# Multi-Objective Bayesian Optimization with Asynchronous Batch Selection

by

Ane Zuniga

B.S., University of the Basque Country (2021)
B.Eng., University of Applied Sciences Mittweida (2021)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024

| | |
|---|---|
| Authored by: | Ane Zuniga<br>Electrical Engineering and Computer Science Department<br>May 17, 2024 |
| Certified by: | Mina Konaković Luković<br>Assistant Professor of Electrical Engineering and Computer Science<br>Thesis Supervisor |
| Accepted by: | Leslie A. Kolodziejski<br>Professor of Electrical Engineering and Computer Science<br>Chair, Department Committee on Graduate Students |

# Multi-Objective Bayesian Optimization with Asynchronous Batch Selection

by

Ane Zuniga

Submitted to the Department of Electrical Engineering and Computer Science
on May 17, 2024 in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

## ABSTRACT

Multi-objective optimization problems are widespread in scientific, engineering, and design fields, necessitating a balance of trade-offs between conflicting objectives. These objectives often represent black-box functions, which are costly and time-consuming to evaluate. Multi-objective Bayesian optimization (MOBO) offers a valuable approach to guide the search for optimal solutions. To enhance efficiency, batch evaluations are employed to test multiple samples simultaneously, aiming to further reduce evaluation times. However, in scenarios involving varying evaluation times, standard batch strategies often lead to suboptimal resource utilization and inefficiencies. Asynchronous evaluations emerge as a promising solution to optimize resource usage under these conditions.

Despite their potential, there has been no prior work or method specifically tailored to address asynchronous evaluations within the MOBO framework. To bridge this critical gap, this thesis proposes a comprehensive adaptation and analysis of existing Bayesian optimization methods for asynchronous MOBO scenarios. It also introduces a novel selection strategy, $\alpha$-HVI, empirically validated through tests on both synthetic and real-world functions.

Thesis supervisor: Mina Konaković Luković
Title: Assistant Professor of Electrical Engineering and Computer Science

# Acknowledgments

First of all, I would like to thank my thesis supervisor, Prof. Mina Konaković Luković, whose guidance and insight have been invaluable throughout this project. Her expertise and encouragement have profoundly shaped my approach and continuously inspired my dedication to research.

I am also grateful to my collaborator, Yunsheng Tian, for our enlightening discussions on the topic and his continuous encouragement throughout the project. His initial explanations helped pave the way for my research, and his support has been a foundation of my progress.

The project that gave rise to these results received the support of a fellowship from ”la Caixa” Foundation (ID 100010434). The fellowship code is LCF/BQ/EU21/11890103.

A special thank you to my labmates at the Algorithmic Design Group. It has been a pleasure and a privilege to spend the past year working and learning alongside such a talented, inspiring, and fun group of individuals.

I am also thankful for the deep friendships I have formed with my Spanish friends I met at MIT. From sailing afternoons to Catan nights, and echo frisbee games, you have made the past year exceptionally memorable and joyful.

Heartfelt thanks go to my family and my best friend on the west coast. Your warm welcomes during my visits, our hikes, and our playful hide-and-seek adventures have been the true highlights of my life outside the academic world.

Eskerrik asko nire gurasoei, nire euskarri izateagatik eta zuengandik jaso dudan maitasunagatik. Itsasoso batek banatzen gaituen arren, gertu sentitzen zaituztet, eta ematen didazuen indarra nire bidearen oinarrizko zati bat da. [1]

Last but certainly not least, I owe a huge debt of gratitude to my brother, Mikel. Your unwavering belief in me, ensuring I always had the right tools and mindset to succeed, has been invaluable. Your continuous support and deep love have truly helped me thrive not only in my studies but also personally.

[1] To my parents, thank you for your unconditional love and support. Despite the physical distance across the ocean, our emotional bond remains strong, and your encouragement has been a fundamental part of my journey.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

In the realms of science and engineering, the pursuit of optimal solutions is a common yet complex challenge. These solutions often need to be derived for black-box functions, which are complex systems or processes. Evaluating these functions is typically costly and time-consuming, which limits the feasible number of experiments. Recently, there has been a significant shift towards using computational methods to enhance the experimental design process, aiming for more effective results.

One method that has gained significant importance in addressing this challenge is *Bayesian optimization* (BO). BO's effectiveness lies in two key aspects: first, it uses a cost-efficient surrogate model to represent the underlying black-box functions based on existing experimental data; second, it employs an acquisition function to navigate the design space and balance the exploration-exploitation trade-off.

Moreover, many problems involve the simultaneous optimization of conflicting objectives. *Multi-objective Bayesian optimization* (MOBO) extends BO's capabilities to address this challenge efficiently. Applications of such approaches are seen in diverse fields, including robot design [1], development of new materials [2][3], drug discovery [4], and hyperparameter tuning for machine learning models [5].

In many real-world experiments, batch evaluations, which involve assessing multiple solutions simultaneously, can help reduce the time and cost of the optimization process. However, these evaluations or tests may take varying amounts of time to complete, such as in scenarios where different simulations or physical experiments have different runtimes. For instance, in aerospace engineering, aerodynamic optimization techniques can lead to varying simulation times depending on airfoil design and conditions [6]; in chemical engineering, the optimization of distillation processes can extend simulation durations due to the complexities of chemical interactions [7]; in computational biology, the time needed for protein folding simulations varies with the complexity of the proteins involved [8]. These examples highlight the necessity for flexible optimization strategies that can adapt to the varying computational demands of different tasks.

Traditional *synchronous* batch settings require waiting until all evaluations within a batch are finished before initiating a new batch. This waiting period can lead to suboptimal resource utilization and extended overall optimization durations.

To address this issue, *asynchronous* evaluations can be employed, where as soon as one evaluation is completed, the next one is promptly initiated. This approach optimizes resource usage and expedites the optimization process by minimizing idle times and aligning resource allocation more effectively with task completion.

In this work, we delve into asynchronous MOBO, an area that has received limited attention in previous literature. To the best of our knowledge, no previous published work has specifically tailored or extensively studied asynchronous evaluations within the MOBO framework.

The key contributions of this thesis can be summarized as follows:

- We adapt existing BO and MOBO methods to effectively address asynchronous MOBO problems. This adaptation process enables us to assess and benchmark performance in this setting.

- We introduce $\alpha$-HVI, a novel selection strategy founded on the hypervolume improvement metric. This approach takes into account both prediction and uncertainty values for points under evaluation with yet unknown objective values.

- We conduct thorough evaluations using both synthetic and real-world test functions, applying both the adapted methods and our proposed approach.

# Chapter 2

# Preliminaries

## 2.1 Multi-objective optimization

*Multi-objective optimization* (MOO) involves optimization problems defined over a continuous $d$ dimensional *design space* $\mathcal{X} \subset \mathbb{R}^d$, where the goal is to simultaneously minimize a set of $m \geq 2$ objective functions $f_1, \ldots, f_m : \mathcal{X} \rightarrow \mathbb{R}$ [9]. We represent the vector of all objectives as $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))$, where $\boldsymbol{x} \in \mathcal{X}$ is a vector of input variables. The *performance space* is an $m$-dimensional image of $\mathcal{X}$, denoted as $\boldsymbol{f}(\mathcal{X}) \subset \mathbb{R}^m$.

Since objectives often conflict with each other, optimal solutions often form a set rather than a single solution. These optimal solutions constitute the *Pareto set*, denoted as $\mathcal{P}_s \subset \mathcal{X}$ in the design space, and the corresponding images in the performance space form the *Pareto front*, represented as $\mathcal{P}_f = \boldsymbol{f}(\mathcal{P}_s) \subset \mathbb{R}^m$. More specifically, a point $\boldsymbol{x}^* \in \mathcal{P}_s$ achieves Pareto optimality when there is no other point $\boldsymbol{x} \in \mathcal{X}$ satisfying $f_i(\boldsymbol{x}^*) \geq f_i(\boldsymbol{x})$ for all $i$ and $f_i(\boldsymbol{x}^*) > f_i(\boldsymbol{x})$ for at least one $i$.

### 2.1.1 Hypervolume indicator

The *hypervolume indicator* [10] serves as a widely adopted metric for assessing the quality of an approximated Pareto front in MOO. The hypervolume $\mathcal{H}(\mathcal{P}_f)$ can be understood as a measure that evaluates the region within an $m$-dimensional performance space covered by a Pareto front approximation $\mathcal{P}_f$ with respect to a reference point $\boldsymbol{r} \in \mathbb{R}^m$. This hypervolume $\mathcal{H}(\mathcal{P}_f)$ is defined as:

$$\mathcal{H}(\mathcal{P}_f) = \int_{\mathbb{R}^m} 1_{\mathcal{H}(\mathcal{P}_f)}(z)\, \mathrm{d}z, \tag{2.1}$$

where $\mathcal{H}(\mathcal{P}_f)$ represents the set $\boldsymbol{z} \in Z$ that satisfy the condition: $\exists 1 \leq i \leq |\mathcal{P}_f| : \boldsymbol{r} \preceq \boldsymbol{z} \preceq \mathcal{P}_f(i)$ where $\mathcal{P}_f(i)$ denotes the $i$-th solution within the Pareto front $\mathcal{P}_f$, $\preceq$ serves as the relational operator indicating objective dominance, and 1 is as a Dirac delta function resulting in a value of 1 if $z \in \mathcal{H}(\mathcal{P}_f)$ and 0 otherwise.

A higher hypervolume indicates that the Pareto front approximation $\mathcal{P}_f$ offers a better representation of the true Pareto front.

### Hypervolume improvement

To quantify the extent of this improvement when a set of new $n$ points $P = \boldsymbol{p}_1, \ldots, \boldsymbol{p}_n \subset \mathbb{R}^m$ is introduced to the existing Pareto front approximation $\mathcal{P}_f$, the concept of *hypervolume improvement* (HVI) is employed. Figure 2.1 illustrates this metric, which measures the increase in hypervolume resulting from the addition of these new points. HVI is defined as:

$$HVI(P, \mathcal{P}_f) = \mathcal{H}(\mathcal{P}_f \cup P) - \mathcal{H}(\mathcal{P}_f). \tag{2.2}$$

## 2.2 Bayesian optimization

Many optimization problems involve the search for a global optimal solution within the realm of a black-box function $f : \mathcal{X} \subset \mathbb{R}^d \rightarrow \mathbb{R}$. These functions are often expensive to evaluate, lacking an analytical form or known derivatives, and the number of function evaluations that can be conducted is limited.

In such scenarios, *Bayesian optimization* (BO) emerges as an effective solution by adeptly

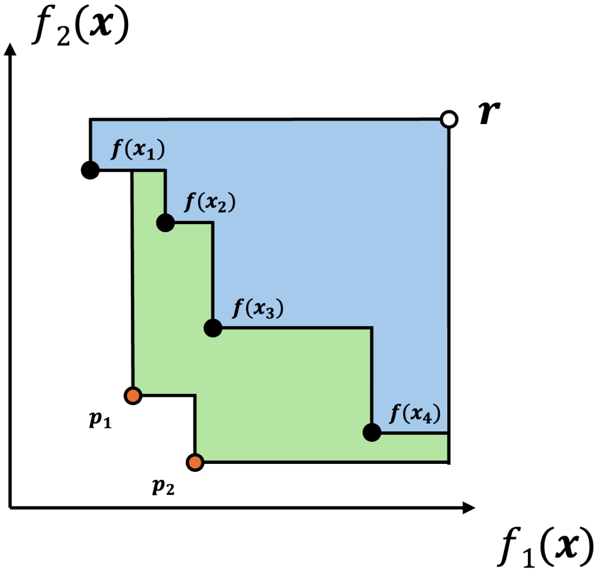

Figure 2.1: Illustration of hypervolume improvement in a two-objective performance space. The hypervolume of evaluated points $\mathbf{f}(\mathbf{x}_1), \mathbf{f}(\mathbf{x}_2), \mathbf{f}(\mathbf{x}_3)$ and $\mathbf{f}(\mathbf{x}_4)$with respect to the reference point $\mathbf{r}$ is shaded in blue. The hypervolume improvement achieved by adding points $\mathbf{p}_1$ and $\mathbf{p}_2$ is shaded in green.

managing the trade-off between exploration (sampling from new and uncertain areas) and exploitation (intensive probing of promising regions to maximize performance).

### 2.2.1 Surrogate model

The *surrogate model* is a pivotal component in BO, serving as an approximation of the unknown objective function $f$. In this context, the Gaussian Process (GP) is a commonly employed surrogate model [11].

A GP over a space $\mathcal{X}$ is a random process $\mathcal{X} \to \mathbb{R}$ characterized by a *mean function* $\mu : \mathcal{X} \to \mathbb{R}$ and a *kernel function* $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. If a function $f$ is sampled from $\mathrm{GP}(\mu, \kappa)$, then $f(\boldsymbol{x})$ is distributed normally $f(\boldsymbol{x}) \sim \mathcal{N}(\mu(\boldsymbol{x}), \kappa(\boldsymbol{x}, \boldsymbol{x}))$ for a finite set of inputs $\boldsymbol{x} \in \mathcal{X}$.

The next step involves training a GP posterior to refine the initial model. This can be accomplished by maximizing the log marginal likelihood, $\log p(\boldsymbol{y}|\boldsymbol{x}, \theta)$, on the available dataset $\{X, Y\}$, where $\theta$ represents the the kernel function parameters. The resulting GP posterior distribution is defined as $f(\boldsymbol{x}) \sim \mathcal{N}(\mu(\boldsymbol{x}), \Sigma(\boldsymbol{x}))$, with the mean function $\mu(\boldsymbol{x}) =$

$m(\boldsymbol{x}) + \boldsymbol{k}\boldsymbol{K}^{-1}Y$ and covariance function $\Sigma(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}\boldsymbol{K}^{-1}\boldsymbol{k}^T$, where $\boldsymbol{k} = k(\boldsymbol{x}, X)$ and $K = k(X, X)$.

### 2.2.2 Acquisition function

The *acquisition function* $\widetilde{f}(\boldsymbol{x})$ is used to select the next point for evaluation. It plays a vital role in guiding the optimization process by balancing the exploration-exploitation trade-off. In the context of single-objective optimization, several popular acquisition functions are commonly employed, such as Expected Improvement (EI) [12], Lower Confidence Bound (LCB) [13], Probability of Improvement (PI) [14], or Thompson Sampling (TS) [15].

In this work we use LCB, which selects the next evaluation point based on the lower confidence bound, a representation of the likely lower limit of the true objective function: $LCB(\boldsymbol{x}) = \mu(\boldsymbol{x}) - \beta^{1/2}\sigma(\boldsymbol{x})$, where $\mu(\boldsymbol{x})$ is the mean predicted value, $\sigma(\boldsymbol{x})$ is the corresponding standard deviation, and $\beta$ is a hyperparameter controlling the exploration-exploitation trade-off.

## 2.3 Multi-objective Bayesian optimization

In *Multi-objective Bayesian optimization* (MOBO), two common approaches are used to handle problems with multiple conflicting objectives.

The first approach consists on using cheap acquisition functions (e.g., EI, LCB, PI, TS) for each objective function. These acquisition functions optimize individual unknown objectives independently. Following that, a population-based multi-objective evolutionary algorithm (e.g., NSGA-II [16], MOEA/D [17]) is employed to identify a set of *candidate points* $\mathcal{C}$ representing Pareto-optimal solutions, considering objective trade-offs. Then, a candidate point $\boldsymbol{x} \in \mathcal{C}$ is selected for evaluation using a chosen selection strategy. This strategy may involve metrics such as HVI or uncertainty-based selection [18].

The second approach consolidates all objectives into a single acquisition function. A well-known example of this is *Expected Hypervolume Improvement* (EHVI) [19], which selects the next evaluation point that maximizes the expected increase in hypervolume over the current Pareto front approximation.

## 2.4 Synchronous vs. Asynchronous Evaluations

In the context of batch optimization, we employ a set of $N$ parallel workers represented as $w \in \mathbb{N}$. When working in the synchronous setting, we wait until all evaluations are completed before proceeding to optimize or send the next batch to evaluate. In contrast, in the asynchronous setting, we dispatch the next evaluation as soon as one evaluation is completed. Consequently, there are data points currently undergoing evaluation for which their respective objective values are yet unknown. We have $b \leq w \in \mathbb{N}$ evaluations in progress which are commonly referred to as *busy points*, denoted as $\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_b$ forming a set $\mathcal{Q} \subset \mathcal{X}$.

Figure 2.2 shows a comparison of synchronous and asynchronous batch evaluations. As it can be observed, asynchronous batch evaluations more efficiently utilizes resources, allowing for a greater number of evaluations within the same timeframe compared to synchronous batch evaluations.

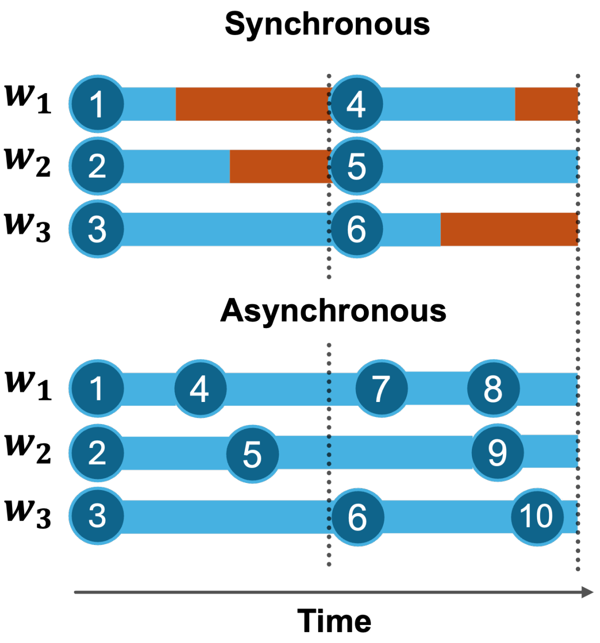

Figure 2.2: Comparison of synchronous and asynchronous batch BO with $N = 3$ workers. Blue bars represent evaluation times and red bars represent waiting times.

# Chapter 3

# Related Work

To the best of our knowledge, there is no previous work specifically tailored to handle asynchronous evaluations in the context of MOBO. In this section, we introduce relevant prior work on asynchronous BO and batch MOBO. Given the absence of prior research in asynchronous MOBO, we delve into various strategies to extend and adapt the methods presented in this section to effectively handle asynchronicity in MOBO, which are described in Chapter 4.

## 3.1 Thompson Sampling

Thompson Sampling (TS) [15] is a probabilistic method used in sequential decision-making processes which operates by sampling potential solutions according to the posterior probability that they are optimal. Specifically, at each step $t+1$, the solution $\boldsymbol{x}_{t+1}$ is drawn from the posterior distribution $p_{\boldsymbol{x}}(\cdot|\mathcal{D}_t)$, where $\mathcal{D}_t = \{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^{t}$ represents the dataset of evaluations completed up to step $t$.

The work of [20] extends single-objective TS to the asynchronous domain, introducing Asynchronous TS (Asy-TS). This variant leverages the inherent randomness of TS to ensure diversity in solution selection during asynchronous evaluations.

In the context of MOO, Thompson Sampling Efficient Multi-objective Optimization

(TSEMO) [21] utilizes TS independently for each objective in conjunction with the HV metric and NSGA-II to effectively conduct MOBO.

## 3.2 Kriging Believer

Kriging Believer (KB) [22] incorporates point hallucination, assuming that busy points have the surrogate's posterior mean value. This assumption is applied prior to fitting the new surrogate model. KB is applicable in both single and multi-objective BO.

## 3.3 Local Penalization and Hard Local Penalization

Local Penalization (LP) [23] was initially introduced as a batch single-objective BO algorithm and later extended to the asynchronous setting [24]. The method relies on emulating the sequential evaluation setting, where upon obtaining experimental results for an input, it is generally expected that the acquisition function will decrease in the neighborhood around that point. This is achieved by defining a new penalized acquisition function $\widetilde{f}'(\boldsymbol{x})$, which is the original acquisition function $\widetilde{f}(\boldsymbol{x})$ multiplied by penalization terms centered at each batch or busy points $\boldsymbol{q} \in \mathcal{Q}$.

The penalizer $\psi$ is built assuming that the function satisfies the Lipschitz continuity condition. A spherical region $\mathbb{S}(\boldsymbol{r}_j, \boldsymbol{q})$ is then considered, centered around $\boldsymbol{q}$. The radius $\boldsymbol{r}_j$ is determined using the Lipschitz continuity property, ensuring that the optimum $f(\boldsymbol{x}^*)$ does not lie inside the spherical region $\mathbb{S}(\boldsymbol{r}_j, \boldsymbol{q})$. It is important to note that this radius $\boldsymbol{r}_j$ is stochastic due to $f$ being modeled by a GP.

[23] define the LP penalizer $\psi_{LP}$ as the probability that a point $\boldsymbol{x} \in \mathcal{X}$ does not lie inside $\mathbb{S}(\boldsymbol{r}_j, \boldsymbol{q})$. $\psi_{LP}$ possesses several properties, including that the penalization region shrinks as the the expected value for $f(\boldsymbol{q})$ gets close to the global optimum or when the Lipschitz constant $L$ increases. [24] argue that, on top of those properties, the extent of penalization should increase as $\boldsymbol{x}$ gets closer to $\boldsymbol{q}$, with $\widetilde{f}(\boldsymbol{x}) = 0$ if $\widetilde{f}(\boldsymbol{q}) \geq 0, \forall \boldsymbol{x} \in \mathcal{X}$ to avoid redundant

sampling, and thus propose an alternative penalizer $\psi_{HLP}$. For more details please refer to [23] and [24].

## 3.4 Highly Parallelizable Pareto Optimization

Highly Parallelizable Pareto Optimization (HIPPO) [25] is a penalization approach for the batch synchronous MOBO scenario, which is integrated into the EHVI acquisition function. [25] argue that, because distant parts of the input domain might correspond to the same area of the Pareto front, diversity in the input space may not achieve diversity along the Pareto front. Therefore, the proposed penalizer $\psi_{HIPPO}$ is based on the performance space distances, i.e. candidate points are penalized based on how close their predictive objectives are in the performance space to batch selected points.

Particularly, HIPPO considers the Mahalanobis distance between candidate and batch points to build $\psi_{HIPPO}$ , which for independent GPs is reduced to the $L^2$ norm normalized by the standard deviation. Similar as LP and HLP, the penalized acquisition function $\widetilde{f}'(\boldsymbol{x})$, is the original acquisition function $\widetilde{f}_{EHVI}(\boldsymbol{x})$ multiplied by penalization terms at each batch point $\boldsymbol{q} \in \mathcal{Q}$. For more details, we refer the reader to [25].

## 3.5 Diversity-Guided Efficient Multi-Objective Optimization

Diversity-Guided Efficient Multi-Objective Optimization (DGEMO) [26] is specifically designed for the synchronous batch MOBO domain. This method integrates both the design and performance spaces by utilizing Karush–Kuhn–Tucker (KKT) conditions. DGEMO categorizes candidate points into diversity regions to optimize the exploration-exploitation balance. DGEMO then selects one point from each region family to include in the batch, based on their expected HVI metric.

# Chapter 4

# Asynchronous MOBO

## 4.1 Adaptation of Prior Work to Asynchronous MOBO

### 4.1.1 Thompson Sampling

To adapt the Asy-TS method for asynchronous MOBO, we directly extend the principles of TSEMO to the asynchronous setting. In this adaptation, we employ the TS acquisition function independently for each objective and conduct MOO using NSGA-II. The next evaluation point is selected from the set of candidate points $\mathcal{C}$ by maximizing the hypervolume improvement metric:

$$\boldsymbol{x}_{next} = \arg\max_{\boldsymbol{x}\in\mathcal{C}} \Big\{ \mathcal{H}(\widetilde{f}(\boldsymbol{x}) \cup \mathcal{P}_f - \mathcal{H}(\mathcal{P}_f) \Big\} \tag{4.1}$$

Notably, as discussed in [20] we take advantage of the intrinsic randomness within TS and completely disregard busy points throughout the asynchronous optimization process.

### 4.1.2 Kriging Believer

To implement KB for asynchronous MOBO, we initially fit the surrogate model with the evaluated dataset up to $t$ evaluations $\mathcal{D}_t = \{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^{t}$ to obtain the posterior mean values for busy points. Next, we update the surrogate model using both the evaluated dataset $\mathcal{D}_t$ and the predictions for busy points $\{(\boldsymbol{q}_i, \boldsymbol{\mu}(\boldsymbol{q}_i))\}_{i=1}^{b}\}$, and we use this updated model for optimization. We employ independent acquisition functions for each objective and leverage the NSGA-II algorithm to generate a set of candidate points $\mathcal{C}$. We select the next evaluation point based on maximizing the HVI metric within $\mathcal{C}$ as described in Eq. 4.1.

### 4.1.3 Local Penalization and Hard Local Penalization

To adapt the single-objective penalization methods LP and HLP for asynchronous MOBO, we initiate the process by defining penalization regions for each objective. These regions are based on the best observed values within the evaluated dataset $\mathcal{D}_t$, and we estimate a global Lipschitz constant for each objective following the methodology outlined in the original LP paper [23]. For each busy point $\boldsymbol{q} \in \mathcal{Q}$, we then independently apply LP or HLP penalization to each objective. The penalized acquisition functions are utilized in conjunction with the NSGA-II to generate a set of candidate points $\mathcal{C}$. The selection of the next point for evaluation is determined by maximizing the HVI metric, as described in Eq. 4.1.

### 4.1.4 Highly Parallelizable Pareto Optimization

When adapting the HIPPO-EHVI method for asynchronous MOBO, we integrate busy points into the model as if they were part of a synchronously selected batch. This integration involves applying the HIPPO penalization technique to these points $\boldsymbol{q} \in \mathcal{Q}$. After penalizing the busy points, we use the standard EHVI acquisition function to determine the most promising candidate for the next evaluation.

### 4.1.5 Diversity-Guided Efficient Multi-Objective Optimization

Our primary focus when adapting DGEMO for asynchronous scenarios centers on the final stage of the MOBO pipeline, where candidate points are categorized into diversity families.

We obtain the surrogate model's posterior mean for both candidate and busy points and determine the candidate point with the closest Euclidean distance within the surrogate's posterior space for each busy point. Subsequently, we cluster the busy points into the same family as this nearest candidate point. We then continue with the DGEMO selection process following the procedures outlined in the original paper.

## 4.2 Proposed method: $\boldsymbol{\alpha}$-HVI

In this section, we present our proposed method for addressing asynchronous evaluations within the MOBO framework, which primarily focuses on the choice of the acquisition function and the final selection stage of the pipeline.

To begin, we fit the GP surrogate model using the dataset of evaluations $\mathcal{D}_t$. Following this, we employ the LCB acquisition function, independently for each of the objectives. The selection of LCB is motivated by our aim to consider uncertainty, and this acquisition function allows us to control the degree of uncertainty we wish to incorporate. We then employ the NSGA-II multi-objective optimizer to generate a set of candidate points $\mathcal{C}$.

For the final selection within $\mathcal{C}$, we propose a novel variant of the HVI metric, $\alpha$-HVI, where we include the acquisition values of busy points together with the current evaluated Pareto front:

$$\alpha\text{-}HVI(\boldsymbol{x}, \mathcal{Q}, \mathcal{P}_f) = \mathcal{H}(\widetilde{f}(\boldsymbol{x}) \cup \mathcal{P}_f \cup \widetilde{\boldsymbol{f}}(\mathcal{Q})) - \mathcal{H}(\mathcal{P}_f \cup \widetilde{\boldsymbol{f}}(\mathcal{Q})) \tag{4.2}$$

Therefore, we select the next point for evaluation $\boldsymbol{x}_{next}$:

$$\boldsymbol{x}_{next} = \arg\max_{\boldsymbol{x} \in \mathcal{C}} \{\alpha\text{-}HVI(\boldsymbol{x}, \mathcal{Q}, \mathcal{P}_f)\} \tag{4.3}$$

$\alpha$-HVI metric is founded on three essential considerations: uncertainty, diversity and batch size awareness.

### 4.2.1 Uncertainty

$\alpha$-HVI recognizes the significance of considering not only the performance predictions of our busy points and candidate points, but also the associated uncertainties. This consideration of uncertainty aligns with the well-known principle of *optimism in the face of uncertainty* [27] in sequential decision-making. This principle claims that exploring areas with higher uncertainty may increase the likelihood of discovering optimal solutions. We achieve this by incorporating the acquisition values $\widetilde{f}(\boldsymbol{x})$ and $\widetilde{\boldsymbol{f}}(\mathcal{Q})$ into the HV calculation in $\alpha$-HVI. This enables us to account for uncertainties and make informed decisions that balance predicted performance and uncertainty levels across the Pareto front.

### 4.2.2 Diversity

Our aim is to avoid selecting points with performance predictions that closely mimic those of busy points. When we actively explore a specific region on the Pareto front, we intentionally refrain from sampling in the immediate vicinity of that performance region to ensure that the selected candidate points provide valuable and diverse information. This deliberate avoidance of busy regions is essential for maintaining diversity and effectiveness in our selection process. $\alpha$-HVI represents an enhancement over HVI, which also avoids sampling near explored regions but does not incorporate uncertainty into its selection process.

By introducing $\widetilde{\boldsymbol{f}}(\mathcal{Q})$ into $\alpha$-HVI, we simultaneously account for both performance improvement and uncertainty. This approach actively discourages the selection of candidate points lying in close proximity to these busy regions due to their expected minimal contribution to HV, while promoting diversity by encouraging sampling in further locations, leading to achieving a widely spread-out Pareto front.

Diversity in the Pareto front is crucial because it ensures that we explore a broader range of trade-offs and objective values, leading to a more comprehensive understanding of the problem space.

### 4.2.3 Batch size awareness

We incorporate awareness of the number of workers into our approach. Our goal is to comprehensively cover as much of the Pareto front as possible, factoring in the available number of workers. $\alpha$-HVI ensures that we make the best use of our resources, adapting to the dynamics of the batch size. In the calculation of the HV in Eq. 4.2, we consider the $\mathcal{Q}$ set with $b$ elements, inherently accounting for the number of workers. This optimization of our selection strategy allows us to efficiently explore the Pareto front while effectively utilizing the available evaluation points.

# Chapter 5

# Experimental Evaluation

We now compare our proposed method, $\alpha$-HVI, with the methods we previously adapted, using both synthetic test functions and real-world benchmark problems.

## 5.1 Benchmark problems

First, we conduct experiments on 6 synthetic multi-objective test functions including DTLZ-2, DTLZ-6 [28], OKA-1, OKA-2 [29], VLMOP-2 [30], ZDT-3 [31], which are widely used in previous literature. The experiments include problems with 2 objectives, and the number of design variables varying from 2 to 6. Second, we adopt 6 real-world engineering design problems which are: gear train design [32], welded beam design [33], four-bar truss design [34], rocket injector design [35], simply I supported beam design [36], and disc brake design [33].

In this section, we briefly introduce the properties of each problem, including the dimensions of the design space $\mathcal{X} \subset \mathbb{R}^d$ and performance space $\boldsymbol{f}(\mathcal{X}) \subset \mathbb{R}^m$ and the reference points we use for calculating the hypervolume indicator. To ensure fairness in our comparisons, we employ a consistent reference point for all algorithms..

The problem descriptions for the 6 synthetic functions and the 6 real-world problems are shown in Table 5.1 and Table 5.2 respectively. For all functions that can work with arbitrary dimensions, we use 6 variables and 2 objectives for consistency in testing.

Table 5.1: Description of synthetic functions.

| Name | d | m | $\boldsymbol{r}$ |
|---|---|---|---|
| DTLZ-2 [28] | 6 | 2 | (1.7435, 1.6819) |
| DTLZ-6 [28] | 6 | 2 | (5.7482, 5.6523) |
| OKA-1 [29] | 2 | 2 | (7.4051, 4.3608) |
| OKA-2 [29] | 3 | 2 | (3.1315, 4.6327) |
| VLMOP-2 [30] | 2 | 2 | (1.0, 1.0) |
| ZDT-3 [31] | 6 | 2 | (0.9699, 6.0236) |

Table 5.2: Description of real-world problems.

| Name | d | m | $\boldsymbol{r}$ |
|---|---|---|---|
| Gear train design [32] | 4 | 3 | (18.0690, 60.0000, 2.1070) |
| Welded beam design [33] | 4 | 3 | (333.9087, 17561.6000, 425021.3810) |
| Four bar truss design [34] | 4 | 2 | (2947.9887, 0.0506) |
| Rocket injector design [35] | 4 | 3 | (1.0020, 1.2433, 1.0941) |
| Simply supported beam design [36] | 4 | 2 | (849.7118, 5.9956) |
| Disc brake design [33] | 4 | 3 | (8.4445, 8.9419, 1430.5745) |

## 5.2 Algorithms and Implementation

We conducted a comprehensive analysis of the performance of the algorithms we adapted, described in Section 4.1, alongside our proposed method. All algorithms, including our own and except async-HIPPO, were implemented within the AutODEx [37] Python codebase, which is built upon pymoo [38], a state-of-the-art Python framework for multi-objective optimization.

In all experiments, we utilize a zero-mean GP surrogate model with a Matern-1/2 kernel. We train the GP posterior to refine the prior model by maximizing the log marginal likelihood on the available dataset.

We perform 10 independent test runs with 10 different random seeds for each problem on each algorithm. For each test run of one problem, we use the same initial set of samples for every algorithm, which is generated by Latin hypercube sampling (LHS) [39] using a same random seed. Particularly, we generate 20 initial samples and do a total of 200 evaluations.

To simulate asynchronicity in our experiments, we assign random evaluation times to each experiment. These times are sampled from a half-normal distribution with a mean of 1 and a standard deviation of $\sigma = \sqrt{\pi/2}$.

## 5.3 Comparison metric

To evaluate the performance of different algorithms, we rely on the hypervolume indicator [10], which is the most widely recognized metric for evaluating MOO problems [40]. We monitor the increase of the hypervolume indicator 2.1.1 over the number of evaluations performed. To ensure a fair comparison across all algorithms, we maintain a consistent reference point and the same initial sample set for each.

## 5.4 Results and Discussion

The test problems we selected aim to assess the effectiveness of our method across a variety of function characteristics including concave, convex, disconnected, and multimodal aspects, as well as varying complexities in design and performance spaces. Our approach shows consistent performance, compared to other algorithms that oscillate on different types of problems.

### 5.4.1 Asynchronous comparison

We initiate our evaluation by focusing on the performance of our method alongside other asynchronous MOBO algorithms in a fixed setup with 15 workers. The improvement in the hypervolume indicator throughout the course of 200 evaluations is shown in Figure 5.1

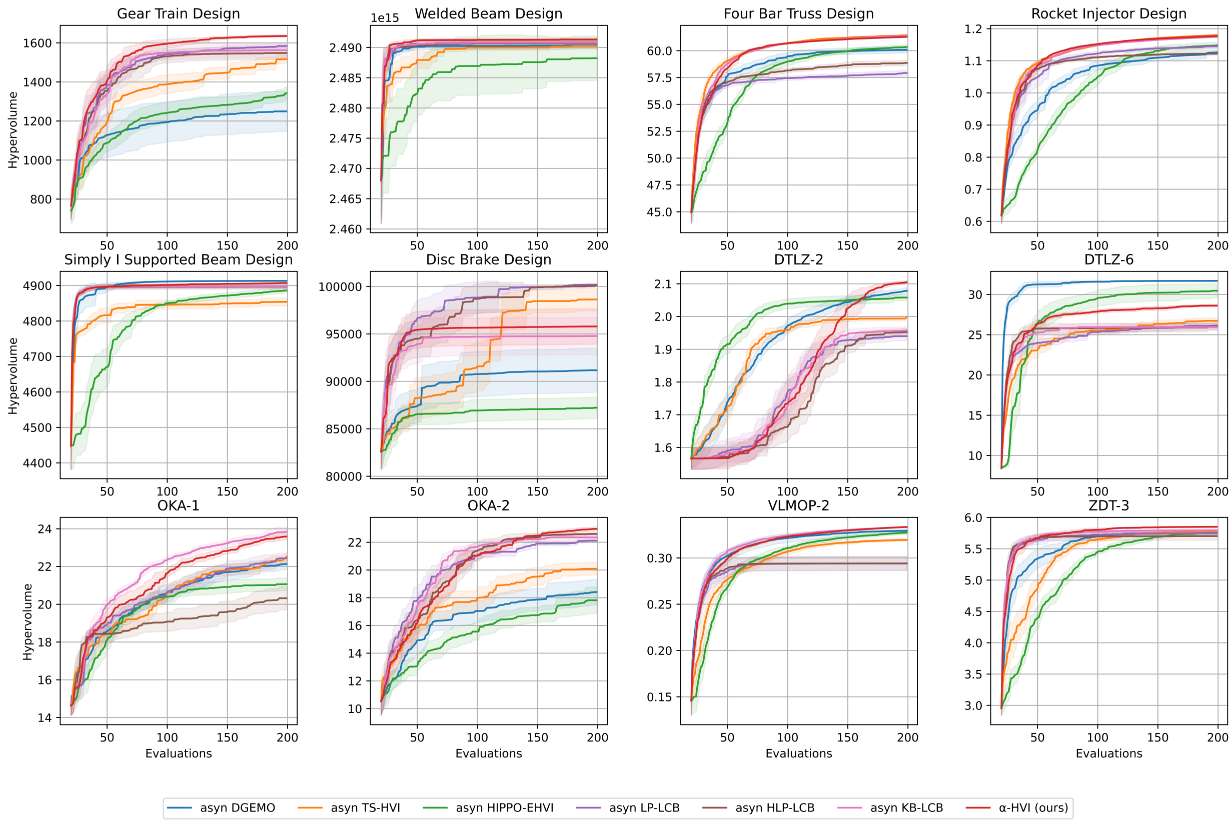

Figure 5.1: Comparison of various algorithms, including our $\alpha$-HVI, across a suite of 6 real-world problems and 6 synthetic test functions. The experiments are conducted under asynchronous conditions with 15 workers and involve up to 200 function evaluations, beginning with an initial set of 20 samples drawn using LHS. The hypervolume indicator is shown relative to the number of function evaluations. Each curve represents the average outcome across 10 different random seeds, with variance indicated by the shaded regions. Our approach, $\alpha$-HVI, consistently demonstrates robust performance, often ranking as the top performer in these tests.

Following this initial comparison, we extend our testing to include a broader range of worker setups: 2, 5, 10, 15, and 30 workers. The results are presented in Figure 5.2 and they are derived from a detailed comparative analysis where we evaluate the performance of each algorithm across all test problems and for each random seed. Specifically, after conducting 200 evaluations for each combination of problem and seed, we identify which method achieves the highest hypervolume indicator, designating it as the best performer for that specific scenario.

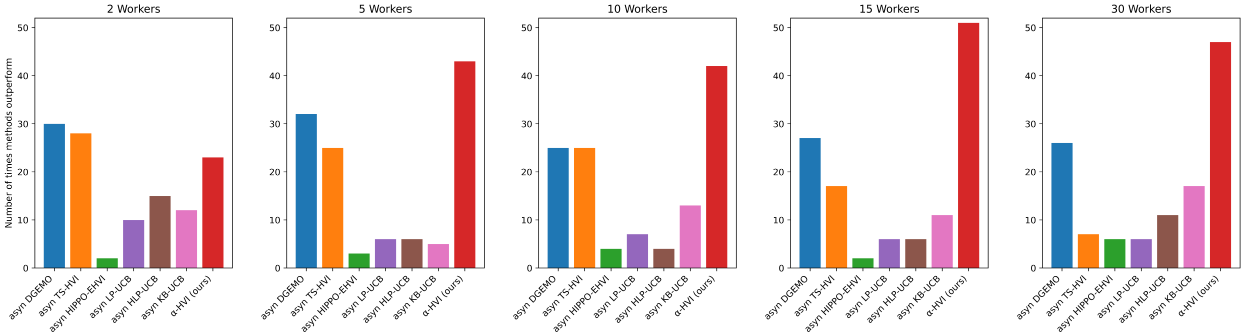

Figure 5.2: Optimization performance summary in an asynchronous setting: Bar lengths indicate the frequency with which each method surpasses others in terms of hypervolume after 200 iterations. This analysis spans a total of 12 synthetic and real-world functions and incorporates results from 10 different random seeds. The analysis assesses the performance effects of varying batch sizes, with number of workers $N$ set at 2, 5, 10, 15, 30. Notably, the results also demonstrate that the effectiveness of our strategy, $\alpha$-HVI, improves with an increasing number of workers.

**Discussion** Next, we provide a detailed discussion of the performance of the adapted baselines and $\alpha$-HVI in the context of asynchronous MOBO evaluations. This discussion focuses on understanding the strengths and weaknesses of each adapted method.

- **Thompson Sampling:** Multi-objective asynchronous TS inherits both the advantages and disadvantages associated with the TS algorithm, particularly in terms of scalability when applied to high-dimensional spaces. One notable advantage of TS is its natural adaptability to the asynchronous MOBO settings, positioning it as a suitable choice for such scenarios.

- **Kriging Believer:** KB relies on the predictions of the surrogate's posterior and performs particularly well in scenarios where uncertainty values are minimal and predictions closely approximate the actual values.

- **Local Penalization and Hard Local Penalization:** The weakness of LP and HLP approaches stems from its reliance on definitions and techniques originally developed for single-objective optimization, where the primary aim is to locate the global minimum. Consequently, penalization regions are constructed with a sole focus on this purpose.

  However, in the context of multi-objective optimization, the goal shifts from the independent optimization of each objective to the identification of points along the Pareto front. Consequently, this approach tends to generate solutions that cluster near the elbows of the Pareto front, where objectives are optimized independently, rather than effectively capturing the entire Pareto front. This can be observed in DTLZ2 problem's performance space plots in Figure 5.3.

- **Highly Parallelizable Pareto Optimization:** The EHVI-HIPPO combination is particularly effective in DTLZ problems but its performance does not consistently reach the same level of effectiveness across other types of problems, indicating some limitations in its general applicability.

- **Diversity-Guided Efficient Multi-Objective Optimization:** The asynchronous version of DGEMO excels in providing diverse Pareto fronts. Please refer to Figure 5.3 for performance space plot for DTLZ2 problem, showing a widespread discovery of the front. DGEMO's adaptability to the asynchronous setting is a promising feature, making it well-suited for such scenarios.

- **$\alpha$-HVI:** Our analysis of the HVI in Fig. 5.1 and rank plots in Fig. 5.2 highlights the stability of our approach. For most of the problems, $\alpha$-HVI consistently exhibits comparable or superior performance, validating its efficacy in asynchronous MOBO scenarios.

  It is also worth noting that the performance of our approach increases with increasing number of workers. We attribute this to one of our essential considerations: batch size awareness. As the batch size increases, our selection strategy effectively allocates more workers along the front, ensuring a good spread. This optimization leads to more uniform exploration and a better-discovered Pareto front, consequently resulting in a higher HV metric.

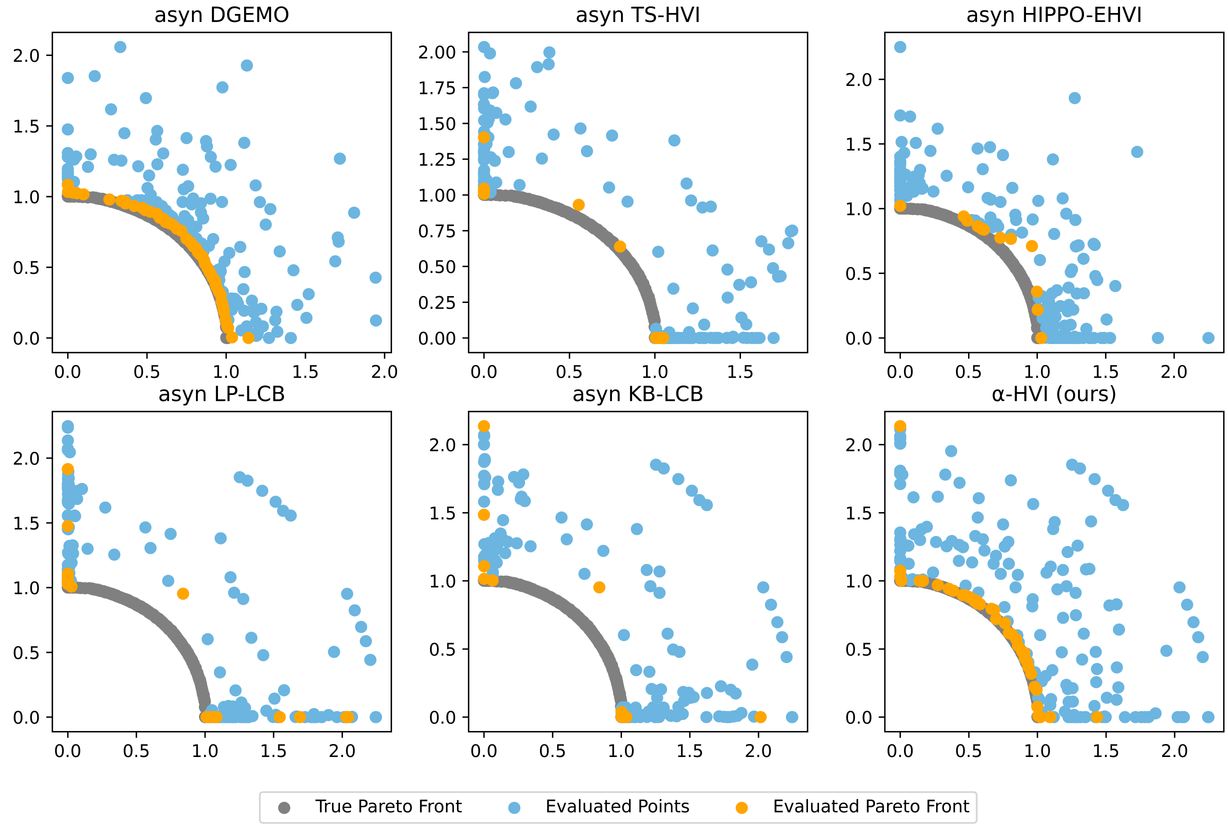

Figure 5.3: Performance space plots of evaluated points by different algorithms on the standard DTLZ2 problem, represented in blue and orange.

**Ablation study** We conducted an ablation study to evaluate the effectiveness of different acquisition functions when integrated with our proposed selection strategy, as detailed in 4.2. This study focused on comparing the LCB acquisition function, our primary choice, with Expected Improvement (EI) and Probability of Improvement (PI). Due to the specific nature of EI and PI acquisition functions, we utilized predictive mean values from the surrogate model instead of acquisition values for the hypervolume calculations. The results of this comparison, illustrated in Figure 5.4, demonstrated that LCB led to more favorable outcomes. The effectiveness of LCB is attributed to its consideration of both the mean and variance in predictions during the selection stage, which is crucial for effectively balancing exploration and exploitation in these optimization environments.

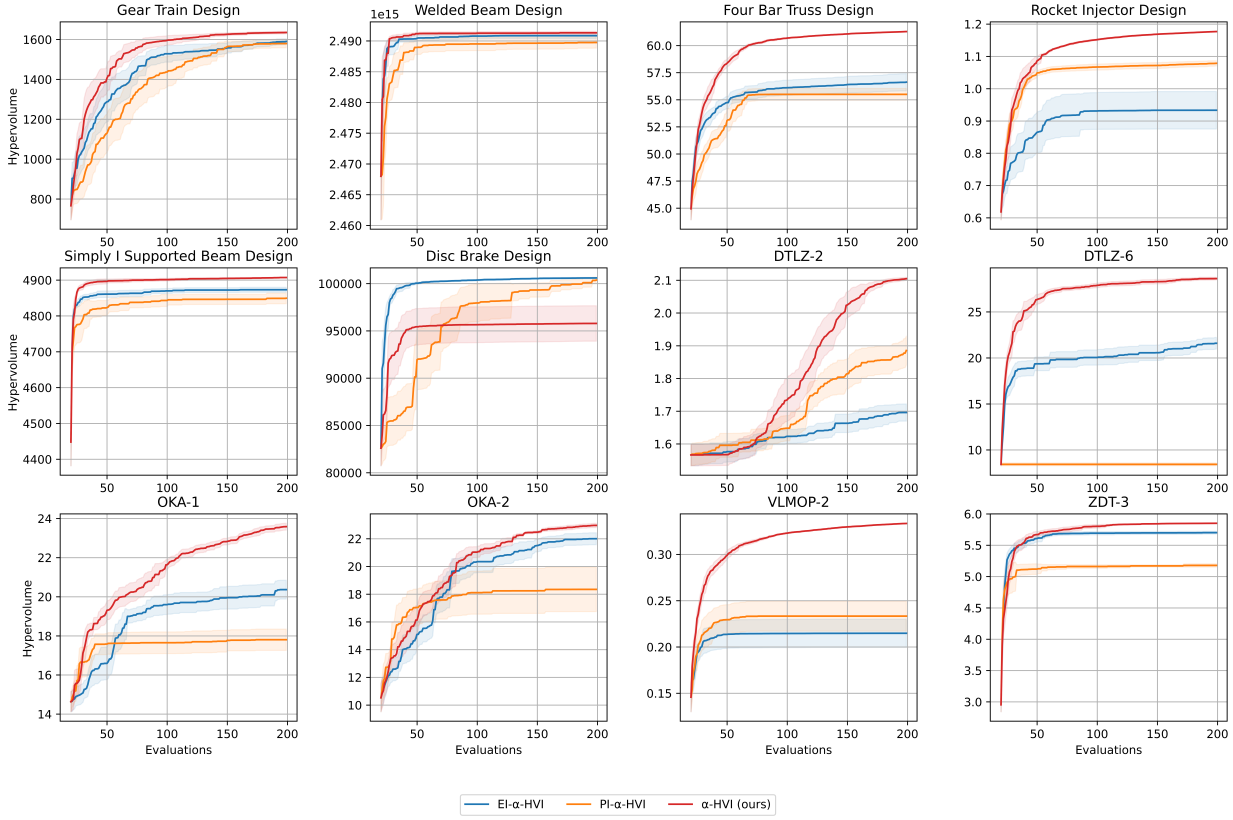

Figure 5.4: Ablation study comparing our algorithm, $\alpha$-HVI, using the LCB acquisition function with asynchronous methods employing PI and EI, across a suite of 6 real-world problems and 6 synthetic test functions. These experiments were conducted under asynchronous conditions with 15 workers, involving up to 200 function evaluations, starting with an initial set of 20 samples generated via LHS. The hypervolume indicator is displayed relative to the number of function evaluations. Each curve represents the average outcome across 10 different random seeds, with variance indicated by the shaded regions. Our approach, $\alpha$-HVI using LCB, demonstrates superior and robust performance.

### 5.4.2 Synchronous comparison

We also conduct a comprehensive analysis by comparing our asynchronous $\alpha$-HVI approach with synchronous batch MOBO baselines, including DGEMO, TS-HVI, HIPPO-EHVI, and synchronous batch LCB-HVI.

Figure 5.5 shows hypervolume plots with respect to time steps. These experiments are conducted using 15 workers over a 300-time-unit duration, and the results are averaged over 10 different random seeds.

**Discussion** While DGEMO and HIPPO-EHVI are particularly effective in addressing DTLZ problems, our asynchronous $\alpha$-HVI method demonstrates consistently superior performance across a broader range of problem contexts. This superior performance is attributed to the $\alpha$-HVI selection strategy and the inherent efficiency of asynchronous evaluations, which allow for more evaluations to be conducted within the same timeframe.

It is also worth noting the comparison of our asynchronous $\alpha$-HVI to its synchronous counterpart using a naive LCB+HVI selection. Our asynchronous approach shows superior performance in all comparisons with this synchronous method, as it is able to conduct more evaluations and better allocate resources, resulting in more effectively explored Pareto fronts.

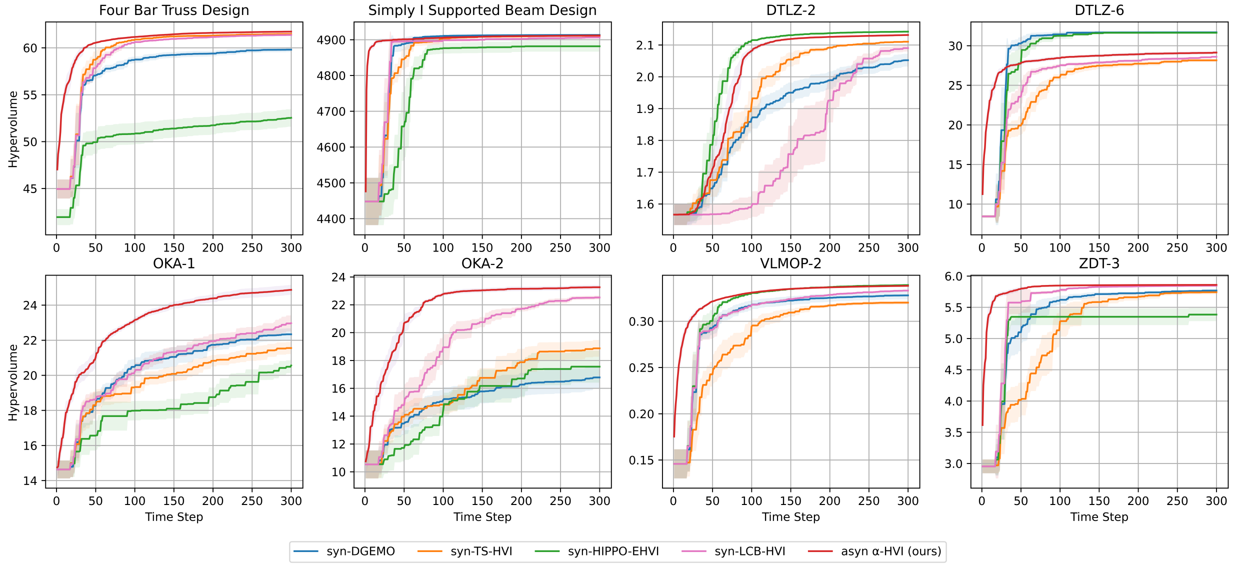

Figure 5.5: Comparison of baseline synchronous evaluations with our asynchronous $\alpha$-HVI method on synthetic test functions and real-world problems. The experiments utilize 15 workers and are conducted over 300 time units. The hypervolume indicator is shown relative to time. Each curve represents the average outcome across 10 different random seeds, with variance indicated by the shaded regions.

# Chapter 6

# Conclusions and Future Work

## 6.1 Conclusions

In this thesis, we explored asynchronous multi-objective Bayesian optimization, an area that has previously received limited attention in the literature. Initially, we adapted existing methods to asynchronous MOBO scenarios. This adaptation not only facilitated benchmarking the performance of these methods but also enabled a comprehensive assessment of their functionality and efficiency under asynchronous conditions.

Additionally, we introduced a novel strategy, $\alpha$-HVI, which is grounded in three essential considerations: uncertainty, diversity of the Pareto front, and batch size awareness. This strategy begins by fitting a Gaussian Process surrogate model using the dataset of evaluations, then applies the Lower Confidence Bound acquisition function independently for each objective. It uses the NSGA-II optimizer to generate a set of candidate points, and finally, makes the selection by leveraging a variant of the hypervolume improvement metric, which we call $\alpha$-HVI, that includes the acquisition values of points currently under evaluation alongside the evaluated Pareto front. This straightforward yet effective method has proven adept at managing the complexities of asynchronous MOBO. Through empirical evaluations conducted on both synthetic and real-world test functions, $\alpha$-HVI consistently demonstrated robust performance, often ranking among the top compared to other adapted methods.

## 6.2 Future work

The research presented in this thesis establishes a solid foundation for further investigation and development within the field of asynchronous MOBO. Given the practical nature of the problems addressed, characterized by multiple objectives and varying evaluation times, it would be particularly beneficial to explore the integration of this methodology with other practical techniques. For instance, incorporating constraint handling capabilities or multi-fidelity modeling (a combination of high-accuracy, high-cost evaluations and lower-accuracy, lower-cost evaluations) could extend the applicability of these methods to a broader range of real-world scenarios.

# References

[1] B. Kaczmarski, D. E. Moulton, A. Goriely, and E. Kuhl, “Bayesian design optimization of biomimetic soft actuators,” *Computer Methods in Applied Mechanics and Engineering*, vol. 408, p. 115 939, 2023.

[2] Y. Zhang, D. W. Apley, and W. Chen, “Bayesian optimization for materials design with mixed quantitative and qualitative variables,” *Scientific reports*, vol. 10, no. 1, p. 4924, 2020.

[3] K. Wang and A. W. Dowling, “Bayesian optimization for chemical products and functional materials,” *Current Opinion in Chemical Engineering*, vol. 36, p. 100 728, 2022.

[4] D. M. Negoescu, P. I. Frazier, and W. B. Powell, “The knowledge-gradient algorithm for sequencing experiments in drug discovery,” *INFORMS Journal on Computing*, vol. 23, no. 3, pp. 346–363, 2011.

[5] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, “Hyperparameter optimization for machine learning models based on bayesian optimization,” *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, 2019.

[6] J. Lou, R. Chen, J. Liu, Y. Bao, Y. You, and Z. Chen, “Aerodynamic optimization of airfoil based on deep reinforcement learning,” *Physics of Fluids*, vol. 35, no. 3, 2023.

[7] J. A. Vazquez–Castillo, J. A. Venegas–Sánchez, J. G. Segovia–Hernández, H. Hernández-Escoto, S. Hernandez, C. Gutiérrez–Antonio, and A. Briones–Ramírez, “Design and optimization, using genetic algorithms, of intensified distillation systems for a class of

quaternary mixtures," *Computers & Chemical Engineering*, vol. 33, no. 11, pp. 1841–1850, 2009.

[8] T. J. Lane, D. Shukla, K. A. Beauchamp, and V. S. Pande, "To milliseconds and beyond: Challenges in the simulation of protein folding," *Current opinion in structural biology*, vol. 23, no. 1, pp. 58–65, 2013.

[9] K. Miettinen, *Nonlinear multiobjective optimization.* Springer Science & Business Media, 1999, vol. 12.

[10] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *IEEE transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[11] C. E. Rasmussen, C. K. Williams, *et al.*, *Gaussian processes for machine learning.* Springer, 2006, vol. 1.

[12] K. Miettinen, *Nonlinear multiobjective optimization.* Springer Science & Business Media, 1999, vol. 12.

[13] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," *ICML'10: Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010.

[14] H. J. Kushner, "A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise," 1964.

[15] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3-4, pp. 285–294, 1933.

[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[17] Q. Zhang and H. Li, “Moea/d: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[18] S. Belakaria, A. Deshwal, N. K. Jayakodi, and J. R. Doppa, “Uncertainty-aware search framework for multi-objective bayesian optimization,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 10 044–10 052.

[19] M. Emmerich, “Single-and multi-objective evolutionary design optimization assisted by gaussian random field metamodels,” *University of Dormund*, 2005.

[20] K. Kandasamy, A. Krishnamurthy, J. Schneider, and B. Póczos, “Parallelised bayesian optimisation via thompson sampling,” in *International conference on artificial intelligence and statistics*, PMLR, 2018, pp. 133–142.

[21] E. Bradford, A. M. Schweidtmann, and A. Lapkin, “Efficient multiobjective optimization employing gaussian processes, spectral sampling and a genetic algorithm,” *Journal of global optimization*, vol. 71, no. 2, pp. 407–438, 2018.

[22] D. Ginsbourger, R. Le Riche, and L. Carraro, “Kriging is well-suited to parallelize optimization,” in *Computational intelligence in expensive optimization problems*, Springer, 2010, pp. 131–162.

[23] J. González, Z. Dai, P. Hennig, and N. Lawrence, “Batch bayesian optimization via local penalization,” in *Artificial intelligence and statistics*, PMLR, 2016, pp. 648–657.

[24] A. S. Alvi, B. Ru, J. Calliess, S. J. Roberts, and M. A. Osborne, “Asynchronous batch bayesian optimisation with improved local penalisation,” *arXiv preprint arXiv:1901.10452*, 2019.

[25] A. Paleyes, H. B. Moss, V. Picheny, P. Zulawski, and F. Newman, “A penalisation method for batch multi-objective bayesian optimisation with application in heat exchanger design,” *ICML2022 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2022.

[26] M. Konakovic Lukovic, Y. Tian, and W. Matusik, "Diversity-guided multi-objective bayesian optimization with batch evaluations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 708–17 720, 2020.

[27] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, pp. 235–256, 2002.

[28] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary multiobjective optimization: theoretical advances and applications*, Springer, 2005, pp. 105–145.

[29] T. Okabe, Y. Jin, M. Olhofer, and B. Sendhoff, "On test functions for evolutionary multi-objective optimization," in *International Conference on Parallel Problem Solving from Nature*, Springer, 2004, pp. 792–802.

[30] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithm test suites," in *Proceedings of the 1999 ACM symposium on Applied computing*, 1999, pp. 351–357.

[31] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.

[32] K. Deb and A. Srinivasan, "Innovization: Innovating design principles through optimization," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 2006, pp. 1629–1636.

[33] T. Ray and K. Liew, "A swarm metaphor for multiobjective design optimization," *Engineering optimization*, vol. 34, no. 2, pp. 141–153, 2002.

[34] F. Cheng and X. Li, "Generalized center method for multiobjective engineering optimization," *Engineering Optimization*, vol. 31, no. 5, pp. 641–661, 1999.

[35] R. Vaidyanathan, K. Tucker, N. Papila, and W. Shyy, "Cfd-based design optimization for single element rocket injector," in *41st Aerospace sciences meeting and exhibit*, 2003, p. 296.

[36] H.-Z. Huang, Y.-K. Gu, and X. Du, "An interactive fuzzy multi-objective optimization method for engineering design," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 5, pp. 451–460, 2006.

[37] Y. Tian, P. V. Konakovic, B. Li, A. Zuniga, M. Foshey, T. Erps, W. Matusik, and M. K. Lukovic, "Autodex: Automated optimal design of experiments platform with data-and time-efficient multi-objective optimization," in *NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2023.

[38] J. Blank and K. Deb, "Pymoo: Multi-objective optimization in python," *Ieee access*, vol. 8, pp. 89 497–89 509, 2020.

[39] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[40] N. Riquelme, C. Von Lücken, and B. Baran, "Performance metrics in multi-objective optimization," in *2015 Latin American computing conference (CLEI)*, IEEE, 2015, pp. 1–11.