b\}$ ,  $R_i$  is the

## 2.4 Evolutionary Multi-objective Optimization Algorithms

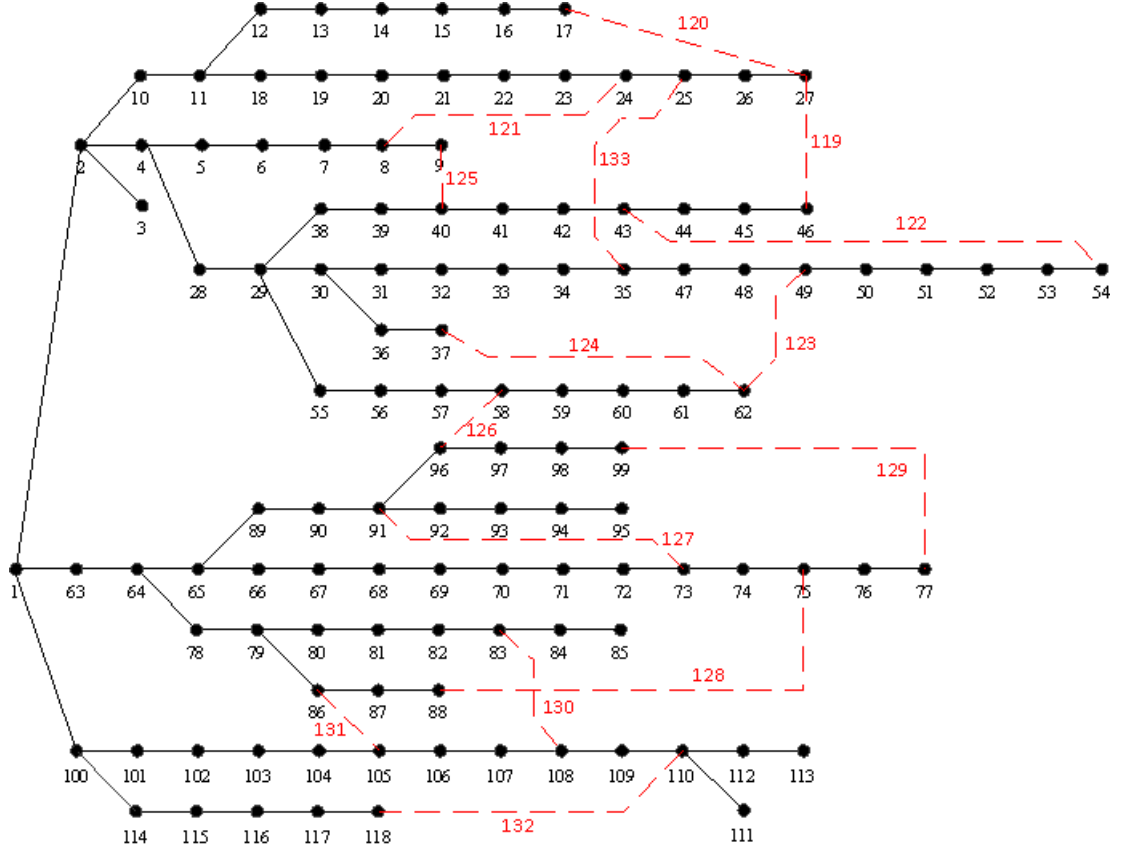


Figure 2.3: Initial configuration of the 119-bus test system.

branch resistance,  $P_i$  and  $Q_i$  are the active power and the inactive power of a branch terminal  $i$ ,  $V_i$  is the terminal node voltage of branch  $i$ ,  $V_i^{min}$  and  $V_i^{max}$  are the minimum and maximum bus voltage of branch  $i$ , respectively,  $k_i$  is the status variable of  $i$ -th switch. If  $k_i$  is 0, then switch  $i$  is open and if  $k_i$  is 1, then switch  $i$  is closed.  $I_i$  is the branch current and  $I_i^{max}$  is the maximum current in branch  $i$ .

The objective function for **minimization of voltage deviation** can be expressed as follows [30][31]:

$$\min f_{VDI} = \max\{|1 - U_{min}|, |1 - U_{max}|\} \quad (4-11)$$

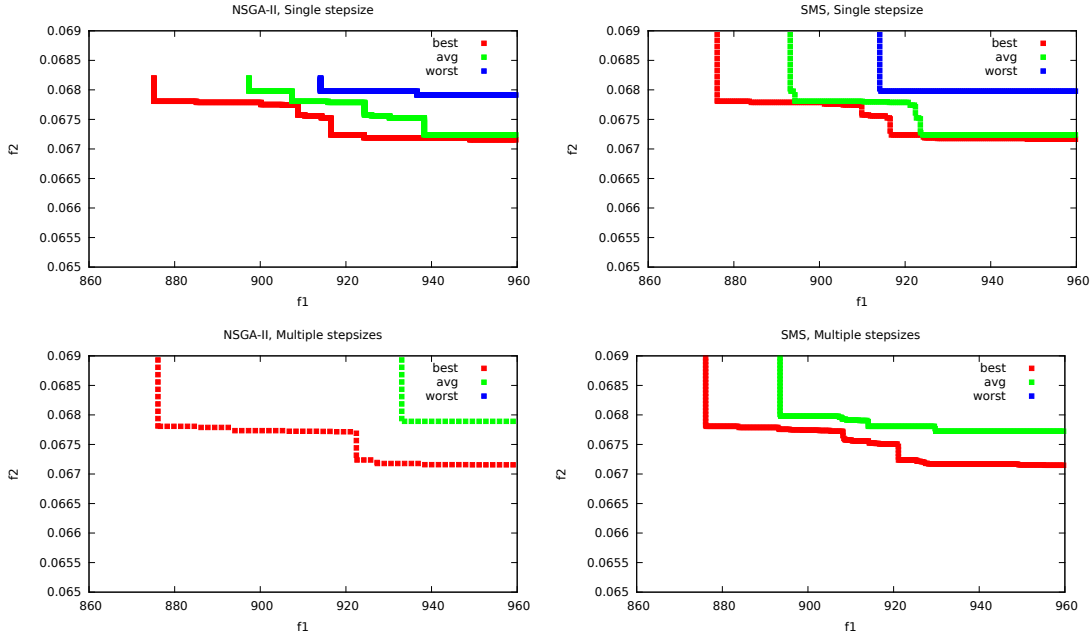
where  $U_{min}$  and  $U_{max}$  are respectively the lowest and highest values of bus voltage which is divided by rated voltage to normalize them to value in  $[0, 1]$ . In this dissertation, Newton's Method based on MATPOWER [32], which is a power flow calculation toolbox on MATLAB, is applied to calculate power loss and

## 2. CONTINUOUS MULTI-OBJECTIVE OPTIMIZATION

voltage deviation. The parameters in Newton's methods are: maximum number of iterations is 20 and termination tolerance on per unit is  $1e-8$ .

In this problem, a multi-objective optimization algorithm is achieved by replacing the selection scheme of a previously single objective optimizer by that of a multi-objective algorithm, namely the  $(\mu + \mu)$  selection of NSGA-II [25] and the  $(\mu + 1)$  selection of SMS-EMOA [19] (which has earlier been used also in Pareto archivers [33]). As a second objective voltage deviation is minimized (see equation 4-11).

As an adaptation, we introduce a variant of SMS-EMOA and NSGA-II with a self-adaptive single step size. Whenever more than five mutations per individual were unsuccessful, the step size was multiplied by a constant factor of  $1/1.2$  (following the  $1/5$ th success rule). The success of a generation was registered if a new non-dominated solution entered the archive of non-dominated solutions among all solutions encountered so far.



**Figure 2.4:** Best, worst, and average attainment curves for the multi-objective optimization of 119 DNRPs. The worst attainment curves of the multiple stepsizes are missing in the below pictures, because they are too far away from the best curves.

The results (attainment curves) for a population size of  $\mu = 30$  and 11 runs per algorithm are shown in Figure 2.4, where  $f1$  and  $f2$  represent voltage deviation

## 2.5 Multi-objective Bayesian Global Optimization

---

and power loss, respectively. Here, attainment curve<sup>1</sup> is a useful tool to assess the statistical performances of stochastic multi-objective optimizers, and visualize the outcome of a series of optimization trails [34, 35, 36].

In the following discussion of Pareto front, we mean the archive of all non-dominated solutions encountered in a single run. SMS-EMOA with a single step size provides the best Pareto fronts in the best case and also in the average case. Interestingly, all strategies find a Pareto front with a concave part, which is interpreted that locally there is a strong conflict between power loss minimization and voltage deviation minimization in this problem. However, the range of voltage deviation is relatively small, so that 'from a distance' the Pareto front has an apparent knee point region. Solutions in this region are recommended as good compromise solutions, whereas points located on the flanks of the Pareto front are not recommended, as small improvements in one objective will cause a large deterioration of the other objective.

## 2.5 Multi-objective Bayesian Global Optimization

In multi-objective optimization, the objective function evaluations are usually costly and evolutionary multi-objective optimization algorithms are usually not efficient to solve these expensive function evaluation problems. This is because EMOAs typically require a large number of function evaluations, furthermore, the Pareto optimality of the solutions cannot be guaranteed with fewer function evaluations. Therefore, EMOA is not recommended when function evaluations are expensive. In such cases, a better idea is perhaps to utilize information from all previous evaluations. This kind of algorithm is called *Bayesian Global Optimization* (BGO), which was proposed by Mockus et al. in [37].

The basic idea of BGO is to use a surrogate model based on Kriging or Gaussian process. A surrogate model reflects the relationship between decision vectors and their corresponding objective values. This surrogate model is learnt from the previous evaluations. For multi-objective problems, the family of these algorithms is called *Multi-Objective Bayesian Global Optimization* (MOBGO). The scheme of a MOBGO algorithm is sequentially updating a surrogate model, instead of 'true' objective functions, by an optimizer and its corresponding objective function value. An optimizer in MOBGO is utilized to search for a promising point  $\mathbf{x}^*$  by maximizing/minimizing an infill criterion according to surrogate models.

---

<sup>1</sup>Attainment curve is also called attainment surface for more than two objectives.

## 2. CONTINUOUS MULTI-OBJECTIVE OPTIMIZATION

---

### 2.5.1 Kriging

Kriging is a statistical interpolation method. Being a Gaussian process based modelling method, it is cheap to evaluate [38]. Kriging has been proven to be a popular surrogate model to approximate noise-free data in computer experiments, where Kriging models are fitted on previously evaluated points and then replace the real time-consuming simulation model [39]. Given a set of  $n$  decision vectors  $\mathbf{X} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)})^T$  in  $m$  dimensional search space, and associated function values  $y(\mathbf{X}) = (y(\mathbf{x}^{(1)}), y(\mathbf{x}^{(2)}), \dots, y(\mathbf{x}^{(n)}))^T$ , Kriging assumes  $y$  to be a realization of a random process  $Y$  and it is of the form [40, 41]:

$$Y(\mathbf{x}) = \mu(\mathbf{x}) + \epsilon(\mathbf{x}) \quad (5-12)$$

where  $\mu(\mathbf{x})$  is estimated mean value over all given sampled points, and  $\epsilon(\mathbf{x})$  is a realization of a normally distributed Gaussian random process with zero mean and variance  $\sigma^2$ . The regression part  $\mu(\mathbf{x})$  approximates globally the function  $Y$  and Kriging/Gaussian process  $\epsilon(\mathbf{x})$  takes local variations into account. Moreover, as opposed to other regression methods, such as supported vector machine (SVM), Kriging/GP also provides an uncertainty qualification of a prediction. The correlation between the deviations at two points ( $\mathbf{x}$  and  $\mathbf{x}'$ ) is defined as:

$$Corr[\epsilon(\mathbf{x}), \epsilon(\mathbf{x}')] = R(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^d R_i(x_i, x'_i) \quad (5-13)$$

Here  $R(.,.)$  is the correlation function, which can be cubic or spline function. Commonly, a Gaussian function (also known as squared exponential) is chosen:

$$R(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^d \exp(-\theta_i(x_i - x'_i)^2) \quad (\theta_i \geq 0)$$

where  $\theta$  are parameters of correlation model and they can be interpreted as measuring the importance of the variable. Then the covariance matrix can be expressed by the correlation function:

$$Cov(\boldsymbol{\epsilon}) = \sigma^2 \boldsymbol{\Sigma}, \quad \text{where} \quad \Sigma_{i,j} = R(\mathbf{x}_i, \mathbf{x}_j)$$

When  $\mu(\mathbf{x})$  is assumed to be an unknown constant, this unbiased prediction is called ordinary Kriging (OK). In OK, the Kriging model determines the hyper-parameters  $\theta = [\theta_1, \theta_2, \dots, \theta_n]$  by maximizing the likelihood on the observed dataset. The expression of the likelihood function is:

$$L = -\frac{n}{2} \ln(\sigma^2) - \frac{1}{2} \ln(|\boldsymbol{\Sigma}|)$$

## 2.5 Multi-objective Bayesian Global Optimization

---

The maximum likelihood estimates of the mean  $\hat{\mu}$  and the variance  $\hat{\sigma}^2$  are given by:

$$\begin{aligned}\hat{\mu} &= \frac{\mathbf{1}_n^T \Sigma^{-1} \mathbf{y}}{\mathbf{1}_n^T \Sigma^{-1} \mathbf{1}_n} \\ \hat{\sigma}^2 &= \frac{1}{n} (\mathbf{y} - \mathbf{1}_n \hat{\mu})^T \Sigma^{-1} (\mathbf{y} - \mathbf{1}_n \hat{\mu})\end{aligned}$$

Then the predictor of mean and variance at point  $x^t$  can be derived and they are shown as follows [41]:

$$\begin{aligned}\mu(\mathbf{x}^t) &= \hat{\mu} + \mathbf{c}^T \Sigma^{-1} (\mathbf{y} - \hat{\mu} \mathbf{1}_n) \\ \sigma^2(\mathbf{x}^t) &= \hat{\sigma}^2 \left[ 1 - \mathbf{c}^T \Sigma^{-1} \mathbf{c} + \frac{1 - \mathbf{c}^T \Sigma^{-1} \mathbf{c}}{\mathbf{1}_n^T \Sigma^{-1} \mathbf{1}_n} \right]\end{aligned}$$

where  $\mathbf{c} = (\text{Corr}[Y(x^t), Y(x_1)], \dots, \text{Corr}[Y(x^t), Y(x_n)])^T$ .

### 2.5.2 Structure of MOBGO

Compared to multi-objective evolutionary algorithms, MOBGO requires only a small budget of function evaluations [10]. As a result, it has already been implemented in the real world optimization for expensive evaluation problems. According to the authors' knowledge, it was for the first time used in the context of airfoil optimization [42]. Later, it was applied in the field of biogas plant controllers [43], in the detection of water quality management [44], in the structural design optimization [45] and could be implemented in the other real-world optimization problems with expensive evaluations.

Multi-Objective Bayesian Global Optimization is assuming that  $d$  objective functions are mutually independent in an objective space. In MOBGO, the Kriging method or Gaussian process can approximate Kriging models  $M$  with respect to objective functions and the uncertainties of the prediction, from the existing evaluated data  $D = ((\mathbf{x}^{(1)}, \mathbf{y}^{(1)} = Y(\mathbf{x}^{(1)})), \dots, (\mathbf{x}^{(\mu)}, \mathbf{y}^{(\mu)} = Y(\mathbf{x}^{(\mu)})))$ . Each objective function at a given point  $\mathbf{x}^{(t)}$  is approximated by a one-dimensional normal distribution, with mean  $\mu$  and standard deviation  $\sigma$ . Then MOBGO can predict the multivariate outputs by means of an independent joint normal distribution with parameters  $\mu_1, \dots, \mu_d$  and  $\sigma_1, \dots, \sigma_d$  at the point  $\mathbf{x}^{(t)}$ .

These predictive means and standard deviations can be used to calculate infill criteria. An *infill criterion* measures how promising a new point is, when compared to a current Pareto-front approximation. With the assistance of a single

## 2. CONTINUOUS MULTI-OBJECTIVE OPTIMIZATION

---

objective optimization algorithm, 'optimal' solution  $\mathbf{x}^*$  can be found according to the score of the infill criterion. This score of the infill criterion is calculated by the predictions of the Kriging models, instead of by the direct objective functions. Then, the 'optimal' solution  $\mathbf{x}^*$  is evaluated, and both the dataset  $D$  and the Pareto-front approximation set  $\mathcal{P}$  are updated.

---

### Algorithm 3: MOBGO algorithm

---

**Input:** Objective functions  $\mathbf{y}$ , initialization size  $\mu$ , termination criterion  $T_c$

**Output:** Pareto-front approximation  $\mathcal{P}$

- 1: Initialize  $\mu$  points  $(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(\mu)})$ ;
  - 2: Evaluate the initial set of  $\mu$  points:  $(\mathbf{y}^{(1)} = \mathbf{y}(\mathbf{x}^{(1)}), \dots, \mathbf{y}^{(\mu)} = \mathbf{y}(\mathbf{x}^{(\mu)}))$ ;
  - 3: Store  $(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(\mu)})$  and  $(\mathbf{y}^{(1)} = \mathbf{y}(\mathbf{x}^{(1)}), \dots, \mathbf{y}^{(\mu)} = \mathbf{y}(\mathbf{x}^{(\mu)}))$  in  $D$ :  
 $D = ((\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \dots, (\mathbf{x}^{(\mu)}, \mathbf{y}^{(\mu)}))$ ;
  - 4: Compute the non-dominated subset of  $D$  and store it in  $\mathcal{P}$ ;
  - 5:  $g = 1$ ;
  - 6: **while**  $g \leq T_c$  **do**
  - 7:   Train surrogate models  $M$  based on  $D$ ;
  - 8:   Use an optimizer to find the promising point  $\mathbf{x}^*$  based on surrogate models  $M$ , with the infill criterion  $C$ ;
  - 9:   Update  $D$ :  $D = D \cup (\mathbf{x}^*, \mathbf{y}(\mathbf{x}^*))$ ;
  - 10:   Update  $\mathcal{P}$  as non-dominated subset of  $D$ ;
  - 11:    $g = g + 1$ ;
  - 12: **end while**
  - 13: Return  $\mathcal{P}$ .
- 

The basic structure of the MOBGO algorithm is shown in Algorithm 3. It mainly contains three parts: initialization, updating and searching, and returning.

Firstly, the dataset  $D$  is initialized and the Pareto-front approximation set  $\mathcal{P}$  is calculated, as shown in Algorithm 3 from Step 1 to Step 5. The initialization of  $D$  contains generation of the decision vectors (Step 1), calculation of the corresponding objective values (Step 2) and storage of this information in data set  $D$  (Step 3). This data set  $D$  will be utilized to build the Kriging models in the second part.

The second part of MOBGO is the main loop, as shown in Algorithm 3 from Step 6 to Step 12. In this main loop, it is started by training the Kriging models  $M$  based on data set  $D$  (Step 6). Please note that  $M$  contains  $d$  independent models for

## 2.5 Multi-objective Bayesian Global Optimization

each objective function, and these models would be used as temporary objective functions instead of 'true' objective functions at Step 7. Then, an optimizer can find the promising point  $\mathbf{x}^*$  by maximizing or minimizing an infill criterion  $C$  (Step 7). Here, an infill criterion is calculated by its corresponding calculation formula, as the inputs of Kriging models  $M$ , Pareto-front approximation  $\mathcal{P}$ , decision vector  $\mathbf{x}$ , etc. In Step 8, a single-objective optimization algorithm is required to find the promising point  $\mathbf{x}^*$  for each temporary objective function – i.e., surrogate model. In this dissertation, the BI-Population CMA-ES has been chosen as the optimizer to find the promising point  $\mathbf{x}^*$ , considering its favorable theoretical properties [46]. After finding the promising point  $\mathbf{x}^*$ , Step 9 and Step 10 will update the dataset  $D$  by adding  $(\mathbf{x}^*, \mathbf{y}(\mathbf{x}^*))$  into  $D$  and update the Pareto-front approximation  $\mathcal{P}$ , respectively. The main loop from Step 6 to Step 12 will not stop until  $g$  meets the termination criterion  $T_c$ .

The last part of MOBGO is the return of Pareto-front approximation  $\mathcal{P}$ .

In single objective Bayesian Global Optimization, some common infill criteria include the *Expected Improvement* (EI) [41, 47], *Probability of Improvement* (PoI) [48, 49], and *Lower Confidence Bounds* (LCB) [50, 51, 52, 53]. In *Multi-Objective Bayesian Global Optimization*, some common *infill criteria* are: *Hypervolume Indicator* (HV) [54], *Probability of Improvement* (PoI) [48, 49, 55], *Hypervolume Improvement* (HVI) [27]<sup>1</sup>, *Euclidean distance-based EI* [55], *Hypervolume Contribution* (HVC) [19], *Expected Hypervolume Improvement* (EHVI) [47, 56], *Tchebycheff aggregation based EI* (EA-EI) [57], *Hypervolumme based PoI* [58], *Truncated Expected Hypervolume Improvement* (TEHVI) [6, 7], and *EI of penalty-based boundary intersection* (PBI-EI) [59].

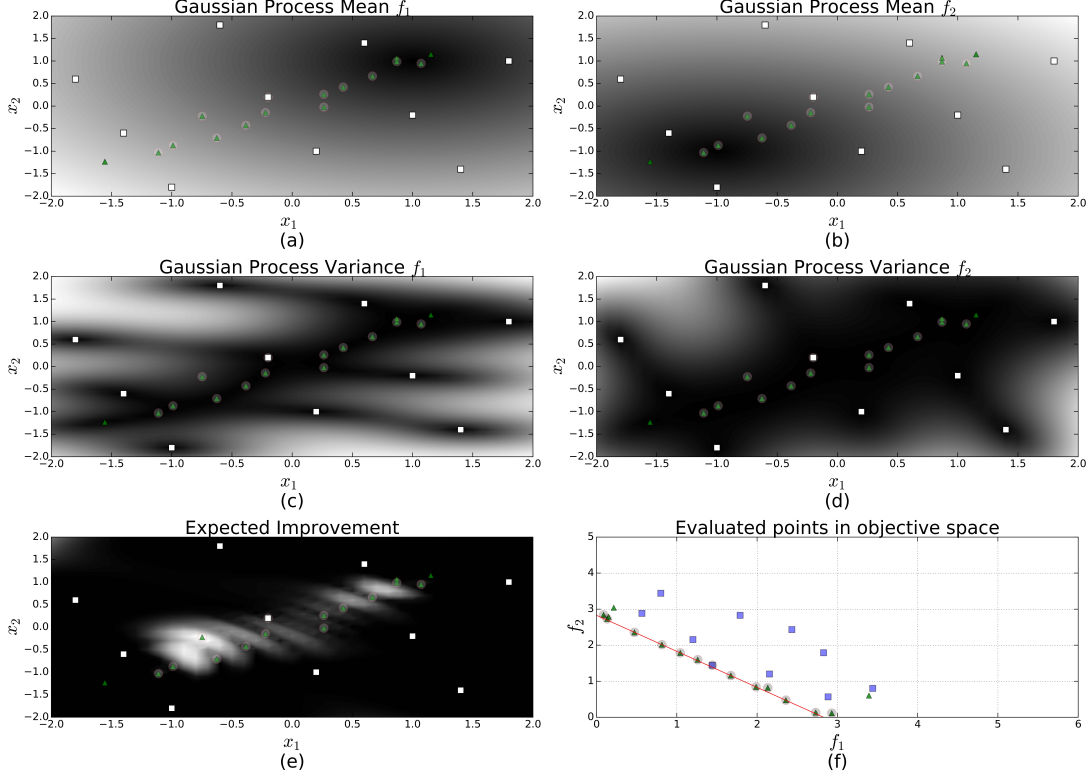
### 2.5.3 Example

The behavior of the BGO based on the expected hypervolume improvement will be illustrated by a single numerical experiment.

The numerical example is visualized in the plots of Figure 2.5. The bicriteria optimization problem from Emmerich et al. [60] is:  $f_1(\mathbf{x}) = \|\mathbf{x} - \mathbf{1}\| \rightarrow \min$ ,  $f_2(\mathbf{x}) = \|\mathbf{x} + \mathbf{1}\| \rightarrow \min$ ,  $\mathbf{x} \in [-2, 2] \times [-2, 2] \subset \mathbb{R}^2$ . The Pareto front is the line segment from  $(0, 2 \cdot \sqrt{2})$  to  $(2 \cdot \sqrt{2}, 0)$ , the efficient set is the line segment that connects  $(-1, -1)$  and  $(1, 1)$ . The metamodel used is a Gaussian random field model with Gaussian correlation function  $\exp(-\theta \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|^2)$ , for  $\mathbf{x}^{(1)} \in \mathbb{R}^m$

<sup>1</sup>The HVI was called the most likely improvement (MLI) in [27].

## 2. CONTINUOUS MULTI-OBJECTIVE OPTIMIZATION



**Figure 2.5:** Example Run of Multicriteria Bayesian Global Optimization.

and  $\mathbf{x}^{(2)} \in \mathbb{R}^m$ , here  $m = 2$ . We set  $\theta = 0.0001$ , which was estimated by maximum likelihood method for initial sample. An initial set of 10 points was evaluated indicated by the dark blue squares in Figure 2.5 (f). From this starting set 15 new points were generated using the expected hypervolume improvement. The maximizer of the expected improvement was found using a uniform grid. In total, each objective function was evaluated 25 times.

The results of the experiment are depicted in plots. In all pictures, points that have been evaluated are indicated by triangles. The points from the initial set are additionally marked by squares. Efficient points are surrounded by circles. The figures in the top row, Figure 2.5 (a) and (b), depict the mean value of the Gaussian random field model at  $\mathbf{x} \in [-2, 2] \times [-2, 2]$  for  $f_1$  and  $f_2$ , respectively. Likewise, the figures in the middle row, Figure 2.5 (c) and (d), depict the variance of the Gaussian random field model at  $\mathbf{x} \in [-2, 2] \times [-2, 2]$  for  $f_1$  and  $f_2$ , respectively. The expected hypervolume improvement values after 25 iterations are shown in Figure 2.5 (e). The final set of points in the objective space and the Pareto front approximation can be found in Figure 2.5 (f). After

only *25 evaluations of the original objective functions*, the algorithm finds a good approximation to the Pareto front.

## 2.6 Summary

Most practical optimization problems involve more than one objective, simply due to the fact that no product, process, or system can be assessed with a single criterion. Since, frequently, there are conflicting criteria, such as minimizing cost and maximizing the quality of a product, Multi-Objective Optimization problems give rise to a set of trade-off optimal (known as Pareto optimal) solutions.

In this chapter, we defined vital terms that are used in MOBGO research, introduced some infill criteria in MOO, described two state-of-the-art EMOAs with a practical problem, and introduced the structure of MOBGO, together with a simple example for the illustration. This chapter only defines the fundamental terminologies of this research. Other related terminologies are represented in each chapter.

# Chapter 3

## Efficient EHVI Calculation

In chapter 2, the basic structure of MOBGO was introduced and some common infill criteria were mentioned. Among these infill criteria in MOO, EHVI outperforms other criteria for its inherent ability to balance exploitation and exploration [61]. However, EHVI is seldom applied in real application because the computational complexity of EHVI is very expensive<sup>1</sup>. In this chapter, an asymptotically optimal algorithm for the computation of the exact expected hypervolume improvement (EHVI) is proposed, based on partitioning the integration volume into a set of axis-parallel slices. Theoretically, the upper bound time complexities are improved from previously  $O(n^3 \log n)$  and  $O(n^4 \log n)$  in [60], for two and three objectives problems respectively, to now  $O(n \log n)$  for both two and three objective problems, which is asymptotically optimal, as we have proved. This scheme is also generalized in the case of high dimension in this chapter.

This chapter mainly contributes to the thesis by introducing the state-of-the-art EHVI calculation methods. This chapter is structured as follows: Section 3.1 provides the definition of EHVI; Section 3.2 explains the reason why EHVI is an important criterion in MOO and introduces some current algorithms to calculate EHVI; Section 3.3 provides the partitioning methods for non-dominated space; Section 3.4 shows the final formula expression of EHVI, based on the partitioning method described in Section 3.3; Section 3.6 shows the EHVI calculation speed comparison and empirical experimental results on benchmarks, with respect to state-of-the-art multi-objective optimization algorithms.

---

<sup>1</sup>The computational complexity of an infill criterion is crucial in multi-objective Bayesian global optimization, because this criterion needs to be called frequently during the execution of such an algorithm.

### 3.1 EHVI Definition

**Definition 3.1** ( $\Delta$  function (see also [2])) *For a given vector of objective function values,  $\mathbf{y} \in \mathbb{R}^d$ ,  $\Delta(\mathbf{y}, \mathcal{P}, \mathbf{r})$  is the subset of the vectors in  $\mathbb{R}^d$  which are exclusively dominated by a vector  $\mathbf{y}$  and not by elements in  $\mathcal{P}$  and that dominate the reference point, in symbols*

$$\Delta(\mathbf{y}, \mathcal{P}, \mathbf{r}) = \lambda_d\{\mathbf{z} \in \mathbb{R} \mid \mathbf{y} \prec \mathbf{z} \text{ and } \mathbf{z} \prec \mathbf{r} \text{ and } \nexists \mathbf{q} \in \mathcal{P} : \mathbf{q} \prec \mathbf{z}\} \quad (1-1)$$

For the simplicity, the notation  $\Delta(\mathbf{y})$  will be used to express  $\Delta(\mathbf{y}, \mathcal{P}, \mathbf{r})$  in this paper.

EHVI is a generalization of EI for multi-objective cases, and it is based on the theory of the HV. Similar to EI, the calculation of EHVI is based on the predictions in the Gaussian random field. EHVI measures how much hypervolume improvement could be achieved by evaluating the new point, considering the uncertainty of the prediction. It is defined as:

**Definition 3.2 (Expected Hypervolume Improvement)** <sup>1</sup> *Given parameters of the multivariate predictive distribution  $\boldsymbol{\mu}$ ,  $\boldsymbol{\sigma}$  and the Pareto-front approximation  $\mathcal{P}$  the expected hypervolume improvement (EHVI) is defined as:*

$$EHVI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) := \int_{\mathbb{R}^d} HVI(\mathcal{P}, \mathbf{y}) \cdot PDF_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \quad (1-2)$$

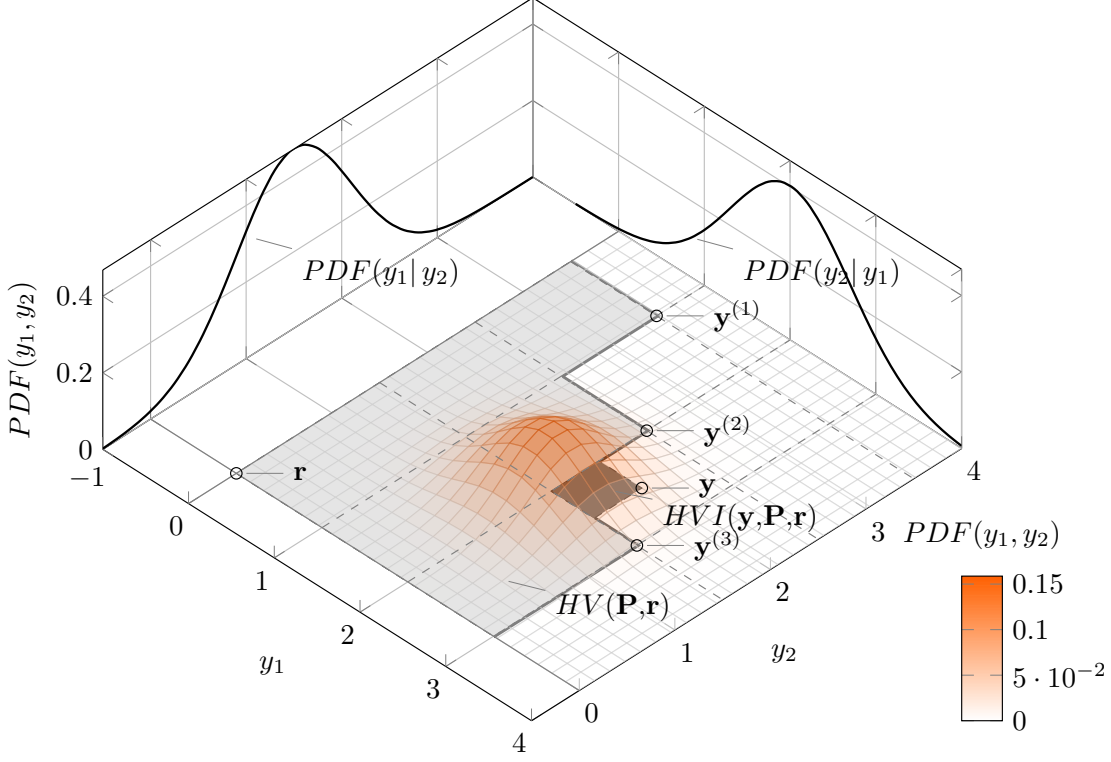
where  $PDF_{\boldsymbol{\mu}, \boldsymbol{\sigma}}$  is the multivariate independent normal distribution for mean values  $\boldsymbol{\mu} \in \mathbb{R}^d$ , and standard deviations  $\boldsymbol{\sigma} \in \mathbb{R}_+^d$ .

**Example 3.1** *An illustration of the 2-D EHVI is shown in Figure 4.1. The light gray area is the dominated subspace of  $\mathcal{P} = \{\mathbf{y}^{(1)} = (3, 1)^\top, \mathbf{y}^{(2)} = (2, 1.5)^\top, \mathbf{y}^{(3)} = (1, 2.5)^\top\}$  cut by the reference point  $\mathbf{r} = (0, 0)^\top$ . The bivariate Gaussian distribution has the parameters  $\mu_1 = 2, \mu_2 = 1.5, \sigma_1 = 0.7, \sigma_2 = 0.6$ . The probability density function (PDF) of the bivariate Gaussian distribution is indicated as a 3-D plot. Here  $\mathbf{y}$  is a sample from this distribution and the area of improvement relative to  $\mathcal{P}$  is indicated by the dark shaded area. The variable  $y_1$  stands for the  $f_1$  value and  $y_2$  for the  $f_2$  value.*

<sup>1</sup>The prediction of  $\boldsymbol{\mu}$  and  $\boldsymbol{\sigma}$  depends on a Kriging model and a target point  $\mathbf{x}$  in the search space. Explicitly, EHVI is dependent on the target point  $\mathbf{x}$ .

### 3. EFFICIENT EHVI CALCULATION

---



**Figure 3.1:** EHVI in 2-D (cf. Example 3.1).

For the convenience of expressing the formula of EHVI and EHVIG in later sections, it is useful to define a function we call  $\Psi_\infty$ .

**Definition 3.3** ( $\Psi_\infty$  function (see also [1])) *Let  $\phi(s) = 1/\sqrt{2\pi}e^{-\frac{1}{2}s^2}$  ( $s \in \mathbb{R}$ ) denote the probability density function (PDF) of the standard normal distribution. Moreover, let  $\Phi(s) = \frac{1}{2} \left(1 + \operatorname{erf}\left(\frac{s}{\sqrt{2}}\right)\right)$  denote its cumulative probability distribution function (CDF), and  $\operatorname{erf}$  is Gaussian error function. The general normal distribution with mean  $\mu$  and standard deviation  $\sigma$  has as PDF,  $\phi_{\mu,\sigma}(s) = \phi_{\mu,\sigma}(s) = \frac{1}{\sigma}\phi\left(\frac{s-\mu}{\sigma}\right)$  and its CDF is  $\Phi_{\mu,\sigma}(s) = \Phi\left(\frac{s-\mu}{\sigma}\right)$ . Then the function  $\Psi_\infty(a, b, \mu, \sigma)$  is defined as:*

$$\begin{aligned} \Psi_\infty(a, b, \mu, \sigma) &= \int_b^\infty (z - a) \frac{1}{\sigma} \phi\left(\frac{z - \mu}{\sigma}\right) dz \\ &= \sigma \phi\left(\frac{b - \mu}{\sigma}\right) + (\mu - a) \left[1 - \Phi\left(\frac{b - \mu}{\sigma}\right)\right] \end{aligned} \quad (1-3)$$

## 3.2 State-of-the-art

In the context of MOBGO, an infill criterion is used to evaluate the improvement for a new point, as introduced in Chapter 2. A common criterion for a single-objective optimization problem is *Expected Improvement* (EI), which was firstly introduced by Mockus et al. [37] in 1978 and it exploits both the Kriging prediction and the variance to give a quantitative measure of the improvements for the points in the search space. Later, EI became more popular due to the work of Jones et al. [41], in which it serves as an infill criterion in the so-called Efficient Global Optimization (EGO) algorithm<sup>1</sup>. In each iteration, EGO evaluates the design point with maximal EI. Its convergence properties are discussed in [62], where a proof of global convergence under mild assumptions on the global covariance and the smoothness of the function is given. Roughly speaking, global convergence occurs due to the fact that EI rewards high variance and also high mean values.

Various generalizations of EI in the field of multi-objective optimization have been discussed in the literature, e.g., [45, 55, 61, 63, 64]. See also [47] for an overview. In the case of multiple objectives, it is possible to consider a Gaussian process model for each objective function separately and independently, resulting in a multivariate distribution with  $d$  mean values  $\mu_i(x)$  and standard deviation  $\sigma_i(x)$ . A key question when generalizing the expected improvement is how to define improvement of a given Pareto-front approximation. In indicator-based multi-objective optimization, the performance of a Pareto-front approximation is assessed by a unary indicator, typically the *Hypervolume Indicator*, which allows a simple generalization of the *Expected Improvement* – the EHVI. EHVI is a straightforward generalization of the *single-objective expected improvement* and was proposed by Emmerich [56] in 2005. Since then, EHVI has been used in Evolutionary Algorithms for airfoil optimization [53] and quantum control [65]. It is also applied in multi-objective generalizations of Bayesian Global Optimization for applications, such as fluid dynamics [42], event controllers in wastewater treatment [44], efficient algorithm tuning [66], electrical component design [58], and bio-fuel power-generation [10]. In all of these applications, the bi-objective EHVI was used. Due to its high computation time for problems which contains three and more objectives, it is not recommended to use EHVI as an infill criterion in such cases. Fast, but imprecise, alternatives were sought [67].

The expected hypervolume improvement (EHVI) is the expected value of the increment of the hypervolume indicator given a Pareto-front approximation and

<sup>1</sup>Efficient Global Optimization is another name of Bayesian Global Optimization.

### 3. EFFICIENT EHVI CALCULATION

---

a predictive multivariate Gaussian distribution predicting the outcome at a new point. When compared to other criteria, EHVI leads to better convergence towards the true Pareto front, and to a higher diversity of the Pareto-front approximation set [6, 58, 67, 68]. However, the calculation of EHVI itself has so far been time-consuming [44, 47, 69], even in the case of two dimensions. It still remains unknown whether the integration algorithms used in the literature achieved optimal performance. Hence, it is important to study whether, and to what extent, the computational efficiency of the exact computation of the EHVI can be further improved. In addition, EHVI is called multiple times in every iteration. For the above reasons, a fast algorithm for computing the EHVI is crucial.

The first method suggested for EHVI calculation was Monte Carlo integration and it was first proposed by Emmerich in [56] and [53]. This method is simple and straightforward. However, the accuracy of EHVI highly depends on the number of the iterations. The first exact EHVI calculation algorithm was derived by Emmerich et al. [60], with the computational complexity  $O(n^3 \log n)$  and  $O(n^4 \log n)$  in the cases of 2-D and 3-D, respectively. Couckuyt et al. introduced a faster exact EHVI calculation algorithm for  $d > 2$  in [58], but did not provide a detailed complexity analysis. Recently, Hupkens et al. reduced the time complexity to  $O(n^2)$  and  $O(n^3)$  [1] for two- and three-dimensional cases, respectively. These algorithms further improved the practical efficiency of EHVI on test data in comparison to [58]. More recently, Emmerich et al. proposed an asymptotically optimal algorithm for the bi-objective case with time complexity  $O(n \log n)$  [2], where  $n$  is the number of non-dominated points in the archive. So far the best known bounds for the time complexity of exact computations have been  $O(n \log n)$  for  $d = 2$ , and  $O(n^3)$  for  $d = 3$ . It is notable that the number of transcendental function evaluations, such as erf and exp, scales only linearly in  $n$  in the algorithm presented in [1]. A lower bound of  $\Omega(n \log n)$  is provided for a given approximation set of size  $n$ . However, it makes sense to assume that non-dominated points are sorted in the first coordinate. In that case, as will be shown, a lower bound of  $\Omega(n)$  still holds.

## 3.3 Non-dominated Space Partitioning Algorithm

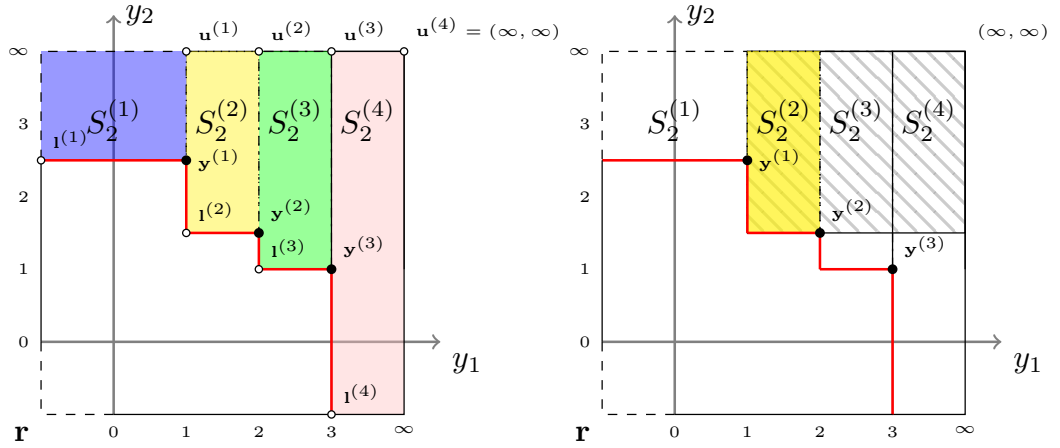
### 3.3.1 Low Dimensional case

**2-D case:** Suppose  $\mathbf{y} = \mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n)}$  and  $d = 2$ , then the integration area (non-dominated area) can be divided into  $n+1$  disjoint integration slices ( $S_2^{(i)}, i =$

### 3.3 Non-dominated Space Partitioning Algorithm

$1, \dots, n+1$ ) by drawing parallel to  $y_2$ -axis lines at each element in  $\mathbf{y}$ , as indicated in Figure 3.2 (left). Then, each integration slice can be expressed by its lower bound ( $\mathbf{l}_2^{(i)}$ ) and upper bound ( $\mathbf{u}_2^{(i)}$ ). In order to define the stripes formally, augment  $\mathcal{P}$  with two sentinels:  $\mathbf{y}^{(0)} = (r_1, \infty)$  and  $\mathbf{y}^{(n+1)} = (\infty, r_2)$ . Then, the integration slices for 2-D case are now defined by:

$$\begin{aligned} S_2^{(i)} = (\mathbf{l}_2^{(i)}, \mathbf{u}_2^{(i)}) &= ((l_1^{(i)}, l_2^{(i)})^T, (u_1^{(i)}, u_2^{(i)})^T) \\ &= ((y_1^{(i-1)}, y_2^{(i)}, (y_1^{(i)}, \infty)^T) \quad i = 1, \dots, N_2 \end{aligned} \quad (3-4)$$



**Figure 3.2:** Left: Partitioning of the integration region into stripes. Right: New partitioning of the reduced integration region after the first iteration of the algorithm.

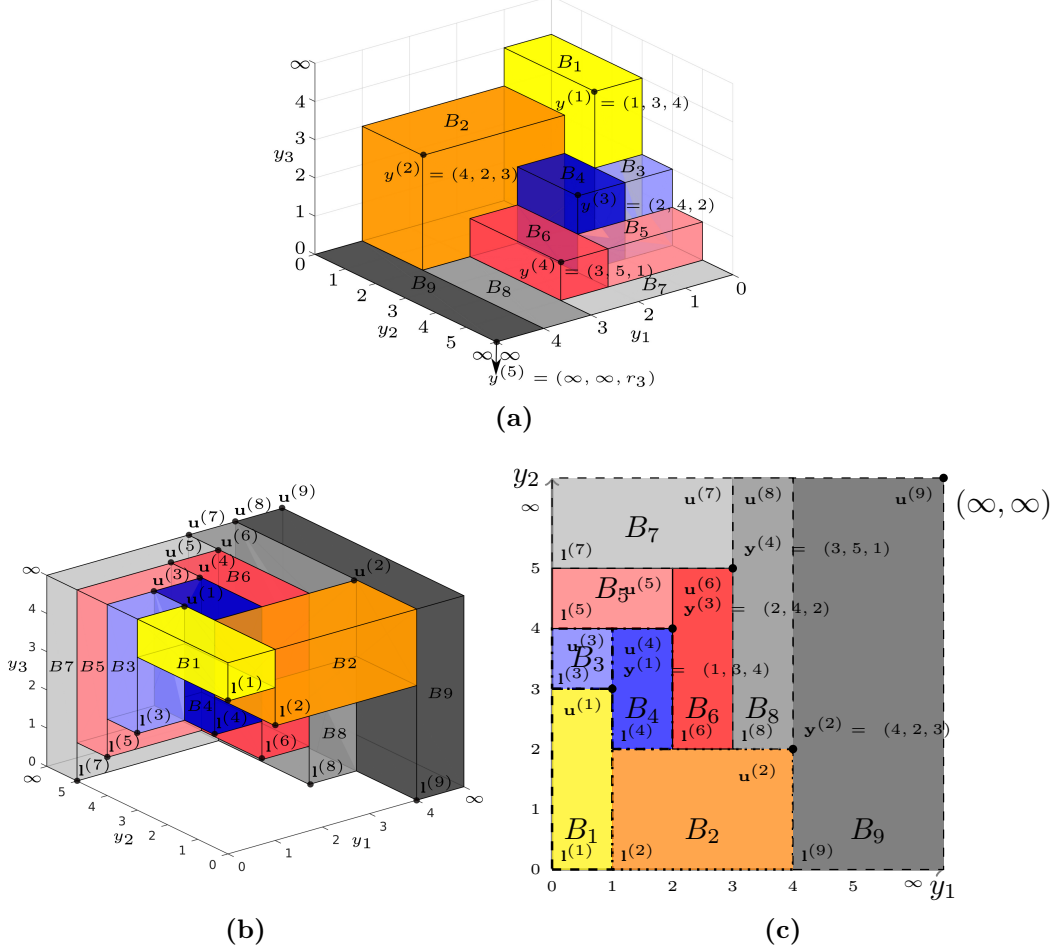
For the 2-D case, it is straightforward that the number of integration slices  $N_2$  is  $n + 1$ .

**3-D case:** Similar to the 2-D partitioning method, in the 3-D case, each integration slice can also be defined by its lower bound ( $\mathbf{l}_3$ ) and upper bound ( $\mathbf{u}_3$ ). Since the upper bound of each integration slice is always  $\infty$  in the  $y_3$  axis, we can describe each integration slice as follows:

$$S_3^{(i)} = (\mathbf{l}_3^{(i)}, \mathbf{u}_3^{(i)}) = ((l_1^{(i)}, l_2^{(i)}, l_3^{(i)})^T, (u_1^{(i)}, u_2^{(i)}, \infty)^T) \quad i = 1, \dots, N_3 \quad (3-5)$$

**Example 3.2** An illustration of integration slices is shown in Figure 3.3. A Pareto front set is composed by  $n = 4$  points ( $\mathbf{y}^{(1)} = (1, 3, 4)$ ,  $\mathbf{y}^{(2)} = (4, 2, 3)$ ,  $\mathbf{y}^{(3)} = (2, 4, 2)$  and  $\mathbf{y}^{(4)} = (3, 5, 1)$ ), and this Pareto front is shown in Figure 3.3 (a).

### 3. EFFICIENT EHVI CALCULATION



**Figure 3.3:** Figure (a): 3-D Pareto-front Approximation. Figure (b): Integration slices in 3-D. Figure (c): The projection of 3-D integration slices into the  $y_1y_2$ -plane, each slice can be described by lower bound and upper bound.

The added point  $\mathbf{y}^{(n+1)}$  is  $\mathbf{y}^{(5)} = (\infty, \infty, r_3)$ . The integration slices in the non-dominated space are represented by boxes in Figure 3.3 (b). Figure 3.3 (c) illustrates the projection onto the  $y_1y_2$ -plane with rectangle slices and  $\mathbf{l}, \mathbf{u}$ . The rectangular slices, which share the similar color but differ in opacity, represent integration slices with the same value of  $y_3$  in their lower bound. For example, the lower bound of the 3-D integration slice  $B_4$  is  $\mathbf{l}_3^{(4)} = (1, 2, 2)$ , and the upper bound of the slice is  $\mathbf{u}_3^{(4)} = (2, 4, \infty)$ .

The basic idea of the efficient partitioning algorithm in 3-D non-dominated space

### 3.3 Non-dominated Space Partitioning Algorithm

is that the transforming the 3-D Pareto front into 2-D Pareto front. This transforming consists of the following steps. Firstly, sort all the  $n$  elements  $(\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n)})$  in Pareto-front approximation set  $\mathcal{P}$  in descending order by coordinate  $y_3$ . Secondly, set a new point  $\mathbf{y}^{(n+1)} = (\infty, \infty, r_3)$ . Thirdly, insert the element  $\mathbf{y}^{(i)}$  ( $i = 1, \dots, n+1$ ) into the 2-D Pareto-front set  $\mathcal{P}'$  one by one using coordinate  $y_1$  and  $y_2$  value<sup>1</sup>, and discard the dominated points  $\mathbf{y}^{[d]}$  which are dominated by the new inserted point  $\mathbf{y}^{(i)}$ . During the third step, when a new point  $\mathbf{y}^{(i)}$  is inserted, one and only one integration box is created on its below left side. When there exists a discarded point  $\mathbf{y}^{[d]}$ , one and only one integration box is created on its above right side. In other words, an integration box is only created when a new point  $\mathbf{y}^{(i)}$  is inserted to the  $\mathcal{P}'$  or a point is  $\mathbf{y}^{[d]}$  is discarded from the  $\mathcal{P}'$ . The third step will not stop until the last point  $\mathbf{y}^{(n+1)}$  is inserted. Since all the elements in  $\mathcal{P}$  are dominated by  $\mathbf{y}^{(n+1)}$  and no point can dominate it in the non-dominated space cut by a reference point  $\mathbf{r}$ , all the elements in  $\mathcal{P}$  will be discarded and  $\mathcal{P}'$  only consist of  $\mathbf{y}^{(n+1)}$  in the last iteration of the third step. Then, the number of integration boxes for 3-D case is the sum of the number of the points that are inserted and the number of the points that are discarded, i.e.,  $N_3 = (n+1) + n = 2n+1$ .

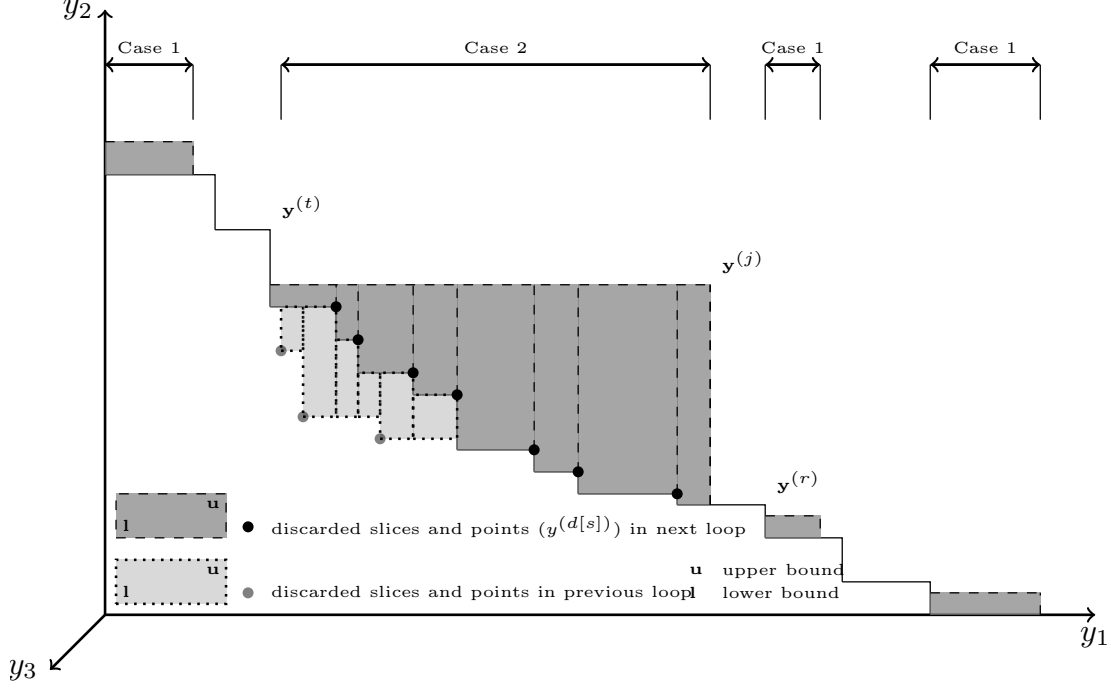
Algorithm 4 describes how to obtain the slices  $S_3^{(1)}, \dots, S_3^{(i)}, \dots, S_3^{(N_3)}$  with the corresponding lower and upper bounds ( $\mathbf{l}_3^{(i)}$  and  $\mathbf{u}_3^{(i)}$ ) and how to compute the integrals for them. The partitioning algorithm is similar to the sweep line algorithm described in [20]. The basic idea of this algorithm is to use an AVL tree to process points in descending order of the  $y_3$  coordinate. For each such point, say  $\mathbf{y}^{(i)}$ , add this point to the AVL tree and find all the points  $(\mathbf{y}^{(d[1])}, \dots, \mathbf{y}^{(d[s])})$  which are dominated by  $\mathbf{y}^{(i)}$  in the  $y_1y_2$ -plane and discard them from the AVL tree. See Figure 3.4 for describing one such iteration. In each iteration,  $s+1$  slices are created using coordinates of the points  $\mathbf{y}^{(t)}, \mathbf{y}^{(d[1])}, \dots, \mathbf{y}^{(d[s])}, \mathbf{y}^{(r)}$ , and  $\mathbf{y}^{(i)}$  as illustrated in Figure 3.4.

Here, the number of the integration slices for 3-D case  $N_3$  is  $2n+1$ , when all points are in general position (the coordinate of each point is different). Otherwise  $2n+1$  provides an upper bound for the obtained number of slices. The reason is as follows: In the algorithm each point  $\mathbf{y}^{(i)}, i = 1, \dots, n$  creates a slice, say slice  $A^{(i)}$ , when it is created and a slice, say slice  $S_3^{(i)}$ , when it is discarded from the AVL tree due to domination by another point, say  $\mathbf{y}^{(s)}$ , in the  $y_1y_2$ -plane.

The two slices are defined as follows  $A^{(i)} = ((y^{(t)}, y_2^{(l2)}, y_3^{(i)}), (y_1^{(u1)}, y_2^{(i)}, \infty))$  whereas

<sup>1</sup>The coordinate value of  $y_3$  is hidden for action of inserting and discarding, but  $y_3$  value still exist.

### 3. EFFICIENT EHVI CALCULATION



**Figure 3.4:** Boundary search for slices in 3-D case.

$y_2^{(l2)}$  is either  $y_2^{(r)}$  if no points are dominated by  $\mathbf{y}^{(i)}$  in the  $y_1y_2$ -plane or  $y_2^{(d[1])}$ , otherwise. Moreover,  $S_3^{(i)} = ((y_1^{(i)}, y_2^{(r)}, y_3^{(s)}), (y_1^{(u)}, y_2^{(s)}, \infty))$ , and  $\mathbf{y}^{(u)}$  denotes either the right neighbour among the newly dominated points in the  $y_1y_2$ -plane, or  $\mathbf{y}^{(s)}$  if  $\mathbf{y}^{(i)}$  is the rightmost point among all newly dominated points. In this way, each slice can be attributed to exactly one point in  $\mathcal{P}$ , except for the slice that is created in the final iteration. In the final iteration one additional point  $\mathbf{y}^{(n+1)} = (\infty, \infty, \infty)$  is added in the  $y_1y_2$ -plane. This point leads to the creation of a slice when it is added, but it adds only a single slice, because it is never discarded. Therefore,  $2n + 1$  slices are created in total.

As opposed to previous techniques, which required grid decomposition of the non-dominated subspace into  $O(n^3)$  integration slices, the new integration technique can make use of efficient partitioning of the dominated space into only  $2n + 1$  axis-aligned integration slices. In practice, the new computation scheme will be of great advantage to making the EHVI and related integrals applicable in multi-objective optimization with three objectives, especially in Bayesian Optimization and surrogate-assisted multi-objective evolutionary algorithms.

### 3.3 Non-dominated Space Partitioning Algorithm

---

#### Algorithm 4: Integration slices acquiring in 3-D Case

---

**Input:**  $(\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n)})$ : mutually non-dominated  $\mathbb{R}^3$ -points sorted by third coordinate ( $y_3$ ) in descending order

**Output:**  $S_3^{(1)}, \dots, S_3^{(i)}, \dots, S_3^{(N_3)}$

```

1:   $\mathbf{y}^{(n+1)} = (\infty, \infty, r_3)$  ;
2:  Initialize AVL tree T for 3-D points
    Insert  $\mathbf{y}^{(1)}, (\infty, r_2, \infty)^T$  and  $(r_1, \infty, \infty)^T$  into T;
3:  Initialize the number of integration slices  $n_b = 1$ ;
4:  Initialize  $EHVI = 0$ ;
5:  for  $i = 2$  to  $n + 1$  do                                     /* Main loop */
6:      Retrieve the following information from tree T:
7:          r: index of the successor of  $\mathbf{y}^{(i)}$  in  $x$ -coordinate (right neighbour);
8:          t: index of the successor of  $\mathbf{y}^{(i)}$  in  $y$ -coordinate (left neighbour);
9:          d[1],  $\dots$ , d[s]: indices of points dominated by  $\mathbf{y}^{(i)}$  in  $y_1y_2$ -plane,
              sorted ascendingly in the first coordinate( $y_1$ );
10:      $S_3^{(n_b)}.l_3 = y_3^{(i)}, \quad S_3^{(n_b)}.u_2 = y_2^{(i)}, \quad S_3^{(n_b)}.u_3 = \infty$  ;
11:     if  $s == 0$  then                                           /* Case 1 */
12:          $S_3^{(n_b)}.l_1 = y_1^{(t)}, \quad S_3^{(n_b)}.l_2 = y_2^{(r)}, \quad S_3^{(n_b)}.u_1 = y_1^{(i)}$ ;
13:          $n_b = n_b + 1$  ;
14:     else                                                       /* Case 2 */
15:         for  $j = 1$  to  $s + 1$  do
16:             if  $j == 1$  then
17:                  $S_3^{(n_b)}.l_1 = y_1^{(t)}, \quad S_3^{(n_b)}.l_2 = y_2^{(d[1])}, \quad S_3^{(n_b)}.u_1 = y_1^{(d[1])}$ ;
18:             else if  $j == s + 1$  then
19:                  $S_3^{(n_b)}.l_1 = y_1^{(d[s])}, \quad S_3^{(n_b)}.l_2 = y_2^{(r)}, \quad S_3^{(n_b)}.u_1 = y_1^{(i)}$ ;
20:             else
21:                  $S_3^{(n_b)}.l_1 = y_1^{(d[j-1])}, \quad S_3^{(n_b)}.l_2 = y_2^{(d[j])}, \quad S_3^{(n_b)}.u_1 = y_1^{(d[j])}$ ;
22:              $n_b = n_b + 1$  ;
23:     Discard  $\mathbf{y}^{(d[1])}, \dots, \mathbf{y}^{(d[s])}$  from tree T;
24:     Insert  $\mathbf{y}^{(i)}$  in tree T.
25: Return  $S_3^{(1)}, \dots, S_3^{(i)}, \dots, S_3^{(N_3)}$ 

```

---

### 3. EFFICIENT EHVI CALCULATION

---

#### 3.3.2 High Dimensional case

In higher dimensional cases, the non-dominated space can be partitioned into axis aligned hyperboxes, similar to 3-D case. In  $d$  dimensional case, the hyperboxes can be denoted by  $S_d^{(1)}, \dots, S_d^{(i)}, \dots, S_d^{N_d}$  with their lower bound  $(\mathbf{l}^{(1)}, \dots, \mathbf{l}^{(N_d)})$  and upper bound  $(\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(N_d)})$ . Here,  $N_d$  is the number of hyperboxes. The hyper-integral box  $S_d^{(i)}$  is defined as:

$$S_d^{(i)} = (\mathbf{l}_d^{(i)}, \mathbf{u}_d^{(i)}) = ((l_1^{(i)}, \dots, l_d^{(i)})^T, (u_1^{(i)}, \dots, \infty)^T) \quad i = 1, \dots, N_d \quad (3-6)$$

An efficient algorithm for partitioning high-dimensional non-dominated space is proposed in this chapter. This new proposed algorithm is based on two state-of-the-art algorithms DKL17 [70] by Dächert et al. and LKF17 [71] by Lacour et al. Here, algorithm DKL17 is an efficient algorithm to locate the local lower bound points in a dominated space for maximization problem, based on a specific neighborhood structure among local lower bounds. Meanwhile, LKF17 is an efficient algorithm to calculate hypervolume improvement by partitioning the dominated space. In other words, LKF17 is also efficient to partition the dominated space and provides the boundary information for each hyperbox in the dominated space.

---

#### Algorithm 5: Partitioning non-dominated space for high dimensional cases

---

**Input:** Pareto-front approximation  $\mathcal{P}$  (maximization problem), a reference point  $\mathbf{r}$

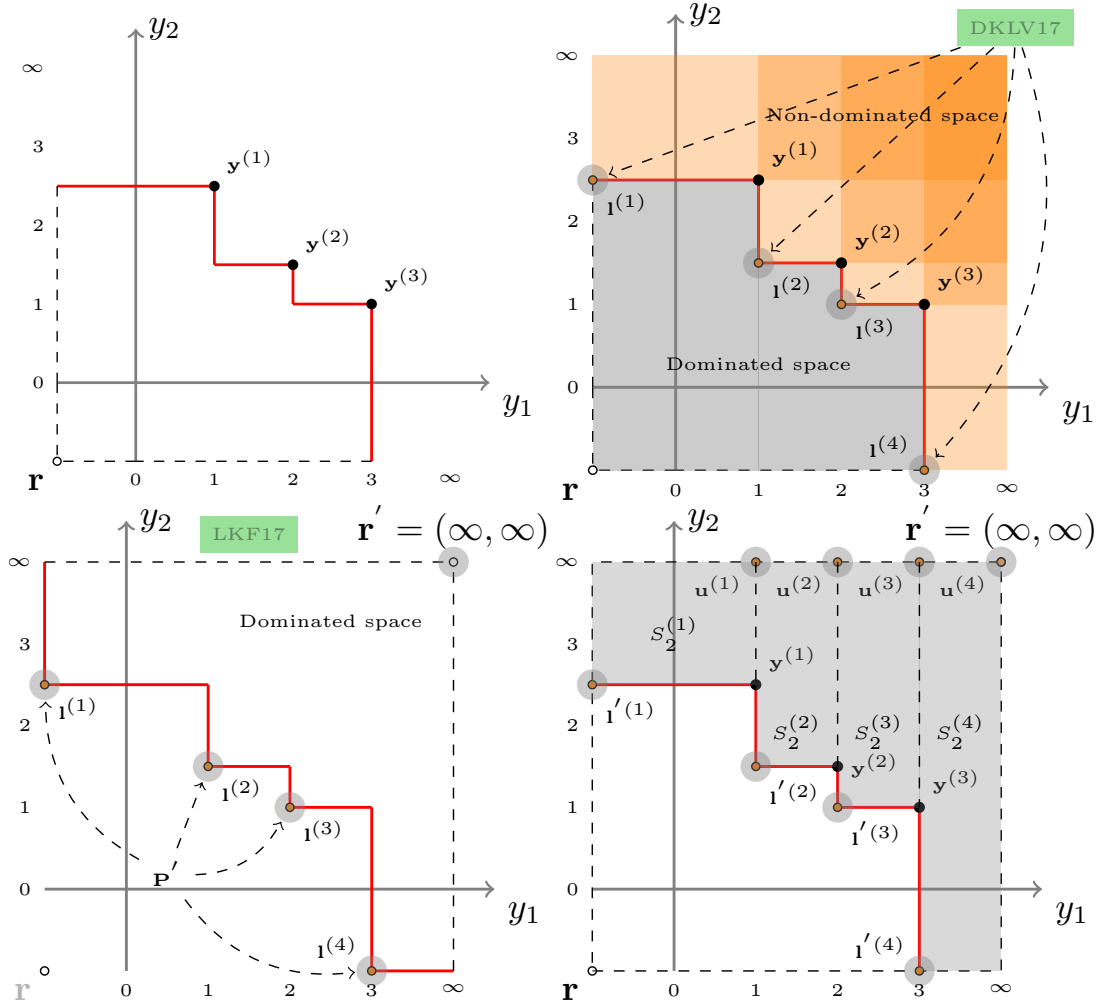
**Output:** Hyperboxes  $S_d$

- 1: Locate local lower bound points  $\mathbf{L}$ :  $\mathbf{L} = DKL17(\mathcal{P}, \mathbf{r})$ ;
  - 2: Set new Pareto front  $\mathcal{P}'$  using  $\mathbf{L}$ :  $\mathcal{P}' = \mathbf{L}$  ;
  - 3: Set a reference point  $\mathbf{r}'$ :  $\mathbf{r}' = \{\infty\}^d$  ;
  - 4: Get local lower bound points  $\mathbf{L}'$  and local upper bound points  $\mathbf{U}'$ :  
 $(\mathbf{L}', \mathbf{U}') = LKF17(\mathcal{P}', \mathbf{r}')$  ;
  - 5:  $S_d = (\mathbf{L}', \mathbf{U}')$  ;
  - 6: Return  $S_d$
- 

The basic idea of the proposed algorithm to partition high-dimensional non-dominated space is transforming the problem of partitioning non-dominated space into the problem of partitioning the dominated space by introducing an intermediate Pareto-front approximation  $\mathcal{P}'$ . This transforming is done by the following

### 3.3 Non-dominated Space Partitioning Algorithm

steps. Suppose that we have a current Pareto-front approximation  $\mathcal{P}$  and we want to partition the non-dominated space of  $\mathcal{P}$ . Firstly, DKL17 is applied to locate the local lower bound points ( $\mathbf{L}$ ) of  $\mathcal{P}$  in dominated space. If we regard the local lower bound points  $\mathbf{L}$  as a new Pareto-front approximation  $\mathcal{P}'$ , the dominated space of  $\mathcal{P}'$  is exact the non-dominated space of  $\mathcal{P}$ . The pseudo code of partitioning non-dominated space for high dimensional cases is shown in Algorithm 5.



**Figure 3.5:** The illustration of partitioning non-dominated space for high dimensional case. Above left: Pareto-front approximation  $\mathcal{P}$ . Above right: Locating  $\mathbf{L}$  points using DKL17. Below left: Partition the dominated space of  $\mathcal{P}'$  using LKF17. Below right: The partitioned non-dominated space of  $\mathcal{P}$ .

**Example 3.3** Figure 3.5 illustrates Algorithm 5. For the 2-D maximization case,

### 3. EFFICIENT EHVI CALCULATION

---

suppose the Pareto-front approximation is  $\mathcal{P}$ , which is composed by  $\mathbf{y}^{(1)} = (1, 2.5)$ ,  $\mathbf{y}^{(2)} = (2, 1.5)$  and  $\mathbf{y}^{(3)} = (3, 1)$ . The reference point is  $\mathbf{r} = (0, 0)$ , see Figure 3.5 (above left). Use DKL17 to locate the local lower bound points  $\mathbf{l}$ , which consist of  $\mathbf{l}^{(1)} = (0, 2.5)$ ,  $\mathbf{l}^{(2)} = (1, 1.5)$ ,  $\mathbf{l}^{(3)} = (2, 1)$  and  $\mathbf{l}^{(4)} = (3, 0)$ , see Figure 3.5 (above right). Regard all the local lower bound points  $\mathbf{l}$  as the elements of a new Pareto-front approximation set  $\mathcal{P}' = (\mathbf{l}^{(1)}, \dots, \mathbf{l}^{(4)})$ . Set a new reference point  $\mathbf{r}' = (\infty, \infty)$  and utilize LKF17 to partition the dominated space of  $\mathcal{P}'$ , considering minimization case, see Figure 3.5 (below left). Then the partitioned non-dominated space of  $\mathcal{P}$  is actually the partitioned dominated space of  $\mathcal{P}'$ , see Figure 3.5 (below right).

### 3.4 Computing the integrals

Before introducing the EHVI formula deviation, it is useful to define an important function  $\vartheta$ :

**Definition 3.4 ( $\vartheta$  function)** Let  $\phi(s) = 1/\sqrt{2\pi}e^{-\frac{1}{2}s^2}$  ( $s \in \mathbb{R}$ ) denote the probability density function (PDF) of the standard normal distribution. Moreover, let  $\Phi(s) = \frac{1}{2} \left( 1 + \operatorname{erf}\left(\frac{s}{\sqrt{2}}\right) \right)$  denote its cumulative probability distribution function (CDF), and  $\operatorname{erf}$  is Gaussian error function. The general normal distribution with mean  $\mu$  and standard deviation  $\sigma$  has as PDF,  $\phi_{\mu,\sigma}(s) = \phi_{\mu,\sigma}(s) = \frac{1}{\sigma} \phi\left(\frac{s-\mu}{\sigma}\right)$  and its CDF is  $\Phi_{\mu,\sigma}(s) = \Phi\left(\frac{s-\mu}{\sigma}\right)$ , the integration box (or hyper-box)  $B_i$  consist of a lower bound point  $\mathbf{l}^{(i)}$  and a upper bound point  $\mathbf{u}^{(i)}$ . Then the function  $\vartheta(l_k^{(i)}, u_k^{(i)}, \sigma_k, \mu_k)$  is defined as:

$$\begin{aligned} \vartheta(l_k^{(i)}, u_k^{(i)}, \sigma_k, \mu_k) &:= \int_{y_k=u_k^{(i)}}^{\infty} \lambda_1[B_i \cap \Delta(y_k)] \cdot PDF_{\mu_k, \sigma_k}(y_k) dy_k \\ &= \int_{y_k=u_k^{(i)}}^{\infty} (u_k^{(i)} - l_k^{(i)}) \cdot PDF_{\mu_k, \sigma_k}(y_k) dy_k \\ &= (u_k^{(i)} - l_k^{(i)}) \cdot \left( 1 - \Phi\left(\frac{u_k^{(i)} - \mu_k}{\sigma_k}\right) \right) \quad \text{where } k = 1, \dots, d-1 \end{aligned} \tag{4-7}$$

In the definition of  $\vartheta$  function,  $\lambda_1[B_i \cap \Delta(y_k)]$  is the *Hypervolume Improvement*

of the  $i^{th}$  integration box in dimension  $k$ , i.e., a 1-D *Hypervolume Improvement*. Considering the partitioning methods in Chapter 3.3,  $\lambda_1[B_i \cap \Delta(y_k)] = |[l_k^{(i)}, u_k^{(i)}] \cap [l_k^{(i)}, y_k]| = \min\{u_k^{(i)}, y_k\} - l_k^{(i)}$ , where  $k = 1, \dots, d$ . The idea of introducing the  $\vartheta$  function is that the improvement for  $\int_{y_k=u_k^{(i)}}^{\infty} \lambda_1[B_i \cap \Delta(y_k)] \cdot PDF_{\mu_k, \sigma_k}(y_k) dy_k$ , where  $k = 1, \dots, d-1$ , is a constant, that is  $\vartheta$  itself. This is very useful during the calculation process of EHVI, because  $\vartheta$  function for each integration box  $B_i$  can be calculated once and reused to save calculation time.

#### 3.4.1 2-D EHVI

According to the definition of the 2-D integration slice in Equation 3.3.1, the *Hypervolume Improvement*  $\mathbf{y} \in \mathbb{R}^2$  for the 2-D case is:

$$\text{HVI}_2(\mathcal{P}, \mathbf{y}, \mathbf{r}) = \sum_{i=1}^{N_2} \lambda_2[S_2^{(i)} \cap \Delta(\mathbf{y}, \mathcal{P}, \mathbf{r})] \quad (4-8)$$

This gives rise to the compact integral for the original EHVI:

$$\text{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) = \int_{y_1=-\infty}^{\infty} \int_{y_2=-\infty}^{\infty} \sum_{i=1}^{N_2} \lambda_2[S_2^{(i)} \cap \Delta(\mathbf{y})] \cdot PDF_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \quad (4-9)$$

Here  $\mathbf{y} = (y_1, y_2)$ , the intersection of  $S_2^{(i)}$  with  $\Delta(y_1, y_2)$  is non-empty if and only if  $(\mathbf{y})$  dominates the lower left corner of  $S_2^{(i)}$ . In other words, if and only if  $\mathbf{y}$  is located in the rectangle with lower left corner  $(l_1^{(i)}, l_2^{(i)})$  and upper right corner  $(\infty, \infty)$ . See Figure 3.2 (right) for an illustration. Therefore:

$$\text{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) = \sum_{i=1}^{N_2} \int_{y_1=l_1^{(i)}}^{\infty} \int_{y_2=l_2^{(i)}}^{\infty} \lambda_2[S_2^{(i)} \cap \Delta(\mathbf{y})] \cdot PDF_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \quad (4-10)$$

In Equation (4-10), the summation is done after integration. This is allowed, because integration is a linear mapping. Moreover, the integration interval  $\int_{y_1=l_1^{(i)}}^{\infty}$  can be divided into  $(\int_{y_1=l_1^{(i)}}^{u_1^{(i)}} + \int_{y_1=u_1^{(i)}}^{\infty})$ , because the *Hypervolume Improvement*  $\lambda_1[S_2^{(i)} \cap \Delta(y_1)]$  differs in these two integration intervals. Then Equation (4-10)

### 3. EFFICIENT EHVI CALCULATION

---

is expressed by:

$$\text{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) = \sum_{i=1}^{N_2} \int_{y_1=l_1^{(i)}}^{u_1^{(i)}} \int_{y_2=l_2^{(i)}}^{\infty} \lambda_2[S_2^{(i)} \cap \Delta(\mathbf{y})] \cdot \text{PDF}_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} + \quad (4-11)$$

$$\sum_{i=1}^{N_2} \int_{y_1=u_1^{(i)}}^{\infty} \int_{y_2=l_2^{(i)}}^{\infty} \lambda_2[S_2^{(i)} \cap \Delta(\mathbf{y})] \cdot \text{PDF}_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \quad (4-12)$$

Here  $\lambda_1[B_i \cap \Delta(y_k)]$  is the *Hypervolume Improvement* in dimension  $k$ , i.e., a 1-D *Hypervolume Improvement*. According to the definition of *Hypervolume Improvement*,  $\lambda_1[B_i \cap \Delta(y_k)]$  is constant and it is  $(u_1^{(i)} - l_1^{(i)})$ . Therefore, the *Expected Improvement* in dimension  $y_1$  is also a constant and it is:  $\vartheta(l_1^{(i)}, u_1^{(i)}, \sigma_1, \mu_1)$ . Recall  $\Psi_{\infty}$  function, then the Equation (4-11) and (4-12) are:

$$\text{Cp.}(4-11) = \sum_{i=1}^{N_2} \left( \Psi_{\infty}(l_1^{(i)}, l_1^{(i)}, \mu_1, \sigma_1) - \Psi_{\infty}(l_1^{(i)}, u_1^{(i)}, \mu_1, \sigma_1) \right) \cdot \Psi_{\infty}(l_2^{(i)}, l_2^{(i)}, \mu_2, \sigma_2) \quad (4-13)$$

$$\text{Cp.}(4-12) = \sum_{i=1}^{N_2} \vartheta(l_1^{(i)}, u_1^{(i)}, \mu_1, \sigma_1) \cdot \Psi_{\infty}(l_2^{(i)}, l_2^{(i)}, \mu_2, \sigma_2) \quad (4-14)$$

#### 3.4.2 3-D EHVI

Given a partitioning of the non-dominated space into integration slices  $S_3^{(1)}, \dots, S_3^{(i)}, \dots, S_3^{(2n+1)}$ , the part of the integral related to each of the integration slices can be computed separately. To see how this can be done, the *Hypervolume Improvement* of a point  $\mathbf{y} \in \mathbb{R}^3$  is rewritten as:

$$\text{HVI}_3(\mathcal{P}, \mathbf{y}, \mathbf{r}) = \sum_{i=1}^{N_3} \lambda_3[S_3^{(i)} \cap \Delta(\mathbf{y})] \quad (4-15)$$

where  $\Delta\mathbf{y}$  is the part of the objective space that is dominated by  $\mathbf{y}$ . The HVI expression in the definition of EHVI in Equation (1-2) can be replaced by  $\text{HVI}_3$  in Equation (4-15):

$$\text{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) = \sum_{i=1}^{N_3} \int_{y_1=l_1^{(i)}}^{\infty} \int_{y_2=l_2^{(i)}}^{\infty} \int_{y_3=l_3^{(i)}}^{\infty} \lambda_3[S_3^{(i)} \cap \Delta(\mathbf{y})] \cdot \text{PDF}_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \quad (4-16)$$

### 3.4 Computing the integrals

Similar to the 2-D case, we can divide the integration interval  $\int_{y_1=l_1^{(i)}}^{\infty}$  and  $\int_{y_2=l_2^{(i)}}^{\infty}$  into  $(\int_{y_1=l_1^{(i)}}^{u_1^{(i)}} + \int_{y_1=u_1^{(i)}}^{\infty})$  and  $(\int_{y_2=l_2^{(i)}}^{u_2^{(i)}} + \int_{y_2=u_2^{(i)}}^{\infty})$ , respectively. Based on this division, Equation (4-16) can be expressed by:

$$\text{Cp.4} - 16 = \sum_{i=1}^{N_3} \int_{y_1=l_1^{(i)}}^{u_1^{(i)}} \int_{y_2=l_2^{(i)}}^{u_2^{(i)}} \int_{y_3=l_3^{(i)}}^{\infty} \lambda_3[S_3^{(i)} \cap \Delta(\mathbf{y})] \cdot \text{PDF}_{\mu, \sigma}(\mathbf{y}) d\mathbf{y} + \quad (4-17)$$

$$\sum_{i=1}^{N_3} \int_{y_1=l_1^{(i)}}^{u_1^{(i)}} \int_{y_2=u_2^{(i)}}^{\infty} \int_{y_3=l_3^{(i)}}^{\infty} \lambda_3[S_3^{(i)} \cap \Delta(\mathbf{y})] \cdot \text{PDF}_{\mu, \sigma}(\mathbf{y}) d\mathbf{y} + \quad (4-18)$$

$$\sum_{i=1}^{N_3} \int_{y_1=u_1^{(i)}}^{\infty} \int_{y_2=l_2^{(i)}}^{u_2^{(i)}} \int_{y_3=l_3^{(i)}}^{\infty} \lambda_3[S_3^{(i)} \cap \Delta(\mathbf{y})] \cdot \text{PDF}_{\mu, \sigma}(\mathbf{y}) d\mathbf{y} + \quad (4-19)$$

$$\sum_{i=1}^{N_3} \int_{y_1=u_1^{(i)}}^{\infty} \int_{y_2=u_2^{(i)}}^{\infty} \int_{y_3=l_3^{(i)}}^{\infty} \lambda_3[S_3^{(i)} \cap \Delta(\mathbf{y})] \cdot \text{PDF}_{\mu, \sigma}(\mathbf{y}) d\mathbf{y} \quad (4-20)$$

Recalling the definition of  $\vartheta$  function and calculation of  $\lambda_1[B_i \cap \Delta(y_k)]$ , component (4-17) can be written as follows:

$$\begin{aligned} \text{Cp.4} - 17 = & \sum_{i=1}^{N_3} (\Psi_{\infty}(l_1^{(i)}, l_1^{(i)}, \mu_1, \sigma_1) - \Psi_{\infty}(l_1^{(i)}, u_1^{(i)}, \sigma_1, \mu_1)) \cdot \\ & (\Psi_{\infty}(l_2^{(i)}, l_2^{(i)}, \mu_2, \sigma_2) - \Psi_{\infty}(l_2^{(i)}, u_2^{(i)}, \sigma_2, \mu_2)) \cdot \Psi_{\infty}(l_3^{(i)}, l_3^{(i)}, \mu_3, \sigma_3) \end{aligned} \quad (4-21)$$

Similar to the derivation of Component (4-17), components (4-18), (4-19) and (4-20) can be written as follows:

$$\begin{aligned} \text{Cp.4} - 18 = & \sum_{i=1}^{N_3} (\Psi_{\infty}(l_1^{(i)}, l_1^{(i)}, \mu_1, \sigma_1) - \Psi_{\infty}(l_1^{(i)}, u_1^{(i)}, \sigma_1, \mu_1)) \cdot \vartheta(l_2^{(i)}, u_2^{(i)}, \sigma_2, \mu_2) \cdot \\ & \Psi_{\infty}(l_3^{(i)}, l_3^{(i)}, \mu_3, \sigma_3) \end{aligned} \quad (4-22)$$

$$\begin{aligned} \text{Cp.4} - 19 = & \sum_{i=1}^{N_3} \vartheta(l_1^{(i)}, u_1^{(i)}, \sigma_1, \mu_1) \cdot (\Psi_{\infty}(l_2^{(i)}, l_2^{(i)}, \mu_2, \sigma_2) - \Psi_{\infty}(l_2^{(i)}, u_2^{(i)}, \sigma_2, \mu_2)) \cdot \\ & \Psi_{\infty}(l_3^{(i)}, l_3^{(i)}, \mu_3, \sigma_3) \end{aligned} \quad (4-23)$$

$$\text{Cp.4} - 20 = \sum_{i=1}^{N_3} \vartheta(l_1^{(i)}, u_1^{(i)}, \sigma_1, \mu_1) \cdot \vartheta(l_2^{(i)}, u_2^{(i)}, \sigma_2, \mu_2) \cdot \Psi_{\infty}(l_3^{(i)}, l_3^{(i)}, \mu_3, \sigma_3) \quad (4-24)$$

The final EHVI formula is the sum of Components (4-21), (4-22), (4-23) and (4-24).

### 3. EFFICIENT EHVI CALCULATION

---

#### 3.4.3 High Dimensional Case

The interval of integration in each coordinate, except the last coordinate, can be divided into two parts:  $[l, u]$  and  $[u, \infty]$ . Therefore, the equation for EHVI for each hyperboxes can be decomposed into  $2^{d-1}$  parts. For the interval of  $[u, \infty]$ , the improvements  $(\lambda_k[S_d^{(i)} \cap \Delta(y_k)])$  are constant numbers, and the  $\Psi$  function can be simplified by calculating function  $\Phi$  and the improvement in these coordinate. For the last coordinate, there is no need to separate the interval, because the improvement in this coordinate  $(\lambda_m[S_d^{(i)} \cap \Delta(y_m)])$  is a variable in  $[l, \infty]$ .

According to the definition of high dimensional integral boxes in Section 3.3.2, the formula of EHVI for a high dimensional case( $d \geq 4$ ) can be calculated by the following equation:

$$\begin{aligned}
 & \text{EHVI}(\mu, \sigma, \mathcal{P}, \mathbf{r}) \\
 &= \sum_{i=1}^{N_d} \int_{y_1=l_1^{(i)}}^{\infty} \cdots \int_{y_d=l_d^{(i)}}^{\infty} \lambda_d[S_d^{(i)} \cap \Delta(y_1, \dots, y_d)] \cdot PDF_{\mu, \sigma}(\mathbf{y}) d\mathbf{y} \\
 &= \sum_{i=1}^{N_d} \left( \left( \int_{y_1=l_1^{(i)}}^{\infty} + \int_{y_1=u_1^{(i)}}^{\infty} \right) \cdots \left( \int_{y_{d-1}=l_{d-1}^{(i)}}^{y_{d-1}=u_{d-1}^{(i)}} + \int_{y_{d-1}=u_{d-1}^{(i)}}^{\infty} \right) \cdot \int_{y_d=l_d^{(i)}}^{\infty} \right) \cdot \lambda_d[S_d^{(i)} \cap \Delta(y_1, \dots, y_d)] \cdot PDF_{\mu, \sigma}(\mathbf{y}) d\mathbf{y} \\
 &= \begin{pmatrix} \int_{y_1=l_1^{(1)}}^{u_1^{(1)}} \int_{y_2=l_2^{(1)}}^{u_2^{(1)}} \cdots \int_{y_{d-2}=l_{d-2}^{(1)}}^{u_{d-2}^{(1)}} \int_{y_{d-1}=l_{d-1}^{(1)}}^{u_{d-1}^{(1)}} \int_{y_d=l_d^{(1)}}^{u_d^{(1)}} \\ \int_{y_1=l_1^{(2)}}^{u_1^{(2)}} \int_{y_2=l_2^{(2)}}^{u_2^{(2)}} \cdots \int_{y_{d-2}=l_{d-2}^{(2)}}^{u_{d-2}^{(2)}} \int_{y_{d-1}=l_{d-1}^{(2)}}^{u_{d-1}^{(2)}} \int_{y_d=l_d^{(2)}}^{u_d^{(2)}} \\ \int_{y_1=l_1^{(3)}}^{u_1^{(3)}} \int_{y_2=l_2^{(3)}}^{u_2^{(3)}} \cdots \int_{y_{d-2}=l_{d-2}^{(3)}}^{u_{d-2}^{(3)}} \int_{y_{d-1}=l_{d-1}^{(3)}}^{u_{d-1}^{(3)}} \int_{y_d=l_d^{(3)}}^{u_d^{(3)}} \\ \vdots \quad \vdots \quad \ddots \quad \vdots \quad \vdots \quad \vdots \\ \int_{y_1=l_1^{(N_d)}}^{\infty} \int_{y_2=l_2^{(N_d)}}^{\infty} \cdots \int_{y_{d-2}=l_{d-2}^{(N_d)}}^{u_{d-2}^{(N_d)}} \int_{y_{d-1}=l_{d-1}^{(N_d)}}^{\infty} \int_{y_d=l_d^{(N_d)}}^{u_d^{(N_d)}} \end{pmatrix} \lambda_d[S_d^{(i)} \cap \Delta(y_1, \dots, y_d)] \cdot PDF_{\mu, \sigma}(\mathbf{y}) d\mathbf{y}
 \end{aligned}$$

### 3. EFFICIENT EHVI CALCULATION

[illegible]

### 3.4 Computing the integrals

$$\begin{aligned}
&= \sum_{i=1}^{N_d} \left( \sum_{j=0}^{2^{d-1}-1} \left( \prod_{k=1}^{d-1} \omega(i, k, C_k^{(j)2}) \cdot \Psi_{\infty}(l_d^{(i)}, l_d^{(i)}, \mu_d, \sigma_d) \right) \right) \\
&= \begin{pmatrix} \omega(i, 1, C_1^{(0)2}) & \omega(i, 2, C_2^{(0)2}) & \cdots & \omega(i, d-2, C_{d-2}^{(0)2}) & \omega(i, d-1, C_{d-1}^{(0)2}) & \Psi_{\infty}(l_d^{(i)}, l_d^{(i)}, \mu_d, \sigma_d) & + \\ \omega(i, 1, C_1^{(1)2}) & \omega(i, 2, C_2^{(1)2}) & \cdots & \omega(i, d-2, C_{d-2}^{(1)2}) & \omega(i, d-1, C_{d-1}^{(1)2}) & \Psi_{\infty}(l_d^{(i)}, l_d^{(i)}, \mu_d, \sigma_d) & + \\ \omega(i, 1, C_1^{(2)2}) & \omega(i, 2, C_2^{(2)2}) & \cdots & \omega(i, d-2, C_{d-2}^{(2)2}) & \omega(i, d-1, C_{d-1}^{(2)2}) & \Psi_{\infty}(l_d^{(i)}, l_d^{(i)}, \mu_d, \sigma_d) & + \\ \omega(i, 1, C_1^{(3)2}) & \omega(i, 2, C_2^{(3)2}) & \cdots & \omega(i, d-2, C_{d-2}^{(3)2}) & \omega(i, d-1, C_{d-1}^{(3)2}) & \Psi_{\infty}(l_d^{(i)}, l_d^{(i)}, \mu_d, \sigma_d) & + \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \omega(i, 1, C_1^{(2^{d-1}-4)2}) & \omega(i, 2, C_2^{(2^{d-1}-4)2}) & \cdots & \omega(i, d-2, C_{d-2}^{(2^{d-1}-4)2}) & \omega(i, d-1, C_{d-1}^{(2^{d-1}-4)2}) & \Psi_{\infty}(l_d^{(i)}, l_d^{(i)}, \mu_d, \sigma_d) & + \\ \omega(i, 1, C_1^{(2^{d-1}-3)2}) & \omega(i, 2, C_2^{(2^{d-1}-3)2}) & \cdots & \omega(i, d-2, C_{d-2}^{(2^{d-1}-3)2}) & \omega(i, d-1, C_{d-1}^{(2^{d-1}-3)2}) & \Psi_{\infty}(l_d^{(i)}, l_d^{(i)}, \mu_d, \sigma_d) & + \\ \omega(i, 1, C_1^{(2^{d-1}-2)2}) & \omega(i, 2, C_2^{(2^{d-1}-2)2}) & \cdots & \omega(i, d-2, C_{d-2}^{(2^{d-1}-2)2}) & \omega(i, d-1, C_{d-1}^{(2^{d-1}-2)2}) & \Psi_{\infty}(l_d^{(i)}, l_d^{(i)}, \mu_d, \sigma_d) & + \\ \omega(i, 1, C_1^{(2^{d-1}-1)2}) & \omega(i, 2, C_2^{(2^{d-1}-1)2}) & \cdots & \omega(i, d-2, C_{d-2}^{(2^{d-1}-1)2}) & \omega(i, d-1, C_{d-1}^{(2^{d-1}-1)2}) & \Psi_{\infty}(l_d^{(i)}, l_d^{(i)}, \mu_d, \sigma_d) & + \end{pmatrix}
\end{aligned}
\tag{4-26}$$

### 3. EFFICIENT EHVI CALCULATION

---

In the component of (4-25), the integral of each dimension  $\int_{y_k=l_k^{(i)}}^{u_k^{(i)}} \lambda_k[S_k^{(i)} \cap \Delta(y_1, \dots, y_k)] \cdot PDF_{\mu, \sigma}(\mathbf{y}) d\mathbf{y}$ ,  $1 \leq k \leq d-1$  has two and only two different expressions ( $\Psi_\infty$  or  $\vartheta$ ), except for the last dimension, the expression of  $\int_{y_d=l_d^{(i)}}^{u_d^{(i)}} \lambda_d[S_d^{(i)} \cap \Delta(y_1, \dots, y_d)] \cdot PDF_{\mu, \sigma}(\mathbf{y}) d\mathbf{y}$  is always  $\Psi_\infty$ . The final expression of EHVI is the sum of the combination of the product of each different expressions. Since the integral of dimension  $1 \leq k \leq d-1$  has two different expressions and dimension  $k = d$  has one expression, the final EHVI expression is the sum of  $2^{d-1}$  terms.

In Equation (4-26),  $j_2$  stands for the binary string of  $j$  in the integer system. The length of  $j_2$  is  $d-1$ .  $C_k^{(j)_2}$  is a binary bit and represents the  $k$ -th bit of  $j$  in binary string. For example, if  $d = 5$ ,  $j = 8$  and  $k = 4$ , then  $j_2 = (1 \ 0 \ 0 \ 0)$  and  $C_k^{(j)_2} = 1$ . Still in Equation (4-26),  $\omega(i, k, C_k^{(j)_2})$  is defined as:

$$\omega(i, k, C_k^{(j)_2}) := \begin{cases} \Psi_\infty(l_k^{(i)}, l_k^{(i)}, \mu_k, \sigma_k) - \Psi_\infty(l_k^{(i)}, u_k^{(i)}, \sigma_k, \mu_k) & \text{if } C_k^{(j)_2} = 0 \\ \vartheta_d(l_k^{(i)}, u_k^{(i)}, \sigma_k, \mu_k) & \text{if } C_k^{(j)_2} = 1 \end{cases} \quad (4-27)$$

Equation (4-26) shows how to calculate EHVI in the case of  $d$  objectives, and based on it, the runtime complexity of the proposed algorithm can be calculated. The exact EHVI is calculated by the sum of  $\prod_{k=1}^{d-1} \omega(i, k, C_k^{(j)_2}) \cdot \Psi_\infty(l_d^{(i)}, l_d^{(i)}, \mu_d, \sigma_d)$  for  $2^{d-1}$  times, which performs  $O(1)$  for each hyperboxes calculation. Currently, the minimum number of hyperboxes  $N_d$ ,  $d \geq 4$  for a non-dominated space is still unknown. It is hypothesized by the author that  $N_d$  is equal to the number of the local lower bound points, which can be calculated by DKL17 algorithm. The upper bound of runtime complexity is  $O(n\tau)$ , where  $O(\tau)$  is the computation complexity of the search algorithm. For the case of  $d = 1, 2, 3$ ,  $O(\tau) \in O(\log n)$ .

### 3.5 Other Related Criterion

*Probability of Improvement* is another important criterion in MOBGO, and it was first introduced by Stuckman [72], and then generalized by Emmerich et al. [53] to multi-objective optimization. It was also considered in MOBGO in Couckuyt et al. [58] and in Keane et al. [55]. It is defined as:

**Definition 3.5 (Probability of Improvement)** *Given parameters of the multivariate predictive distribution  $\mu, \sigma$  and the Pareto-front approximation  $\mathcal{P}$ , the*

Probability of Improvement (*PoI*) is defined as:

$$PoI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}) := \int_{\mathbb{R}^d} PDF_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \quad (5-28)$$

where  $PDF_{\boldsymbol{\mu}, \boldsymbol{\sigma}}$  is the multivariate independent normal distribution for mean values  $\boldsymbol{\mu} \in \mathbb{R}^d$ , and standard deviations  $\boldsymbol{\sigma} \in \mathbb{R}_+^d$ .

According to the partitioning method in Section 3.4.3, the calculation of PoI can be achieved by the following expression:

$$\begin{aligned} PoI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}) &= \int_{y_1=-\infty}^{\infty} \cdots \int_{y_d=-\infty}^{\infty} PDF_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) dy_1 \cdots dy_d \\ &= \sum_{i=1}^{N_d} \prod_{j=1}^d \Phi\left(\frac{u_j^{(i)} - \mu_j}{\sigma_j}\right) - \Phi\left(\frac{l_j^{(i)} - \mu_j}{\sigma_j}\right) \end{aligned} \quad (5-29)$$

Here,  $N_d$  is the number of integration slices, and  $N_2 = n + 1$ ,  $N_3 = 2n + 1$  for 2-D and 3-D cases respectively. Since PoI is a reference-free indicator<sup>1</sup>, reference point  $\mathbf{r} = \{-\infty\}^d$  should be set in order to obtain the correct boundary information  $(\mathbf{l}_d, \mathbf{u}_d)$ .

## 3.6 Empirical Experiments

### 3.6.1 Speed Comparison

Three EHVI calculation algorithms, CDD13 [58], IRS\_fast [1] and KMAC<sup>1</sup> [4, 5], are compared using the same benchmarks in this experiment. The test benchmarks from Emmerich and Fonseca [20] are used to generate Pareto-front sets. The Pareto-front sets and evaluated points were randomly generated based on CONVEXSPHERICAL and CONCAVESPHERICAL functions.

The parameters:  $\sigma_d = 2.5$ ,  $\mu_d = 10$ ,  $d = 2, \dots, 5$  were used in the experiments. Pareto front sizes  $|P| \in \{10, 20, \dots, 200\}$  and the number of predictions (candidate points) Batch Size<sup>1</sup>  $\in \{1\}$  are used together with  $\sigma_d$  and  $\mu_d$ . Ten trials were

---

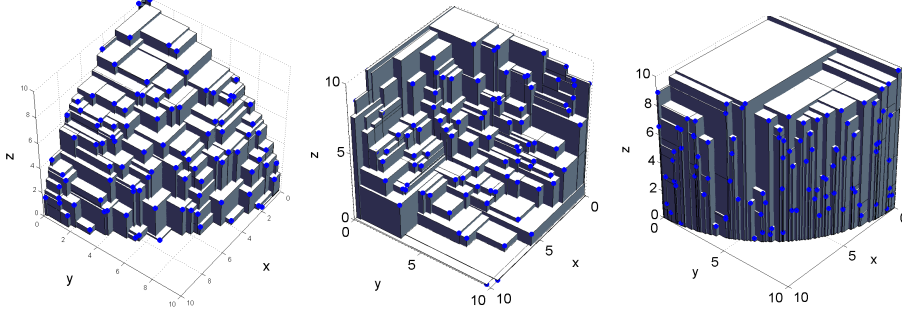
<sup>1</sup>This means that the integration space for PoI is unbounded and covers the entire non-dominated space.

<sup>1</sup>KMAC stands for the authors' given names.

<sup>1</sup>Batch Size means the number of the evaluated points under the same Pareto-front approximation set.

### 3. EFFICIENT EHVI CALCULATION

---



**Figure 3.6:** Randomly generated fronts of type CONVEXSPHERICAL, CONCAVE-SPHERICAL, and CLIFF3D from [20] with  $|P| = 100$  (left, middle and right).

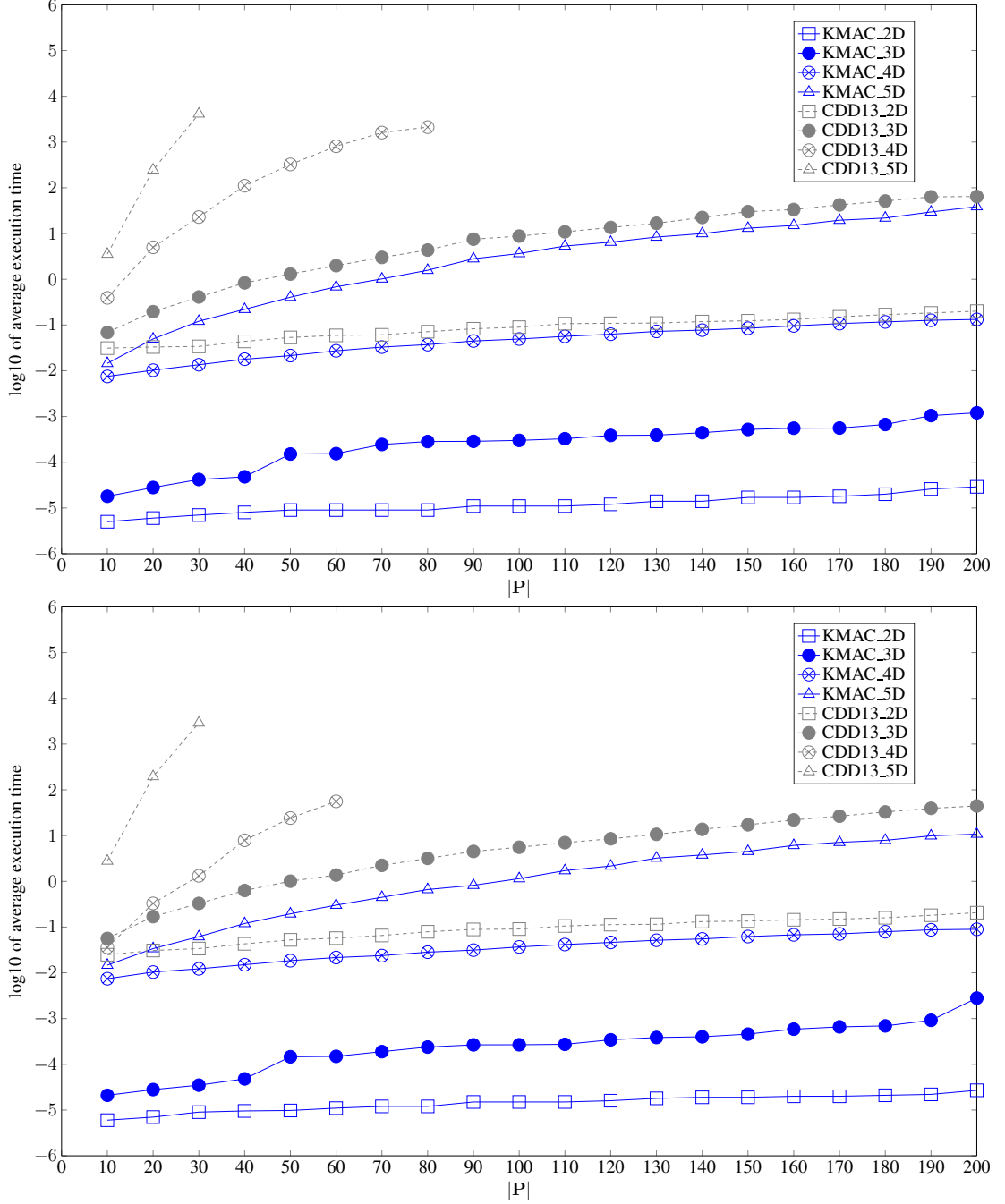
randomly generated by the same parameters, and average runtimes (10 runs) for whole trails with the same parameters were computed. All the experiments were performed on the same computer and the hardware were: Intel(R) Xeon(R) CPU I7 3770 3.40GHz, RAM 16GB. The operating system was Ubuntu 16.04 LTS (64 bit), and software were gcc 4.9.2 with compiler flag -Ofast, except for SUMO code, MATLAB 8.4.0.150421 (R2014b), 64 bit. The experiments were set to halt if the algorithms could not finish the EHVI computation within 30 minutes. The results are shown in Figure 3.7.

The experimental results in Figure 3.7 show that KMAC is much faster than CDD13, especially when  $|P|$  is increased. Sometimes, we need to calculate EHVI for multiple points under the same Pareto-front set and test whether the algorithms handle this problem efficiently. Since execution time would be increased dramatically when Batch Size is increased for high dimension ( $d \geq 4$ ), here we only consider the 3-D case. Table 3.1 shows the experimental results with different Batch Size.

The parameters:  $\sigma = (2.5, 2.5, 2.5)$ ,  $\mu = (10, 10, 10)$  were used in the experiments. Pareto front sizes  $|P| \in \{10, 100, 1000\}$  and the number of predictions (candidate points) or Batch Size  $\in \{1, 10, 100, 1000\}$  are used together with  $\sigma$  and  $\mu$ . Ten trials were randomly generated by the same parameters, and average runtimes (10 runs) for the whole 10 trails with the same parameters were computed. The data for 3-D case with  $|P| = 100$  are visualized in Figure 3.6, and these figures are originally from [20]. All the experiments were run on the same hardware: Intel(R) Xeon(R) CPU E5-2667 v2 3.30GHz, RAM 48GB. The operating system was Ubuntu 12.04 LTS (64 bit), and the compiler was gcc 4.9.2 with compiler flag -Ofast, except for SUMO code, MATLAB 8.4.0.150421 (R2014b), 64 bit. The experiments were set to halt if the algorithms could not finish the EHVI

### 3.6 Empirical Experiments

computation within 3 hours. The results are shown in Table 3.1.



**Figure 3.7:** Speed comparison of EHVI calculation. Above: concave random Pareto front set; Below: convex random Pareto front set.

### 3. EFFICIENT EHVI CALCULATION

**Table 3.1:** Empirical comparisons of strategies for 3-D EHVI calculation.

Type	$ P $	Batch Size	Time Average (s)		KMAC
			CDD13 [58]	IRS_fast [1]	
CONVEX	10	1	0.13785	0.00037	<b>0.00005</b>
CONVEX	10	10	0.14090	0.00056	<b>0.00021</b>
CONVEX	10	100	0.16500	0.00304	<b>0.00095</b>
CONVEX	10	1000	0.69104	0.02778	<b>0.00754</b>
CONVEX	100	1	13.97556	0.05337	<b>0.00038</b>
CONVEX	100	10	17.05551	0.13730	<b>0.00099</b>
CONVEX	100	100	45.90095	0.93196	<b>0.00831</b>
CONVEX	100	1000	422.31263	8.38585	<b>0.06462</b>
CONVEX	1000	1	>3 hours	94.72402	<b>0.00390</b>
CONVEX	1000	10	>3 hours	155.77306	<b>0.01067</b>
CONVEX	1000	100	>3 hours	795.11319	<b>0.06517</b>
CONVEX	1000	1000	>3 hours	2838.31854	<b>0.53801</b>
CONCAVE	10	1	0.11209	0.00026	<b>0.00007</b>
CONCAVE	10	10	0.12790	0.00054	<b>0.00014</b>
CONCAVE	10	100	0.14002	0.00294	<b>0.00077</b>
CONCAVE	10	1000	0.36697	0.02597	<b>0.00840</b>
CONCAVE	100	1	10.62329	0.04895	<b>0.00031</b>
CONCAVE	100	10	12.63582	0.12927	<b>0.00146</b>
CONCAVE	100	100	27.51827	0.85124	<b>0.00768</b>
CONCAVE	100	1000	314.32314	7.67280	<b>0.06285</b>
CONCAVE	1000	1	>3 hours	91.51055	<b>0.00332</b>
CONCAVE	1000	10	>3 hours	149.58491	<b>0.01079</b>
CONCAVE	1000	100	>3 hours	744.46691	<b>0.06696</b>
CONCAVE	1000	1000	>3 hours	2499.29737	<b>0.50981</b>
CLIFF3D	10	1	0.12514	0.00026	<b>0.00007</b>
CLIFF3D	10	10	0.13222	0.00055	<b>0.00013</b>
CLIFF3D	10	100	0.14432	0.00278	<b>0.00075</b>
CLIFF3D	10	1000	0.44964	0.02725	<b>0.00761</b>
CLIFF3D	100	1	10.90605	0.04730	<b>0.00029</b>
CLIFF3D	100	10	12.85031	0.12709	<b>0.00112</b>
CLIFF3D	100	100	44.79395	0.80735	<b>0.00689</b>
CLIFF3D	100	1000	679.51368	7.46205	<b>0.06099</b>
CLIFF3D	1000	1	>3 hours	136.37944	<b>0.00344</b>
CLIFF3D	1000	10	>3 hours	165.34537	<b>0.01007</b>
CLIFF3D	1000	100	>3 hours	731.03794	<b>0.06480</b>
CLIFF3D	1000	1000	>3 hours	2543.16864	<b>0.51032</b>

## 3.6 Empirical Experiments

The results show that the proposed algorithm, KMAC, is the fastest one for all the test problems. Empirical comparisons on randomly generated Pareto fronts of different shape show that the new algorithm is by a factor of 7 to  $3.9 \times 10^4$  faster than previously published implementations in the 3-D case.

### 3.6.2 Benchmark Performance

Five state-of-the-art algorithms are compared in this section, they are: EHVI-MOBGO, PoI-MOBGO, NSGA-II [25], NSGA-III [73][74] and SMS-EMOA [19]. The benchmarks are DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5 and DTLZ7. The parameter settings for all these test algorithms are shown in Table 3.2. Here, EHVI-EGO and PoI-EGO were only tested with evaluation budget as 300, because these two algorithms are time-consuming<sup>1</sup>. The reference points for each benchmark are shown in Table 3.3. Each setting was repeated ten times.

**Table 3.2:** Algorithm Parameter Settings.

	EHVI-MOBGO	PoI-MOBGO	NSGA-II	NSGA-III	SMS-EMOA
$\mu$	30	30	30	/	30
$\lambda$	1	1	30		/
Evaluation	300	300	300/2000	300/2000	300/2000
Divisions_outer	/	/	/	12	/
$p_c$	/	/	0.9		0.9
$p_m$	/	/	1/6		1/6
Platform	MATLAB	MATLAB	MATLAB	Python	MATLAB

**Table 3.3:** Reference Points.

	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ7
REF	(400,400,400)	(2.5,2.5,2.5)	(1500,1500,1500)	(2.5,2.5,2.5)	(11,11,11)	(1,1,10)

The final Pareto fronts were evaluated by *Hypervolume*. The empirical experimental results, with respect to statistical mean and standard deviation, are shown in Table 3.4 and 3.5. MOBGO based algorithms perform better than EAs (NSGA-II, NSGA-III and SMS-EMOA), despite the fact that the evaluation budget was increased to 2000. Among EHVI-MOBGO and PoI-MOBGO, EHVI-MOBGO

<sup>1</sup>Updating Kriging model and finding the optimal solution using CMA-ES are expensive.

### 3. EFFICIENT EHVI CALCULATION

**Table 3.4:** Empirical Comparisons.

Algorithm Eval.	MOBGO		EAs			EAs		
	EHVI 300	PoI 300	NSGA2 300	NSGA3 300	SMS-EMOA 300	NSGA2 2000	NSGA3 2000	SMS-EMOA 2000
DTLZ1	6.38174E+7	6.31810E+7	6.34392E+7	6.36393E+7	6.04776E+7	6.39818E+7	6.39986E+7	6.39985E+7
	6.39662E+7	6.33303E+7	6.32707E+7	6.38025E+7	6.18341E+7	6.39958E+7	6.39990E+7	6.39932E+7
	6.39672E+7	6.35670E+7	6.31858E+7	6.35901E+7	6.14465E+7	6.39958E+7	6.39991E+7	6.39995E+7
	6.39729E+7	6.35214E+7	6.37121E+7	6.36915E+7	6.29733E+7	6.39993E+7	6.39984E+7	6.39995E+7
	6.39722E+7	6.36307E+7	6.37121E+7	6.36159E+7	5.99827E+7	6.39840E+7	6.39992E+7	6.39801E+7
	6.39833E+7	6.33890E+7	6.30707E+7	6.36498E+7	6.23334E+7	6.39969E+7	6.39999E+7	6.39977E+7
	6.39776E+7	6.34682E+7	6.36282E+7	6.32780E+7	6.22081E+7	6.39969E+7	6.39990E+7	6.39951E+7
	6.39790E+7	6.28641E+7	6.34714E+7	6.37243E+7	6.18742E+7	6.39919E+7	6.39996E+7	6.39951E+7
	6.39723E+7	6.35972E+7	6.34714E+7	6.34629E+7	6.18742E+7	6.39764E+7	6.39928E+7	6.39926E+7
	6.39791E+7	6.34263E+7	6.36216E+7	6.32948E+7	6.03677E+7	6.39947E+7	6.39985E+7	6.39946E+7
mean	<b>6.39587E+7</b>	6.33975E+7	6.34583E+7	6.35749E+7	6.15372E+7	6.39914E+7	<b>6.39984E+7</b>	6.39946E+7
std.	<b>4.99505E+4</b>	2.30970E+5	2.22275E+5	1.75929E+5	9.64758E+5	7.78434E+3	<b>2.01767E+3</b>	5.68083E+3
DTLZ2	1.50123E+1	1.49961E+1	1.35131E+1	1.45495E+1	1.28434E+1	1.39116E+1	1.49691E+1	1.47435E+1
	1.50293E+1	1.49869E+1	1.35313E+1	1.43761E+1	1.36917E+1	1.35030E+1	1.49737E+1	1.46302E+1
	1.50133E+1	1.50001E+1	1.36346E+1	1.44741E+1	1.29611E+1	1.40951E+1	1.49713E+1	1.42623E+1
	1.50302E+1	1.50036E+1	1.33895E+1	1.41762E+1	1.31983E+1	1.40951E+1	1.49893E+1	1.47413E+1
	1.50178E+1	1.49990E+1	1.33895E+1	1.42859E+1	1.24597E+1	1.38371E+1	1.49763E+1	1.47010E+1
	1.50288E+1	1.49972E+1	1.34931E+1	1.43059E+1	1.28168E+1	1.44331E+1	1.49705E+1	1.48084E+1
	1.50325E+1	1.50030E+1	1.34931E+1	1.45558E+1	1.29619E+1	1.41515E+1	1.49737E+1	1.46117E+1
	1.50263E+1	1.50075E+1	1.28980E+1	1.44279E+1	1.30089E+1	1.34422E+1	1.49844E+1	1.46629E+1
	1.50263E+1	1.49913E+1	1.29148E+1	1.44522E+1	1.31828E+1	1.34422E+1	1.49747E+1	1.47318E+1
	1.49865E+1	1.49901E+1	1.39749E+1	1.46875E+1	1.34046E+1	1.43841E+1	1.49816E+1	1.47118E+1
mean	<b>1.50203E+1</b>	1.49975E+1	1.34232E+1	1.44291E+1	1.30529E+1	1.39295E+1	<b>1.49765E+1</b>	1.46605E+1
std.	1.02673E-2	<b>5.15545E-3</b>	2.20199E-1	1.14701E-1	2.53151E-1	3.02290E-1	<b>5.18375E-3</b>	9.54696E-2

outperforms PoI-MOBGO in most cases, except for DTLZ4 and DTLZ5. The reason is that PoI is a reference-free indicator, and it considers all the possibilities of an evaluated point in the subspace which dominates  $\mathcal{P}$ . Compared to PoI, however, EHVI only considers the subspace, which is dominated by  $\mathcal{P}$  and is cut by a reference point  $\mathbf{r}$ . In other words, EHVI cannot indicate any improvement of an evaluated point in the rest of non-dominated space, which is cut by a reference point.

## 3.7 Summary

This chapter described the *Expected Hypervolume Improvement* as the criterion used in MOBGO. The exact calculation of EHVI in the 2-D and the 3-D cases was introduced with the computational complexity of  $O(n \log n)$ . Compared to [1], the computational complexity is improved by the factor  $n^2 / \log n$  for 2-D and 3-D cases. This meets the lower bound for the time complexity of the EHVI com-

### 3.7 Summary

**Table 3.5:** Empirical Comparisons.

Algorithm Eval.	MOBGO		EAs			EAs		
	EHVI 300	PoI 300	NSGA2 300	NSGA3 300	SMS-EMOA 300	NSGA2 2000	NSGA3 2000	SMS-EMOA 2000
DTLZ3	3.37465E+9	3.37431E+9	3.35455E+9	3.36011E+9	3.27636E+9	3.37471E+9	3.37498E+9	3.37459E+9
	3.37462E+9	3.37188E+9	3.34970E+9	3.35426E+9	3.27336E+9	3.37477E+9	3.37498E+9	3.37494E+9
	3.37405E+9	3.36607E+9	3.33783E+9	3.36719E+9	3.26915E+9	3.37446E+9	3.37499E+9	3.37413E+9
	3.37433E+9	3.34920E+9	3.35935E+9	3.36343E+9	3.28850E+9	3.37443E+9	3.37498E+9	3.37440E+9
	3.37460E+9	3.36956E+9	3.33889E+9	3.36678E+9	3.26551E+9	3.37478E+9	3.37500E+9	3.37469E+9
	3.37455E+9	3.37459E+9	3.36864E+9	3.35810E+9	3.34960E+9	3.37471E+9	3.37498E+9	3.37467E+9
	3.37466E+9	3.37371E+9	3.35894E+9	3.36364E+9	3.20942E+9	3.37498E+9	3.37499E+9	3.37469E+9
	3.37452E+9	3.37278E+9	3.36435E+9	3.36349E+9	3.30056E+9	3.37499E+9	3.37500E+9	3.37496E+9
	3.37465E+9	3.36970E+9	3.36268E+9	3.36183E+9	3.30523E+9	3.37361E+9	3.37499E+9	3.37413E+9
	3.37443E+9	3.37337E+9	3.36111E+9	3.35554E+9	3.30145E+9	3.37484E+9	3.37498E+9	3.37467E+9
mean	<b>3.37451E+9</b>	3.36952E+9	3.35561E+9	3.36144E+9	3.28391E+9	3.37463E+9	<b>3.37499E+9</b>	3.37459E+9
std.	<b>1.41276E+5</b>	4.75297E+6	8.29054E+6	3.54949E+6	2.51542E+7	2.75889E+5	<b>5.94791E+3</b>	2.20550E+5
DTLZ4	1.38157E+1	1.45550E+1	1.17541E+1	1.08884E+1	9.55215E+0	1.47844E+1	1.49959E+1	1.48530E+1
	1.36793E+1	1.44860E+1	9.30831E+0	1.16457E+1	1.08876E+1	1.32053E+1	1.36099E+1	9.33000E+0
	1.30832E+1	1.45194E+1	9.33753E+0	1.18060E+1	8.85919E+0	1.47539E+1	1.35738E+1	1.31180E+1
	1.38658E+1	1.48104E+1	1.29697E+1	1.28858E+1	9.20911E+0	9.32150E+0	1.49421E+1	1.32434E+1
	1.36991E+1	1.45245E+1	1.31589E+1	1.21828E+1	1.06302E+1	1.35101E+1	1.49398E+1	1.48101E+1
	1.44512E+1	1.42640E+1	1.42940E+1	1.32904E+1	8.86695E+0	1.45135E+1	1.49798E+1	1.32412E+1
	1.42156E+1	1.45332E+1	1.44560E+1	1.19631E+1	8.91678E+0	1.47941E+1	1.35860E+1	1.34922E+1
	1.35453E+1	1.38725E+1	1.30766E+1	1.24391E+1	9.20227E+0	9.30187E+0	1.30364E+1	1.33983E+1
	1.38884E+1	1.44282E+1	1.30079E+1	1.26858E+1	9.16894E+0	1.48205E+1	1.36039E+1	1.33954E+1
	1.37201E+1	1.45676E+1	1.27012E+1	1.29606E+1	1.11599E+1	1.29413E+1	1.49813E+1	1.36090E+1
mean	1.37964E+1	<b>1.44561E+1</b>	1.24064E+1	1.22748E+1	9.64531E+0	1.31946E+1	<b>1.42249E+1</b>	1.32491E+1
std.	2.50980E-1	<b>1.60714E-1</b>	1.36386E+0	5.77561E-1	7.48353E-1	1.60385E+0	<b>7.42880E-1</b>	8.12735E-1
DTLZ5	1.31781E+3	1.31885E+3	1.29108E+3	1.31652E+3	1.28384E+3	1.31670E+3	1.31872E+3	1.31681E+3
	1.31609E+3	1.31883E+3	1.31533E+3	1.31687E+3	1.31455E+3	1.31755E+3	1.31888E+3	1.31803E+3
	1.31703E+3	1.31885E+3	1.31556E+3	1.31650E+3	1.30785E+3	1.31759E+3	1.31878E+3	1.31815E+3
	1.31754E+3	1.31885E+3	1.30788E+3	1.31583E+3	1.28715E+3	1.31804E+3	1.31889E+3	1.31744E+3
	1.31741E+3	1.31882E+3	1.31565E+3	1.31663E+3	1.29529E+3	1.31714E+3	1.31887E+3	1.31827E+3
	1.31706E+3	1.31883E+3	1.30716E+3	1.31333E+3	1.30770E+3	1.31731E+3	1.31881E+3	1.31844E+3
	1.31711E+3	1.31878E+3	1.31571E+3	1.31034E+3	1.30873E+3	1.31675E+3	1.31875E+3	1.31785E+3
	1.31802E+3	1.31884E+3	1.31501E+3	1.31654E+3	1.31039E+3	1.31769E+3	1.31876E+3	1.31813E+3
	1.31735E+3	1.31882E+3	1.31564E+3	1.31595E+3	1.30349E+3	1.31785E+3	1.31881E+3	1.31832E+3
	1.31742E+3	1.31883E+3	1.30992E+3	1.31640E+3	1.29023E+3	1.31760E+3	1.31883E+3	1.31795E+3
mean	1.31728E+3	<b>1.31883E+3</b>	1.31089E+3	1.31549E+3	1.30092E+3	1.31742E+3	<b>1.31881E+3</b>	1.31794E+3
std.	3.69904E-1	<b>1.43867E-2</b>	5.50786E+0	1.46033E+0	9.43520E+0	3.57031E-1	<b>4.52836E-2</b>	3.43975E-1
DTLZ7	5.09516E+0	4.40942E+0	-1.64477E+0	1.94194E+0	-3.70070E+0	-1.36049E+0	4.90652E+0	1.97405E+0
	5.17729E+0	4.03866E+0	-2.52335E+0	2.08332E+0	-5.84918E+0	-4.14008E+0	5.08696E+0	4.34746E-2
	5.15481E+0	4.28059E+0	-4.49931E+0	2.70067E+0	-4.80969E+0	1.48952E+0	4.66387E+0	2.56120E-1
	4.89191E+0	4.02969E+0	-1.46708E+0	1.18938E+0	-4.38641E+0	2.25216E+0	4.71677E+0	1.98754E-1
	5.06238E+0	4.42348E+0	-1.54354E+0	2.12846E+0	-4.43218E+0	-4.30157E+0	4.99910E+0	1.60383E+0
	4.93139E+0	4.22680E+0	-2.94045E+0	1.27511E+0	-6.29530E-1	-1.04859E+0	5.08663E+0	2.61729E+0
	5.16743E+0	4.52667E+0	-3.76357E+0	2.76559E+0	-3.21233E+0	-2.91679E+0	4.95629E+0	1.78240E+0
	5.02720E+0	4.16430E+0	-2.93363E+0	2.23855E+0	-4.61563E+0	-9.60784E-1	4.66007E+0	1.01542E+0
	5.06158E+0	4.31499E+0	-4.74578E+0	1.94897E+0	-4.83294E+0	-3.53033E+0	4.67642E+0	5.15294E-1
	5.08646E+0	4.06894E+0	-5.35736E+0	1.59954E-1	-4.29017E+0	-6.53071E-1	4.91579E+0	1.06682E-1
mean	<b>5.08646E+0</b>	4.06894E+0	-5.35736E+0	1.59954E-1	-4.29017E+0	-6.53071E-1	<b>4.91579E+0</b>	1.06682E-1
std.	<b>7.59397E-2</b>	1.42676E-1	9.78696E-1	5.81027E-1	1.02525E+0	1.76415E+0	<b>1.61886E-1</b>	7.87267E-1

### 3. EFFICIENT EHVI CALCULATION

---

putation for  $d = 2, 3$ , shown by reduction of the *Hypervolume Indicator* problem, see [1]. Thus, the algorithm is asymptotically optimal and the time complexity of 2-D and 3-D EHVI computation is in  $\Theta(n \log n)$ . For the arbitrary dimensional case when  $d \geq 2$ , the formula for exact EHVI calculation is generalized in this chapter. In the speed-comparison experiments, the average execution time of KMAC is compared with that of CDD13. The experimental results show that KMAC is much faster than CDD13, especially for high dimensional cases.

This chapter also compared EHVI-MOBGO with other state-of-the-art multi-objective optimization algorithms. Multi-objective Bayesian global optimization algorithms yield better results, compared to evolutionary multi-objective optimization algorithms. Among multi-objective Bayesian global optimization algorithms, the Pareto-front approximation sets generated by EHVI-MOBGO are usually closer to the true Pareto front. However, PoI-MOBGO is better than EHVI-MOBGO when dealing with DTLZ4 and DTLZ5 problems. The reason is that PoI is a reference-free criterion and EHVI is a reference-based criterion, and EHVI only implies the improvement in the non-dominated space which is cut above by the reference point. A remedy to this problem can be achieved by setting a large reference point or using dynamic reference point. The reference point cannot be too large, otherwise, EHVI at any evaluated points would be similar, even the same, which is due to the numerical stability.

# Chapter 4

## TEHVI Calculation<sup>1</sup>

In optimization with expensive black box evaluations, the expected improvement algorithm (also called *efficient global optimization*) is a commonly applied method. It uses Gaussian Processes (or Kriging) to build a model of the objective function and uses the expected improvement as an infill criterion, taking into account both – predictive mean and variance. EI has been generalized to multi-objective optimization using the expected hypervolume improvement, which measures the expected gain in the hypervolume indicator of a Pareto front approximation, as shown in Chapter 3. However, this criterion assumes an unbounded objective space even if it is often known a-priori that the objective function values are within a prescribed range, e.g., lower bounded by zero. Taking advantage of such a-priori knowledge, this chapter introduces the truncated expected hypervolume improvement and a multi-objective efficient global optimization method that is based on TEHVI. This chapter shows how to compute the truncated expected hypervolume improvement exactly and efficiently. Then it is tested as an infill criterion in efficient global optimization. It is shown that it can effectively make use of a-priori knowledge and achieve better results in cases where such knowledge is given. The usefulness of the new approach is demonstrated on benchmark examples. The empirical studies in this chapter are confined to the bi-objective case.

This chapter is structured as follows: Section 4.1 introduces the motivations of the TEHVI research; Section 4.2 provides the definition of TEHVI and the formula to calculate exact TEHVI, including asymptotic complexity analysis and CPU time assessment; Sections 4.3 and 4.4 show the experimental settings and empirical

---

<sup>1</sup>This chapter only considers minimization problems.

## 4. TEHVI CALCULATION

---

experimental results, respectively.

### 4.1 Motivations

Many algorithms exist in the multi-objective optimization field. For the evolutionary algorithm class, NSGA-II [75] and SMS-EMOA [19] are two well-known algorithms. Surrogate-model based optimization strategies, which replace exact evaluations by approximations learned from past evaluations, are another important branch. Compared to evolutionary algorithms, surrogate-model based algorithms need a small budget of function evaluations. Because of this, they are often used in the real world design optimization with expensive evaluations.

A simple and common sequential optimization scheme is to sequentially update the surrogate model by evaluations the points that are promising due to the prediction. The prospects of a new point are assessed by the so-called infill criterion. In the context of Gaussian process models, the expected improvement criterion is a commonly applied infill criterion. It takes into account both the predictive mean and the predictive variance of the surrogate model. Therefore it promotes evaluations in less explored regions that have a higher predictive variance. In single-objective optimization the expected improvement was introduced in 1978 by Mockus et al. [37], and became more popular due to the work of Jones et al. [41]. It was generalized to the expected hypervolume improvement (EHVI) for multi-objective optimization by Emmerich [56].

There are alternative generalizations of the expected improvement in the field of multi-objective optimization. Among them, EHVI has a good convergence to a diverse approximation of the Pareto front, however, exact calculation of EHVI used to be time-consuming [69]. Recently, new algorithms for computing EHVI in the bi-objective EHVI-EGO have been found. Hupkens et al. improved the time complexity to  $O(n^2)$  [1]. More recently, Emmerich et al. devised an asymptotically optimal algorithm with time complexity  $O(n \log n)$  [2] in the bi-objective case, where  $n$  is the number of non-dominated points in the archive. This makes EHVI-EGO competitive with other techniques that use fast computable infill criteria, in particular SMS-EGO.

Besides its time consumption, EHVI does not take into account some known domain information of the objective function. Expected hypervolume improvement (EHVI) is the expected increment of the hypervolume indicator, which is related to the current approximation of the Pareto front and a predictive multivariate

Gaussian distribution of a new point [10]. That is to say, EHVI is based on the assumption that the objective values follow a normal distribution, and prediction of the objective values are from minus infinity to plus infinity. However, in some cases, we already know an approximation range of the objective value. For instance, in a PID parameter tuning problem, the rising time is always a positive value. In this case, we assumed surrogate-model based algorithms could converge to true Pareto front faster, if the multivariate Gaussian distribution in EHVI is truncated to the objective function's co-domain.

For using the co-domain information, this chapter introduces the truncated expected hypervolume improvement (TEHVI) based on EHVI. TEHVI is based on the normal distribution, which is truncated by the objective value domain. In terms of Bayesian reasoning, it uses the conditional distribution given the a-priori knowledge. This knowledge is about the true output of the objective function within a prescribed range. Practically speaking, the idea behind TEHVI is to focus sampling on more relevant parts of the search space by taking into account a-priori knowledge on objective function value ranges. It is hypothesized that this will speed up the convergence of the Pareto front.

This section mainly discusses the formula to calculate truncated expected hypervolume improvement. It will also have asymptotically optimal time  $\Theta(n \log n)$ , and empirical validation and speed comparison between Monte Carlo method and the TEHVI exact method.

## 4.2 TEHVI Definition

**Definition 4.1 (Truncated Expected Hypervolume Improvement)** <sup>2</sup> *Given parameters of the multivariate predictive distribution  $\boldsymbol{\mu}$ ,  $\boldsymbol{\sigma}$  and the Pareto-front approximation  $\mathbf{P}$ , a preferred multidimensional range  $[\mathbf{A}, \mathbf{B}] = [A_1, B_1] \times \cdots \times [A_d, B_d] \subset \mathbb{R}^d$  in the objective space. Suppose an objective value vector  $\mathbf{y}$  follows the truncated normal distribution and lies within an interval  $\mathbf{y} \in (\mathbf{A}, \mathbf{B})$ , where  $-\infty \leq \mathbf{A} < \mathbf{B} \leq \infty$ , then the truncated expected hypervolume improvement (TEHVI) is defined as:*

$$TEHVI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{P}, \mathbf{r}, \mathbf{A}, \mathbf{B}) := \int_{\mathbf{y} \in [\mathbf{A}, \mathbf{B}]} HVI(\mathbf{P}, \mathbf{y}) \cdot TPDF_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \quad (2-1)$$

---

<sup>2</sup>The prediction of  $\boldsymbol{\mu}$  and  $\boldsymbol{\sigma}$  depends on a Kriging model and a target point  $\mathbf{x}$  in the search space. Explicitly, TEHVI is dependent on the target point  $\mathbf{x}$ .

## 4. TEHVI CALCULATION

---

where  $TPDF_{\mu, \sigma}$  is the truncated multivariate independent normal distribution with mean values  $\mu \in \mathbb{R}^d$  and standard deviations  $\sigma \in \mathbb{R}_+^d$ .

Here,  $TPDF_{\mu, \sigma}$  is the probability density function for the event that  $\mathbf{y}$  is the result of the function evaluation, given function evaluations that are not within the range  $[\mathbf{A}, \mathbf{B}]$  would be rejected. Due to rejection of values outside the range, in Bayesian reasoning we could use the information  $\mathbf{y} \in [\mathbf{A}, \mathbf{B}]$  (after acceptance) as a Bayesian prior.

**Example 4.1** An illustration of the EHVI is shown in Figure 4.1. The light gray area is the dominated subspace of  $\mathbf{P} = \{\mathbf{y}^{(1)} = (3, 1)^\top, \mathbf{y}^{(2)} = (2, 1.5)^\top, \mathbf{y}^{(3)} = (1, 2.5)^\top\}$  cut by the reference point  $\mathbf{r} = (0, 0)^\top$ . The bivariate Gaussian distribution has the parameters  $\mu_1 = 2, \mu_2 = 1.5, \sigma_1 = 0.7, \sigma_2 = 0.6$ . The truncated probability density function (TPDF) of the bivariate Gaussian distribution is indicated as a 3-D plot, with truncated domains  $[\mathbf{A}, \mathbf{B}] = [\mathbf{1}, \infty]$ . Here  $\mathbf{y}$  is a sample from this distribution and the area of improvement relative to  $\mathbf{P}$  is indicated by the dark shaded area. The variable  $y_1$  stands for the  $f_1$  value and  $y_2$  for the  $f_2$  value.

### 4.2.1 Formula Derivation

For the aim of calculating the truncated expected hypervolume improvement, we need to define the truncated PDF ( $\phi_T$ ) and truncated CDF ( $\Psi_T$ ) functions first. Suppose the co-domain of a truncated normal distribution is  $[A, B]$ , where  $-\infty \leq A < B \leq \infty$ . From the definition of truncated normal distribution,  $\phi_T$  and  $\Phi_T$  functions are defined as follows[76]:

$$\phi_T(x) = \begin{cases} 0 & \text{if } x \leq A \text{ or } x \geq B \\ \mathbf{Z}(A, B) \cdot \frac{\phi(\frac{x-\mu}{\sigma})}{\sigma} & \text{if } A < x < B \end{cases} \quad (2-2)$$

$$\Phi_T(x) = \begin{cases} 0 & \text{if } x \leq A \\ \mathbf{Z}(A, B) \cdot [\Phi(\frac{x-\mu}{\sigma}) - \Phi(\frac{A-\mu}{\sigma})] & \text{if } A < x < B \\ 1 & \text{if } x \geq B \end{cases} \quad (2-3)$$

$$\text{where: } \mathbf{Z}(A, B) = 1/[\Phi(\frac{B-\mu}{\sigma}) - \Phi(\frac{A-\mu}{\sigma})] \quad (2-4)$$

Due to the independence of the multivariate Gaussian distributions for each objective, the product of the truncated distributions can be computed using Fubini's

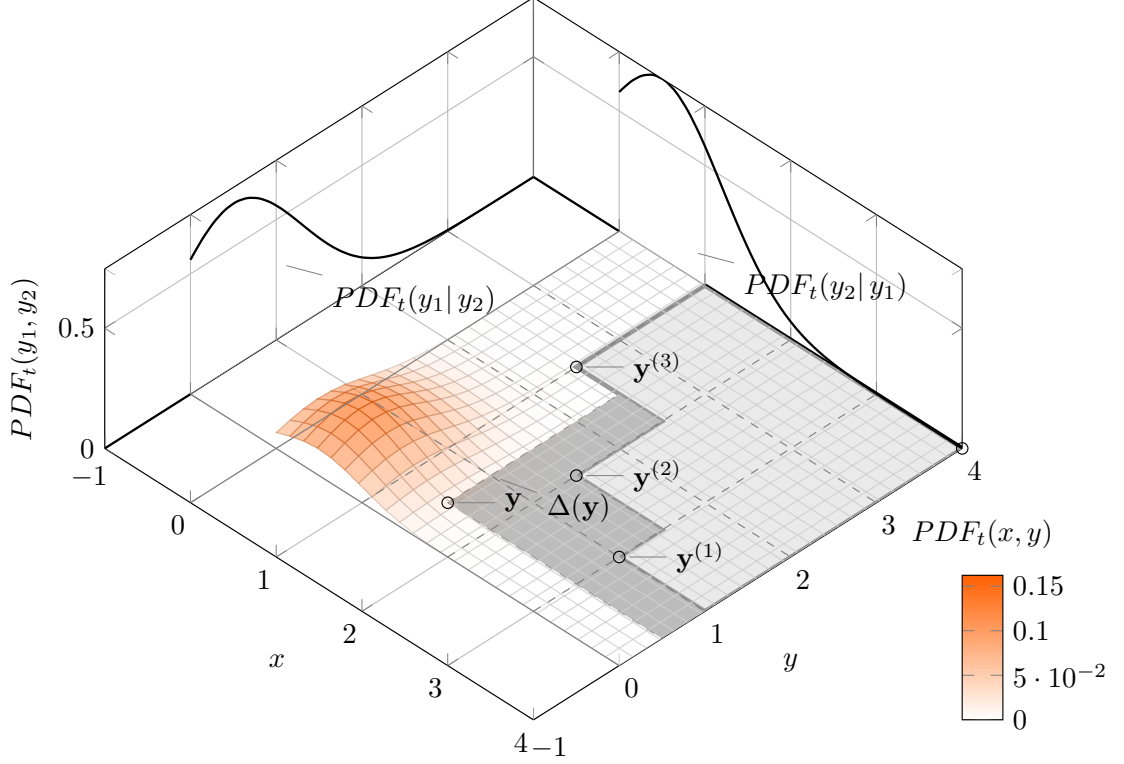


Figure 4.1: TEHVI in 2-D (cf. Example 4.1).

law. What needs to be done is to replace  $\Phi$  by  $\Phi_T$  and  $\Psi$  by  $\Psi_T$  in the original computation, with interval boundaries chosen according to the interval boundaries. Based on  $\Theta(n \log n)$  2-D EHVI formula (minimization case) in A.4, the TEHVI formula can be written as:

$$\begin{aligned}
 & \text{TEHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, Y, \mathbf{r}, A, B) \\
 &= \int_{y_1=-\infty}^{\infty} \int_{y_2=-\infty}^{\infty} \sum_{i=1}^{n+1} \lambda_2[S_i \cap \Delta(y_1, y_2)] \cdot PDF_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \\
 &= \sum_{i=1}^{n+1} (y_1^{(i-1)} - y_1^{(i)}) \cdot \Phi_T \left( \frac{y_1^{(i)} - \mu_1}{\sigma_1} \right) \cdot \Psi_T(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) + \\
 & \quad \sum_{i=1}^{n+1} \left( \Psi_T(y_1^{(i-1)}, y_1^{(i-1)}, \mu_1, \sigma_1) - \Psi_T(y_1^{(i-1)}, y_1^{(i)}, \mu_1, \sigma_1) \right) \times \\
 & \quad \Psi_T(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2)
 \end{aligned} \tag{2-5}$$

According to the definition of truncated normal distribution and the formula of

#### 4. TEHVI CALCULATION

---

normal *exipsi* function ( $\Psi$ ), the truncated *exipsi* function ( $\Psi_T$ ) for minimization problems can be derived by the following procedure:

$$\begin{aligned}
\Psi_T(a, b, \mu, \sigma, A, B)_{min} &= \int_A^b (a - z) \frac{1}{\sigma} \phi_T\left(\frac{z - \mu}{\sigma}\right) dz \\
&= \int_A^b (a - z) \cdot \mathbf{Z}(A, B) \cdot \frac{1}{\sigma} \phi\left(\frac{z - \mu}{\sigma}\right) dz \\
&= \mathbf{Z}(A, B) \cdot \int_A^b (a - z) \frac{1}{\sigma} \phi\left(\frac{z - \mu}{\sigma}\right) dz \\
&= \mathbf{Z}(A, B) \cdot \left[ \left( \sigma \phi\left(\frac{b - \mu}{\sigma}\right) + (a - \mu) \Phi\left(\frac{b - \mu}{\sigma}\right) - \right. \right. \\
&\quad \left. \left( \sigma \phi\left(\frac{A - \mu}{\sigma}\right) + (a - \mu) \Phi\left(\frac{A - \mu}{\sigma}\right) \right) \right] \tag{2-6}
\end{aligned}$$

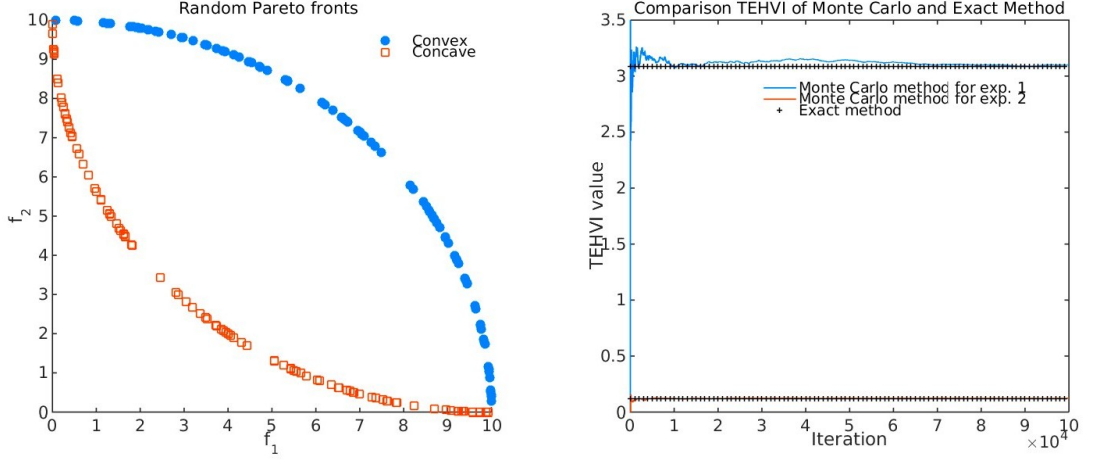
In some cases, it is difficult to transform maximization problems to minimization problems. For solving it, the truncated *exipsi* function for maximization problems is necessary and it is:

$$\begin{aligned}
\Psi_T(a, b, \mu, \sigma, A, B)_{max} &= \int_b^B (a - z) \frac{1}{\sigma} \phi_T\left(\frac{z - \mu}{\sigma}\right) dz \\
&= \mathbf{Z}(A, B) \cdot \left[ \left( \sigma \phi\left(\frac{b - \mu}{\sigma}\right) + (\mu - a) (1 - \Phi\left(\frac{b - \mu}{\sigma}\right)) - \right. \right. \\
&\quad \left. \left( \sigma \phi\left(\frac{B - \mu}{\sigma}\right) + (\mu - a) (1 - \Phi\left(\frac{B - \mu}{\sigma}\right)) \right) \right] \tag{2-7}
\end{aligned}$$

##### 4.2.2 Computational Speed Test

In this subsection the computational speed of TEHVI computation is assessed when there are different population sizes. For validation purposes, the results are compared with results from Monte Carlo integration. The acceptance-rejection method [77] was used as the sampling strategy in Monte Carlo method. Samples out of the feasible interval range were rejected. The Monte Carlo method was allowed to run for 100,000 iterations. All the experiments were performed on the same computer: Intel(R) i7-3770 CPU @ 3.40GHz, RAM 16GB. The operating system was Ubuntu 14.04 LTS (64 bit), compiler was gcc 4.9.2 with flag '-Ofast' for exact method, and platform was MATLAB 8.4.0.150421 (R2014b), 64 bit for Monte Carlo method.

Fig 4.2, the left subfigure is the randomly generated Pareto fronts with the type of CONVEXSPHERICAL and CONCAVESPHERICAL in the 2-D case from [20]. The



**Figure 4.2:** Left: Randomly generated fronts with  $|P| = 100$ . Right: TEHVI comparison of Monte Carlo method and exact method.

right subfigure shows the convergence of Monte Carlo integration to the Pareto fronts. The evaluated points are (9.9955, 0.3001) and (8.4149, 0.1264) for experiment 1 (convex Pareto front) and experiment 2 (concave Pareto front), respectively. The reference point for both experiments was (15,15). It shows that the TEHVI value based on Monte Carlo method is similar to exact method after 50,000 iterations. However, the Monte Carlo method needs more iterations to generate a sufficiently accurate TEHVI value.

Table 4.1 shows the empirical speed experiments between exact TEHVI calculation method and Monte Carlo method. The parameters:  $\sigma_m = 2.5$ ,  $\mu_m = 10$  and  $m = 2$  were used to randomly generate Pareto fronts in the experiments. Pareto front sizes  $|P| \in \{10, 100, 1000\}$  and different number of predictions (candidate points) of Batch Size are used together with  $\sigma_m$  and  $\mu_m$ . In the experiments, *batch mode* means that the Pareto front population does not change and computations of transcendental functions (erf, exp) on grid coordinates can be re-used. This results in a significant speed-up in the empirical performance, although the time complexity is not affected. 10 trials were randomly generated by the same parameters, and average runtimes (10 runs) for the whole trails with the same parameters were computed. It shows that the exact and efficient TEHVI calculation method is fast, even for large population sizes of 1000 points it can be computed in ca. 1 second. It is also fast when compared to an imprecise Monte Carlo method. Note that this study should not be used as a speed comparison, as the Monte Carlo method is not precise. However, it indeed shows that the time consumption of the Monte Carlo method increases quickly.

## 4. TEHVI CALCULATION

---

**Table 4.1:** TEHVI computation speed experiments results.

Type	$ P $	Batch Size	Time_average (s)	
			Exact	Monte Carlo
CONVEX	10	1	0.00005	3.82615
CONVEX	10	10	0.00019	37.56125
CONVEX	10	100	0.00166	379.19625
CONVEX	10	1000	0.02030	> 10 min
CONVEX	100	1	0.00022	13.34105
CONVEX	100	10	0.00165	140.10564
CONVEX	100	100	0.01654	> 10 min
CONVEX	100	1000	0.16084	> 10 min
CONVEX	1000	1	0.00208	203.11692
CONVEX	1000	10	0.01644	> 10 min
CONVEX	1000	100	0.16746	> 10 min
CONVEX	1000	1000	1.69174	> 10 min
CONCAVE	10	1	0.00006	3.89077
CONCAVE	10	10	0.00016	39.15924
CONCAVE	10	100	0.00142	389.16426
CONCAVE	10	1000	0.01585	> 10 min
CONCAVE	100	1	0.00023	13.95431
CONCAVE	100	10	0.00150	138.14462
CONCAVE	100	100	0.01457	> 10 min
CONCAVE	100	1000	0.14779	> 10 min
CONCAVE	1000	1	0.00203	204.58791
CONCAVE	1000	10	0.01494	> 10 min
CONCAVE	1000	100	0.14657	> 10 min
CONCAVE	1000	1000	1.53391	> 10 min

### 4.3 Experimental Setup

In the comparison, nine test problems are used. They are: BK1 [78], SSFYY1 [79], ZDT1, ZDT2, ZDT3, ZDT4, ZDT6[80], generalized Schaffer problem (GSP) [81] and PID parameter tuning problem.

The parameters for the algorithms are shown in Table 4.2.

**Table 4.2:** Parameter settings.

Algorithm	$\mu$ /Initial Population	$\lambda$	iteration	$p_c$	$p_m$
EHVI-EGO	30	/	200	/	/
TEHVI-EGO	30	/	200	/	/
NSGA-II	30	30	200	0.9	1/N
SMS-EMOA	30	/	200	0.9	1/N

The TEHVI-EGO boundary  $(A, B)$  for all the experiments are  $(0, \infty)$ , except for the ZDT3 problem. Since the lower bound of ZDT3 is close to  $-1$ , the TEHVI-EGO boundary for ZDT3 was set to  $(-1, \infty)$ . For the generalized Schaffer problem, the parameter  $\gamma$  was set as  $\gamma = 0.4$ . All the experiments were repeated 5 times.

## 4.4 Empirical Results

Table 4.3 shows the results for all the test problems. TEHVI-EGO and EHVI-EGO are better than the other two algorithms. Among the EGO based algorithms, TEHVI-EGO performs slightly better than EHVI-EGO. The reason for these results is that all the fitness values for all the problems are positive, except for the ZDT3, and the truncation forces the optimization algorithm to focus on the fitness spaces in the positive domain. Figure 4.3 shows the Pareto fronts generated by 4 different algorithms for generalized Schaffer problem (GSP) in the left. The Pareto fronts generated by TEHVI-EGO and EHVI-EGO are much closer to true Pareto front than the other two algorithms. Compared to the performance of EHVI-EGO with respect to HV, HV of TEHVI-EGO Pareto front is slightly bigger than that of EHVI-EGO. The interval boundaries for TEHVI-EGO are set to  $A = 0, B = \infty$  in Figure 4.3 (right). This is based on the assumption that only the lower bound of the fitness value is known.

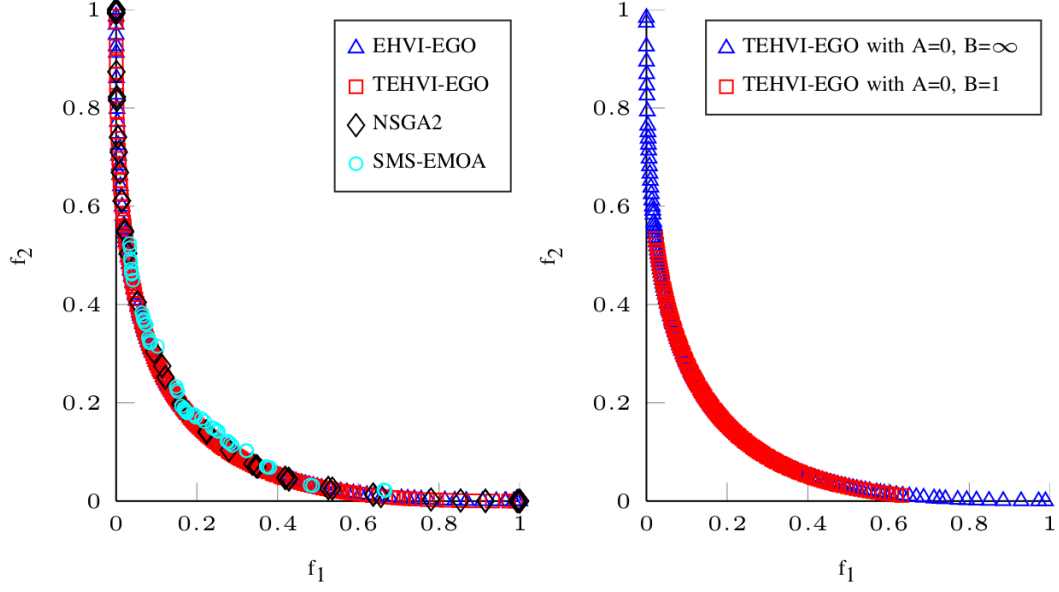
However, the strategy for setting the interval boundary is tricky. In Figure 4.3, the right plot shows the Pareto fronts generated by TEHVI-EGO with different interval boundary. In this case, the (red) squared Pareto front focuses on the knee points with more points but can not explore the extreme boundary of the true Pareto front well, when compared to the (blue) triangle one. Meanwhile,

#### 4. TEHVI CALCULATION

**Table 4.3:** Empirical comparison of different algorithms for test problems.

Test Function	Methods	reference point	Pareto front size				HV			
			max	min	mean	std	max	min	mean	std
BK1	EHVI-EGO	[60 60]	126	118	121.6	3.5071	3176.3411	3175.8606	3176.0758	0.2057
BK1	TEHVI-EGO	[60 60]	131	120	<b>124.6</b>	4.6690	<b>3176.4359</b>	3175.8961	<b>3176.1843</b>	0.2451
BK1	NSGA-II	[60 60]	58	53	55.6	2.0736	3132.1585	3126.3921	3129.2289	2.3525
BK1	SMS-EMOA	[60 60]	51	50	50.8	0.4472	3035.8162	2889.2727	2964.0079	65.9014
SSFYY1	EHVI-EGO	[5 5]	99	91	93.8	3.1145	20.7376	20.7308	20.7345	0.0027
SSFYY1	TEHVI-EGO	[5 5]	100	88	<b>95</b>	5.0990	<b>20.7431</b>	20.7269	<b>20.7355</b>	0.0062
SSFYY1	NSGA-II	[5 5]	57	53	55.6	1.6733	20.3667	20.1943	20.2895	0.0662
SSFYY1	SMS-EMOA	[5 5]	51	43	47.4	4.0373	18.5052	16.2884	17.6106	0.8399
ZDT1	EHVI-EGO	[15 15]	64	43	<b>56.2</b>	8.0436	224.6547	224.6409	224.6468	0.0064
ZDT1	TEHVI-EGO	[15 15]	71	40	52.6	11.4149	<b>224.6569</b>	224.6461	<b>224.6502</b>	0.0044
ZDT1	NSGA-II	[15 15]	45	39	43.2	2.4900	224.6172	220.7671	222.7442	1.7513
ZDT1	SMS-EMOA	[15 15]	51	48	49.4	1.1402	218.6193	217.7475	218.2609	0.3504
ZDT2	EHVI-EGO	[15 15]	27	24	26	1.2247	224.3152	224.2933	224.3099	0.0093
ZDT2	TEHVI-EGO	[15 15]	29	27	28.2	1.0954	<b>224.3168</b>	224.3151	<b>224.3161</b>	0.0008
ZDT2	NSGA-II	[15 15]	57	44	<b>47.8</b>	5.4037	224.0502	210.1391	213.4889	5.9302
ZDT2	SMS-EMOA	[15 15]	22	10	16	5.6569	189.7940	176.3485	184.7327	6.0880
ZDT3	EHVI-EGO	[15 15]	20	12	17.8	3.4928	235.8485	235.8015	235.8328	0.0196
ZDT3	TEHVI-EGO	[15 15]	32	15	21	6.8191	<b>235.8495</b>	235.8039	<b>235.8359</b>	0.0183
ZDT3	NSGA-II	[15 15]	47	42	<b>44.2</b>	1.9235	235.7032	221.2648	228.3704	5.3756
ZDT3	SMS-EMOA	[15 15]	31	16	24	5.7879	207.4997	202.1116	204.0837	2.1858
ZDT4	EHVI-EGO	[15 15]	7	3	4.4	1.6733	224.3295	204.4661	217.2593	8.0295
ZDT4	TEHVI-EGO	[15 15]	11	6	7.4	2.0736	<b>224.5959</b>	217.9858	<b>221.3740</b>	2.9304
ZDT4	NSGA-II	[15 15]	47	37	<b>43.2</b>	4.1473	215.6850	184.2184	203.9832	12.6032
ZDT4	SMS-EMOA	[15 15]	20	5	13	5.3385	145.3309	116.0149	127.4335	36.2023
ZDT6	EHVI-EGO	[15 15]	25	16	19.6	3.7815	218.7974	218.7442	218.7843	0.0228
ZDT6	TEHVI-EGO	[15 15]	32	19	23.3	6.1305	<b>218.8301</b>	218.7871	<b>218.8095</b>	0.0178
ZDT6	NSGA-II	[15 15]	41	25	<b>35.2</b>	7.2938	218.6877	155.9356	198.5166	26.2609
ZDT6	SMS-EMOA	[15 15]	22	7	11.6	6.0249	145.3309	116.0149	127.4335	11.8642
GSP	EHVI-EGO	[5 5]	167	140	161.4	11.9708	24.9066	24.9063	24.9065	0.0001
GSP	TEHVI-EGO	[5 5]	169	154	<b>166</b>	6.7082	<b>24.9066</b>	24.9066	<b>24.9066</b>	<0.0001
GSP	NSGA-II	[5 5]	60	56	58	2.0000	24.8933	24.8838	24.8903	0.0040
GSP	SMS-EMOA	[5 5]	51	50	50.8	0.4472	24.8605	24.6154	24.7519	0.1140

the HV value of the red squared Pareto front is smaller than blue triangle one. The reason for the difference is that the precise integration domain  $(0, 1)$  is much smaller than  $(0, \infty)$ . This could lead to the low probability of exploration at the extreme boundary, which is close to infinity, and high probability of sampling the area on the Pareto front, which is closer to the minimization point (in this case, this point is  $(0, 0)$ ).



**Figure 4.3:** Left: The results of one trial experiment for GSP in Table 4.3. Right: TEHVI-EGO with different interval boundary.

## 4.5 Summary

In this chapter, we introduced an exact method for the calculation of the truncated expected hypervolume improvement and investigated different multi-objective optimization algorithms for the benchmarks and the optimization of a controller. In particular, the two state-of-the-art evolutionary algorithms (NSGA-II and SMS-EMOA) were compared with multi-objective efficient global optimization algorithms (EHVI-EGO and TEHVI-EGO), which utilize a surrogate model of the objective function. Among the 9 test problems, TEHVI-EGO yielded better results than the other three algorithms, with respect to HV.

As the TEHVI only calculates the EHVI in a particular domain, and can force the algorithm on exploring in this domain, TEHVI-EGO exhibits poor performance of exploring the extreme boundaries for the Pareto fronts, when the interval co-domain is set as the boundary of the fitness. However, in this case, TEHVI-EGO, compared to EHVI-EGO, can focus on the knee point of the Pareto fronts, which could be used when a particular domain of a Pareto front is attractive.

#### 4. TEHVI CALCULATION

---

To summarize, based on the result of this study, we recommend using TEHVI-EGO when boundaries of domain regions are known, because it shows a good performance more consistently when compared to the other algorithms. In addition, it can focus on the Pareto front in a particular domain. For the further work, it is recommended to research the gradient of TEHVI.

## Chapter 5

# Preference-Based Multi-Objective Optimization

The ultimate goal of multi-objective optimization is to provide potential solutions to a decision maker. Usually, what concerns a decision maker concerns is a Pareto front in an interesting/preferred region, instead of the whole Pareto front. In this chapter, a method for effectively approximating a Pareto-front approximation set in the preferred region, based on multi-objective efficient global optimization (EGO), is introduced. EGO uses Gaussian processes (or Kriging) to build a model of the objective function. Our variant of EGO uses truncated expected hypervolume improvement (TEHVI) as an infill criterion, which takes into consideration predictive mean, variance and preference region in the objective space. Compared to expected hypervolume improvement (EHVI), the probability density function in TEHVI follows a truncated normal distribution. This chapter proposes a TEHVI method that makes it possible to set a region of interest on the Pareto front and focuses the search effectively on this preferred region. An expression for the exact and efficient computation of the TEHVI for truncation over a two dimensional range is derived, and benchmark results on standard bi-objective problems for small budget of evaluations are computed, which confirms that the new approach is more effective.

## 5. PREFERENCE-BASED MULTI-OBJECTIVE OPTIMIZATION

---

### 5.1 Background

Most optimization problems involve multiple objectives that need to be considered simultaneously and under certain constraints. Unlike single objective optimization, the result consists of multiple trade-off solutions called Pareto front. Over the last 20 years, Evolutionary Multi-objective Optimization (EMO) has demonstrated a great success in approximating the whole Pareto front. However, the ultimate goal of multi-objective optimization is to assist the decision maker to choose the most suitable solution in a preferred region. Therefore, the preference-based multi-objective optimization is a hot topic recently. It utilizes the preference information offered by the DM, such as weights, reference points, trade-off constraints, to guide the search towards the Region of Interest (ROI) on a Pareto front. The overview of existing approaches has already been provided in [82, 83, 84]. Depending on when the DM can participate in the optimization process, preference-based EMO can be categorized into three types: an *a-priori*, *a-posteriori*, *interactive*. In *a-priori* method, preference information from DM is provided before the search process, on the other hand in *a-posteriori* approaches; DM preferences are incorporated after the search. *Interactive* approaches make it possible to adapt the preference during optimization by having an interaction between the DM and EMO algorithms. The method proposed in this chapter belongs to the interactive methods.

According to the preference information offered by the DM, Bechikh et al. classified existing methods into weight-based approaches, solution ranking-based approaches, objective ranking-based approaches, reference point-based approaches, trade-off-based approaches, and outranking-based approaches [85]. There are also other methods to express preference, for example, utility function [86], lexicographic order [87], and preference region [88]. In the following, we restrict the summary of the state-of-art to approaches that use objective space region (or preferred region) as preference articulation.

Desirability Functions (DFs) is widely used to specify the preferences by transforming the objective values into a decision maker's satisfaction level, considering its simple and intuitive meaning. DFs can nonlinearly map the objectives in a desired region into the domain  $[0,1]$ , based on the DFs' values of exemplary objective levels. Thus, an increasing desirability of the solution can reflect an increase of objective quality. By changing the values of objectives corresponding to 0 (least favored) and 1 (most favored), the DF can focus on different regions of the PF. It has already been successfully combined with NSGA-II [89], MOPSO [90], SMS-EMOA [91] on both benchmark problems and practical tuning problems

from machining. Karahan and Köksalan devised a territory defining steady-state elitist evolutionary algorithm (TDEA) [92], which defines a territory around each solution to prevent crowding. They also proposed a preference-based approach, called prTDEA, to assign different sizes of territories for preferred regions and non-preferred regions. Preferred regions have smaller territories so that a denser coverage could be achieved. An interactive version of this method has been proposed in [93].

In [94], an interactive decision-making approach is embedded in the preference-inspired co-evolutionary algorithm (PICEA-g). The DM can easily brush his/her preferred region in the objective space without specifying any parameters. Goal vectors are generated according to this region and co-evolved with solution vectors, in order to achieve solutions in the ROI brushed by the DM. Other methods include weighted hypervolume [95] and hyperplane construction [96], all of which can be used to specify preferred regions.

Among the existing preference-based multiobjective optimization methods, surrogate-assisted optimization is rarely used. Moreover, the combination of TEHVI is meaningful and feasible when a DM has a vague idea about his/her preference region in the objective space, because TEHVI has an inherent ability to explore a certain region in the objective space. Specifying an interval region for each objective is also referred to as brushing or zooming, and it has already been applied in the context of interactive multicriteria optimization with non-expensive function evaluations by decision makers [94]. Here we introduce such techniques to surrogate assisted multiobjective optimization.

## 5.2 Algorithms

### 5.2.1 TEHVI-EGO for Preference-based Multi-Objective Optimization

For the aim of obtaining a preferred Pareto front, TEHVI-EGO is used in this chapter to solve this problem. The details of TEHVI calculation can be found in Chapter 4. Truncated domains  $[\mathbf{A}, \mathbf{B}]$  are chosen according to a preferred region in objective space. The definition of preferred region is:

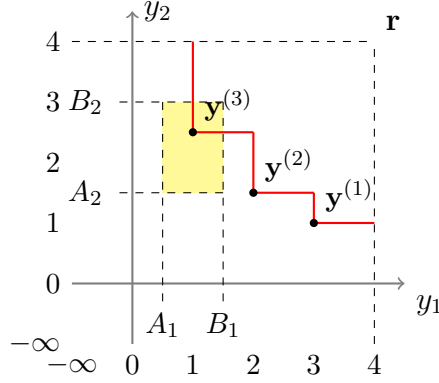
**Definition 5.1 (Preferred region)** *Given an objective space and two vectors, say  $\mathbf{A} \subset \mathbb{R}^2$  and  $\mathbf{B} \subset \mathbb{R}^2$ , a preferred region (PR) is the area bounded by  $\mathbf{A}$  and*

## 5. PREFERENCE-BASED MULTI-OBJECTIVE OPTIMIZATION

$\mathbf{B}$  in the objective space:

$$\text{PR} = \left[ \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}, \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} \right] \quad (2-1)$$

An example of preferred region is shown in Figure 5.1. The yellow region is a preferred region, and the boundary of this region is set for the truncated region for each objective function.



**Figure 5.1:** A preferred region for 2-D case.

The pseudocode for TEHVI-EGO with a preferred region is shown in Algorithm 6. It is important to compute a preliminary approximation of the Pareto front (line 3-6) before the preference region is set. This way the probability is increased and new non-dominated points can be found in this region. The interaction approach could be compared to 'zooming in' the preferred region or seen as a kind of brushing technique.

### 5.2.2 Preferred region with EAs

The concept of a preferred region can be integrated to the pre-selection criterion in Bayesian Global Optimization, and works well in Evolutionary Multi-objective Optimization. Our preferred region based Evolutionary Multi-Objective Algorithm model works reliably when the DM wants to concentrate only on those regions of the  $\mathcal{P}$  which really interests him/her. For algorithms in this model, which include T-SMS-EMOA, T-R2-EMOA and T-NSGA-II (where T stands for preferred region), three ranking criteria (1. non-dominated sorting; 2. performance

**Algorithm 6: TEHVI-EGO**


---

**Input:** initialization size  $\mu$ , termination criterion  $T_c$ , truncated boundary  $[\mathbf{A}, \mathbf{B}]$ , parameter  $ap$

**Output:** Preferred Pareto front  $\mathcal{PF}$

- 1: Initialize  $\mu$  points and Pareto front  $\mathcal{PF}$ ;
- 2:  $g = 1$ ;
- 3: **while**  $g \leq ap$  **do**
- 4:     Set infill criterion as  $EHVI$ ;
- 5:     Find the optimal point using CMA-ES;
- 6:     Update  $\mathcal{PF}$  and  $g = g + 1$ ;
- 7: DM interaction: Set  $[\mathbf{A}, \mathbf{B}]$  as truncated boundary;
- 8: **while**  $ap < g \leq T_c$  **do**
- 9:     Set infill criterion as  $TEHVI_{(\mathbf{A}, \mathbf{B})}$ ;
- 10:     Find the optimal point by maximizing TEHVI (e.g., using CMA-ES);
- 11:     Update  $\mathcal{PF}$  and  $g = g + 1$ ;
- 12: Return  $\mathcal{PF}$

---

indicator (Hypervolume in T-SMS-EMOA or R2 in T-R2-EMOA) or crowding distance in T-NSGA-II; 3. the Chebyshev distance to the preferred region) work together to achieve a well-converged and well-distributed set of Pareto optimal solutions in the preferred region using preference information provided by the DM. Non-dominated sorting is used as the first level ranking criterion, performance indicator or crowding distance as the second and the Chebyshev distance as the third level ranking criterion. The Chebyshev distance speeds up evolution toward the preferred region and is computed as the distance to the center of the preferred region.

The hypervolume, R2 indicator or crowding distance is chosen as the second level ranking criterion, which is used as a diversity mechanism and is measured based on coordinate transformations using desirability functions (DFs). The concept of desirability was introduced by Harrington [97] in the context of multi-objective industrial quality control and the approach of expressing the preferences of the DM using DFs is suggested by Wagner and Trautmann [91]. DFs map the objective values to desirabilities, which are normalized values in the interval  $[0,1]$ , where the larger the value, the more satisfying the quality of the objective value. The Harrington DF [97] and Derringer-Suich DF [98] are two most common types

## 5. PREFERENCE-BASED MULTI-OBJECTIVE OPTIMIZATION

---

of DFs and both of them result in biased distributions of the solutions on the  $\mathcal{P}$  through mapping the objective values to desirabilities based on preference information. In our algorithm model, we use a simple type of DFs, which classifies the domain of the objective function into only two classes, "unacceptable" and "acceptable". In this approach we have:

$$D(x) = \begin{cases} 1 & x \text{ is in the preferred region,} \\ 0 & x \text{ is not in the preferred region.} \end{cases}$$

The desirability here is for a solution, it is not necessary to consider desirability by each objective because the goal of our algorithm is to *zoom in* the preferred region. Therefore, we treat solutions out of the preferred region as unacceptable solutions and assign their desirabilities to be 0; at the same time, we assume that all solutions inside the preferred region are of equal importance, (i.e. acceptable) and assign their desirabilities to be 1. There is no further bias on the points in the preferred region, however, if other types of DFs are integrated into the new algorithms, it is possible to generate solutions of different distributions in the preferred region concerning the specified preferences.

For solutions with desirability 0, their second level ranking criterion is assigned to be 0. For solutions with desirability 1, their second level ranking criterion needs to be calculated further. Because only solutions in the preferred region are retained, a way is derived to simplify the calculation of the indicator values or to realize a reference point free version of indicators [99], which is based on coordinate transformation. The preferred region is treated as a new coordinate space of which the origin being the lower bound. For the maximization problem in T-SMS-EMOA or the minimization problem in T-R2-EMOA, a coordinate transformation is performed for the  $i$ -th objective as:

$$Ct_i(x) = f_i(x) - LB(f_i)$$

For the minimization problems in T-SMS-EMOA or the maximization problems in T-R2-EMOA, coordinate transformation is performed for the  $i$ -th objective as:

$$Ct_i(x) = UB(f_i) - (f_i(x) - LB(f_i))$$

where  $LB(f_i)$  and  $UB(f_i)$  are the lower bound and upper bound of the  $i$ -th objective in the preferred region, which is predefined by the DM.

The reason for distinguishing the maximization from the minimization problems when performing coordinate transformation is that the origin of the new coordinate space (i.e., the lower bound of the target region) is used as the reference point when calculating the indicator values. In T-SMS-EMOA, the worst point in the target region is chosen as the reference point when calculating hypervolume. On the other hand, the ideal point is chosen as the reference point when calculating R2 indicator in T-R2-EMOA. After coordinate transformation, the calculation of the second-ranking criterion is implemented only in the target region instead of the whole coordinate system. It does make sense because the target region is the desired space to the DM. No reference point is needed in the calculation of crowding distance, and therefore, both formulas of coordinate transformation can be chosen in T-NSGA-II.

The shape of the target region is not necessarily rectangular; it could be a circle, an ellipse or any other shape, as long as the shape can sufficiently reflect whether a solution is in the target region or not. For instance, if the DM wants the solutions to be restricted to a sphere, he/she can specify the center point and radius of the sphere and our algorithms can obtain the approximation set of the  $\mathcal{P}$  in the sphere.

**T-SMS-EMOA** The details of T-SMS-EMOA are given in Algorithm 7. The framework of T-SMS-EMOA is based on SMS-EMOA. However, after fast non-dominated sorting, all the solutions in the worst ranked front are separated into two parts (acceptable and unacceptable) by the DF. Solutions in part 1 have desirability 0 and their hypervolume contributions are assigned to be 0; solutions in part 2 have desirability 1 and coordinate transformation is conducted on each objective of each solution in this part. After that, their hypervolume contributions are calculated in the new coordinate system and the origin in the new coordinate system is adopted as the reference point. The other difference between T-SMS-EMOA and SMS-EMOA is the involvement of the Chebyshev distance. In the early iterations, it is unlikely to exist individuals in the preferred region, the Chebyshev distance works on attracting solutions towards the preferred region.

**T-R2-EMOA** The details of T-R2-EMOA are given in Algorithm 8. R2-EMOA is extended to T-R2-EMOA in the same way as SMS-EMOA is extended to T-SMS-EMOA. The formula of coordinate transformation used in T-R2-EMOA, however, is opposite to the formula used in T-SMS-EMOA for the same problem, since the origin of the new coordinate system is used as the reference point in the measure of both hypervolume indicator in T-SMS-EMOA and R2 indicator in T-R2-EMOA.

## 5. PREFERENCE-BASED MULTI-OBJECTIVE OPTIMIZATION

---



---

### Algorithm 7: T-SMS-EMOA

---

```

 $P_0 \leftarrow \text{init}(\ )$  /*Initialise random population*/
 $t \leftarrow 0$ 
repeat
     $q_{t+1} \leftarrow \text{generate}(P_t)$  /*generate offspring by variation*/
     $P_t = P_t \cup \{q_{t+1}\}$ 
     $\{R_1, \dots, R_v\} \leftarrow \text{fast-nondominated-sorting}(P_t)$ 
     $\forall x \in R_v : \text{compute } D_{Ch}(x)$  /* Chebyshev distance */
     $R_v = R_{v1} \cup R_{v2}$  /*separate acceptable and unacceptable parts:
     $\forall x \in R_{v1} : D(x) = 0; \forall x \in R_{v2} : D(x) = 1$  */
     $\forall x \in R_{v1} : HC(x) = 0$ 
     $R_{v2} \leftarrow \text{Coordinate Transformation}(R_{v2})$ 
     $\forall x \in R_{v2} : HC(x) = HV(R_{v2}) - HV(R_{v2} \setminus x)$ 
    if unique  $\text{argmin}\{HC(x) : x \in R_v\}$  exists
         $x^* = \text{argmin}\{HC(x) : x \in R_v\}$ 
    else
         $x^* = \text{argmax}\{D_{Ch}(x) : x \in R_v\}$  /*in case of tie, choose randomly*/
     $P_{t+1} = P \setminus \{x^*\}$ 
     $t \leftarrow t + 1$ 
until termination condition fulfilled

```

---

**T-NSGA-II** The details of T-NSGA-II are given in Algorithm 9. In T-NSGA-II, the size of the offspring population is the same as the size of the parent population. The next population is generated by choosing the best half solutions from the merged parent and offspring population: starting with points in the first non-domination front, continuing with points in the second non-domination front, and so on. Picking points in the descending order of crowding distance when all points in one non-domination front cannot be fully accommodated in  $P_{t+1}$  and picking points in the descending order of the Chebyshev distance when all the points with the same crowding distance can not be accommodated in  $P_{t+1}$ . Unlike T-SMS-EMOA and T-R2-EMOA, no reference point is needed in T-NSGA-II.

---

**Algorithm 8:** T-R2-EMOA

---

```

 $P_0 \leftarrow \text{init}()$  /*Initialise random population*/
 $t \leftarrow 0$ 
repeat
   $q_{t+1} \leftarrow \text{generate}(P_t)$  /*generate offspring by variation*/
   $P_t = P_t \cup \{q_{t+1}\}$ 
   $\{R_1, \dots, R_v\} \leftarrow \text{fast-nondominated-sorting}(P_t)$ 
   $\forall x \in R_v : \text{compute } D_{Ch}(x)$  /* Chebyshev distance */
   $R_v = R_{v1} \cup R_{v2}$  /*separate acceptable and unacceptable parts:
   $\forall x \in R_{v1} : D(x) = 0; \forall x \in R_{v2} : D(x) = 1$  */
   $\forall x \in R_{v1} : r(x) = 0$ 
   $R_{v2} \leftarrow \text{Coordinate Transformation}(R_{v2})$ 
   $\forall x \in R_{v2} : r(x) = R2(P \setminus \{x\}; \Lambda; i)$  /*  $i$ : ideal point*/
  if unique  $\text{argmin}\{r(x) : x \in R_v\}$  exists
     $x^* = \text{argmin}\{r(x) : x \in R_v\}$ 
  else
     $x^* = \text{argmax}\{D_{Ch}(x) : x \in R_v\}$  /*in case of tie, choose randomly*/
   $P_{t+1} = P \setminus \{x^*\}$ 
   $t \leftarrow t + 1$ 
until termination condition fulfilled

```

---

## 5.3 Empirical Experiments

### 5.3.1 TEHVI assisted EGO

**Experimental Setup** All the experiments were based on the same computer and the hardware were: Intel(R) i7-3770 CPU @ 3.40GHz, RAM 16GB. The operating system was Ubuntu 14.04 LTS (64 bit), and software were gcc 4.9.2 with compiler flag -Ofast for exact TEHVI calculation, and MATLAB 8.4.0.150421 (R2014b), 64 bit for EGO. The benchmarks were: ZDT1, ZDT2 and the generalized Schaffer problem (GSP) [81], with the parameter of  $\gamma = 0.4$ . Each experiment was repeated once. The preference regions were set as in Table 5.1: The number of initial points for all the experiments was set to 30, and the iteration number was set to 300. After initialization, 30 iterations based on TEHVI with prefer-

## 5. PREFERENCE-BASED MULTI-OBJECTIVE OPTIMIZATION

---

### Algorithm 9: T-NSGA-II

---

```

 $P_0 \leftarrow \text{init}()$  /*Initialise random population*/
 $t \leftarrow 0$ 
repeat
   $Q_t \leftarrow \text{generate}(P_t)$  /*generate offsprings by variation*/
   $P_t = P_t \cup Q_t$ 
   $\forall x \in P_t$  : compute  $D_{Ch}(x)$  /* Chebyshev distance */
   $\{R_1, \dots, R_v\} \leftarrow \text{fast-nondominated-sorting}(P_t)$ 
  for  $i = \text{rank } 1, \dots, v$  do
     $R_i = R_{i1} \cup R_{i2}$  /*separate acceptable and unacceptable parts*/
     $\forall x \in R_{i1}$  :  $D_c(x) = 0$  /*  $D_c$ : crowding distance*/
     $R_{i2} \leftarrow \text{Coordinate Transformation}(R_{i2})$ 
     $\forall x \in R_{i2}$  : compute  $D_c(x)$ 
   $P_{t+1} \leftarrow \text{half of } P_t \text{ based on rank, } D_c \text{ and then } D_{Ch}$ 
   $t \leftarrow t + 1$ 
until termination condition fulfilled

```

---

**Table 5.1:** Parameter settings.

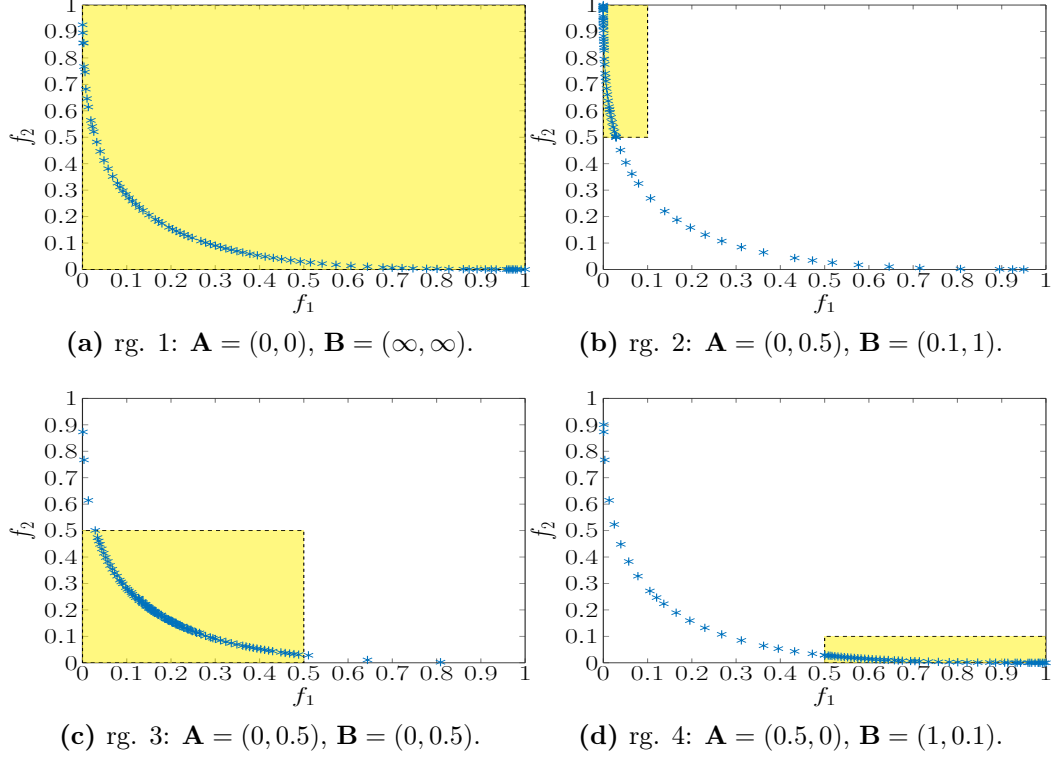
Benchmark	rg. 1		rg. 2		rg. 3		rg. 4	
	$[A_1, B_1]$	$[A_2, B_2]$	$[A_1, B_1]$	$[A_2, B_2]$	$[A_1, B_1]$	$[A_2, B_2]$	$[A_1, B_1]$	$[A_2, B_2]$
GSP	$[0, \infty]$	$[0, \infty]$	$[0, 0.1]$	$[0.5, 1]$	$[0, 0.5]$	$[0, 0.5]$	$[0.5, 1]$	$[0, 0.1]$
ZDT1	$[0, \infty]$	$[0, \infty]$	$[0, 0.5]$	$[0.5, 1]$	$[0, 0.5]$	$[0, 0.5]$	$[0.5, 1]$	$[0, 0.5]$
ZDT2	$[0, \infty]$	$[0, \infty]$	$[0, 0.5]$	$[0.5, 1]$	$[0, 0.5]$	$[0, 0.5]$	$[0.5, 1]$	$[0, 0.5]$

ence region of  $\mathbf{A} = (0, 0)^T$  and  $\mathbf{B} = (\infty, \infty)^T$  were performed for obtaining a preliminary Pareto front approximation. Then the rest of 240 iterations results, which based on the truncated domain with the precise preferred regions, is shown in Table 5.2.

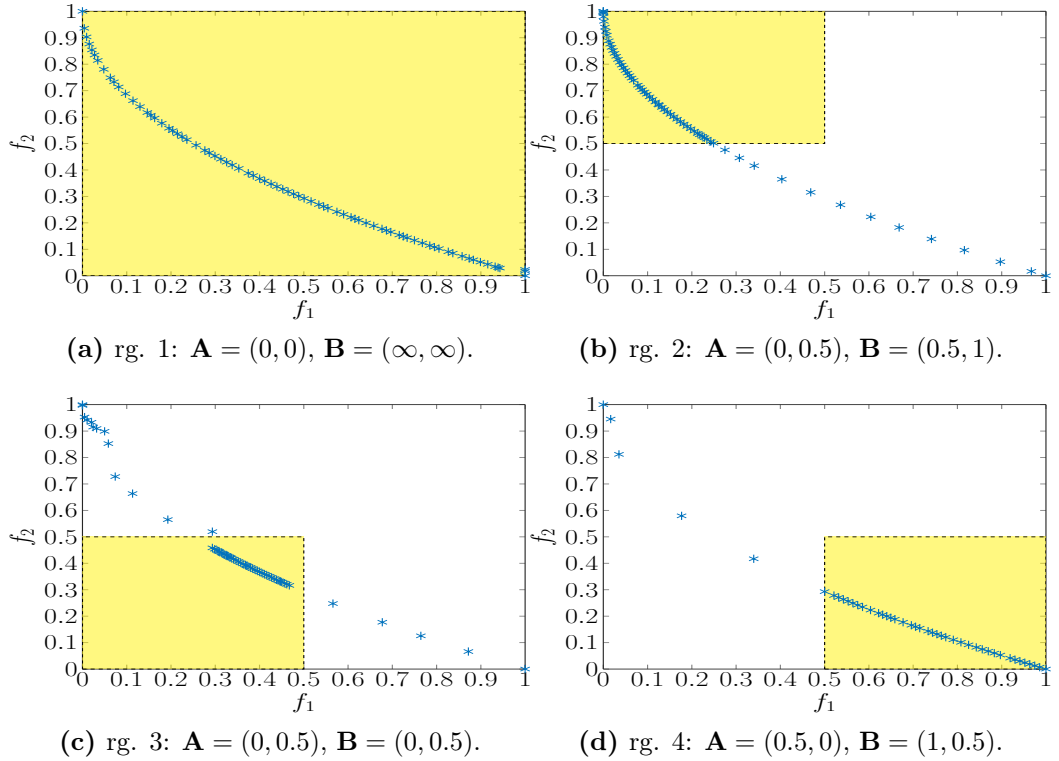
**Empirical Results** The experimental results for GSP, ZDT1 and ZDT2 problems are shown in Figure 5.2, 5.3 and 5.4 respectively. Each upper left subfigure is a Pareto front without preferred region (or, a preferred region is set as  $(\mathbf{A}, \mathbf{B}) = (\mathbf{0}^T, \infty^T)$ ). The yellow region in each figure ((b),(c) and(d)) represents the preference region.

The experiments show that most elements in a Pareto front concentrate on the

### 5.3 Empirical Experiments

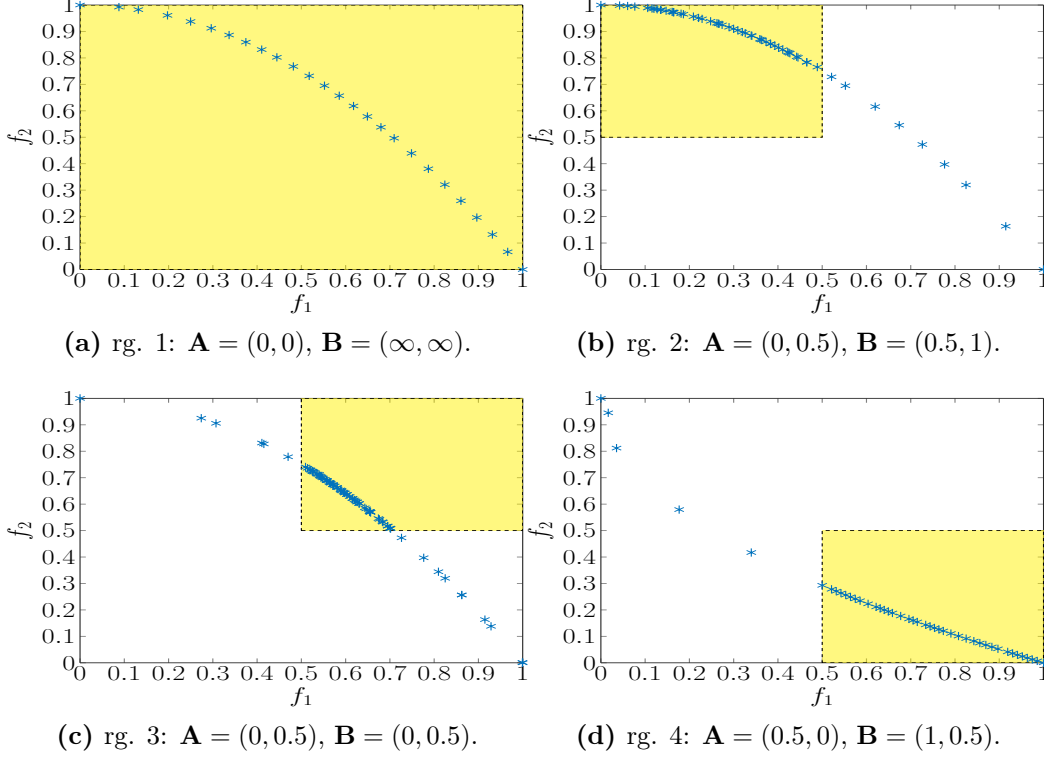


**Figure 5.2:** The preferred Pareto front for GSP problem.



**Figure 5.3:** The preferred Pareto front for ZDT1 problem.

## 5. PREFERENCE-BASED MULTI-OBJECTIVE OPTIMIZATION



**Figure 5.4:** The preferred Pareto front for ZDT2 problem.

**Table 5.2:** The size of Pareto front analysis.

Benchmark	$n$			
	rg. 1	rg. 2	rg. 3	rg. 4
GSP	112	64	115	70
ZDT1	77	78	58	80
ZDT2	112	64	115	70

corresponding preferred region, and they are adjacent to the true Pareto front. Moreover, the Pareto front approximation in a preferred region also covers the extreme boundary in this area. Since the truncated probability density function is not zero only in the truncated domain, TEHVI is not zero only in a preferred region and zero outside of the preferred region. Because of this, EGO can intensively explore the preferred region in the objective space. Some solutions exist in the outside preferred region. This is reasonable, considering the initialization and only 30 precise evaluations. Moreover, all these procedures are explorations

in the whole objective space. In the case of ZDT1 problem, the Pareto front with the preferred region in Figure 5.3 (b) can not explore the extreme boundary in the preferred region. This is probably caused by a failure of the CMA-ES [100] to locate points in this narrow part. It might be advisable to widen the interval range in such cases.

### 5.3.2 Preferred region based on EAs<sup>1</sup>

**Experimental Settings** In this section, simulations are conducted to demonstrate the performance of the algorithms, namely T-SMS-EMOA, T-R2-EMOA, and T-NSGA-II. In all simulations, we use the SBX operator with an index of 15 and polynomial mutation with an index 20 [101]. The crossover and mutation probabilities are set to 1 and  $1/N$ , respectively.

We conduct experiments on some benchmark problems, including ZDT, DTLZ and knapsack problems, to investigate the performance of the new algorithms. All experiments were run on a personal laptop with i5-5257U @ 2.7 GHz and 8G RAM. The population size and the number of evaluations are chosen to be dependent on the complexity of the test problem. Table 5.3 shows the population size and the number of evaluations (NOEs) we use on different test problems.

**Table 5.3:** Population Size and Number of Evaluations.

Problems	Population Size	NOEs
ZDT1	100	10000
ZDT2-3	100	20000
DTLZ1-2	100	30000
knapsack-250-2	200	200000
knapsack-500-2		
knapsack-250-3	250	500000
knapsack-500-3		

## Experimental Results

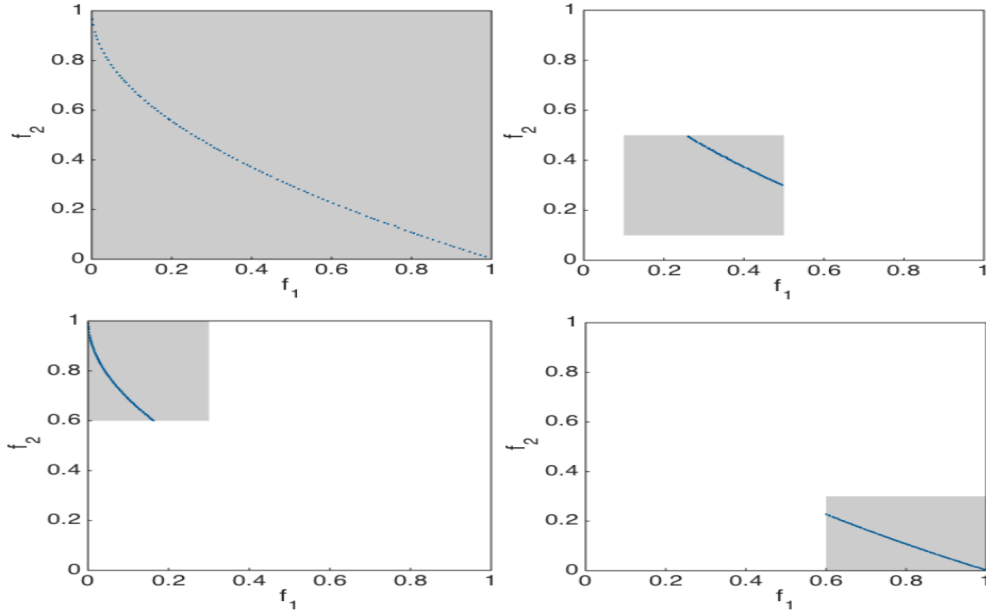
**Two-Objective ZDT Test Problems** In this section, we consider three ZDT test problems. First, we consider the 30-variable ZDT1 problem. This problem

---

<sup>1</sup>This part of work is mainly done by Yali Wang.

## 5. PREFERENCE-BASED MULTI-OBJECTIVE OPTIMIZATION

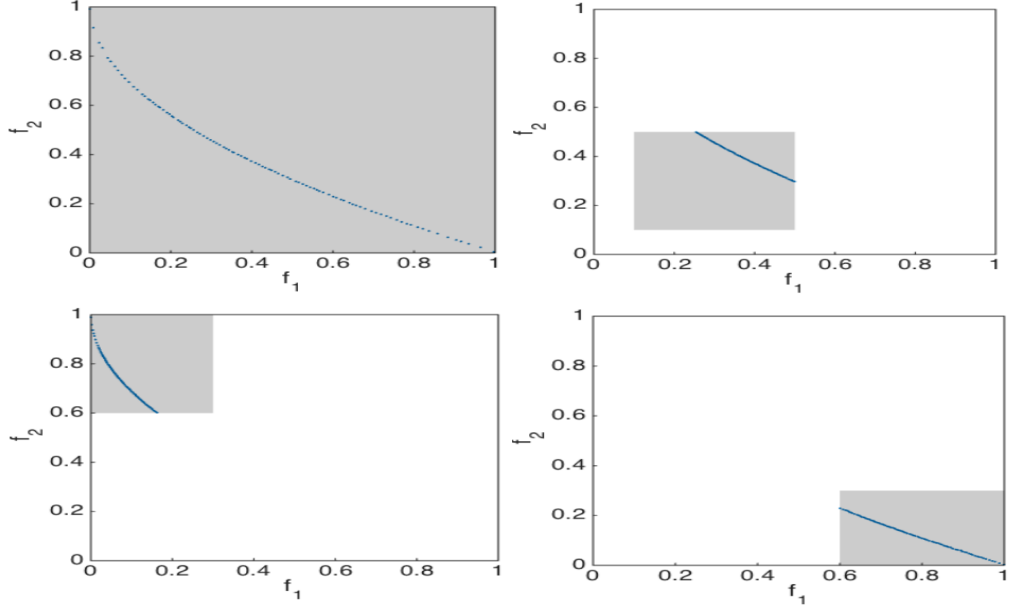
has a convex Pareto optimal front which is a connected curve and can be determined by  $f_2(x) = 1 - \sqrt{f_1(x)}$ . The true  $\mathcal{P}$  spans continuously in  $f_1 \in [0, 1]$ . Four different preferred regions are chosen to observe the performance of T-SMS-EMOA, T-R2-EMOA and T-NSGA-II in Figure 5.5, Figure 5.6 (a) and Figure 5.6 (b), respectively. The first preferred region covers the entire  $\mathcal{P}$  with the lower bound (0,0) and the upper bound (1,1). The second preferred region restricts preferred solutions to the central part of the  $\mathcal{P}$  and its lower bound is (0.1,0.1), and upper bound is (0.5,0.5). The third and fourth preferred regions take two ends of the  $\mathcal{P}$ , respectively and have their lower bounds to be (0,0.6) and (0.6,0), upper bounds to be (0.3,1) and (1,0.3), respectively.



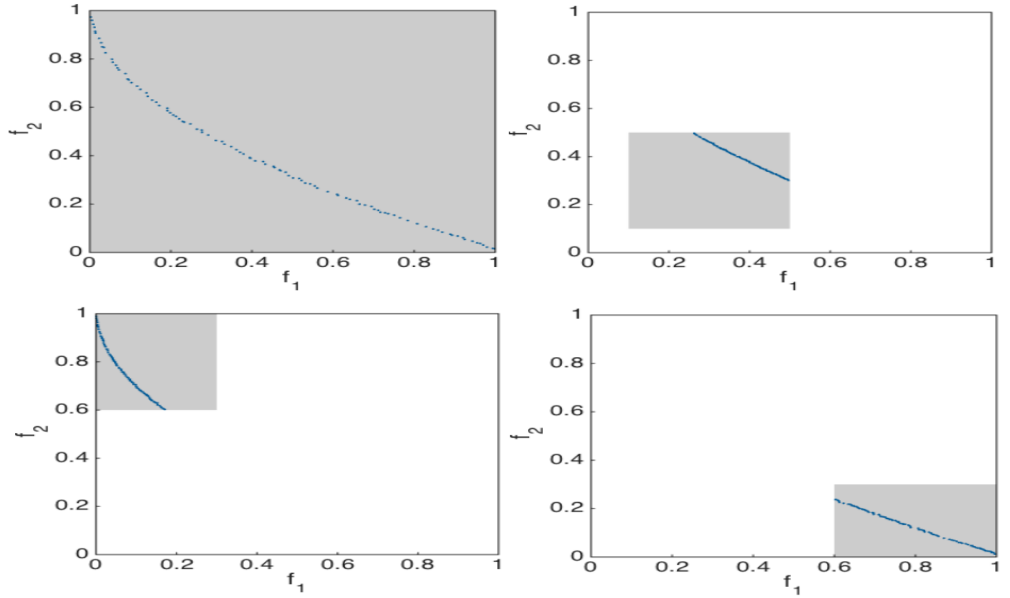
**Figure 5.5:** Representative  $\mathcal{P}$  approximations of T-SMS-EMOA on ZDT1. The different preferred regions are highlighted by gray boxes and their lower and upper bounds are: upper left graph: (0,0)(1,1), upper right graph: (0.1,0.1)(0.5,0.5), lower left graph: (0,0.6)(0.3,1), lower right graph: (0.6,0)(1,0.3).

Figure 5.5 and Figure 5.6 ((a) and (b)) show  $\mathcal{P}$  approximations obtained from the algorithms on the four different preferred regions in a random single run. It is observed that all three algorithms can find well-distributed and well-converged solutions on the  $\mathcal{P}$  in the preferred regions and no outliers exist. The solution set obtained by T-SMS-EMOA is more uniform than the solution sets obtained by the other two algorithms. It is also observable that R2 indicator has a bias towards the center of the PF.

### 5.3 Empirical Experiments



(a) Representative  $\mathcal{P}$  approximations of T-R2-EMOA on ZDT1. The different preferred regions are highlighted by gray boxes and their lower and upper bounds are: upper left graph:  $(0,0)(1,1)$ , upper right graph:  $(0.1,0.1)(0.5,0.5)$ , lower left graph:  $(0,0.6)(0.3,1)$ , lower right graph:  $(0.6,0)(1,0.3)$ .



(b) Representative  $\mathcal{P}$  approximations of T-NSGA-II on ZDT1. The different preferred regions are highlighted by gray boxes and their lower and upper bounds are: upper left graph:  $(0,0)(1,1)$ , upper right graph:  $(0.1,0.1)(0.5,0.5)$ , lower left graph:  $(0,0.6)(0.3,1)$ , lower right graph:  $(0.6,0)(1,0.3)$ .

**Figure 5.6:** The preferred Pareto fronts for ZDT1 problem.

## 5. PREFERENCE-BASED MULTI-OBJECTIVE OPTIMIZATION

We examine the performance of new algorithms using the hypervolume metric. The hypervolume is calculated within the preferred region by normalizing the values of each objective to the values between 0 and 1 and using the lower bound of the preferred region as the reference point for the maximization problem and the upper bound of a preferred region as the reference point for the minimization problem. Table 5.4 shows the median of hypervolume over 30 runs. The statistical results correspond to the observation that T-SMS-EMOA outperforms T-R2-EMOA and T-NSGA-II slightly. The original SMS-EMOA, R2-EMOA and NSGA-II are also involved in the comparison, and the results of the original MOEAs are obtained by presenting constraints in the description of a problem. Although the results of our algorithms are not better than those of original MOEAs with constraints on the range of objectives, experiments show that our algorithms can reduce computation time dramatically.

**Table 5.4:** The median of hypervolume and average computation time (Sec.) on ZDT1 with respect to different preferred regions.

New Algorithms		T-SMS-EMOA	T-R2-EMOA	T-NSGA-II
preferred region	Metric			
(0,0)	HV	0.6580	0.6566	0.6425
(1,1)	Time	24.99	74.01	0.21
(0.1,0.1)	HV	0.1640*	0.1638*	0.1543
(0.5,0.5)	Time	10.30	23.61	0.19
(0,0.6)	HV	0.8110	0.8097	0.7936
(0.3,1)	Time	12.86	31.78	0.20
(0.6,0)	HV	0.6255*	0.6233*	0.6079
(1,0.3)	Time	11.45	27.92	0.21
Original Algorithms		SMS-EMOA	R2-EMOA	NSGA-II
(0,0)	HV	0.6621	0.6610	0.6609
(1,1)	Time	108.57	314.99	0.25
(0.1,0.1)	HV	0.1694	0.1693	0.1690
(0.5,0.5)	Time	106.32	274.05	0.23
(0,0.6)	HV	0.8197	0.8185	0.8191
(0.3,1)	Time	105.73	271.00	0.21
(0.6,0)	HV	0.6364	0.6348	0.6356
(1,0.3)	Time	101.82	283.3	0.22

In the table, the symbol of “\*” on the values for the same preferred region means the medians of these algorithms are significantly indifferent. The Mann-Whitney U test (also called the Mann-Whitney-Wilcoxon (MWW), Wilcoxon rank-sum test, or Wilcoxon-Mann-Whitney test) is used to determine whether the medians of different algorithms for the same problem are significantly indifferent.

Next, we consider the 30-variable ZDT2 and ZDT3 problems. ZDT2 has a non-convex Pareto optimal front, and ZDT3 has a disconnected Pareto optimal front, which consists of five non-contiguous convex parts. Circle preferred regions are adopted in the case of ZDT2 and ZDT3 problems. A circle with a center point (1,0) and radius 0.5 intersects the whole  $\mathcal{P}$  of ZDT2 at its one end, and a circle with a center point (0.6,0.5) and radius 0.3 intersects the whole  $\mathcal{P}$  at its central part. The two different circles are chosen as examples for preferred regions on the ZDT2 problem. Experiments for a circle with a center point (0.3,0.1) and radius 0.3 as the preferred region are conducted on the ZDT3 problem.

Figure 5.7 shows PF approximation of T-SMS-EMOA in these preferred regions. Similar figures can also be achieved by T-R2-EMOA and T-NSGA-II. Orange points denote the results obtained by means of T-SMS-EMOA on provided preference information. Approximated optimal  $\mathcal{P}$  of ZDT2 problem for 100 blue points are from [102]. Statistical results of the median of hypervolume for three algorithms (T-SMS-EMOA, T-R2-EMOA and T-NSGA-II) for 30 independent runs in each preferred region are shown in Table 5.5.

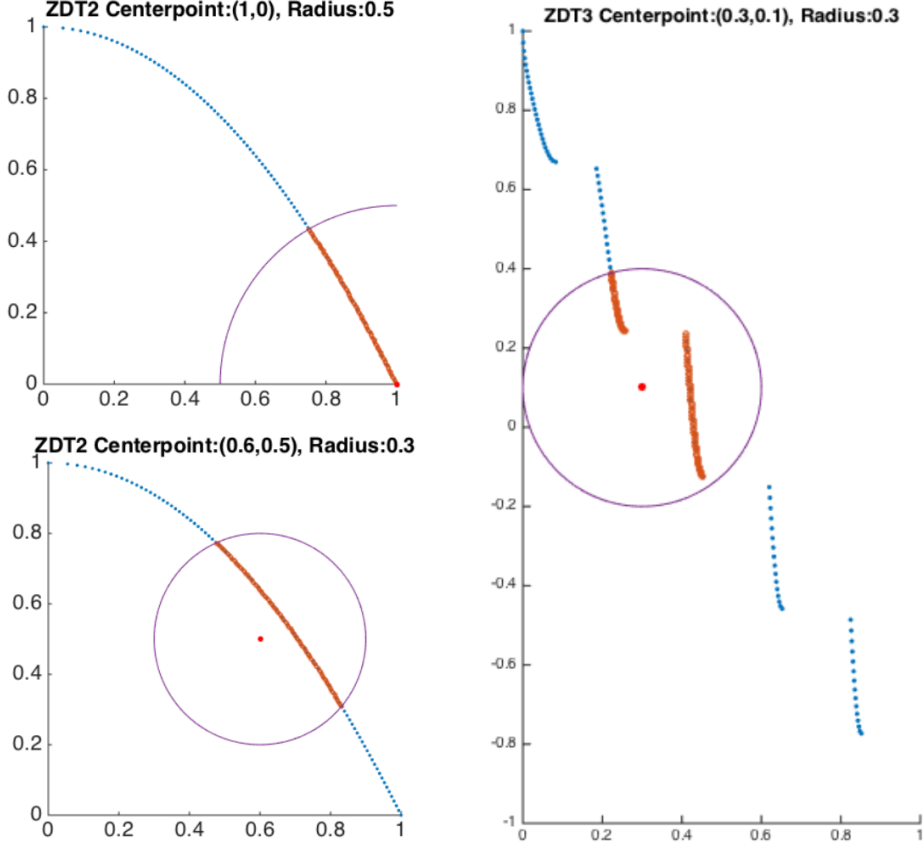
**Table 5.5:** The median of hypervolume on ZDT2 and ZDT3 with respect to different circle preferred regions.

MOEA	T-SMS-EMOA	T-R2-EMOA	T-NSGA-II
preferred region			
ZDT2 (1,0) 0.5	0.3168	0.3167	0.3159
ZDT2 (0.6,0.5) 0.3	0.3257	0.3256	0.3234
ZDT3 (0.3,0.1) 0.3	0.3377	0.3375	0.3365

## 5.4 Summary

This chapter introduced two main approaches to solve the preference-based multi-objective optimization problems. The first approach is using TEHVI as the pre-selection criterion in MOBGO. The basic idea behind TEHVI-EGO is straightfor-

## 5. PREFERENCE-BASED MULTI-OBJECTIVE OPTIMIZATION



**Figure 5.7:** Representative  $\mathcal{P}$  approximations of T-SMS-EMOA on ZDT2 and ZDT3. The preferred regions are purple circles and center points are red points.

ward: using the lower and upper bounds in THEIV to define the preferred region in the objective space, TEHVI can lead the algorithm to explore more solutions in this preferred region. TEHVI-EGO for solving the problem of preference-based multiobjective optimization is introduced. The concept of a preferred region is introduced by means of the corresponding truncated domain in TEHVI (interval range in objective space). Then, EGO can intensively explore the preferred region in objective space, and obtain preferred parts of the Pareto front. Empirical experiments were based on GSP, ZDT1 and ZDT2 problems, and show that a Pareto front can effectively converge to a preferred region through the proposed method. Compared to non-preference method, the population of the points in a preference region is larger, which means more choices are provided in this preference region. The computational cost remains small –  $O(n \log n)$  for a current

Pareto front consisting of  $n$  points. In summary, TEHVI-EGO shows a robust capability to search Pareto front segments in a particularly preferred region.

Inspired by the definition of TEHVI, preferred region based EAs are proposed and introduced in the second approach. In this chapter, a region-based multi-objective evolutionary algorithm model is also proposed. Three algorithms named T-SMS-EMOA, T-R2-EMOA and T-NSGA-II have been instantiated when combining the algorithm model with original SMS-EMOA, R2-EMOA and NSGA-II algorithms. These new algorithms have been applied to some continuous benchmark problems with two objectives. Experimental results show that our algorithms can guide the search toward the preferred region on the Pareto optimal front. No outlier appears on a reasonable number of function evaluations. Although our new algorithms presented similar performance with the original MOEAs on tested problems by integrating accessorial constraints in the problem description, our algorithms save computational efforts by guiding the search towards the preferred region without exploring the whole set of Pareto optimal solutions. On the contrary, in the case of original MOEAs, the increase in the number of constraints leads to the decrease of the search ability. Moreover, the proposed algorithms exhibit the trend of behaving better with the increase in the number of objectives, compared to the original MOEAs. The experiments on many-objective problems (i.e., problems with four or more objectives) should be conducted in future work.

# Chapter 6

## EHVI Gradient Calculation

The Expected Hypervolume Improvement (EHVI) is a frequently used infill criterion in Multi-Objective Bayesian Global Optimization (MOBGO), due to its good ability to lead the exploration. Recently, the computational complexity of EHVI calculation is reduced to  $O(n \log n)$  both in 2-D and 3-D cases. However, the maximization of EHVI, which is carried out in each iteration of the algorithm, still requires a significant amount of time. This chapter introduces a formula for the Expected Hypervolume Improvement Gradient (EHVIG) and proposes an efficient algorithm to calculate EHVIG. The new criterion (EHVIG) is utilized by two different strategies in this chapter. Firstly, it enables gradient ascent methods to be used in MOBGO. Moreover, since the EHVIG of an optimal solution should be a zero vector, it can be regarded as a stopping criterion in global optimization, e.g., an Evolution Strategy. Empirical experiments are performed on seven benchmark problems. The experimental results show that the second proposed strategy, using EHVIG as a stopping criterion, can outperform the normal MOBGO on the problems, where the optimal solutions are located in the interior of the search space. For the remaining test problems, EHVIG can still perform better when gradient projection is applied.

This chapter mainly discusses the computation of the 2-D EHVIG and how to apply EHVIG in MOBGO by two approaches: using EHVIG in gradient ascent algorithm for local search and using EHVIG as a stopping criterion in an evolutionary algorithm. The chapter is structured as follows: Section 6.1 describes the motivations of the EHVIG research; Section 6.2 introduces the definition of the EHVIG and proposes an efficient algorithm to calculate 2-D EHVIG, including a computational performance assessment between the proposed efficient exact

calculation method and numerical calculation method in 2-D EHVI case; Section 6.3 introduces the gradient method using EHVI in MOBGO; Section 6.4 illustrates how to utilize EAs (CMA-ES in this chapter) assisted by the stopping criterion EHVI in MOBGO; Section 6.5 shows the empirical experimental results.

## 6.1 Motivations

Compared to EAs, MOBGO still performs much slower using the infill criterion EHVI, because EHVI needs to be called many times in the process of searching for the optimal point based on the Kriging models. Since the calculation of the EHVI can be formulated in a closed form, it is possible to analyze its differentiability. It is easy to see, that all components of the EHVI expression are differentiable. However, a precise formula of the EHVI has not been derived until now. Once the formula of EHVI is derived, it could speed up the MOBGO in the process of searching for the optimal point by using the gradient ascent algorithm to maximize EHVI or using it as a stopping criterion in EAs.

## 6.2 Expected Hypervolume Improvement Gradient (EHVIG)

Considering the definition of the EHVI in Equation (2-1) and the efficient algorithm to calculate 2-D EHVI (minimization case) in A.4, the EHVI is differentiable with respect to the predictive mean and its corresponding standard deviation, which are again differentiable with respect to the input vector (or target point) in the search space. The EHVIG is the first order derivative of the EHVI with respect to a target point  $\mathbf{x}$  under consideration in the search space. It is defined as:

**Definition 6.1 (Expected Hypervolume Improvement Gradient)** <sup>1</sup> *Given parameters of the multivariate predictive distribution  $\boldsymbol{\mu}$ ,  $\boldsymbol{\sigma}$  at a target point  $\mathbf{x}$  in the search space, the Pareto-front approximation  $\mathcal{P}$ , and a reference point  $\mathbf{r}$ , the*

<sup>1</sup>The prediction of  $\boldsymbol{\mu}$  and  $\boldsymbol{\sigma}$  depends on a Kriging model and a target point  $\mathbf{x}$  in the search space. Explicitly, EHVIG is dependent on the target point  $\mathbf{x}$ .

## 6. EHVI GRADIENT CALCULATION

---

expected hypervolume improvement gradient (EHVIG) at  $\mathbf{x}$  is defined as:

$$EHVIG(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) = \frac{\partial \left( \int_{\mathbb{R}^d} HVI(\mathcal{P}, \mathbf{y}) \cdot PDF_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \right)}{\partial \mathbf{x}} = \frac{\partial (EHVI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}))}{\partial \mathbf{x}} \quad (2-1)$$

According to the definition of EHVIG in Equation (2-1) and the efficient algorithm to calculate EHVI in Equation (A-12), we can substitute the Equation (A-12) into Equation (2-1), say that the formula of EHVIG for 2-D case can be expressed as:

$$EHVIG(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) = \frac{\partial (EHVI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}))}{\partial \mathbf{x}} \\ = \frac{\partial \left( \sum_{i=1}^{n+1} (y_1^{(i-1)} - y_1^{(i)}) \cdot \Phi\left(\frac{y_1^{(i)} - \mu_1}{\sigma_1}\right) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) \right)}{\partial \mathbf{x}} + \quad (2-2)$$

$$\frac{\partial \left( \sum_{i=1}^{n+1} (\Psi(y_1^{(i-1)}, y_1^{(i-1)}, \mu_1, \sigma_1) - \Psi(y_1^{(i-1)}, y_1^{(i)}, \mu_1, \sigma_1)) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) \right)}{\partial \mathbf{x}} \quad (2-3)$$

For the Terms (2-2) and (2-3), the prerequisite of calculating these two Terms is to calculate the gradient of the  $\Psi$  function and of the  $\Phi(\frac{y-\mu}{\sigma})$  function. The final expressions for  $\frac{\partial \Psi(a, b, \mu, \sigma)}{\partial \mathbf{x}}$  and  $\frac{\partial \Phi(\frac{y-\mu}{\sigma})}{\partial \mathbf{x}}$  are shown in Equation (2-4) and Equation (2-5), respectively. For detailed proofs, please refer to the Appendix (A.3) in this dissertation.

$$\frac{\partial \Psi(a, b, \mu, \sigma)}{\partial \mathbf{x}} = \left( \frac{b-a}{\sigma} \cdot \phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{b-\mu}{\sigma}\right) \right) \cdot \frac{\partial \mu}{\partial \mathbf{x}} + \\ \phi\left(\frac{b-\mu}{\sigma}\right) \cdot \left( 1 + \frac{(b-\mu)(b-a)}{\sigma^2} \right) \cdot \frac{\partial \sigma}{\partial \mathbf{x}} \quad (2-4)$$

$$\frac{\partial \Phi(\frac{y-\mu}{\sigma})}{\partial \mathbf{x}} = \phi\left(\frac{y-\mu}{\sigma}\right) \cdot \left( \frac{\mu-y}{\sigma^2} \cdot \frac{\partial \sigma}{\partial \mathbf{x}} - \frac{1}{\sigma} \cdot \frac{\partial \mu}{\partial \mathbf{x}} \right) \quad (2-5)$$

By substituting Equations (2-4) and (2-5) into Term (2-2) with applying the chain

## 6.2 Expected Hypervolume Improvement Gradient (EHVIG)

---

rule, Term (2-2) can be expressed by:

$$\begin{aligned}
& \frac{\partial \left( \sum_{i=1}^{n+1} (y_1^{(i-1)} - y_1^{(i)}) \cdot \Phi\left(\frac{y_1^{(i)} - \mu_1}{\sigma_1}\right) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) \right)}{\partial \mathbf{x}} \\
&= \sum_{i=1}^{n+1} (y_1^{(i-1)} - y_1^{(i)}) \cdot \frac{\partial \left( \Phi\left(\frac{y_1^{(i)} - \mu_1}{\sigma_1}\right) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) \right)}{\partial \mathbf{x}} \\
&= \sum_{i=1}^{n+1} (y_1^{(i-1)} - y_1^{(i)}) \cdot \\
&\quad \left( \phi\left(\frac{y_1^{(i)} - \mu_1}{\sigma_1}\right) \cdot \left( \frac{\mu_1 - y_1^{(i)}}{\sigma_1^2} \cdot \frac{\partial \sigma_1}{\partial \mathbf{x}} - \frac{1}{\sigma_1} \cdot \frac{\partial \mu_1}{\partial \mathbf{x}} \right) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) + \right. \\
&\quad \left. \left( 0 - \Phi\left(\frac{y_2^{(i)} - \mu_2}{\sigma_2}\right) \cdot \frac{\partial \mu_2}{\partial \mathbf{x}} \right) + \phi\left(\frac{y_2^{(i)} - \mu_2}{\sigma_2}\right) \cdot (1 + 0) \cdot \frac{\partial \sigma_2}{\partial \mathbf{x}} \cdot \Phi\left(\frac{y_1^{(i)} - \mu_1}{\sigma_1}\right) \right) \\
&= \sum_{i=1}^{n+1} (y_1^{(i-1)} - y_1^{(i)}) \cdot \\
&\quad \left( \phi\left(\frac{y_1^{(i)} - \mu_1}{\sigma_1}\right) \cdot \left( \frac{\mu_1 - y_1^{(i)}}{\sigma_1^2} \cdot \frac{\partial \sigma_1}{\partial \mathbf{x}} - \frac{1}{\sigma_1} \cdot \frac{\partial \mu_1}{\partial \mathbf{x}} \right) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) + \right. \\
&\quad \left. \left( \phi\left(\frac{y_2^{(i)} - \mu_2}{\sigma_2}\right) \cdot \frac{\partial \sigma_2}{\partial \mathbf{x}} - \Phi\left(\frac{y_2^{(i)} - \mu_2}{\sigma_2}\right) \cdot \frac{\partial \mu_2}{\partial \mathbf{x}} \right) \cdot \Phi\left(\frac{y_1^{(i)} - \mu_1}{\sigma_1}\right) \right) \tag{2-6}
\end{aligned}$$

Similar to the derivation of Term (2-2), Term (2-3) can be expressed by:

$$\begin{aligned}
& \frac{\partial \left( \sum_{i=1}^{n+1} \left( \Psi(y_1^{(i-1)}, y_1^{(i-1)}, \mu_1, \sigma_1) - \Psi(y_1^{(i-1)}, y_1^{(i)}, \mu_1, \sigma_1) \right) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) \right)}{\partial \mathbf{x}} \\
&= \sum_{i=1}^{n+1} \left( \frac{\partial \left( \Psi(y_1^{(i-1)}, y_1^{(i-1)}, \mu_1, \sigma_1) - \Psi(y_1^{(i-1)}, y_1^{(i)}, \mu_1, \sigma_1) \right)}{\partial \mathbf{x}} \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) + \right. \\
&\quad \left. \frac{\partial \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2)}{\partial \mathbf{x}} \cdot \left( \Psi(y_1^{(i-1)}, y_1^{(i-1)}, \mu_1, \sigma_1) - \Psi(y_1^{(i-1)}, y_1^{(i)}, \mu_1, \sigma_1) \right) \right) \\
&= \sum_{i=1}^{n+1} \left( \frac{\partial \left( \Psi(y_1^{(i-1)}, y_1^{(i-1)}, \mu_1, \sigma_1) \right)}{\partial \mathbf{x}} \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) - \right. \\
&\quad \frac{\partial \left( \Psi(y_1^{(i-1)}, y_1^{(i)}, \mu_1, \sigma_1) \right)}{\partial \mathbf{x}} \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) + \\
&\quad \left. \frac{\partial \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2)}{\partial \mathbf{x}} \cdot \left( \Psi(y_1^{(i-1)}, y_1^{(i-1)}, \mu_1, \sigma_1) - \Psi(y_1^{(i-1)}, y_1^{(i)}, \mu_1, \sigma_1) \right) \right)
\end{aligned}$$

## 6. EHVI GRADIENT CALCULATION

---

$$\begin{aligned}
&= \sum_{i=1}^{n+1} \left( \left( \phi\left(\frac{y_1^{(i-1)} - \mu_1}{\sigma_1}\right) \cdot \frac{\partial \sigma_1}{\partial \mathbf{x}} - \Phi\left(\frac{y_1^{(i-1)} - \mu_1}{\sigma_1}\right) \cdot \frac{\partial \mu_1}{\partial \mathbf{x}} \right) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) - \right. \\
&\quad \left( \left[ \frac{y_1^{(i)} - y_1^{(i-1)}}{\sigma_1} \cdot \phi\left(\frac{y_1^{(i)} - \mu_1}{\sigma_1}\right) - \Phi\left(\frac{y_1^{(i)} - \mu_1}{\sigma_1}\right) \right] \cdot \frac{\partial \mu_1}{\partial \mathbf{x}} + \left[ \phi\left(\frac{y_1^{(i)} - \mu_1}{\sigma_1}\right) \cdot \right. \right. \\
&\quad \left. \left. \left( 1 + \frac{(y_1^{(i)} - \mu_1)(y_1^{(i)} - y_1^{(i-1)})}{\sigma_1^2} \right) \right] \cdot \frac{\partial \sigma_1}{\partial \mathbf{x}} \right) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) + \\
&\quad \left( \phi\left(\frac{y_2^{(i)} - \mu_2}{\sigma_2}\right) \cdot \frac{\partial \sigma_2}{\partial \mathbf{x}} - \Phi\left(\frac{y_2^{(i)} - \mu_2}{\sigma_2}\right) \cdot \frac{\partial \mu_2}{\partial \mathbf{x}} \right) \cdot \\
&\quad \left. \left( \Psi(y_1^{(i-1)}, y_1^{(i-1)}, \mu_1, \sigma_1) - \Psi(y_1^{(i-1)}, y_1^{(i)}, \mu_1, \sigma_1) \right) \right) \quad (2-7)
\end{aligned}$$

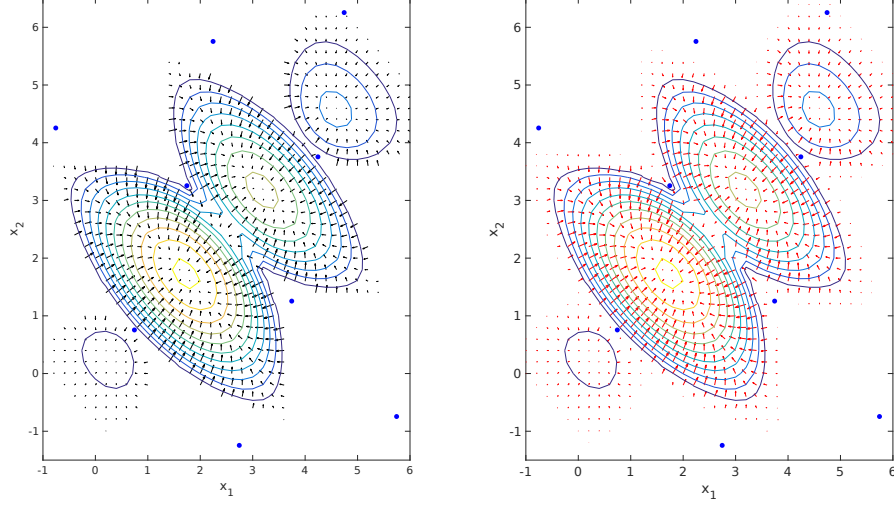
Then, the EHVI is the sum of Terms (2-6) and (2-7). In these two Terms,  $\frac{\partial \mu_i}{\partial \mathbf{x}}$  and  $\frac{\partial \sigma_i}{\partial \mathbf{x}}$  ( $i = 1, 2$ ) are the first order derivatives of the Kriging predictive means and standard deviations at a target point  $\mathbf{x}$ , respectively. These parameters can be precisely calculated by means of a Kriging model. For the details of the formulas and how to calculate these parameters, please refer to [103].

**Performance Assessment** The performance assessment of the EHVI will be illustrated by a single numerical experiment. The bi-criteria optimization problem is:  $y_1(\mathbf{x}) = \|\mathbf{x} - \mathbf{1}\| \rightarrow \min$ ,  $y_2(\mathbf{x}) = \|\mathbf{x} + \mathbf{1}\| \rightarrow \min$ ,  $\mathbf{x} \in [-1, 6] \times [-1, 6] \subset \mathbb{R}^2$  [2]. Figure 6.1 shows the landscape of EHVI, in which the evaluated points are marked by blue circles. The EHVI calculated by the exact method is indicated by the black arrow in the left figure. The EHVI calculated by the numerical method is indicated by the red arrows in the right figure. The landscapes of EHVI in both figures are very similar, however, there exist some slight differences between them, while very small and caused by numerical errors.

### 6.3 Gradient Ascent Algorithm

Previously, the optimizer *opt* in Algorithm 3 was chosen as CMA-ES [104], which is a state-of-the-art heuristic global optimization algorithm. Since the formula of 2-D EHVI is derived in this chapter, a gradient ascent algorithm can replace CMA-ES to speed up the process of finding an optimal point  $x^*$ .

Many gradient ascent algorithms (GAAs) exist. The conjugate gradient algorithm is one of the most efficient algorithms among them. However, it cannot solve the



**Figure 6.1:** The landscape of EHVI. Left: computed using exact calculation algorithm, Right: computed using numerical calculation method.

constrained problems, and this is the reason why we exclude it in this chapter. For the other GAAs, the general formula of computing the next solution is:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + s \cdot \nabla F(\mathbf{x}^{(t)}) \quad (3-8)$$

where  $\mathbf{x}^{(t)}$  is the current solution,  $\mathbf{x}^{(t+1)}$  is the updated solution,  $s$  is the stepsize, and  $\nabla F(\cdot)$  is the gradient of the objective functions or of the infill criterion. In this chapter,  $\nabla F$  is EHVI.

Another important aspect is that the starting point is very important to the performance of GAAs. In order to improve the probability of finding the globally optimal point, CMA-ES was used to initialize the starting points in this chapter. The structure of gradient ascent based search algorithm is shown in Algorithm 10.

## 6.4 Stopping Criterion – EHVI

Traditionally, when EAs are searching for the optimal point  $\mathbf{x}^*$ , convergence velocity and some other statistical criteria are used to determine whether the EAs should stop/restart or not. These criteria can balance the quality of the performance and efficiency of the execution time to some degree, but not optimally.

## 6. EHVI GRADIENT CALCULATION

---



---

### Algorithm 10: Gradient Ascent Based Search Algorithm

---

**Input:** Kriging Models  $M_1, \dots, M_d$ , Pareto-front approximation  $\mathcal{P}$ ,  
reference point  $\mathbf{r}$ , number of clusters  $N_c$

**Output:** Optimal solution  $\mathbf{x}^*$

- 1: Initialize  $\lambda$  points using CMA-ES with 15 iterations;
  - 2: Cluster  $\lambda$  points into  $N_c$  clusters  $G_1, \dots, G_{N_c}$ ;
  - 3: **for**  $i = 1$  **to**  $i \leq N_c$  **do**
  - 4:     Update starting point  $\mathbf{x}^s$ ,  $\mathbf{x}^s = \text{mean}(G_i)$  ;
  - 5:     Calculate the optimal point  $\mathbf{x}^{*i}$  using simple gradient ascent algorithm  
          and the starting point  $\mathbf{x}^s$ ;
  - 6:     Calculate the corresponding EHVI value  $EHVI^i$
  - 7: Find the optimal point  $\mathbf{x}^*$  among  $\mathbf{x}^{*1}, \dots, \mathbf{x}^{*N_c}$  ;
  - 8: Return  $\mathbf{x}^*$ ;
- 

Because all these criteria are blind to whether an individual is already the optimal or not.

Considering the gradient of the optimal point in the search space should be a zero vector and EHVI can be exactly calculated, EHVI can be used as a stopping/restart criterion in EAs when they are searching for the optimal point with the EHVI as the infill criterion. Theoretically speaking, the EHVI should be maximized during the procedure, therefore, this strategy should also be required to check the negative value of the second derivative of EHVI at this point. However, this is omitted due to the complexities. The structure of CMA-ES assisted by EHVI is shown in Algorithm 11.

## 6.5 Experimental Results

**Experimental Settings** The benchmarks were well-known test problems: BK1 [78], SSFYY1 [79], ZDT1, ZDT2, ZDT3 [80] and the generalized Schaffer problem [81] with different parameter settings for  $\gamma$ . All these benchmarks were employed by using different searching strategies in MOBGO, as shown in Table 6.1. Each trail was repeated for 10 times. All the experiments were finished on the same computer: Intel(R) i7-3770 CPU @ 3.40GHz, RAM 16GB. The operating system was Ubuntu 16.04 LTS (64 bit) and platform was MATLAB 8.4.0.150421

(R2014b), 64 bit.

---

**Algorithm 11: CMA-ES Assisted by EHVI**


---

**Input:** Kriging Models  $M_1, \dots, M_d$ , Pareto-front approximation  $\mathcal{P}$ ,  
reference point  $\mathbf{r}$ , restart number  $N_r$

**Output:** Optimal solution  $\mathbf{x}^*$

```

1: for  $i = 1$  to  $i \leq N_r$  do
2:   Initialize parameters in CMA-ES;
3:    $flag = 1$  ;
4:   for  $flag \geq \epsilon$  do
5:     Get offspring by normal CMA-ES with default parameters ;
6:     Select the best individual  $\mathbf{x}^{*i}$  from the offspring ;
7:     Predict the mean value  $\boldsymbol{\mu}^*$  and the standard deviation  $\boldsymbol{\sigma}^*$  at  $\mathbf{x}^{*i}$  ;
8:      $flag = Sum(|EHVI(\mathbf{x}^{*i}, \boldsymbol{\mu}^*, \boldsymbol{\sigma}^*, \mathcal{P}, \mathbf{r})|)$  ;
9: Find the optimal point  $\mathbf{x}^*$  among  $\mathbf{x}^{*1}, \dots, \mathbf{x}^{*N_r}$  ;
10: Return  $\mathbf{x}^*$ ;
```

---

**Table 6.1:** CMA-ES Parameter Settings.

	$\epsilon$	$N_r$	Stopping Criterion	Max Iter.	Gradient Decent	$N_c$
Alg. 1	/	3	Default	2000	No	/
Alg. 2	$10^{-5}$	3	EHVI	2000	No	/
Alg. 3	/	0	Default	15	Yes	4
Alg. 4	/	0	Default	15	No	/
Alg. 5	$10^{-5}$	3	EHVI projection	2000	No	/

Table 6.2 shows the final experimental results. The final performance on each algorithm is evaluated by HV and execution time. The highest value of HV on each test problem is indicated in bold, and the smallest value of the standard deviation of HV is also shown in bold. For the execution time, both the least execution time and smallest standard deviation of time, among Alg. 1, Alg. 2 and Alg. 3 are indicated in bold.

Here, Alg. 4 (original CMA-ES with no restart mechanism and with a max iteration of 15) is a control group for Alg. 3 to test whether the GAA works as

## 6. EHVI GRADIENT CALCULATION

---

predicted or not. Since there is no new mechanism added to Alg. 4 and max iteration is too small, the performance of Alg. 4 is indeed worse than the other three algorithms. Hence, there is no need to compare the execution time of Alg. 4 with others.

From Table 6.2, it can be seen that Alg. 3, using GAA for searching an optimal point and CMA-ES for the initialization of the starting points, can improve the final performance a little bit, compared to Alg. 4. However, it can not outperform the original CMA-ES (Alg. 1). The reason is related to the starting points in the GAA, that is: GAA is very sensitive to the starting point and the starting points generated by CMA-ES with 15 iterations are located at the local optimal area.

Compared to original CMA-ES (Alg. 1), Alg. 2 (CMA-ES using EHVI as the stopping criterion) outperforms Alg. 1 on BK1, SSFYY1, GSP, and GSP12. Among these four test problems, the execution time of Alg. 2 is much faster than Alg. 1 in the cases of the SSFYY1 and GSP problems. When applying EHVI as a stopping criterion in Alg. 2, algorithm CMA-ES can terminate the loop earlier when the EHVI of one individual is a zero vector, and therefore some execution time can be saved. In other words, while original CMA-ES does not know whether a current individual is already the optimal solution or not, EHVI can be used as a criterion to check this individual. For the BK1 and GSP12 problems, Alg. 2 needs more time, but the performance of Alg. 2 is better than Alg. 1.

On ZDT series problems, however, the performance of Alg. 2 is worse than Alg. 1. An explanation of this phenomenon is that the optimal solutions for ZDT series problems are located on the boundary of the search space. According to the definition of the gradient, EHVI would be infeasible at these boundaries, and thus EHVI would mislead CMA-ES to search the optimal solution. A remedy to improve the performance of Alg. 2 is applying the projection of EHVI to check whether an individual is optimal or not on the boundaries, instead of EHVI. Here, the projection of EHVI is the orthogonal projection of EHVI onto the active constraint boundary. Since we are only dealing with box constraints, all the components of the gradient that correspond to active boundaries in the same dimension are set to zero. In Table 6.2, compared to Alg. 2 in ZDT series problems, Alg. 5 is assisted by the projection of EHVI and can reach Pareto front approximations closer to the true ones with less execution time. For ZDT1 and ZDT2 problems, the average HV values of Alg. 5 are even better than Alg. 1 with less execution time.

## 6.5 Experimental Results

**Table 6.2:** Experimental Results.

Benchmark	Ref.			Alg. 1	Alg. 2	Alg. 3	Alg. 4	Alg. 5
BK1	(60,60)	Time(mins)	mean	<b>6.2817</b>	13.4433	8.0933	0.4350	/
			std.	<b>0.6480</b>	1.0280	0.8803	0.0166	/
		HV	mean	3175.7582	<b>3175.9683</b>	3166.4668	3133.8960	/
			std.	0.3620	<b>0.2940</b>	3.6840	6.0266	/
SSFYY1	(5,5)	Time(mins)	mean	13.1067	<b>4.7667</b>	7.2550	0.4233	/
			std.	5.4001	<b>0.3306</b>	0.3705	0.0117	/
		HV	mean	20.7096	<b>20.7098</b>	20.5474	20.0187	/
			std.	0.0069	<b>0.0035</b>	0.0361	0.1284	/
ZDT1	(11,11)	Time(mins)	mean	82.9317	76.9400	<b>15.0667</b>	6.6133	34.8383
			std.	38.5988	12.1167	<b>8.2437</b>	4.2966	14.7293
		HV	mean	120.6491	120.6488	120.6268	120.6275	<b>120.6498</b>
			std.	0.0055	<b>0.0052</b>	0.0069	0.0066	0.0063
ZDT2	(11,11)	Time(mins)	mean	40.3233	39.6800	<b>6.8889</b>	2.0983	33.8407
			std.	7.1394	6.1038	<b>0.1332</b>	0.1628	2.3391
		HV	mean	120.3025	120.2965	120.1151	119.2155	<b>120.3159</b>
			std.	0.0130	<b>0.0067</b>	0.3474	2.9890	0.0127
ZDT3	(11,11)	Time(mins)	mean	53.6267	45.9850	<b>8.5450</b>	2.8550	13.3067
			std.	8.5955	8.8638	<b>0.5120</b>	0.4217	9.0423
		HV	mean	<b>128.7486</b>	128.4772	127.7556	127.4168	128.6857
			std.	<b>0.0079</b>	0.7747	1.2385	1.2383	0.1029
GSP	(5,5)	Time(mins)	mean	46.4850	<b>7.5017</b>	13.3167	0.5283	/
			std.	40.2517	<b>0.3572</b>	0.7771	0.0112	/
		HV	mean	24.9066	<b>24.9066</b>	24.9055	24.9050	/
			std.	0.0001	<b>0.0000</b>	0.0001	0.0001	/
GSP12	(5,5)	Time(mins)	mean	20.3167	20.6650	<b>13.7200</b>	4.6867	/
			std.	<b>0.4215</b>	0.7123	0.4407	0.1403	/
		HV	mean	24.3914	<b>24.3930</b>	24.3883	24.3848	/
			std.	0.0034	0.0019	<b>0.0016</b>	0.0013	/

### 6.6 Summary

This chapter introduced an efficient algorithm to exactly calculate the 2-D EHVI and applied EHVI in *Multi-Objective Bayesian Global Optimization* using two different strategies in the process of searching for the optimal solution: using EHVI as a stopping criterion in the original CMA-ES and using gradient ascent algorithm (CMA-ES used here to initialize the starting points).

The empirical experimental results show that the gradient ascent based algorithm is much faster than original CMA-ES, but it has an obvious drawback: it gets easily stuck at a locally optimal point. Another strategy, taking EHVI as the stopping criterion in CMA-ES, can improve the quality of the final Pareto front and reduce some execution time, compared to original CMA-ES on the problem whose optimal points are not at the boundaries in the search space. This strategy does not work on ZDT series problems because EHVI cannot be calculated at the boundaries in the search space. However, a useful remedy to this strategy is the projection of EHVI.

# Chapter 7

## Applications

This chapter presents and analyses an engineered expected hypervolume improvement (EHVI) algorithm, which aims to solve the problem of PID parameter tuning and the optimization problem of controlling the substrate feed of a bio-gas plant. The EHVI is the expected value of the increment of the hypervolume indicator given a Pareto front approximation and a predictive multivariate Gaussian distribution of a new point. To solve this problem,  $\delta$ -metric selection-based efficient global optimization (SMS-EGO), EHVI based efficient global optimization (EHVI-EGO) and SMS-EMOA are used and compared in both the PID parameter tuning problem and for bio-gas plant feed optimization. The results of the experiments show that surrogate model based algorithms perform better than SMS-EMOA, and the performance of EHVI-EGO is slightly better than SMS-EGO.

This chapter is structured as follows: Section 7.1 describes the backgrounds of the bio-plant and PID parameter tuning problems. Section 7.2 introduces the PID parameter tuning task, the multi-objective nonlinear model predictive control approach as well as, the biogas process model. Section 7.3 and Section 7.4 discuss experimental studies and results.

### 7.1 Backgrounds

Based on REN21's (Renewable Energy Policy Network for the 21st Century) 2014 report, renewable energy contributed 19 percent to worldwide energy consumption and 22 percent to worldwide electricity generation in 2012 and 2013,

## 7. APPLICATIONS

---

respectively. Among renewable energy, the use of renewable energy in the form of biogas grew at a rapid pace in the past decade. Biogas is produced in bio-gas plants, where an anaerobic degradable substrate is decomposed by anaerobic bacteria in an oxygen-free environment. The main ingredient of biogas is methane, which can be used for further utilization, such as energy production.

Maximizing the yield and maintaining the bio-gas plant stability are crucial for commercial use. To this ends, bio-gas plants should work at optimal operating points. This can be achieved by adjusting the substrate mixture and by tracking the optimal setpoints [105]. However, due to the high dimensional nonlinearity of the anaerobic digestion process and due to lacking reliable measurement sensors on most full-scale bio-gas plants [106], predicting the biogas throughput and designing the optimal feedback control are challenging tasks in the field of anaerobic digestion.

Batstone et al. [107] proposed the Anaerobic Digestion Model No.1 (ADM1) in 2002, and we use it as dynamic simulation model in multi-objective nonlinear model predictive control (MONMPC). In MONMPC, there is an upper-level controller which generates the optimal methane setpoint and the corresponding optimal substrate feed that is passed through to a lower-level controller, which tracks a directly measurable process value (or setpoint). MONMPC is chosen due to the multi-objective nature of bio-gas plant operation, *maximizing the profits* and *minimizing the ecological footprint* [43].

For multi-objective black-box optimization, many algorithms exist. Evolutionary algorithms for solving these problems exist, e.g., NSGA-II [75] and SMS-EMOA [19], which, however, typically require many function evaluations ( $\gg 1000$ ) to find good approximations to Pareto fronts. In controller optimization, expensive evaluations of black-box objective functions pose typical challenges. In this chapter, we focus, therefore, on optimization with a small budget of function evaluations and use surrogate-model based optimization strategies [108], which replace exact evaluations by approximations learned from past evaluations. Recently, this idea has been generalized to multi-objective optimization (e.g. [109, 110, 111]). Algorithms that generalize efficient global optimization are S-Metric Selection EGO (SMS-EGO) [50] and Expected Hypervolume Improvement-based EGO (EHVI-EGO) [42, 60, 61]. Recently, the runtime efficiency of the exact infill criterion in the bi-objective EHVI-EGO was improved from  $O(n^3 \log n)$  to  $O(n^2)$  [42], where  $n$  is the number of points in the archive. This makes it competitive with other techniques that use computable infill criteria, in particular, SMS-EGO.

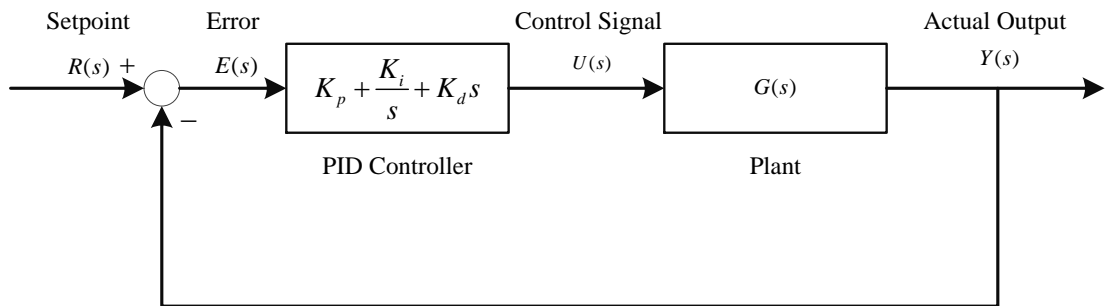
The specific contributions of this chapter are as follows: In this chapter, we

will compare algorithms in the application domain of controller optimization. In this chapter, we will provide a first application study with the new Fast EHVI-EGO. The contribution of this chapter is also to discuss concisely the definition of multi-objective model-predictive control optimization in bio-gas plant – more precisely for minimizing stage cost and terminal cost of a bio-gas plant – and of multi-objective parameter tuning of PID controllers.

## 7.2 Problem Definition

### 7.2.1 PID Parameter Tuning

This benchmark on PID parameter tuning is taken from [112]. The three parameters in PID controller are: proportionality  $K_p$ , integral  $K_i$  and derivative  $K_d$ . The transfer function of PID controller for a continuous system can be defined as:  $Y(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s$ , where  $E(s)$  and  $U(s)$  represent error signal and control signal, respectively. The basic idea of PID controller is attempting to minimize an error ( $E(s)$ ) by adjusting the process control inputs. The process of PID controller can be described as follows: when a setpoint is set or  $E(s)$  exists,  $E(s)$  will be calculated by the difference between the setpoint and actual output, and a PID controller will generate a new control signal ( $U(s)$ ) based on  $E(s)$ . Then the new control signal  $U(s)$  is applied to the plant model, and the new actual output and  $E(s)$  are generated again. The structure of a PID control is shown in Figure 7.1.



**Figure 7.1:** The structure of PID control.

**Problem and Fitness function**

The chosen transfer functions modelling the plant in this chapter are:

The chosen transfer functions modelling the plant in this chapter are:

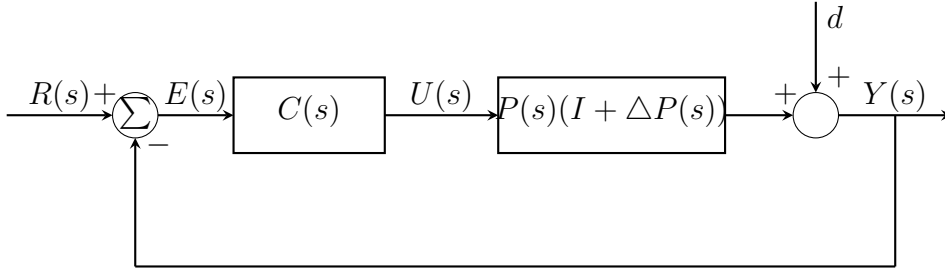
## 7. APPLICATIONS

$$G_1(s) = \frac{25.2s^2 + 21.2s + 3}{s^5 + 16.58s^4 + 25.41s^3 + 17.18s^2 + 11.70s + 1} \quad (2-1)$$

$$G_2(s) = \frac{4.228}{(s + 0.5)(s^2 + 1.64s + 8.456)} \quad (2-2)$$

The step response of these two plants is analyzed with the criteria of *settling time* ( $t_s$ ) and *percentage overshoot* ( $PO$ ). *Settling time* ( $t_s$ ) is defined as time elapsed from the application of an ideal instantaneous step input to the time, at which the output has entered error band with 2% in this chapter, while *percentage overshoot* ( $PO$ ) refers to the percentage of an output exceeding its final steady-state value.

### 7.2.2 Robust PID Tuning



**Figure 7.2:** Feedback control system with plant perturbation and external disturbance.

The structure of the feedback controller is shown in Fig 7.2, where  $R(s)$  is the reference input signal,  $E(s)$  represents error signal,  $C(s)$  is the transfer function of the controller,  $U(s)$  is control signal,  $P(s)$  stands for controlled plant,  $\Delta P(s)$  is the plant perturbation,  $d(t)$  is the external disturbance and  $Y(s)$  is the output of the system. For PID controller, three parameters are consisted in  $C(s)$ : proportionality  $B_2$ , integral  $B_1$  and derivative  $B_0$ , and the transfer function of PID controller for a continuous system can be defined as:  $C(s) = \frac{B_2s^2 + B_1s + B_0}{s}$ . The basic idea of PID controller is to attempt to minimize an error ( $E(s)$ ) by adjusting the process control inputs.

The benchmark for PID tuning is taken from [113] [114]. The transfer function of the plant is given as follows:

$$P(s) = \left( \frac{\frac{-33.98}{(98.02s + 1)(0.42s + 1)}}{\frac{-18.85}{(75.43s + 1)(0.30s + 1)}} \quad \frac{\frac{32.63}{(99.6s + 1)(0.35s + 1)}}{\frac{34.84}{(110.5s + 1)(0.03s + 1)}} \right) \quad (2-3)$$

**Objective functions** Two criteria were used here: balanced performance criterion  $J_\infty = (J_a^2 + J_b^2)^{1/2}$  [115] and interval squared error  $J_2 = \int_0^\infty e^T(t)e(t)dt$ . For  $J_\infty$ ,  $J_a$  and  $J_b$  are defined as follows:  $J_a^2 = \|W_1(s)T(s)\|_\infty$ ,  $J_b^2 = \|W_2(s)S(s)\|_\infty$ . Here,  $W_1(s)$  is the assumed boundary of plant perturbation  $\Delta P(s)$ ,  $W_2(s)$  is a stable weighting function matrix and they are defined in [115]:

$$W_1(s) = \frac{100s + 1}{s + 1000} \times I_{2 \times 2}, \quad (2-4)$$

$$W_2(s) = \frac{s + 1000}{1000s + 1} \times I_{2 \times 2}. \quad (2-5)$$

$T(s)$  and  $S(s)$  are the sensitivity and complementary sensitivity functions of the system, respectively, and they can be calculated by:

$$S(s) = (I + P(s)C(s))^{-1}, \quad (2-6)$$

$$T(s) = P(s)C(s)(I + P(s)C(s))^{-1}. \quad (2-7)$$

### 7.2.3 Bio-gas Plant Optimization

Consider a bio-gas plant fed with  $n_u \in \mathbb{N}$  substrates. Its  $n_s \in \mathbb{N}$  dimensional system state is symbolized by  $\mathbf{x} : \mathbb{R}_0^+ \rightarrow \mathcal{S}$  and its substrate feed by  $\mathbf{u} : \mathbb{R}_0^+ \rightarrow \mathcal{U}$ , with  $\mathcal{S} \subseteq \mathbb{R}^{n_s}$  and  $\mathcal{U} \subseteq \mathbb{R}^{n_u}$  denoting the state and input space, respectively. In multi-objective nonlinear model predictive control, a time dependent ( $t \in \mathbb{R}_0^+$ ) optimization problem that is defined over a prediction horizon  $T_p \in \mathbb{R}^+$  is solved at every discrete time instant  $t_k := k \cdot \delta$ , with sampling time  $\delta \in \mathbb{R}^+$  and  $k = 0, 1, 2, \dots$ , [116]. The objective is to minimize a two-dimensional objective function  $\mathbf{J} : \mathcal{S} \times \mathcal{U} \rightarrow \mathbb{R}^2$ , which depends on the open loop state  ${}^o\mathbf{s} : \mathbb{R}_0^+ \rightarrow \mathcal{S}$  and the open loop substrate feed  ${}^o\mathbf{u} : \mathbb{R}_0^+ \rightarrow \mathcal{U}$  of the controlled bio-gas plant, approximately modeled by a set of nonlinear differential equations  ${}^o\mathbf{s}'(t) = \mathbf{f}({}^o\mathbf{s}(t), \mathbf{u}(t))$ , called the bio-gas plant model  $\mathbf{f} : \mathcal{S} \times \mathcal{U} \rightarrow \mathbb{R}^{n_s}$ . The optimization problem is solved by choosing the optimal substrate feed  ${}^o\mathbf{u}$  over a control horizon  $T_c \in \mathbb{R}^+$ ,  $\delta \leq T_c \leq T_p$ . The MONMPC problem can be stated as

## 7. APPLICATIONS

---

follows:

For each  $k = 0, 1, 2, \dots$  set  $t_k = k \cdot \delta$  and solve:

$$\begin{aligned}
 & \min_{\mathbf{u}(\cdot) \in \mathcal{U}} \mathbf{J}(\mathbf{s}(\tau), \mathbf{u}(\tau)) \\
 \text{s.t. } & \mathbf{s}'(\tau) = \mathbf{f}(\mathbf{s}(\tau), \mathbf{u}(\tau)), \\
 & \mathbf{s}(t_k) = \mathbf{x}(t_k), \\
 & \mathbf{s}(\tau) \in \mathcal{S}, \forall \tau \in [t_k, t_k + T_p], \\
 & \mathbf{u} : [t_k, t_k + T_c] \rightarrow \mathcal{U}, \\
 & \mathbf{u}(\tau) = \mathbf{u}(t_k + T_c), \forall \tau \in (t_k + T_c, t_k + T_p].
 \end{aligned} \tag{2-8}$$

Since the objective function, i. e.  $\mathbf{J} := (J_1, J_2)^T$ , is vector valued, there is not one single optimal solution, but rather many trade-off solutions, which are all Pareto optimal with respect to (2-8) and collected in the so-called Pareto optimal set  $\mathcal{P}_k^* \subset \mathcal{U}$ , [117]. The trade-off solution applied to the plant,  $\mathbf{u}_k^*$ , is given by a weighted sum,  $\varpi_1, \varpi_2 \in \mathbb{R}$ :

$$\mathbf{u}_k^* := \arg \min_{\mathbf{u} \in \mathcal{P}_k^*} \sum_{i_o=1}^2 \varpi_{i_o} \cdot J_{i_o}(\mathbf{s}, \mathbf{u}), \tag{2-9}$$

and then applied for the duration of the sampling time  $\delta$ :

$$\mathbf{u}(t) = \mathbf{u}_k^*(t), \quad t \in [t_k, t_k + \delta). \tag{2-10}$$

**Objective functions** The objective function components  $J_1$  and  $J_2$  are defined as follows. The first component of the objective function is defined as:

$$J_1 := \frac{1}{T_p} \cdot \int_{t_k}^{t_k + T_p} F_1(\tau) \, d\tau + T_{p,1}, \tag{2-11}$$

It is the average of the negative financial profit  $E_{\text{plant}} := (\text{benefit} - \text{cost})$  obtained by operating the bio-gas plant over the prediction horizon  $T_p$ , with the first component of the stage cost defined as:

$$F_1(\tau) := -E_{\text{plant}}(\mathbf{s}(\tau), \mathbf{u}(\tau)). \tag{2-12}$$

The cost function is defined by the sum of the substrate and energy costs and the benefit function is defined by the profit obtained after selling the produced electrical and thermal energy, which, in Germany, is determined by the Renewable Energy Sources Act - EEG [118]. The minus sign in eq. (2-12) is added because the optimization problem in eq. (2-8) is formulated as a minimization problem.

In eq. (2-11) the first component of the terminal cost  $T_{p,1}$  is used, which is defined as:

$$T_{p,1} := \kappa_{T,1} \cdot F_1(t_k + T_p), \quad (2-13)$$

with the weighting factor  $\kappa_{T,1} \in \mathbb{R}^+$ .

The second component of the objective function  $J_2$ ,

$$J_2 := \frac{1}{T_p} \cdot \int_{t_k}^{t_k+T_p} F_2(\tau) d\tau + \int_{t_k}^{t_k+T_p} \|\mathbf{u}'(\tau)\|_2^2 d\tau + T_{p,2}, \quad (2-14)$$

contains a weighted sum of all  $n_c \in \mathbb{N}_0$  boundary conditions that are active over the prediction horizon  $T_p$ , defined in the second component of the stage cost  $F_2$ :

$$F_2(\tau) := \sum_{i_c=1}^{n_c} \kappa_{i_c} \cdot \text{bound}_{i_c}({}^o\mathbf{s}(\tau), {}^o\mathbf{u}(\tau)). \quad (2-15)$$

Furthermore,  $J_2$  contains the integral over the change of the open loop control input  ${}^o\mathbf{u}$  and the terminal penalty term  $T_{p,2}$  with the weighting factor  $\kappa_{T,2} \in \mathbb{R}^+$ :

$$T_{p,2} := \kappa_{T,2} \cdot F_2(t_k + T_p). \quad (2-16)$$

In eq. (2-15) the weights  $\kappa_{i_c} \in \mathbb{R}^+$  are normalized,  $\sum_{i_c=1}^{n_c} \kappa_{i_c} = 1$ , and the boundary conditions are defined as:

$$\text{bound}_{i_c} : \mathcal{S} \times \mathcal{U} \rightarrow \begin{cases} 0 < \dots \leq \left(1 \text{ or } \frac{4.6851^2}{6}\right) & \text{if active,} \\ 0 & \text{else.} \end{cases} \quad (2-17)$$

Such that all constraints are smooth, some of the them are implemented using the Tukey biweight function  $\rho_T : \mathbb{R} \rightarrow \mathbb{R}^+$ , which is defined as, with  $C_T := 4.6851$  [119]:

$$\rho_T(u_T) := \begin{cases} \frac{C_T^2}{6} \left[ 1 - \left( 1 - \left( \frac{u_T}{C_T} \right)^2 \right)^3 \right] & |u_T| \leq C_T, \\ \frac{C_T^2}{6} & \text{else.} \end{cases} \quad (2-18)$$

In eq. (2-18),  $u_T \in \mathbb{R}$  must be replaced by the difference between the constrained value and its boundary condition.

Examples for the  $n_c$  constraint functions  $\text{bound}_{i_c}$ ,  $i_c = 1, \dots, n_c$ , are upper and lower boundaries for VFA/TA, COD degradation rate, pH value, OLR, HRT,  $\text{NH}_4\text{-N}$  and VFA (for details see [43]).

## 7. APPLICATIONS

---

### Bio-gas Process:

In the simulation experiments performed in this chapter, a model of a bio-gas plant is used. This model and the real plant are described here. The modeled bio-gas plant is a full-scale agricultural bio-gas plant with an electrical power of 500 kW, located in Germany. The plant is configured as a two-stage system with a primary digester (1st) ( $V_{\text{liq}} = 1977 \text{ m}^3$ ) and a secondary (or post) digester ( $V_{\text{liq}} = 4182 \text{ m}^3$ ), whereas the secondary digester also serves as a final storage tank. A pumping station offers the possibility of interchanging sludge between both digesters. The first digester is mainly fed with maize silage, swine, and cattle manure as well as grass silage. The secondary digester is not fed. The produced biogas is burned in two CHPs with an electrical power of 250 kW each. The produced electrical power is injected into the local grid, which is enumerated by the EEG 2009. Both digesters are heated with the thermal energy produced by the CHPs and are operated at about 40 °C.

The simulation model is implemented in MATLAB<sup>®</sup> in a self-developed toolbox, which is freely available under the terms of the GNU GPL.

### 7.3 Experimental Settings

**PID Parameter Tuning** For the PID controller problem, the following parameters were used. The search space for the three parameters ( $K_p$ ,  $K_i$  and  $K_d$ ) in  $G_1(s)$  [120] and  $G_2(s)$  [112] is shown in Table 7.1.

**Table 7.1:** Parameter setting.

	$G_1(s)$			$G_2(s)$		
	SMS-EMOA	SMS-EGO	EHVI-EGO	SMS-EMOA	SMS-EGO	EHVI-EGO
$K_p$	[0,10]	[0,10]	[0,10]	[0,10]	[0,10]	[0,10]
$K_i$	[0,13]	[0,13]	[0,13]	[0,6]	[0,6]	[0,6]
$K_d$	[0,18]	[0,18]	[0,18]	[0,6]	[0,6]	[0,6]
$n_{\text{pop}}$	30	32	32	30	32	32
$n_{\text{eval}}$	90	50	50	90	50	50

**Robust PID tuning** The parameters for the algorithms are shown in Table 7.2.

**Table 7.2:** Parameter settings.

Algorithm	$\mu$ /Initial Population	$\lambda$	iteration	pc	pm
EHVI-EGO	30	/	200	/	/
TEHVI-EGO	30	/	200	/	/
NSGA-II	30	30	200	0.9	1/N
SMS-EMOA	30	/	200	0.9	1/N

The TEHVI-EGO boundary  $((A, B))$  for all the experiments are  $(0, \infty)$ , except for ZDT3 problem. Since the lower bound of ZDT3 is close to  $-1$ , the TEHVI-EGO boundary for ZDT3 was set to  $(-1, \infty)$ . For the generalized Schaffer problem, the parameter  $\gamma$  was set to  $\gamma = 0.4$ . All the experiments were repeated five times.

**Bio-gas Plant Optimization** The simulation of biogas is composed by two stages: intermediate process and steady state. In the first stage, the objective functions are  $J_1$  and  $J_2$  described in section 7.2. For steady state, the objective functions are  $F_1$  and  $F_2$  described in section 7.2. Three different initial substrate feeds, as shown in Table 7.3, were set to test whether the MONMPC is robust against initial substrate feed. SMS-EGO and EHVI-EGO were used and compared, the parameters of these two algorithms can be found in Table 7.3, where  $T_p$ ,  $T_c$  and  $\delta$  are the control parameters and represent prediction horizon, control horizon, and control sampling time, respectively.

## 7.4 Experimental Results

### 7.4.1 PID Parameter Tuning

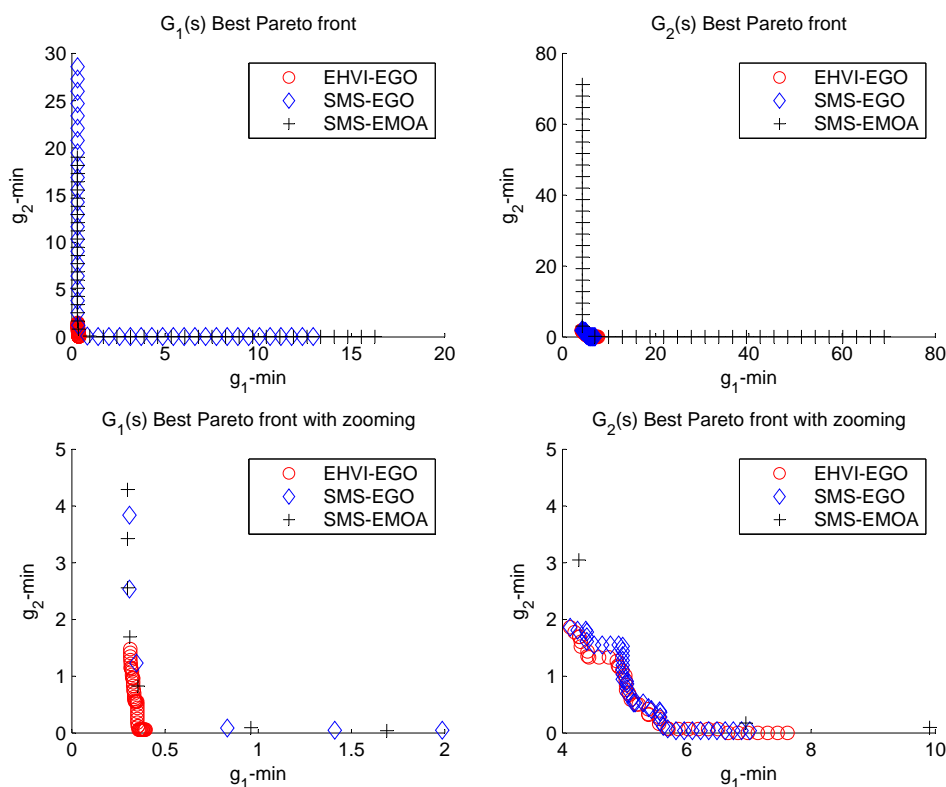
For PID parameter tuning, the final best and average non-dominated Pareto fronts in the 20 runs are shown in Figures 7.3 and 7.4, where  $g_1$  and  $g_2$  represent *settling time* ( $t_s$ ) and *percentage overshoot* ( $PO$ ), respectively. The search space for  $G_1(s)$  and  $G_2(s)$  are described in Table 7.1.

Figures 7.3 and 7.4 refer to the best and the average Pareto fronts in all runs respectively, and these are generated by using attainment curves of toolbox *plot-atta* [34, 36].

## 7. APPLICATIONS

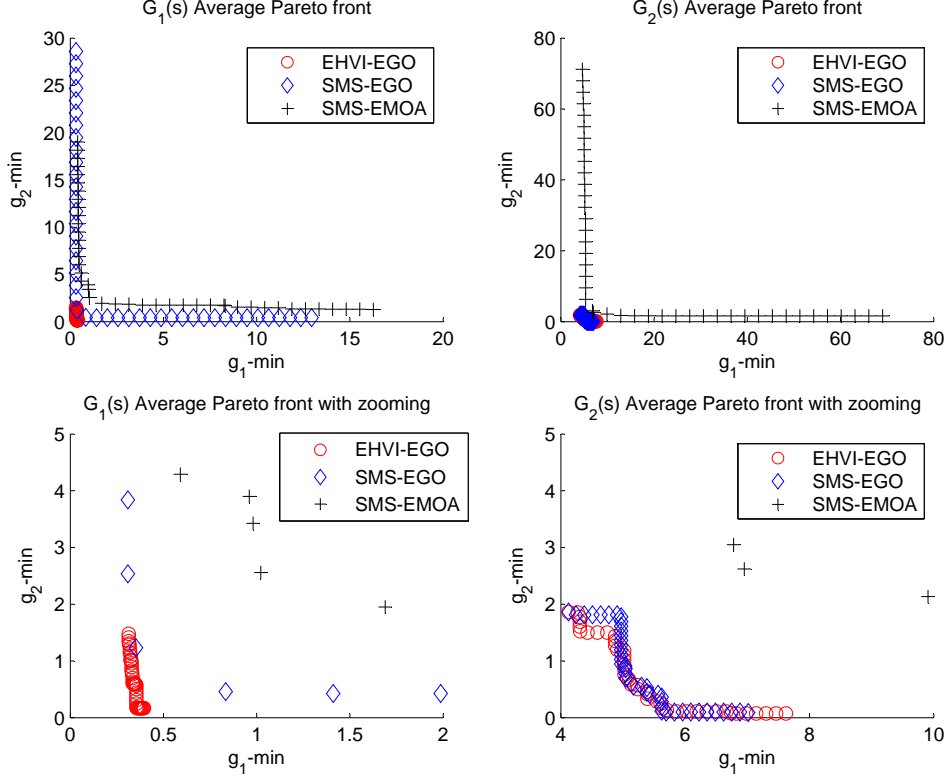
**Table 7.3:** Parameter sets for all experiments.

Component	Test A	Test B	Test C	LB	UB	unit
$Q_{\text{maize}}$	15	5	40	0	30	$\text{m}^3/\text{d}$
$Q_{\text{manure}}$	10	5	30	5	15	$\text{m}^3/\text{d}$
$Q_{\text{grass}}$	2	0	10	0	30	$\text{m}^3/\text{d}$
Exp. no.	$T_p/\text{d}$	$T_c/[\text{d}]$	$\delta/[\text{d}]$	$n_{\text{pop}}$	$n_{\text{eval}}$	Method
1	150	10	10	32	50	SMS-EGO
2	150	10	10	32	40	SMS-EGO
3	150	10	10	32	60	SMS-EGO
4	150	10	10	32	50	EHVI-EGO
5	150	10	10	32	40	EHVI-EGO
6	150	10	10	32	60	EHVI-EGO
7	150	10	10	20	60	SMS-EMOA
8	150	10	10	25	75	SMS-EMOA
9	150	10	10	20	80	SMS-EMOA



**Figure 7.3:** Best Pareto fronts.

## 7.4 Experimental Results



**Figure 7.4:** Average Pareto fronts.

Comparing the best Pareto fronts generated by surrogate-model-based algorithms and model-free algorithm in Figure 7.3, almost all the elements in SMS-EMOA best Pareto front are dominated by those generated by SMS-EGO and EHVI-EGO, and surrogate model based algorithms (EHVI-EGO and SMS-EGO) outperform model-free algorithm (SMS-EMOA). Comparing the best Pareto front generated by the EHVI-EGO and SMS-EGO in  $G_1(s)$  and  $G_2(s)$ , EHVI-EGO is slightly more robust to reach the “ideal” Pareto front as close as possible than the SMS-EGO.

The difference between best and average Pareto fronts in EHVI-EGO and SMS-EGO are not obviously. This means that when compared to model-free optimization algorithms, surrogate-model-based algorithms can faster converge to the “ideal” Pareto front with fewer evaluations, and EHVI-EGO is more robust to converge than SMS-EGO within a certain number of evaluations.

For statistic analysis, hypervolume in all the experiments are analysed in Table 7.4. The reference points for  $G_1(s)$  and  $G_2(s)$  in Table 7.4 are  $[5, 5]$  and  $[10, 5]$ , respectively.

## 7. APPLICATIONS

**Table 7.4:** HV Comparison.

	$G_1(s)$				$G_2(s)$			
	AVG	MIN	MAX	$\sigma$	AVG	MIN	MAX	$\sigma$
EHVI-EGO	23.38	23.33	23.40	0.02	27.34	26.87	27.62	0.25
SMS-EGO	21.26	14.03	23.37	3.45	27.23	26.04	27.48	0.32
SMS-EMOA	18.95	13.11	22.50	2.81	19.01	3.08	27.49	7.26

From Table 7.4, surrogate-model-based algorithms are more robust compared to a model-free algorithm (SMS-EMOA). EHVI-EGO provides a higher average value of HV and lower value of corresponding standard deviation than those of the other two algorithms. Therefore, in the case of PID parameter tuning problem, EHVI-EGO is more robust and performs better than SMS-EGO and SMS-EMOA.

Figures 7.5 and 7.6 show the step response for  $G_1(s)$  and  $G_2(s)$ . In Figure 7.3, each set of PID parameters corresponds to the first suggested point sorted by NSGA-II [75]. The computation time  $t_s$ , corresponds with EHVI-EGO is faster than that of SMS-EGO and SMS-EMOA, and  $PO$  using EHVI-EGO is also smaller than the other two algorithms.

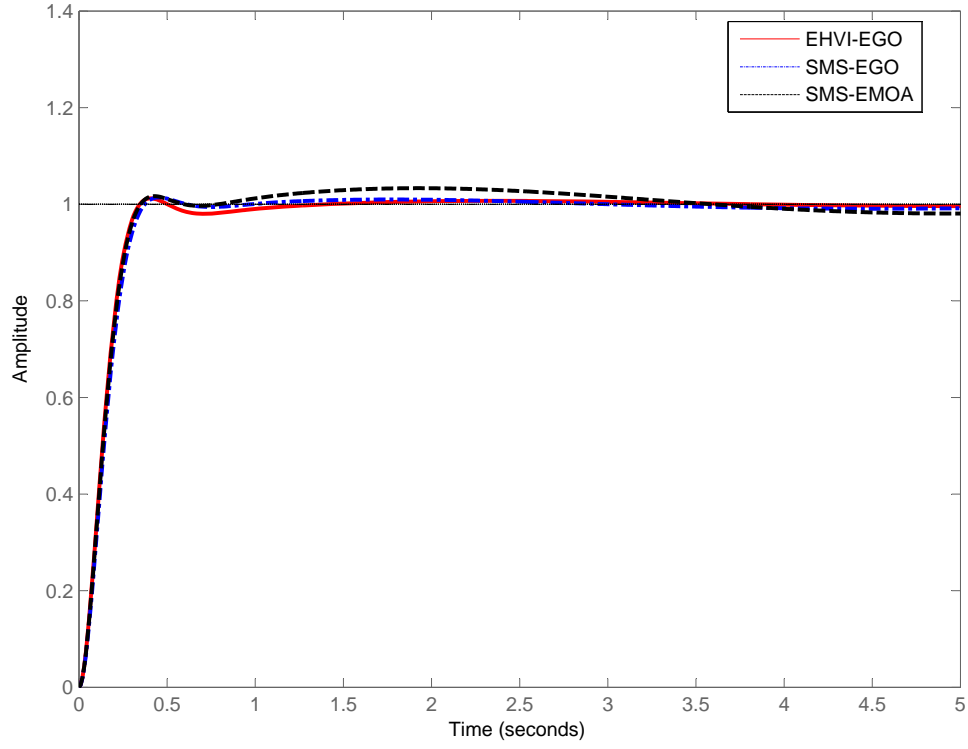
### 7.4.2 Robust PID Tuning

**Table 7.5:** Robust PID parameter tuning.

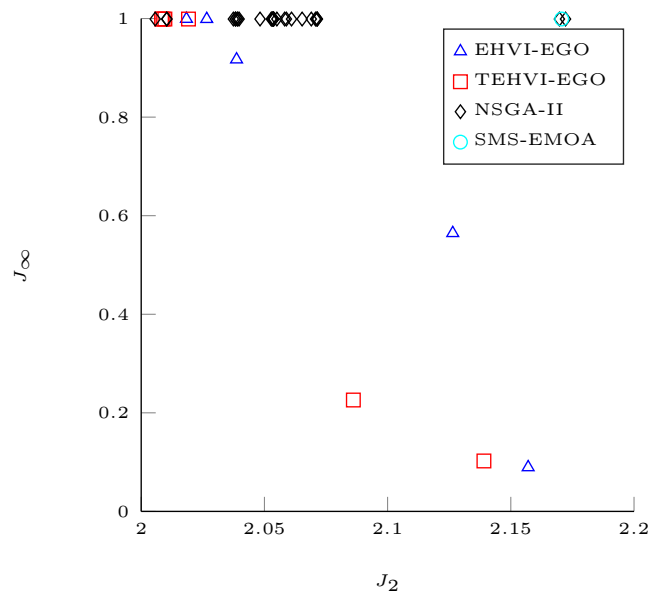
Test Function	Methods	reference point	Pareto front size				HV			
			max	min	mean	std	max	min	mean	std
PID	EHVI-EGO	[30 2]	6	3	4.6	1.1402	53.7859	32.5078	47.4223	8.6584
PID	TEHVI-EGO	[30 2]	5	3	4.2	0.8367	<b>53.9046</b>	38.6312	<b>49.0419</b>	6.3154
PID	NSGA-II	[30 2]	54	36	<b>45</b>	8.3066	28.0222	27.9868	28.0054	0.0142
PID	SMS-EMOA	[30 2]	9	1	5.2	3.0332	53.3783	27.4147	36.8671	11.8578

Table 7.5 shows the result of the robust PID parameter tuning problem. Figure 7.7 shows the best Pareto fronts of each method for this problem. It shows that TEHVI-EGO can explore more non-dominated candidates than the other algorithms, and almost all the other Pareto fronts are dominated by the (red) squared Pareto front of EHVI-EGO.

## 7.4 Experimental Results

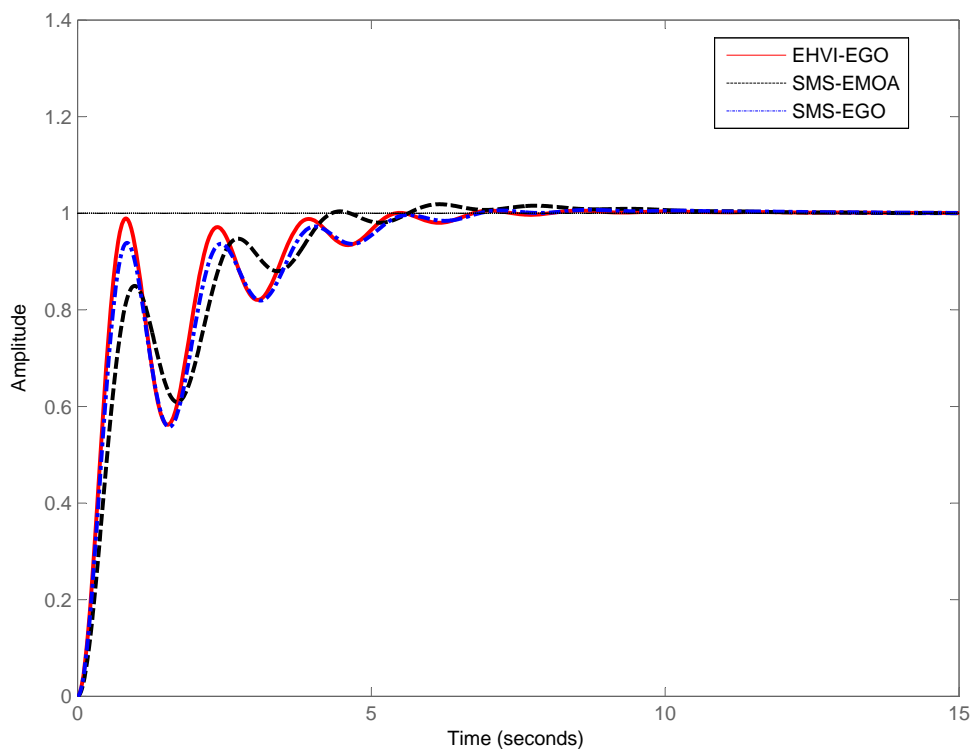


**Figure 7.5:**  $G_1(s)$  step response. For EHVI-EGO  $K_p = 1.95$ ,  $K_i = 3.07$ ,  $K_d = 4.61$ , for SMS-EGO  $K_p = 2.25$ ,  $K_i = 2.92$ ,  $K_d = 4.20$ , and for SMS-EMOA  $K_p = 2.00$ ,  $K_i = 4.04$ ,  $K_d = 4.48$ .



**Figure 7.7:** The best Pareto fronts for PID problem.

## 7. APPLICATIONS



**Figure 7.6:**  $G_2(s)$  step response. For EHVI-EGO  $K_p = 2.71$ ,  $K_i = 1.05$ ,  $K_d = 2.13$ , for SMS-EGO  $K_p = 1.99$ ,  $K_i = 0.98$ ,  $K_d = 1.35$ , and for SMS-EMOA  $K_p = 2.35$ ,  $K_i = 0.96$ ,  $K_d = 1.97$ .

**Table 7.6:** PID parameters in Figure 7.8.

Algorithm	B2		B1		B0	
EHVI-EGO	$\begin{bmatrix} -146.8052 & -183.1330 \\ -39.5844 & 38.28307 \end{bmatrix}$		$\begin{bmatrix} -58.8757 & -39.9390 \\ 39.1692 & 164.6072 \end{bmatrix}$		$\begin{bmatrix} -29.1297 & 168.1379 \\ 11.2049 & 148.3339 \end{bmatrix}$	
THEIV-EGO	$\begin{bmatrix} -110.9551 & 197.8837 \\ 198.4738 & 199.5100 \end{bmatrix}$		$\begin{bmatrix} 45.5191 & 194.3873 \\ 80.6678 & -188.7033 \end{bmatrix}$		$\begin{bmatrix} -30.7334 & 151.2778 \\ 76.1469 & 35.5006 \end{bmatrix}$	
NSGA-II	$\begin{bmatrix} -68.0186 & 199.9837 \\ 83.4419 & 21.5003 \end{bmatrix}$		$\begin{bmatrix} -15.0820 & 199.8181 \\ 117.6667 & 200 \end{bmatrix}$		$\begin{bmatrix} 74.1652 & 139.4757 \\ 58.1943 & 94.3620 \end{bmatrix}$	
SMS-EMOA	$\begin{bmatrix} -44.6694 & 198.9674 \\ 113.6318 & 198.1877 \end{bmatrix}$		$\begin{bmatrix} -123.9725 & 69.6727 \\ -44.6645 & 149.6474 \end{bmatrix}$		$\begin{bmatrix} -34.5741 & 158.8530 \\ -10.7696 & 116.9635 \end{bmatrix}$	

Figure 7.8 shows the step response with plant perturbation, and the PID param-

eters are shown in Table 7.6. The objective values for each method are: EHVI-EGO = (2.1264, 0.5646), TEHVI-EGO = (2.1391, 0.1023), NSGA-II = (2.1722, 0.9990) and SMS-EMOA = (2.1707, 0.9991). It is clear that the EGO-based controller outputs are slightly better than those generated by the other two algorithms. For output  $y_2$ , the overshoot of the TEHVI-EGO step response is bigger than the other three responses. This is acceptable, considering that the objective functions do not have the overshoot criterion, and the overshoot is also less than 20%.

### 7.4.3 Bio-plant Optimization

For the bio-gas plant, two strategies of setting reference point are used: fixed and dynamic reference points. The fixed reference point is always (0.1, 1.6), while, the dynamic reference point is determined on the fly by the algorithm and is calculated by adding one to the maximum value of the existing Pareto fronts. The initial substrate feeds and parameters in the experiments are represented in Table 7.3. Bio-gas plant simulation costs a lot of time (about 15-24 hours), and thereby, we did the experiments only once.

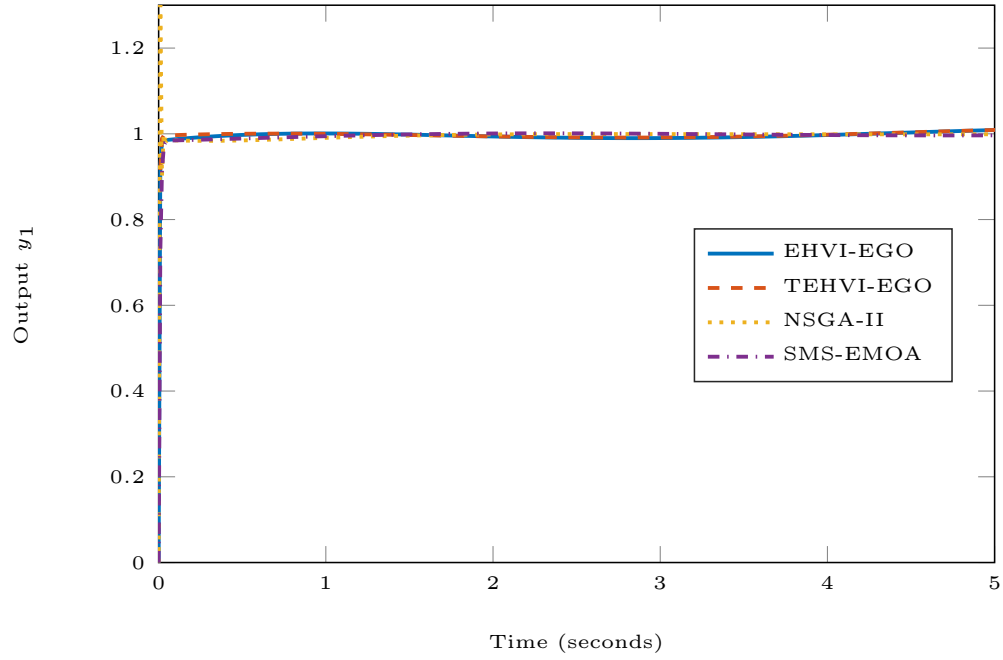
Figure 7.9 illustrates the final intermediate Pareto fronts with dynamic reference point and initial substrate feeds of experiments A and C, respectively.

In Figure 7.9, nearly all elements in the SMS-EGO and SMS-EMOA intermediate Pareto fronts are dominated by those in EHVI-EGO intermediate Pareto fronts. When compared the performance of SMS-EGO with that of SMS-EMOA, SMS-EMOA is a little bit better than SMS-EGO at the cost of 150, 525 and 300 more evaluations in Exp. 1&7, 2&8 and 3&9, respectively. The choice of the reference point is very important in these experiments. The HV differs a lot when choosing different reference points. Take an example in Figure 7.9 (a). When compared the red and blue Pareto fronts in the right subfigure, only one element in EHVI-EGO Pareto front is dominated by that of SMS-EGO, and HV for A3 and A6 are 0.1381 and 0.1584, respectively with the same reference point of (-1.8, 0.7). However, HV for A3 and A6 are 3.3995 and 3.3758 using the reference point of (0.1, 1.6). Hence, the criterion of HV is very sensitive to the reference point, and the bad choice of reference point can mislead the final decision, even when the results are very clear.

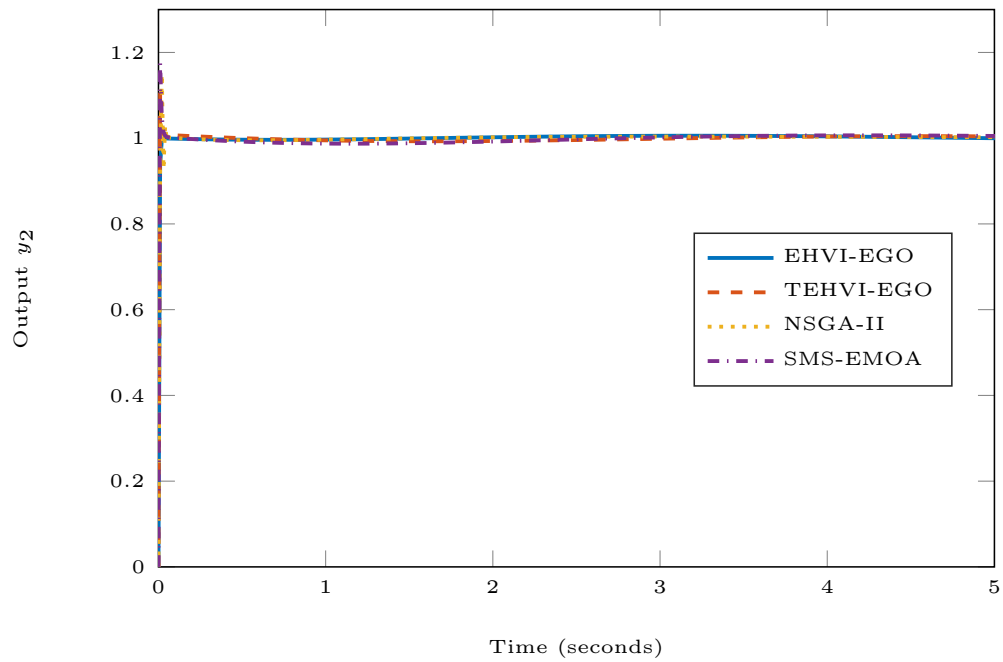
Figure 7.10 shows the steady Pareto front in Exp. A and C. The performance gap between all the three algorithms is not obvious. Even some few elements in the Pareto front, generated by EHVI-EGO, are dominated by those of SMS-EMOA and SMS-EMOA. When compared the intermediate with steady Pareto fronts us-

## 7. APPLICATIONS

---

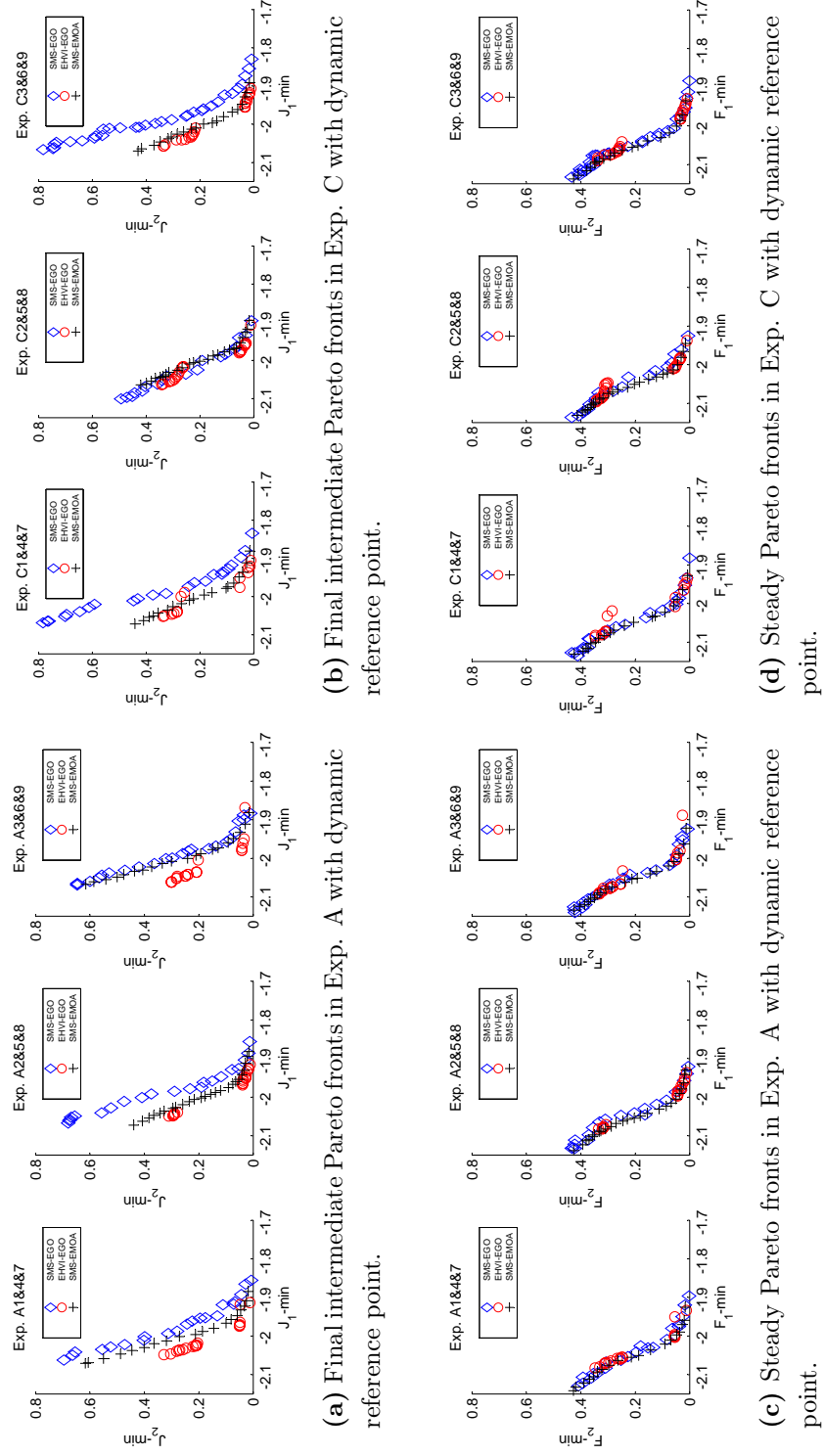


(a) The step response for output  $y_1$ .



(b) The step response for output  $y_2$ .

**Figure 7.8:** Step responses.



**Figure 7.9:** Experimental results on Exp. A & C, using dynamic reference point strategy.

## 7. APPLICATIONS

ing a dynamic reference point, the performance gap of the three algorithms diminished for steady state. That is to say, the predictions by surrogate-model-based algorithms still process a slight difference to the real data, for steady state.

Figure 7.10 (a) and (b) are both the intermediate Pareto fronts using a fixed reference point with different initial substrate feed strategies (A and B). Figure 7.10 (c) and (d) are the steady Pareto fronts based on fixed reference point.

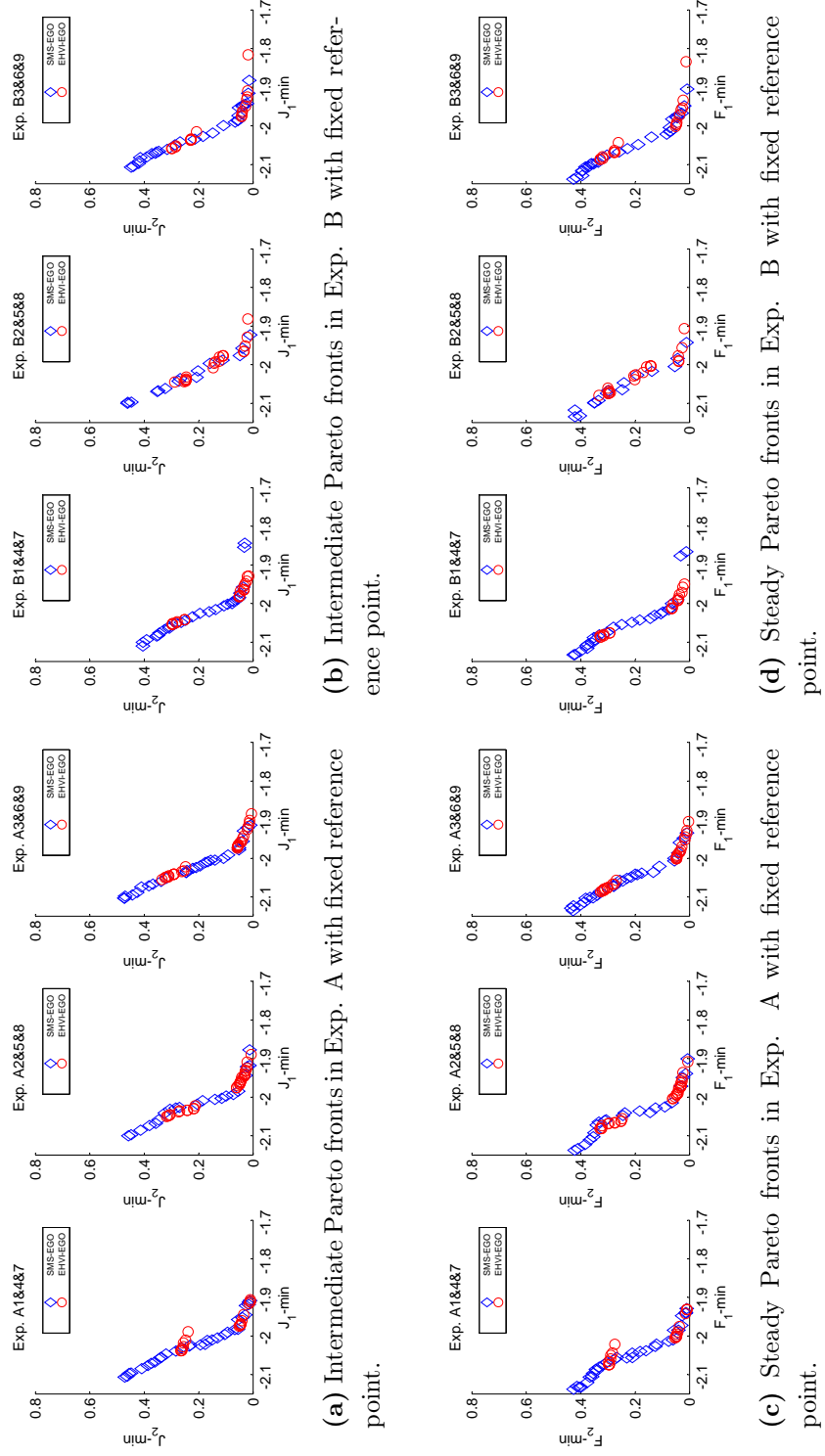
For the fixed reference point strategy, the performance of EHVI-EGO is similar to SMS-EGO. The reason of this could be the bad choice of the reference point. Some actual better Pareto fronts are regarded as the worse ones in process optimization, due to the improper reference point. However, it still needs more experiments to verify this assumption.

Comparing all the intermediate Pareto fronts with a dynamic and fixed reference point, EHVI-EGO outperforms SMS-EGO when using a dynamic reference point, while SMS-EGO is better than EHVI-EGO when using a fixed reference point. This means that SMS-EGO is more robust against the choice of a reference point than EHVI-EGO. Another evidence of this conclusion can be found in Table 7.7: the number of elements in EHVI-EGO intermediate Pareto front is always smaller than that of SMS-EGO, and EHVI-EGO standard deviation is larger than SMS-EGO's.

**Table 7.7:** Number of elements in intermediate Pareto fronts.

Ref.	Exp.	EHVI-EGO				SMS-EGO			
		AVG	MIN	MAX	$\sigma$	AVG	MIN	MAX	$\sigma$
Dynamic	<i>A</i>	14.51	6	23	4.04	20.49	7	27	5.09
Dynamic	<i>C</i>	16.71	5	28	5.86	21.87	11	35	5.34
Fixed	<i>A</i>	19.29	7	33	8.15	22.53	11	31	4.50
Fixed	<i>B</i>	16.04	9	24	3.38	20.73	8	29	6.45

Contrary to PID parameter tuning, the Pareto fronts in a bio-gas plant are more similar. This is because the total function evaluations numbers within the simulation in the bio-gas plant are much larger than the corresponding ones in PID parameter tuning. For the bio-gas plant,  $T_p = 150$ ,  $T_c = 10$  and  $n_{\text{eval}}$  is the function evaluation number in each control period. Therefore, the total function evaluations number in the whole simulation period is 15 times  $n_{\text{eval}}$ .



**Figure 7.10:** Experimental results on Exp. A & B, using fixed reference point strategy.

## 7. APPLICATIONS

---

### 7.5 Summary

In this chapter, we investigated different evolutionary multi-criterion optimization algorithms for the optimization of PID controllers and bio-gas plants optimization. The focus of our study was hypervolume based methods for problems that allow only a small ( $\approx 100$ ) number of exact objective function evaluations. In particular, a state-of-the-art evolutionary algorithm (SMS-EMOA) was compared to multi-objective efficient global optimization algorithms, SMS-EGO and EHVI-EGO, which utilize a surrogate model of the objective functions. In the problem domain of PID controller tuning, the results were, as expected, that the surrogate model assisted strategies yielded better results than the SMS-EMOA. Among them, the EHVI-EGO performs slightly better than the SMS-EGO, but it is more sensitive to the change of the reference point.

However, in the case of the optimization of the bio-gas plant, the result seems to be less consistent. While also in this case, for intermediate control optimization the EHVI-EGO shows the best performance, there is no big gap between all the algorithms performances. Also the SMS-EMOA, which uses no surrogate model, manages to produce a diverse Pareto-front that in terms of convergence is only slightly worse than that of EHVI-EGO. More surprisingly, the gap between the algorithms performances almost diminishes when it comes to applying the optimized control in steady state. The consistency of the results by the three algorithms shows that the solution is probably close to the true Pareto front.

Between EHVI-EGO and TEHVI-EGO, TEHVI-EGO outperforms EHVI-EGO in the case of robust PID tuning problem, as the TEHVI only considers the EHVI in a certain domain, and therefore, it can force the algorithm on exploring in this domain. Summarizing, based on the result of this study we recommend using TEHVI-EGO when the a-priori knowledge of the objective functions is available. Otherwise, we recommend to use EHVI-EGO, because it shows a good performance more consistently as compared to the other algorithms. For further work, it is recommended to research the robust setting for the reference point.

## Chapter 8

# Conclusions and Outlooks

Multi-objective Bayesian global optimization is an effective method when the objective functions cannot be directly measured or evaluation of the objective functions is too expensive. The basic concept behind MOBGO is using Kriging/Gaussian Process to build the independent models based on the relationship between inputs and outputs of a real problem. After this step, an optimization algorithm is applied to find an optimal solution based on the Kriging models and an infill criterion. Then, the optimal solution will be evaluated by the real objective functions to update the Kriging models. This loop will be repeated until it meets the stop criterion.

Compared to evolutionary algorithms, Bayesian global optimization is more efficient when dealing with expensive function evaluation problems. However, there still exists much space to improve the efficiency of MOBGO. The efficiency of MOBGO is mainly determined by three aspects: the computational complexity of updating a Kriging model, the computational complexity of infill criteria, and efficiency of a single objective optimization to find the optimal solution based on the Kriging models. Targeting at reducing the computational complexity of EHVI calculation and improving the performance of MOBGO in this dissertation, an efficient EHVI calculation algorithm, two new infill criterion (TEHVI and EHVIG) were proposed, and a thorough research on them was conducted.

Section 8.1 provides the conclusions drawn from works in this dissertation and Section 8.2 discusses an outlook and directions of the future research.

## 8. CONCLUSIONS AND OUTLOOKS

---

### 8.1 Conclusions

During the process of searching the optimal solution, the infill criterion in MOBGO acts as a pre-selection criterion and the maximizing of the infill criterion is equivalent to achieving an optimal Pareto front set. The infill criterion in MOBGO is very crucial, considering that it merges each objective into a single value. Then, according to the value of this infill criterion, the optimization algorithm can select the optimal solution by choosing the solution which has the maximum/minimum infill criterion value. Among the infill criteria, EHVI performs much better than the other criteria, because it can balance the exploration and exploitation well in the objective space. However, EHVI is rarely applied in real applications for its high computational complexity. An efficient EHVI calculation algorithm is proposed. The computational complexity of EHVI is decreased to  $\Theta(n \log n)$ , for both 2-D and 3-D cases, by an efficient non-dominate space partitioning method and a new EHVI calculation formula. It is also proven that the number of partitioned slices in non-dominated space is always  $n + 1$  and  $2n + 1$  for 2-D and 3-D, respectively. For the high dimensional cases ( $d \geq 4$ ), the proposed algorithm is much faster than the previous methods. Moreover, the proposed algorithm can also be extended to other integral-based criteria calculation, like PoI, TEHVI and EHVIG.

The inherent assumption of EHVI is that each objective function follows a normal distribution and ranges from  $-\infty$  to  $\infty$ . However, the rough range of an objective function is obtained from experts in some real applications, and this a-prior knowledge was not utilized in EHVI. To solve this problem, a new criterion, TEHVI derived from EHVI, was proposed in this dissertation. Compared to EHVI, TEHVI follows a truncated normal distribution within a lower bound and an upper bound for each objective function. The experimental results show that TEHVI outperforms EHVI on benchmarks and PID robust parameter tuning problems.

Moreover, TEHVI can also be used to solve the preference-based multi-objective optimization problems, since the boundary information is related to the objective space and can be set by a decision maker according to his/her preferences. Inspired by TEHVI, THV is proposed and applied to solve preference-based multi-objective optimization problems. Compared to TEHVI, THV can be used to any evolutionary algorithms, while TEHVI can only be utilized in BGO. The experimental results that show both of two approaches are effective.

Based on parts of the works in this dissertation, the computation of EHVI and TEHVI is considered to be efficient, as illustrated in Chapter 3 and Chapter

4. However, the maximization of EHVI still requires significantly amount of time in MOBGO, because EHVI needs to be calculated for many times in each iteration. Since the landscape of a Kriging model is continuous, and considering the definition of EHVI that the integral of hypervolume improvement times its corresponding probability density function, EHVI should be differentiable at a target point, and EHVI is proposed to speed up the convergence of the optimizer in MOBGO by using GAA. However, compared to other optimizers (CMA-ES, GA), GAA is much faster, but it is very easy to be stuck at a local optimal point. To improve the convergence of the optimizer, EHVI is applied as a stopping criterion in EAs, as the gradient of EHVI at an optimal point should be a zero vector and EHVI can force an EA to stop earlier when an EA find the optimal solution. The experimental results show that using EHVI as a stopping criterion needs less execution time and improves the quality of the final Pareto front approximation on the benchmarks.

## 8.2 Outlooks

This dissertation aims to improve the efficiency and effectiveness of MOBGO by reducing the computational complexity of integral-based infill criterion and proposing two new infill criteria. Some open questions related to this dissertation for future research are:

**Improving the efficiency of EHVI calculation for  $d \geq 4$  case:** The partitioning algorithm of EHVI for high dimensional case ( $d \geq 4$ ) is based on two state-of-the-art algorithms, DKLV17 and LKF17. These two algorithms are very efficient to partition the dominated space, but not non-dominated space. Therefore, the research of efficient partitioning non-dominated space is highly recommended.

**Multivariate surrogate models with output dependence assumption:** Currently, Kriging/Gaussian processes build a surrogate model for each objective function and surrogate models are independent. However, in some real applications, the objective functions are dependent. This can lead to inappropriate representations of joint uncertainty. The potential solutions could be achieved by principal component analysis (PCA) and borrowing the idea of decoupling method in control theory.

**Reference point free in infill criteria:** Setting a reference point for hypervolume-based criteria is very tricky. For a minimization problem, a big reference point

## 8. CONCLUSIONS AND OUTLOOKS

---

can more easily lead an algorithm to focus on the extreme points. On the other hand, the optimization algorithm would omit the extreme points if the reference point is small. Recently, a dynamic reference point strategy, which uses a big reference point in the early iteration and decreases the reference point in the later iterations, is applied in some papers. This strategy is useful but will also lead to another question: how to set the dynamic reference point strategy effectively? Actually, the most effective solution to this problem is to remove the concept of a reference point of an infill criterion theoretically, and this will be an interesting research topic in the future.

# Appendix A

## A.1 Symbols

## APPENDIX

**Table A.1:** Notations.

Symbol	Type	Description	Ref.
$m$	$\mathbb{N}^+$	Dimension of a search space	Eq. (2.1)
$d$	$\mathbb{N}^+$	Dimension of an objective space	Eq. (2.1)
$\mathbb{S}$	$\mathbb{R}^m$	Search space	Eq. (2.1)
$\mathcal{X}$	$\subseteq \mathbb{S}$	Feasible set in $\mathbb{S}$	Eq. (2.1)
$\mathbf{x}$	$\in \mathcal{X}$	Decision vector	Eq. (2.1)
$\mathbf{X}$	$\subset \mathcal{X}$	Decision vector set	Eq. (2.2)
$\mathbf{y}_i$	$\mathbb{R}$	The $i$ -th objective function	Eq. (2.1)
$\mathbf{y}$	$\mathbb{R}^d$	The objective functions	Def. (2.1)
$y_i(\mathbf{x})$	$\mathbb{R}$	The $i$ -th objective value of $\mathbf{x}$	Def. (2.1)
$\mathcal{P}^*$	$(\mathbb{R}^d)^n$	Pareto front	Def. (2.3)
$\mathcal{P}$	$(\mathbb{R}^d)^n$	Pareto front approximation	Def. (2.4)
$n$	$\mathbb{N}^+$	Number of the points in $\mathcal{P}$ or $\mathcal{P}^*$	Def. (2.5)
$\mathbf{r}$	$\mathbb{R}^d$	Reference point	Def. (2.5)
$y_i$	$\mathbb{N}^+$	The $i$ -th objective space	Fig. 2.2
$\mu$	$\mathbb{N}^+$	Population size	Alg. 1
$p_m$	$0 < p_m < 1$	Mutation rate	Alg. 1
$p_c$	$0 < p_c < 1$	Crossover rate	Alg. 1
$R_i$	$\mathbb{R}^+$	Branch resistance	Def. 4-8
$f_{loss}$	$\mathbb{R}^+$	Active power loss	Def. 4-8
$P_i$	$\mathbb{R}^+$	Active power	Def. 4-8
$Q_i$	$\mathbb{R}^+$	Inactive power	Def. 4-8
$V_i$	$\mathbb{R}^+$	Node voltage	Def. 4-8
$I_i$	$\mathbb{R}^+$	Branch current	Def. 4-8
$f_{VDI}$	$\mathbb{R}^+$	Voltage deviation	Def. 4-11
$\boldsymbol{\mu}$	$\mathbb{R}^d$	Mean values of predictive distribution	Alg. 3
$\boldsymbol{\sigma}$	$(\mathbb{R}_0^+)^d$	Standard deviations of predictive distribution	Alg. 3
$\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n)}$	$\mathbb{R}^d$	The vectors in $\mathcal{P}$ , where $\mathcal{P} = (\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n)})$	Fig. 2.2
$S_d$	$(\mathbb{R}^d)^2$	Integration slices for $d$ dimension	Def. 3.3.1
$\mathbf{l}_d^{(1)}, \dots, \mathbf{l}_d^{(N_d)}$	$\mathbb{R}^d$	Lower bound of integration boxes	Def. 3.3.1
$\mathbf{u}_d^{(1)}, \dots, \mathbf{u}_d^{(N_d)}$	$\mathbb{R}^d$	Upper bound of integration boxes	Def. 3.3.1
$N_d$	$\mathbb{N}^+$	Number of integration boxes	Def. 3.3.1
$\mathbf{x}^*$	$\mathbb{R}^d$	An optimal point in search space	Alg. 3
$D$	$\mathbb{R}^{(m+d)}$	Training data set	Alg. 3

## A.2 Abbreviations

**Table A.2:** Abbreviations-I.

Abb.	Full Name
AVL	Adelson-Velskii and Landis
BGO	Bayesian Global Optimization
CDF	Cumulative Density Function
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
DKLV	Dächert, Klamroth, Lacour and Vanderpooten
LKF	Lacour, Klamroth and Fonseca
DF(s)	Desirability Function(s)
DM	Decision Maker
DNRP	Distribution Network Reconfiguration Problem
EA(s)	Evolutionary Algorithm(s)
EGO	Efficient Global Optimization
EHVI	Expected Hypervolume Improvement
EHVIG	Expected Hypervolume Improvement Gradient
EI	Expected Improvement
EMOA(s)	Evolutionary Multi-objective Optimization Algorithm(s)
GA	Genetic Algorithm
GAA	Gradient Ascent Algorithm
GSP	Generalized Schaffer Problem
HV	Hypervolume Indicator
HVI	Hypervolume Improvement
HVC	Hypervolume Contribution
LCB	Lower Confidence Bound
MLI	Most Likely Improvement
MONMPC	Multi-Objective Nonlinear Model Predictive Control
MOO	Multi-objective Optimization
MOPSO	Multi-Objective Particle Swarm Optimization
NOEs	Number of Evaluations
NSGA-II	Non-dominated Sorting Genetic Algorithm II

**Table A.3:** Abbreviations-II.

Abb.	Full Name
OK	Ordinary Kriging
PCA	Principal Component Analysis
PDF	Probability Density function
PICEA-g	Preference-Inspired Co-Evolutionary Algorithm
PID	Proportional Integral Derivative
PMX	Partial-Mapped Crossover
PO	Percentage Overshoot
PoI	Probability of improvement
PR	Preferred Region
REN21	Renewable Energy Policy Network for the 21st Century
ROI	Region of Interest
SMS-EMOA	<i>S</i> -metric Selection Evolutionary Multi-Optimization Algorithm
SMS-EGO	<i>S</i> -metric Selection Efficient Global Optimization
TCDF	Truncated Cumulative Density Function
TEHVI	Truncated Expected Hypervolume Improvement
THV	Truncated Hypervolume
TPDF	Truncated Probability Density Function
$\lambda$	Lebesgue Measure

### A.3 EHVIG Formula Derivation

$$1. \phi'(x) = -x\phi(x) \quad (\text{A-1})$$

$$2. \Phi'(x) = \phi(x) \quad (\text{A-2})$$

$$3. \frac{\partial \Phi(\frac{y-\mu}{\sigma})}{\partial \mathbf{x}} = \phi(\frac{y-\mu}{\sigma}) \cdot (\frac{\mu-y}{\sigma^2} \cdot \frac{\partial \sigma}{\partial \mathbf{x}} - \frac{1}{\sigma} \cdot \frac{\partial \mu}{\partial \mathbf{x}}) \quad (\text{A-3})$$

Using the chain rule and quotient rule, considering that  $y$  does not depend on  $\mathbf{x}$ , we get the statement in (A-3):

$$\frac{\partial \Phi(\frac{y-\mu}{\sigma})}{\partial \mathbf{x}} = \phi(\frac{y-\mu}{\sigma}) \cdot \frac{\partial (\frac{y-\mu}{\sigma})}{\partial \mathbf{x}} = \phi(\frac{y-\mu}{\sigma}) \cdot \frac{(\frac{\partial y}{\partial \mathbf{x}} - \frac{\partial \mu}{\partial \mathbf{x}})\sigma - (y-\mu)\frac{\partial \sigma}{\partial \mathbf{x}}}{\sigma^2}$$

After tidying up, we get a statement in (A-3):

$$\frac{\partial \Phi(\frac{y-\mu}{\sigma})}{\partial \mathbf{x}} = \phi(\frac{y-\mu}{\sigma}) \cdot (\frac{\mu-y}{\sigma^2} \cdot \frac{\partial \sigma}{\partial \mathbf{x}} - \frac{1}{\sigma} \cdot \frac{\partial \mu}{\partial \mathbf{x}})$$

$$4. \frac{\partial \Psi(a, b, \mu, \sigma)}{\partial \mathbf{x}} = \left( \frac{b-a}{\sigma} \cdot \phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{b-\mu}{\sigma}) \right) \cdot \frac{\partial \mu}{\partial \mathbf{x}} + \phi(\frac{b-\mu}{\sigma}) \cdot \left( 1 + \frac{(b-\mu)(b-a)}{\sigma^2} \right) \cdot \frac{\partial \sigma}{\partial \mathbf{x}}$$

Using the product rule and considering  $a$  and  $b$  do not depend on  $\mathbf{x}$ , we get the statement:

$$\frac{\partial \Psi(a, b, \mu, \sigma)}{\partial \mathbf{x}} = \frac{\partial \Psi(a, b, \mu, \sigma)}{\partial \mu} \cdot \frac{\partial \mu}{\partial \mathbf{x}} + \frac{\partial \Psi(a, b, \mu, \sigma)}{\partial \sigma} \cdot \frac{\partial \sigma}{\partial \mathbf{x}} \quad (\text{A-4})$$

Substituting Equation (A-7) into  $\frac{\partial \Psi(a, b, \mu, \sigma)}{\partial \mu}$  and  $\frac{\partial \Psi(a, b, \mu, \sigma)}{\partial \sigma}$ , using the chain rule, quotient rule, and product rule, the statements of  $\frac{\partial \Psi(a, b, \mu, \sigma)}{\partial \mu}$  and  $\frac{\partial \Psi(a, b, \mu, \sigma)}{\partial \sigma}$  are:

## APPENDIX

---

$$\begin{aligned}
\frac{\partial \Psi(a, b, \mu, \sigma)}{\partial \mu} &= \frac{\partial [\sigma \cdot \phi(\frac{b-\mu}{\sigma}) + (a - \mu) \cdot \Phi(\frac{b-\mu}{\sigma})]}{\partial \mu} \\
&= \sigma \cdot \frac{\partial \phi(\frac{b-\mu}{\sigma})}{\partial \mu} + (-1) \cdot \Phi(\frac{b-\mu}{\sigma}) + (a - \mu) \cdot \frac{\partial \Phi(\frac{b-\mu}{\sigma})}{\partial \mu} \\
&= \frac{b-\mu}{\sigma} \cdot \phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{b-\mu}{\sigma}) + [-\frac{a-\mu}{\sigma} \cdot \phi(\frac{b-\mu}{\sigma})] \\
&= \frac{b-a}{\sigma} \cdot \phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{b-\mu}{\sigma}) \tag{A-5}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \Psi(a, b, \mu, \sigma)}{\partial \sigma} &= \frac{\partial [\sigma \cdot \phi(\frac{b-\mu}{\sigma}) + (a - \mu) \cdot \Phi(\frac{b-\mu}{\sigma})]}{\partial \sigma} \\
&= \phi(\frac{b-\mu}{\sigma}) + \sigma \cdot \frac{\partial \phi(\frac{b-\mu}{\sigma})}{\partial \sigma} + (a - \mu) \cdot \frac{\partial \Phi(\frac{b-\mu}{\sigma})}{\partial \sigma} \\
&= \phi(\frac{b-\mu}{\sigma}) + (\frac{b-\mu}{\sigma})^2 \cdot \phi(\frac{b-\mu}{\sigma}) + (-\frac{(a-\mu) \cdot (b-\mu)}{\sigma^2} \cdot \phi(\frac{b-\mu}{\sigma})) \\
&= \phi(\frac{b-\mu}{\sigma}) + \frac{(b-\mu) \cdot (b-a)}{\sigma^2} \cdot \phi(\frac{b-\mu}{\sigma}) \\
&= \phi(\frac{b-\mu}{\sigma}) (1 + \frac{(b-\mu) \cdot (b-a)}{\sigma^2}) \tag{A-6}
\end{aligned}$$

After substituting Equations (A-5) and (A-6) into (A-4), then we get statement in (A-4):

$$\begin{aligned}
\frac{\partial \Psi(a, b, \mu, \sigma)}{\partial \mathbf{x}} &= \left( \frac{b-a}{\sigma} \cdot \phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{b-\mu}{\sigma}) \right) \cdot \frac{\partial \mu}{\partial \mathbf{x}} + \\
&\quad \phi(\frac{b-\mu}{\sigma}) \cdot \left( 1 + \frac{(b-\mu)(b-a)}{\sigma^2} \right) \cdot \frac{\partial \sigma}{\partial \mathbf{x}}
\end{aligned}$$

## A.4 2-D EHVI Formula (Minimization Case)

**Definition A.1 ( $\Psi$  function)** Let  $\phi(s) = 1/\sqrt{2\pi}e^{-\frac{1}{2}s^2}$ ,  $s \in \mathbb{R}$  denote the PDF of the standard normal distribution and  $\Phi(s) = \frac{1}{2} \left(1 + \operatorname{erf}\left(\frac{s}{\sqrt{2}}\right)\right)$  denote its cumulative probability distribution function (CDF). The general normal distribution with mean  $\mu$  and variance  $\sigma$  has the PDF  $\phi_{\mu,\sigma}(s) = \frac{1}{\sigma}\phi\left(\frac{s-\mu}{\sigma}\right)$  and the CDF  $\Phi_{\mu,\sigma}(s) = \Phi\left(\frac{s-\mu}{\sigma}\right)$ . Then the function  $\Psi$  is defined as:

$$\begin{aligned}\Psi(a, b, \mu, \sigma) &= \int_{-\infty}^b (a - z) \frac{1}{\sigma} \phi\left(\frac{z - \mu}{\sigma}\right) dz \\ &= \sigma \phi\left(\frac{b - \mu}{\sigma}\right) + (a - \mu) \Phi\left(\frac{b - \mu}{\sigma}\right).\end{aligned}\tag{A-7}$$

It partitions the integration domain into  $n + 1$  disjoint rectangular stripes  $S_1, \dots, S_{n+1}$ , see Figure A.1 for an illustration. For this, we augment the set  $P$  by two points  $\mathbf{y}^{(0)} = (r_1, -\infty)$  and  $\mathbf{y}^{(n+1)} = (-\infty, r_2)$ . The stripes are now defined by:

$$S_i = \left( \begin{pmatrix} y_1^{(i)} \\ -\infty \end{pmatrix}, \begin{pmatrix} y_1^{(i-1)} \\ y_2^{(i)} \end{pmatrix} \right), \quad i = 1, \dots, n + 1.\tag{A-8}$$

Suppose  $\mathbf{Y}$  are the sorted non-dominated vectors of the current Pareto front approximation  $\mathcal{P}$ . A formula will be derived that consists of  $n + 1$  integrals, as indicated in Figure A.1. The HVI of a point  $\mathbf{y} \in \mathbb{R}^2$  can be expressed by:

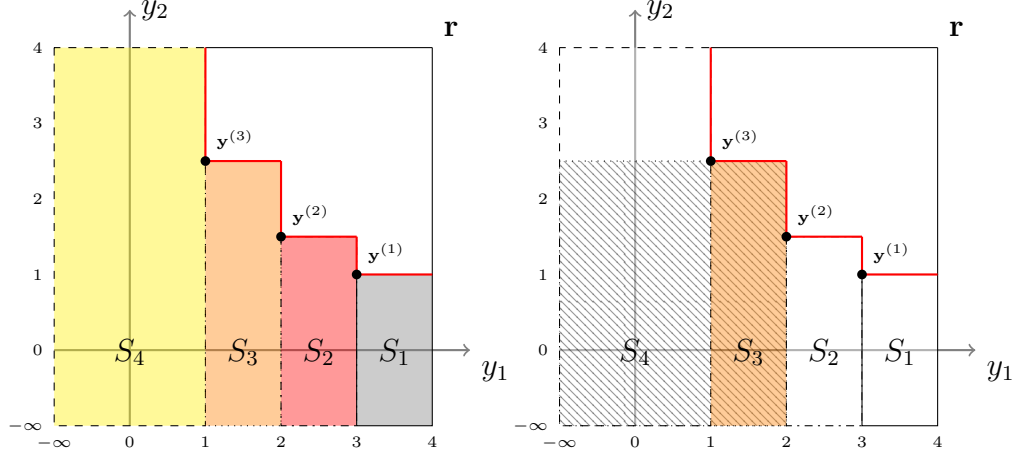
$$\text{HVI}(\mathbf{y}, \mathbf{Y}, \mathbf{r}) = \sum_{i=1}^{n+1} \lambda_2[S_i \cap \Delta(\mathbf{y})].\tag{A-9}$$

This gives rise to the compact integral for the original EHVI,  $\mathbf{y} = (y_1, y_2)$ :

$$\begin{aligned}\text{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{Y}, \mathbf{r}) &= \\ \int_{y_1=-\infty}^{\infty} \int_{y_2=-\infty}^{\infty} \sum_{i=1}^{n+1} \lambda_2[S_i \cap \Delta((y_1, y_2))] \cdot \text{PDF}_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y}\end{aligned}\tag{A-10}$$

It is observed that the intersection of  $S_i$  with  $\Delta((y_1, y_2))$  is non-empty if and only if  $\mathbf{y} = (y_1, y_2)$  dominates the upper right corner of  $S_i$ . In other words, if and only

## APPENDIX



**Figure A.1:** Partitioning of the integration region into stripes. Right: New partitioning of the reduced integration region after first iteration of the algorithm.

if  $\mathbf{y}$  is located in the rectangle with lower left corner  $(y_1^{(i)}, -\infty)$  and upper right corner  $(y_1^{(i-1)}, y_2^{(i)})$ . Therefore:

$$\begin{aligned} & \text{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{Y}, \mathbf{r}) \\ &= \sum_{i=1}^{n+1} \int_{y_1=-\infty}^{y_1^{(i-1)}} \int_{y_2=-\infty}^{y_2^{(i)}} \lambda_2[S_i \cap \Delta((y_1, y_2))] \cdot \text{PDF}_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \end{aligned} \quad (\text{A-11})$$

$$\begin{aligned} &= \sum_{i=1}^{n+1} \int_{y_1=-\infty}^{y_1^{(i)}} \int_{y_2=-\infty}^{y_2^{(i)}} \lambda_2[S_i \cap \Delta((y_1, y_2))] \cdot \text{PDF}_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} + \\ & \sum_{i=1}^{n+1} \int_{y_1=y_1^{(i)}}^{y_1^{(i-1)}} \int_{y_2=-\infty}^{y_2^{(i)}} \lambda_2[S_i \cap \Delta((y_1, y_2))] \cdot \text{PDF}_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \\ &= \sum_{i=1}^{n+1} (y_1^{(i-1)} - y_1^{(i)}) \cdot \Phi\left(\frac{y_1^{(i)} - \mu_1}{\sigma_1}\right) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) + \\ & \sum_{i=1}^{n+1} \left( \Psi(y_1^{(i-1)}, y_1^{(i-1)}, \mu_1, \sigma_1) - \Psi(y_1^{(i-1)}, y_1^{(i)}, \mu_1, \sigma_1) \right) \times \\ & \quad \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) \end{aligned} \quad (\text{A-12})$$

Since integration is a linear mapping, it is allowed to do the summation after integration in Eq. (A-11). The integrals are now over a rectangular region and can be solved using the function  $\Psi$  as detailed in [2].

# Bibliography

- [1] Hupkens, I., Deutz, A., Yang, K., Emmerich, M.: Faster exact algorithms for computing expected hypervolume improvement. In Gaspar-Cunha, A., Henggeler Antunes, C., Coello, C.C., eds.: International Conference on Evolutionary Multi-Criterion Optimization, Cham, Springer (2015) 65–79
- [2] Emmerich, M., Yang, K., Deutz, A., Wang, H., Fonseca, C.M.: A multi-criteria generalization of bayesian global optimization. In Pardalos, P.M., Zhigljavsky, A., Žilinskas, J., eds.: Advances in Stochastic and Deterministic Global Optimization. Springer, Berlin, Heidelberg (2016) 229–243
- [3] Yang, K., Emmerich, M.T., Li, R., Wang, J., Bäck, T.: Power distribution network reconfiguration by evolutionary integer programming. In Bartz-Beielstein, T., Branke, J., Filipič, B., Smith, J., eds.: International Conference on Parallel Problem Solving from Nature–PPSN XIII, Cham, Springer (2014) 11–23
- [4] Yang, K., Emmerich, M., Deutz, A., Fonseca, C.M.: Computing 3-D expected hypervolume improvement and related integrals in asymptotically optimal time. In Trautmann, H., Rudolph, G., Klamroth, K., Schütze, O., Wiecek, Margaret Jin, Y., Grimme, C., eds.: International Conference on Evolutionary Multi-Criterion Optimization, Cham, Springer (2017) 685–700
- [5] Yang, K., Deutz, A., Fonseca, C.M., Bäck, T., Emmerich, M.: Efficient exact computation of expected hypervolume improvement in Bayesian global optimization. *Journal of Global Optimization* (Submitted)
- [6] Yang, K., Deutz, A., Yang, Z., Back, T., Emmerich, M.: Truncated expected hypervolume improvement: Exact computation and application. In: 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE (2016) 4350–4357

## BIBLIOGRAPHY

---

- [7] Yang, K., Li, L., Deutz, A., Bäck, T., Emmerich, M.: Preference-based multiobjective optimization using truncated expected hypervolume improvement. In: 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), IEEE (2016) 276–281
- [8] Wang, Y., Li, L., Yang, K., Emmerich, M.: A new approach to target region based multiobjective evolutionary algorithms. In: 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE (2017) 1757–1764
- [9] Yang, K., Emmerich, M., Bäck, T., Deutz, A.: Multi-objective Bayesian global optimization using expected hypervolume improvement gradient. Swarm and Evolutionary Computation (Submitted)
- [10] Yang, K., Gaida, D., Bäck, T., Emmerich, M.: Expected hypervolume improvement algorithm for PID controller tuning and the multiobjective dynamical control of a biogas plant. In: 2015 IEEE Congress on Evolutionary Computation (CEC). (2015) 1934–1942
- [11] Yang, Z., Wang, H., Yang, K., Back, T., Emmerich, M.: SMS-EMOA with multiple dynamic reference points. In: 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), IEEE (2016) 282–288
- [12] Marler, R.T., Arora, J.S.: The weighted sum method for multi-objective optimization: new insights. Structural and Multidisciplinary Optimization **41** (2010) 853–862
- [13] Deb, K., Sindhya, K., Hakanen, J.: Multi-objective optimization. In Sengupta, R.N., Dutta, J., Gupta, A., eds.: Decision Sciences: Theory and Practice. CRC Press (2016) 145–184
- [14] Coello Coello, C.A.: Evolutionary multi-objective optimization: Basic concepts and some applications in pattern recognition. In Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Ben-Youssef Brants, C., Hancock, E.R., eds.: Proceedings of the Third Mexican conference on Pattern recognition, Berlin, Heidelberg, Springer (2011) 22–33
- [15] Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation **3** (1999) 257–271
- [16] Beume, N., Fonseca, C.M., López-Ibáñez, M., Paquete, L., Vahrenhold, J.: On the complexity of computing the hypervolume indicator. IEEE Transactions on Evolutionary Computation **13** (2009) 1075–1082

- [17] Chan, T.M.: Klee's measure problem made easy. In: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS), IEEE (2013) 410–419
- [18] Emmerich, M., Deutz, A., Klinkenberg, J.W.: The computation of the expected improvement in dominated hypervolume of Pareto front approximations. Technical Report, Leiden University **34** (2008)
- [19] Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* **181** (2007) 1653–1669
- [20] Emmerich, M.T.M., Fonseca, C.M.: Computing hypervolume contributions in low dimensions: Asymptotically optimal algorithm and complexity results. In Takahashi, R.H.C., Deb, K., Wanner, E.F., Greco, S., eds.: *International Conference on Evolutionary Multi-Criterion Optimization*, Berlin, Heidelberg, Springer (2011) 121–135
- [21] Bäck, T., Fogel, D.B., Michalewicz, Z.: *Handbook of Evolutionary Computation*. 1st edn. IOP Publishing Ltd., Bristol, UK, UK (1997)
- [22] Bäck, T.: *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, Oxford, UK (1996)
- [23] Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* **2** (1994) 221–248
- [24] Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J.J., Schwefel, H.P., eds.: *International Conference on Parallel Problem Solving from Nature-PPSN VI*, Berlin, Heidelberg, Springer (2000) 849–858
- [25] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6** (2002) 182–197
- [26] Emmerich, M., Deutz, A., Li, R., Kruisselbrink, J.: Getting lost or getting trapped: On the effect of moves to incomparable points in multiobjective hill climbing. In: *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation (GECCO)*, New York, USA, ACM (2010) 1963–1966

## BIBLIOGRAPHY

---

- [27] Emmerich, M., Beume, N., Naujoks, B.: An EMO algorithm using the hypervolume measure as selection criterion. In Coello, C.A.C., Aguirre, A.H., Zitzler, E., eds.: International Conference on Evolutionary Multi-Criterion Optimization, Berlin, Heidelberg, Springer (2005) 62–76
- [28] Zhang, D., Fu, Z., Zhang, L.: An improved ts algorithm for loss-minimum reconfiguration in large-scale distribution systems. *Electric Power Systems Research* **77** (2007) 685–694
- [29] Qin, Y., Wang, J.: Distribution network reconfiguration based on particle clonal genetic algorithm. *Journal of Computers* **4** (2009) 813–820
- [30] Chiou, J.P., Chang, C.F., Su, C.T.: Variable scaling hybrid differential evolution for solving network reconfiguration of distribution systems. *IEEE Transactions on Power Systems* **20** (2005) 668–674
- [31] Niknam, T.: A new hybrid algorithm for multi-objective distribution feeder reconfiguration. *Cybernetics and Systems* **40** (2009) 508–527
- [32] Zimmerman, R.D., Murillo-Sánchez, C.E., Thomas, R.J.: MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems* **26** (2011) 12–19
- [33] Knowles, J., Corne, D.: Properties of an adaptive archiving algorithm for storing nondominated vectors. *IEEE Transactions on Evolutionary Computation* **7** (2003) 100–116
- [34] Fonseca, C.M., Fleming, P.J.: On the performance assessment and comparison of stochastic multiobjective optimizers. In Voigt, H.M., Ebeling, W., Rechenberg, I., Schwefel, H.P., eds.: International Conference on Parallel Problem Solving from Nature IV, Berlin, Heidelberg, Springer (1996) 584–593
- [35] Grunert da Fonseca, V., Fonseca, C., Hall, A.: Inferential performance assessment of stochastic optimisers and the attainment function. In Zitzler, E., Thiele, L., Deb, K., Coello Coello, C.A., Corne, D., eds.: International Conference on Evolutionary Multi-Criterion Optimization, Berlin, Heidelberg, Springer (2001) 213–225
- [36] Knowles, J.: A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. In: 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), IEEE (2005) 552–557

- [37] Mockus, J., Tiešis, V., Žilinskas, A.: The application of Bayesian methods for seeking the extremum. In Dixon, L., Szegö, G., eds.: *Towards Global Optimization*. Volume 2. North-Holland, Amsterdam (1978) 117–131
- [38] Li, R., Emmerich, M.T.M., Eggermont, J., Bovenkamp, E.G.P., Back, T., Dijkstra, J., Reiber, J.H.C.: Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis. In: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. (2008) 2764–2771
- [39] Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. *Statistical Science* (1989) 409–423
- [40] Chugh, T.: Handling expensive multiobjective optimization problems with evolutionary algorithms. PhD thesis, Faculty of Information Technology, University of Jyväskylä (2017)
- [41] Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. *Journal of Global optimization* **13** (1998) 455–492
- [42] Łaniewski-Wołk, Ł., Obayashi, S., Jeong, S.: Development of expected improvement for multi-objective problems. In: *Proceedings of 42nd Fluid Dynamics Conference/Aerospace Numerical, Simulation Symposium* (CD ROM), Varna, Bulgaria (2010)
- [43] Gaida, D.: Dynamic real-time substrate feed optimization of anaerobic co-digestion plants. PhD thesis, Leiden Institute of Advanced Computer Science (LIACS), Faculty of Science, Leiden University (2014)
- [44] Zaefferer, M., Bartz-Beielstein, T., Naujoks, B., Wagner, T., Emmerich, M.: A case study on multi-criteria optimization of an event detection software under limited budgets. In Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J., eds.: *International Conference on Evolutionary Multi-Criterion Optimization*, Berlin, Heidelberg, Springer (2013) 756–770
- [45] Shimoyama, K., Sato, K., Jeong, S., Obayashi, S.: Updating kriging surrogate models based on the hypervolume indicator in multi-objective optimization. *Journal of Mechanical Design* **135** (2013) 094503
- [46] Hansen, N.: Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In: *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers (GECCO)*, New York, USA, ACM (2009) 2389–2396

## BIBLIOGRAPHY

---

- [47] Wagner, T., Emmerich, M., Deutz, A., Ponweiser, W.: On expected-improvement criteria for model-based multi-objective optimization. In Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G., eds.: International Conference on Parallel Problem Solving from Nature–PPSN XI. Springer, Berlin, Heidelberg (2010) 718–727
- [48] Kushner, H.J.: A new method of locating the maximum point of an arbitrary multi-peak curve in the presence of noise. *Journal of Basic Engineering* **86** (1964) 97–106
- [49] Ulmer, H., Streichert, F., Zell, A.: Evolution strategies assisted by gaussian processes with improved preselection criterion. In: 2003 IEEE Congress on Evolutionary Computation (CEC). Volume 1., IEEE (2003) 692–699
- [50] Ponweiser, W., Wagner, T., Biermann, D., Vincze, M.: Multiobjective optimization on a limited budget of evaluations using model-assisted  $\mathcal{S}$ -metric selection. In Rudolph, G., Jansen, T., Beume, N., Lucas, S., Poloni, C., eds.: International Conference on Parallel Problem Solving from Nature–PPSN X, Berlin, Heidelberg, Springer (2008) 784–794
- [51] Dennis, J.E., Torczon, V.: Managing approximation models in optimization. In Alexandrov, N., Hussaini, M.Y., eds.: Multidisciplinary design optimization: State-of-the-art, Philadelphia, PA: SIAM (1997) 330–347
- [52] Fleischer, M.: The measure of Pareto optima applications to multi-objective metaheuristics. In Fonseca, C.M., Fleming, P.J., Zitzler, E., Thiele, L., Deb, K., eds.: International Conference on Evolutionary Multi-Criterion Optimization, Berlin, Heidelberg, Springer (2003) 519–533
- [53] Emmerich, M., Giannakoglou, K.C., Naujoks, B.: Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation* **10** (2006) 421–439
- [54] Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms a comparative case study. In Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.P., eds.: International Conference on Parallel Problem Solving from Nature–PPSN V, Berlin, Heidelberg, Springer (1998) 292–301
- [55] Keane, A.J.: Statistical improvement criteria for use in multiobjective design optimization. *American Institute of Aeronautics and Astronautics (AIAA) Journal* **44** (2006) 879–891
- [56] Emmerich, M.: Single-and multi-objective evolutionary design optimization

- assisted by Gaussian random field metamodels. PhD thesis, Fachbereich Informatik, Chair of Systems Analysis, University of Dortmund (2005)
- [57] Zhang, Q., Liu, W., Tsang, E., Virginas, B.: Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation* **14** (2010) 456–474
- [58] Couckuyt, I., Deschrijver, D., Dhaene, T.: Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization. *Journal of Global Optimization* **60** (2014) 575–594
- [59] Namura, N., Shimoyama, K., Obayashi, S.: Expected improvement of penalty-based boundary intersection for expensive multiobjective optimization. *IEEE Transactions on Evolutionary Computation* **PP** (2017) 1–1
- [60] Emmerich, M., Deutz, A.H., Klinkenberg, J.W.: Hypervolume-based expected improvement: Monotonicity properties and exact computation. In: 2011 IEEE Congress on Evolutionary Computation (CEC), IEEE (2011) 2147–2154
- [61] Shimoyama, K., Sato, K., Jeong, S., Obayashi, S.: Comparison of the criteria for updating kriging response surface models in multi-objective optimization. In: 2012 IEEE Congress on Evolutionary Computation (CEC), IEEE (2012) 1–8
- [62] Vazquez, E., Bect, J.: Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and Inference* **140** (2010) 3088–3095
- [63] Knowles, J., Hughes, E.J.: Multiobjective optimization on a budget of 250 evaluations. In Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E., eds.: *International Conference on Evolutionary Multi-Criterion Optimization*, Berlin, Heidelberg, Springer (2005) 176–190
- [64] Svenson, J., Santner, T.: Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models. *Computational Statistics & Data Analysis* **94** (2016) 250–264
- [65] Shir, O.M., Emmerich, M., Bäck, T., Vrakking, M.J.: The application of evolutionary multi-criteria optimization to dynamic molecular alignment. In: 2007 IEEE Congress on Evolutionary Computation, IEEE (2007) 4108–4115
- [66] Koch, P., Wagner, T., Emmerich, M.T., Bäck, T., Konen, W.: Efficient

## BIBLIOGRAPHY

---

- multi-criteria optimization on noisy machine learning problems. *Applied Soft Computing* **29** (2015) 357–370
- [67] Luo, C., Shimoyama, K., Obayashi, S.: Kriging model based many-objective optimization with efficient calculation of expected hypervolume improvement. In: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE (2014) 1187–1194
- [68] Rahat, A.A.M., Everson, R.M., Fieldsend, J.E.: Alternative infill strategies for expensive multi-objective optimisation. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), New York, USA, ACM (2017) 873–880
- [69] Shimoyama, K., Jeong, S., Obayashi, S.: Kriging-surrogate-based optimization considering expected hypervolume improvement in non-constrained many-objective test problems. In: 2013 IEEE Congress on Evolutionary Computation (CEC), IEEE (2013) 658–665
- [70] Dächert, K., Klamroth, K., Lacour, R., Vanderpooten, D.: Efficient computation of the search region in multi-objective optimization. *European Journal of Operational Research* **260** (2017) 841–855
- [71] Lacour, R., Klamroth, K., Fonseca, C.M.: A box decomposition algorithm to compute the hypervolume indicator. *Computers & Operations Research* **79** (2017) 347–360
- [72] Stuckman, B.E.: A global search method for optimizing nonlinear systems. *IEEE Transactions on Systems, Man, and Cybernetics* **18** (1988) 965–977
- [73] Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation* **18** (2014) 577–601
- [74] Hadka, D.: Platypus - Multiobjective optimization in Python. (2015) <https://github.com/Project-Platypus/Platypus>
- [75] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6** (2002) 182–197
- [76] Robert, C.P.: Simulation of truncated normal variables. *Statistics and computing* **5** (1995) 121–125

- [77] Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57** (1970) 97–109
- [78] Binh, T.T., Korn, U.: An evolution strategy for the multiobjective optimization. In: *The Second International Conference on Genetic Algorithms (Mendel 96)*, Czech Republic, Brno (1996) 23–28
- [79] Shim, M.B., Suh, M.W., Furukawa, T., Yagawa, G., Yoshimura, S.: Pareto-based continuous evolutionary algorithms for multiobjective optimization. *Engineering Computations* **19** (2002) 22–48
- [80] Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation* **8** (2000) 173–195
- [81] Emmerich, M.T., Deutz, A.H.: Test problems based on Lamé superspheres. In Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T., eds.: *International Conference on Evolutionary Multi-Criterion Optimization*, Berlin, Heidelberg, Springer (2007) 922–936
- [82] Branke, J.: Consideration of partial user preferences in evolutionary multiobjective optimization. In Branke, J., Deb, K., Miettinen, K., Słowiński, R., eds.: *Multiobjective Optimization: Interactive and Evolutionary Approaches*, Berlin, Heidelberg, Springer (2008) 157–178
- [83] Purshouse, R.C., Deb, K., Mansor, M.M., Mostaghim, S., Wang, R.: A review of hybrid evolutionary multiple criteria decision making methods. In: *2014 IEEE Congress on Evolutionary Computation (CEC)*, IEEE (2014) 1147–1154
- [84] Li, L., Yevseyeva, I., Basto-Fernandes, V., Trautmann, H., Jing, N., Emmerich, M.: Building and using an ontology of preference-based multiobjective evolutionary algorithms. In Trautmann, H., Rudolph, G., Klamroth, K., Schütze, O., Wiecek, M., Jin, Y., Grimme, C., eds.: *International Conference on Evolutionary Multi-Criterion Optimization*, Springer (2017) 406–421
- [85] Bechikh, S., Kessentini, M., Said, L.B., Ghédira, K.: Chapter four-preference incorporation in evolutionary multiobjective optimization: A survey of the state-of-the-art. *Advances in Computers* **98** (2015) 141–207
- [86] Fowler, J.W., Gel, E.S., Köksalan, M.M., Korhonen, P., Marquis, J.L., Wallenius, J.: Interactive evolutionary multi-objective optimization for quasi-concave preference functions. *European Journal of Operational Research* **206** (2010) 417 – 425

## BIBLIOGRAPHY

---

- [87] Castro-Gutiérrez, J., Landa-Silva, D., Moreno-Pérez, J.: Dynamic lexicographic approach for heuristic multi-objective optimization. In: Proceedings of the Workshop on Intelligent Metaheuristics for Logistic Planning (CAEPIA-TTIA), Spain, Springer (2009) 153–163
- [88] Friedrich, T., Kroeger, T., Neumann, F.: Weighted preferences in evolutionary multi-objective optimization. In Wang, D., Reynolds, M., eds.: AI 2011: Advances in Artificial Intelligence. Springer, Berlin, Heidelberg (2011) 291–300
- [89] Trautmann, H., Mehnen, J.: Preference-based pareto optimization in certain and noisy environments. *Engineering Optimization* **41** (2009) 23–38
- [90] Mostaghim, S., Trautmann, H., Mersmann, O.: Preference-based multi-objective particle swarm optimization using desirabilities. In Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G., eds.: International Conference on Parallel Problem Solving from Nature–PPSN XI. Springer (2010) 101–110
- [91] Wagner, T., Trautmann, H.: Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions. *IEEE Transactions on Evolutionary Computation* **14** (2010) 688–701
- [92] Karahan, İ., Köksalan, M.: A territory defining multiobjective evolutionary algorithms and preference incorporation. *IEEE Transactions on Evolutionary Computation* **14** (2010) 636–664
- [93] Köksalan, M., Karahan, I.: An interactive territory defining evolutionary algorithm: iTDEA. *IEEE Transactions on Evolutionary Computation* **14** (2010) 702–722
- [94] Wang, R., Purshouse, R.C., Fleming, P.J.: "Whatever works best for you"—A new method for a priori and progressive multi-objective optimisation. In Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J., eds.: International Conference on Evolutionary Multi-Criterion Optimization, Berlin, Heidelberg, Springer (2013) 337–351
- [95] Brockhoff, D., Bader, J., Thiele, L., Zitzler, E.: Directed Multiobjective Optimization Based on the Weighted Hypervolume Indicator. *Journal of Multi-Criteria Decision Analysis* **20** (2013) 291–317
- [96] Narukawa, K., Setoguchi, Y., Tanigaki, Y., Olhofer, M., Sendhoff, B., Ishibuchi, H.: Preference representation using gaussian functions on a hyperplane in evolutionary multi-objective optimization. *Soft Computing* (2015) 1–25

- [97] Harrington, E.: The desirability function. *Industrial Quality Control* **21** (1965) 494–498
- [98] Suich, R., Derringer, G.C.: Is the regression equation adequate?: One criterion. *Technometrics* (1977) 213–216
- [99] Emmerich, M.T., Deutz, A.H., Yevseyeva, I.: On reference point free weighted hypervolume indicators based on desirability functions and their probabilistic interpretation. *Procedia Technology* **16** (2014) 532–541
- [100] Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation* **9** (2001) 159–195
- [101] Justesen, P.D.: Multi-objective optimization using evolutionary algorithms. University of Aarhus, Department of Computer Science, Denmark (2009)
- [102] Zitzler, E., et al.: Density and approximations of  $\mu$ -distribution for different testproblems, Computer Engineering and Networks Laboratory (TIK), ETH Zürich (2009) <http://people.ee.ethz.ch/~sop/download/supplementary/testproblems/>
- [103] Nielsen, H.B., Lophaven, S.N., Søndergaard, J.: DACE, a MATLAB Kriging toolbox. Informatics and mathematical modelling. Lyngby–Denmark: Technical University of Denmark, DTU (2002)
- [104] Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation* **11** (2003) 1–18
- [105] Gaida, D., Wolf, C., Bäck, T., Bongards, M.: Nonlinear model predictive substrate feed control of biogas plants. In: 2012 20th Mediterranean Conference on Control Automation (MED). (2012) 652–657
- [106] Gaida, D., Luis, S.B.A., Wolf, C., Bäck, T., Bongards, M., McLoone, S.: Optimal control of biogas plants using nonlinear model predictive control. In: IET Irish Signals and Systems Conference (ISSC), Dublin, IEEE (2011) 219–224
- [107] Batstone, D.J., Keller, J., Angelidaki, I., Kalyuzhnyi, S.V., Pavlostathis, S.G., Rozzi, A., Sanders, W.T.M., Siegrist, H., Vavilin, V.A.: Anaerobic Digestion Model No.1 (ADM1). 1st edn. IWA Publishing, London (2002)
- [108] Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing* **9** (2005) 3–12

## BIBLIOGRAPHY

---

- [109] Emmerich, M., Naujoks, B.: Metamodel assisted multiobjective optimisation strategies and their application in airfoil design. In Parmee, I.C., ed.: Adaptive computing in design and manufacture VI, London, Springer (2004) 249–260
- [110] Knowles, J.: ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* **10** (2006) 50–66
- [111] Voutchkov, I., Keane, A.: Multi-objective optimization using surrogates. In Tenne, Y., Goh, C.K., eds.: Computational Intelligence in Optimization, Berlin, Heidelberg, Springer (2010) 155–175
- [112] Saad, M.S., Jamaluddin, H., Darus, I.Z.: PID controller tuning using evolutionary algorithms. *Wseas transactions on Systems and Control* **7** (2012) 139–149
- [113] Chen, B.S., Cheng, Y.M.: A structure-specified  $H^\infty$  optimal control design for practical applications: a genetic approach. *IEEE Transactions on Control Systems Technology* **6** (1998) 707–718
- [114] Zhao, S.Z., Iruthayarajan, M.W., Baskar, S., Suganthan, P.N.: Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization. *Information Sciences* **181** (2011) 3323–3335
- [115] Ho, S.J., Ho, S.Y., Hung, M.H., Shu, L.S., Huang, H.L.: Designing structure-specified mixed  $H_2/H^\infty$  optimal controllers using an intelligent genetic algorithm IGA. *IEEE Transactions on Control Systems Technology* **13** (2005) 1119–1124
- [116] Findeisen, R., Imsland, L., Allgöwer, F., Foss, B.A.: State and output feedback nonlinear model predictive control: An overview. *European Journal of Control* **9** (2003) 190–206
- [117] Coello Coello, C.A.: Evolutionary multi-objective optimization: Basic concepts and some applications in pattern recognition. In Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Ben-Youssef Brants, C., Hancock, E.R., eds.: Proceedings of the Third Mexican conference on Pattern recognition, Berlin, Heidelberg, Springer (2011) 22–33
- [118] BMU: Gesetz für den vorrang erneuerbarer energien (erneuerbare-energien-gesetz - eeg). (2009) <http://www.bmu.de/english/renewable-energy/downloads/doc/47883.php>

- [119] Comport, A.I., Marchand, E., Pressigout, M., Chaumette, F.: Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE Transactions on Visualization and Computer Graphics* **12** (2006) 615–628
- [120] Chiha, I., Liouane, N., Borne, P.: Tuning PID controller using multiobjective ant colony optimization. *Applied Computational Intelligence and Soft Computing* **2012** (2012) 11–18

# English Summary

*Evolutionary algorithms* (EAs) and *Bayesian Global Optimization* (BGO) are two main branches in the field of multi-objective optimization. The first is based on the paradigm of encoding natural evolution for computational observation; the latter uses statistical models in optimization. However, in the field of multi-objective optimization, evolutionary multi-objective optimization algorithm (EMOA) approach as commonly used requires a large number of objective function evaluations and therefore is inefficient when dealing with expensive function evaluation problems. This problem can be solved by using the efficient method of Bayesian Global Optimization. When applying this method in multi-objective optimization, we have a new challenge to meet: the execution time of *multi-objective Bayesian Global Optimization* (MOBGO) itself is still too long, even though it only requires a few function evaluations. The reason for the high cost of MOBGO is two-fold: on the one hand, MOBGO requires an infill criterion to be calculated many times, but the computational complexity of an infill criterion has so far been very high. Another reason is that the optimizer, which aims at searching for an optimal solution according to the surrogate models, is not sufficiently effective.

For the aim of improving the performance of one of the most common infill criteria, *Expected Hypervolume Improvement* (EHVI), a new efficient algorithm is proposed in this thesis. This new efficient algorithm is based on an efficient partitioning algorithm for the non-dominated space and on a newly derived EHVI calculation formula. The computational complexity of the proposed algorithm for exact EHVI calculation is  $O(n \log n)$  both in the cases of two and three objectives. It is shown that this time complexity can no longer be improved. In the case of three objectives, the execution time of the new algorithm is nearly forty thousand times faster than that of the previous algorithm. Moreover, this algorithm can also be extended for other exact calculations of infill criteria, for instance, *Probability of Improvement* (PoI) and *Truncated Expected Hypervolume*

*Improvement (TEHVI).*

To make full use of a-priori knowledge of objective functions, whenever it is available, TEHVI is proposed as a new infill criterion in this thesis. In the definition of TEHVI, the *probability density function* follows a truncated normal distribution, where the truncated domain is determined by the range of the objective functions. The exact calculation of TEHVI is derived from that of EHVI and the computational complexity of TEHVI is  $O(n \log n)$  in the case of two objectives. Since the TEHVI is non-zero only in the valid truncated domain, it can lead the optimizer to the valid domain in the objective space for the aim of searching for the optimal solution. Therefore, compared to EHVI, the TEHVI can generate a better Pareto-front approximation. Moreover, the TEHVI can be utilized to solve the preference-based multi-objective optimization problems, when the truncated domain is set to be the preferred-region in the objective space.

To improve the effectiveness of the optimizer in MOBGO, another infill criterion, namely, Expected Hypervolume Improvement Gradient (EHVIG), is introduced in this thesis. EHVIG is the gradient of EHVI, which makes it possible to apply a gradient ascent algorithm, instead of an EA, in MOBGO as the optimizer. However, a well-known drawback of a gradient ascent (descent) algorithm is that it can easily get stuck at a local optimal solution. This problem is solved in this thesis by introducing the EHVIG as a new stopping criterion in an EA. The basic idea of this method is straightforward: once the EHVIG of one individual in an EA is very close to or even equal to a zero vector, this individual can be regarded as the optimal solution and it becomes unnecessary to continue the following iterations in the EA.

In this thesis, both state-of-the-art EMOAs (NSGA-II and SMS-EMOA) and MOBGO based algorithms are implemented and compared in different practical applications. These applications include the PID parameter tuning problems, the robust PID parameter tuning problems with disturbance, and the bio-gas plant optimization problems. In view of the better performance of the MOBGO based algorithms, it is recommended to use TEHVI-EGO and EHVI-EGO in practical applications.

# Nederlandse Samenvatting

*Evolutionaire algoritmen* (EAs) en *Bayesiaanse Globale Optimalisatie* (BGO) zijn de twee hoofdtakken op het gebied van optimalisatie voor meerdere doelstellingen. De eerstgenoemde tak is gebaseerd op het idee om berekeningen te beschrijven als natuurlijke evolutie, de tweede tak maakt gebruik van statistische modellen voor optimalisatie. Echter, op het gebied van optimalisatie voor meerdere doelstellingen, maken evolutionaire meerdere doelstellingen optimalisatie algoritmen (EMOA) in hun standaard vorm vaak gebruik van een groot aantal evaluaties van de doel-functies, waardoor deze aanpak inefficiënt is in het geval van dure evaluaties. Dit probleem kan worden verholpen door middel van de efficiënte Bayesiaanse Globale Optimalisatie methode. Bij het gebruik van deze methode in optimalisatie voor meerdere doelstellingen is er een nieuwe uitdaging: de executie tijd van *meerdere doelstellingen Bayesiaanse Globale Optimalisatie* (MOBGO) zelf is nog steeds te lang, ondanks dat er maar een klein aantal evaluaties nodig is. Er zijn twee redenen dat MOBGO duur is: ten eerste, voor MOBGO is het noodzakelijk dat een zogeheten opvul criterium vaak berekend wordt, maar tot nog toe is de reken complexiteit van een opvul criterium erg duur. Een tweede reden is dat de optimalisator, die met behulp van surrogaat modellen de optimale oplossing probeert te vinden, niet effectief genoeg is.

Om de efficiëntie te verbeteren van een van de meest voorkomende opvul criteria, *Verwachte Hypervolume Verbetering* (EHVI), wordt er in dit proefschrift een nieuw efficiënt algoritme voorgesteld. Dit nieuwe efficiënte algoritme is gebaseerd op een efficiënt partitionerings algoritme voor de niet-gedomineerde ruimte, en op een nieuw afgeleide formule om de EHVI te berekenen. De reken complexiteit van het voorgestelde algoritme voor precieze berekening van de EHVI is  $O(n \log n)$ , voor zowel twee als drie doelstellingen. Het wordt aangetoond dat de tijd-complexiteit niet verder verbeterd kan worden. In geval van drie doelstellingen is de executie tijd van het nieuwe algoritme bijna veertigduizend keer sneller dan het vorige algoritme. Bovendien kan dit algoritme ook uitgebreid worden voor

de exacte berekening van andere opvul criteria, bijvoorbeeld *Kans op Verbetering* (PoI) en *Getrunceerde Verwachte Hypervolume Verbetering* (TEHVI).

Om volledig gebruik te kunnen maken van a-priorische kennis van doel-functies, wanneer beschikbaar, wordt TEHVI in dit proefschrift voorgesteld als nieuw opvul criterium. In de definitie van TEHVI volgt de waarschijnlijkheidsdichtheidsfunctie een getrunceerde normaalverdeling, waar het getrunceerde deel afhangt van het bereik van de doel-functies. De precieze berekening van TEHVI is afgeleid van die van EHVI, en de reken complexiteit van TEHVI is  $O(n \log n)$  voor twee doelstellingen. Aangezien TEHVI alleen niet-nul is in het valide getrunceerde domein kan het de optimalisator naar het valide domein in de doel-ruimte leiden, met het doel de optimale oplossing te vinden. Daardoor kan TEHVI, in verhouding tot EHVI, een betere Pareto-front benadering vinden. Verder kan TEHVI ook toegepast worden op voorkeur-gebaseerde meerdere doelstellingen optimalisatie problemen, wanneer het getrunceerde domein gedefinieerd wordt als de voorkeur regio in de doel-ruimte.

Om de effectiviteit van de optimalisator in MOBGO te verbeteren wordt er nog een opvul criterium geïntroduceerd in dit proefschrift, namelijk de *Verwachte Hypervolume Verbetering Gradiënt* (EHVIG). EHVIG is de gradiënt van EHVI, waardoor het mogelijk is een gradiënt stijgingsalgoritme, in plaats van een EA, als optimalisator te gebruiken in MOBGO. Echter, een bekend nadeel aan het gradiënt stijgingsalgoritme is dat het gemakkelijk kan blijven hangen in een lokaal optimum. Dit probleem wordt in dit proefschrift verholpen door EHVIG als een nieuw stopzettings criterium toe te passen in een EA. De kern van dit idee is simpel: wanneer de EHVIG van een individu in een EA heel dichtbij, of zelfs gelijk aan, nul is, kan dit individu als de optimale oplossing beschouwd worden, waardoor het onnodig wordt om verder te gaan met de iteraties van het EA.

In dit proefschrift zijn zowel state-of-the-art EMOAs (NSGA-II en SMS-EMOA) en MOBGO gebaseerde algoritmen geïmplementeerd en vergeleken in verschillende praktische toepassingen. Het betreft toepassingen voor het PID parameter afstellings probleem, het robuuste PID parameter afstellings probleem met verstoring, en het bio-gas fabriek optimalisatie probleem. Gezien de betere prestatie van de MOBGO gebaseerde algoritmen wordt er aanbevolen om TEHVI-EGO en EHVI-EGO te gebruiken voor praktische applicaties.

# Curriculum Vitae

Kaifeng Yang was born on December 23rd, 1987 in Fufeng, Shaanxi, People's Republic of China. He received his Bachelor degree in Automation at School of Information Science and Engineering at Central South University in July 2010. In his bachelor thesis, he investigated fuzzy controller on an intelligent car, which can navigate by itself, using Freescale and STMicroelectronics microcontroller unit(s) under the supervision of Dr. Ji Wang.

He was recommended to the same department as a Master student under the supervision of Dr. Ji Wang in September 2010. During his Master degree, he won the first prize in Freescale Cup Intelligent Car in 2011. For his master thesis, he implemented a genetic algorithm to solve the power network reconfiguration problem for single- and bio- objective optimization. In 2013, he finished his graduate studies successfully and received his MSc in Control Engineering in July 2013.

In September 2013, he joined the research group headed by Prof. Thomas Bäck. He started his Ph.D. research at the Leiden Institute of Advanced Computer Science (LIACS), under the supervision of Prof. Thomas Bäck and Dr. Michael Emmerich. His Ph.D. research was founded by China Scholarship Council (CSC). His research primarily focuses on hypervolume-based multi-objective Bayesian global optimization and preference-based multi-objective optimization. He also participated in a short project of predictive analytics in LIACS. Besides his research, he also acted as a teaching assistant for several courses in LIACS, including Evolutionary Algorithms and Natural Computing.