Bayesian Bilevel Optimization

Vedat Dogan
119222557

Thesis submitted for the degree of
Doctor of Philosophy

NATIONAL UNIVERSITY OF IRELAND, CORK

COLLEGE OF SCIENCE, ENGINEERING AND FOOD SCIENCE

SCHOOL OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
INSIGHT CENTRE FOR DATA ANALYTICS

September 2023

Head of School: Professor Utz Roedig

Supervisors: Dr. Steven Prestwich
Dr. Nic Wilson

Contents

List of Figures

List of Tables

I, Vedat Dogan, certify that this thesis is my own work and has not been submitted for another degree at University College Cork or elsewhere.

Vedat Dogan

To those who accompanied me throughout this process with special mention to my family for their continuous support.

Acknowledgements

Firstly, I wish to express my eternal gratitude to my main supervisor Steven Prestwich for the invaluable opportunity he offered me, his time, patience, and continuous support of my study. PhD has been a life-changing experience, and it would not be possible without his guidance. I truthfully thank him for being so supportive and encouraging. I also would like to thank my advisor Nic Wilson for his support and enlightening discussions during this journey.

I have thoroughly enjoyed my time at the Insight Research Centre of Data Analytics and Confirm Smart Manufacturing Centre. I would like to sincerely thank Barry O'Sullivan for widening my research perspective for his guidance and moreover for his support besides the hard work in providing such opportunities for us. I sincerely thank all the help provided by the administrative staff Caitriona, Eleanor, and Linda. I feel privileged to have worked alongside such professionals, many of whom I am happy to call my lifelong friends.

I have met lots of bright people during this journey. We shared this experience with them alongside interesting discussions, challenges, lunches, gatherings, travels, barbeques and many more. I would love to call many of them a friend more than a colleague and thanks to them for sharing this experience with me during these years.

Hazal and Jeremy, we met in Ireland and they became my little family. We had an amazing and incredible time together, and I am sure we will be there for each other forever. Especially, I would like to thank Hazal for her invaluable support, several hours-long talks (about anything), her trust in me and her great friendship. Whenever I needed her, she was there for me and I believe this achievement would not be the same without her. We enjoyed a lot of barbeques, trips, karaoke nights, picnics and dinners together. I am grateful to have such a friend in my life.

Begum Genc deserves special thanks. I feel lucky and am grateful to have met with her on the first day in Ireland. I owe her a lot for her guidance, our discussions and support, and I will be grateful to her forever. I felt her support always with me whenever I felt overwhelmed and stuck. She is an amazing friend and a brilliant person, and this achievement would not be the same without her. Also, the Little Italy crew (Andrea, Diego and Federico), are bright people that I met at the beginning of my PhD journey in Ireland. I would like to thank them for being a part of this period of my life with their support and friendship.

My family's support has been invaluable in my life, and of course during this journey. I especially would like to thank my brother, Sedat, for being my support all of my life and for being there always. I feel extremely lucky to have him as a brother and it gives a considerable relief to me knowing that he will be there forever. I cannot express my gratitude to my Father, Izzet, who was my constant support at each stage of my life. To my mother, Halime, thank you for being there always for me and being sure that I have everything I need. I learn from them a lot and I will be grateful all of my life for it.

I wholeheartedly thank Julie and I would like to say that her support was always with me during this time. She was always there for me whenever I need and it will be invaluable for me. It is hard to express all my feelings here, but I just want to say that it would not be the same without her support. I am grateful to my true friends Samet, Enes and Zelal for to being with me during this journey. Samet deserves special thanks here, he was a friend who encouraged me during the most challenging times. I feel lucky to have them and feel grateful to know I will have them always.

This dissertation would not have been possible without the financial support of the Science Foundation Ireland Grant No. 16/RC/3918 which is co-funded under the European Regional Development Fund.

Vedat Dogan
Cork, September 2023

Abstract

In this dissertation, we focus on improving bilevel optimization through several approaches developed during research. Bilevel optimization problems consist of upper-level and lower-level optimization problems connected hierarchically. Upper-level and lower-level problems are also referred to as the leader and the follower problems in the literature. The leader must solve a constrained optimization problem in which some decisions are made by the follower. Both have their objectives and constraints. The follower's problem appears as a constraint of the leader. Such problems are used to model practical applications in real life where authority is realizable only if the corresponding follower objective is optimum. There are several practical applications in the fields of engineering, environmental economics, defence industry, transportation and logistics that have nested structures that are fitted to these types of modelling. As the complexity and the size of such problems have increased over the years, the design of efficient algorithms for bilevel optimization problems has become a critical issue and an important research topic.

The nested structure of bilevel optimization problems comes with several interesting challenges to the problem of algorithm design. Naturally, the problem requires an optimization problem at the lower level for each upper-level solution. That makes the problem computationally expensive. Also, an upper-level solution is considered feasible only if the corresponding lower-level solutions are the global optimum of its objective. Therefore, not accurate lower-level solutions might lead to an objective value that is better than the true optimum at the upper level. It comes with another challenge for the selection strategies of solutions and naturally the performance including the computational effort.

There are several approaches proposed in the literature for solving bilevel problems. Most focus on special cases, for example, a large set of exact methods was introduced to solve small linear bilevel optimization problems. Another approach is to replace the lower-level problem with its Karush-Kuhn-Tucker conditions, reducing the bilevel problem to a single-level optimization problem. A popular approach is to use nested evolutionary search to explore both parts of the problem, but this is computationally expensive and does not scale well. The works presented in this dissertation are directed to improve the bilevel optimization process in terms of accuracy and required function evaluations by developing and implementing the black-box approach to the upper-level problem. First, we focused on extracting knowledge of previous iterations and conditioned lower-level decisions to improve upper-level search. Then, we attempt to improve upper-level search by using the multi-objective acquisition function in the Bayesian optimization process and use the benefit of multi-objective optimization literature for using different search strategies at the upper-level search. After, the multi-objective bilevel optimization problems are investigated and a Bayesian approach is developed to approximate the Pareto-optimal front of the problem. Both single and multi-objective optimization problems are investigated in this dissertation. The experiments are conducted on a comprehensive suite of mathematical test problems that are available in the literature as well as some real-world problems. The performance is compared with existing methods. It is observed that the proposed approaches achieve a promising balance between accuracy and computational expanse, therefore it is suitable for applying the proposed approaches to several real-life applications in different fields.

Chapter 1

Introduction

1.1 Motivation

Many practical problems in applied science, economics, statistics and engineering are posed in terms of optimization. It is very often that the objectives of real-world optimization problems are outputs of complex systems where the interactions between the collaborative sub-systems limit the feasible set of the problem. For instance, let us consider an example from the area of transportation.

Assume that a government operates a large network of highways or a private company wants to optimize the tolls to maximize its earnings. A high amount of toll would discourage users from using the network and it results in decreasing revenue due to the low volume. A low amount of toll would increase the volume but it might not be optimal for maximizing the earnings. The government or a private company that wants to maximize earnings, can not write its optimization problem without considering the highway users under such situations. When we consider the highway users for maximizing the revenue of the government, the two-level optimization problem that is hierarchical needs to be written. The government needs to construct its optimization problem for maximizing the revenue with an extra constraint represented by the highway user's optimization problem. It consists of an objective for minimizing the generalized cost for highway users. It might contain the travel distance, toll amount or travel time. In this way, the government can examine the response of highway users to any amount of toll during the optimization process. In this problem, the upper-level is the government which can be referred to as a leader and the highway users can be referred to as followers which is a constraint of the government problem in the lower-level. This problem is known as *a bilevel problem* in mathematical optimization.

In particular, such problems can be characterized by several objective functions that have to be optimized at the same time. They are called *multi-objective problems*. When we consider the example above, a highway user may want to simultaneously minimize energy consumption and maximize comfort at the lower-level while the government might want to minimize air pollution while maximizing revenue. Generally, there is no identical optimum solution. In this kind of problem, the goal is to approximate the optimal compromises which are called Pareto points. The set of the Pareto points is called *the Pareto set*.

Another motivating example is in the context of homeland security [Low06]. The mission of homeland security is to ensure that the country is secure, safe and resilient against terrorism and other attacks. While a government is doing that, it needs to be achieved cost-effectively and to try not to sacrifice ease of customs operations for the public. So multiple objectives need to be considered to solve this problem. One of the ways is to make informed guesses on the previous actions of the attackers, which comes at the lower-level in the bilevel optimization structure. Therefore, the government as an upper-level decision-maker, needs to model the attacker's behaviour during the decision-making to prevent an attack. The problems with multiple objectives on one or both upper and lower-levels are called *multi-objective bilevel problems*. The work presented in this dissertation proposes new approaches for solving single and multi-objective bilevel problems.

The main goal of this dissertation is to improve the algorithmic performance of single and multi-objective bilevel optimization problems in terms of computational cost and accuracy. These kinds of problems have a significant role in application in real life, especially considering their nested nature. Some of them are discussed above and more can be found in [SMD17b]. A broad range of works presented in the literature handle these problems with linear and quadratic nature by using exact mathematical programming techniques [Dem02]. However, they generally need some assumptions in their mathematical properties such as objective linearity, convexity, differentiability, etc. Not requiring mathematical properties of the problems are particularly important for many real-world applications as it may be needed some black-box computer simulations to evaluate the objectives. Considering this, there are some approaches developed over the years that have been reviewed in [Dem20]. They tend to use hybrid methods including surrogate models and evolutionary algorithms combined with other exact strategies. Even though metaheuristics such as evolutionary algorithms are successful in solving bilevel optimization problems, the number of function evaluations used to solve the optimization problem is relatively high. In particular, population-based approaches require a large number of function evaluations during the search coming from the nature of the algorithms.

The presented work in this dissertation tackles this obstacle in the literature and contributes towards this ongoing research and development to improve algorithmic performance by using a black-box approach at the upper-level in the bilevel optimization problems. In Section 1.2, we present the scope of the research of this dissertation.

1.2 Scope of Research

Several kinds of bilevel problems are possible depending on the number and the nature of the objective functions such as single-objective, multi-objective, optimistic/pessimistic approaches, etc. Each cluster of bilevel problems may have a different solution method than the others. Within the scope of this dissertation, the following characteristics have been assumed.

- This dissertation focused on single or multi-objectives at both levels. Also, mainly the variables are considered to be real-valued rather than discrete. However, the active research of applied methods in this dissertation makes it straightforward to extend the work for discrete variables.
- All objective functions of upper-level problems are assumed to be black-box, meaning there are no apriori known properties of the problems used to inform the search, such as linearity, convexity, differentiability, etc. It enables us to deal with a much wider range of problems considering analytical algorithms which might need some assumptions or conditions.
- The computational effort is measured using only the number of evaluations for both single and multi-objective bilevel problems. As bilevel problems are computationally expensive, it is a common method to measure the computational cost. Fitness score is the main accuracy metric for single-objective bilevel problems while hypervolume improvement and in-

verted generational distance are for multi-objective bilevel problems.

1.3 Thesis Statement and Contributions

In this dissertation, we focused on introducing advanced black-box approaches by Bayesian optimization for solving single and multi-objective bilevel problems. We stated the thesis that is defended in this dissertation and presented our approach. Also, we provided a list of our supporting publications for each point.

Thesis. *Bilevel problems are widely studied hierarchical optimization problems that require a high number of computations. Several classical and hybrid approaches have been developed over the years. However, the required assumptions are generally strict and computational cost is high. Therefore, the need is to consider computationally effective approaches to solve these problems and to avoid the necessary assumptions such as linearity, convexity, differentiability, etc. We claim that the black-box approach at the upper-level problems achieves relatively better optimal results with less computational cost. We developed and proposed advanced Bayesian optimization-based algorithms for solving the upper-level problem that have not been considered in this field before. In this way, the decision-makers do not need to know the properties of the true objective function and can deal with objective uncertainty.*

Discussion. In this dissertation, first, we focused on single-objective optimization problems. In chapter 3, we introduced a conditional Bayesian optimization algorithm for bilevel optimizations, called ConBaBo. It uses the previous lower-level decisions as a condition to improve upper-level search during the optimization process. The approach is inspired by previous work on conditional optimization [PKG20] and standard Bayesian optimization for bilevel problems [KDBN17]. The first appearance of our work related to the proposal of the novel concept can be found in the following conference publication:

> *Vedat Dogan, Steven Prestwich. 2023. Bilevel Optimization by Conditional Bayesian Optimization. In Machine Learning, Optimization, and Data Science: 8th International Conference, LOD 2023, Grasmere, Lake District, UK, September 22–26, 2023.*

Then, as ConBaBo succeeded in solving bilevel problems while improving the computational performance, we aimed to improve algorithmic performance by multi-objective acquisition function approach and presented the MACBP algorithm in Chapter 4. The details of the presented approach can be found in the following conference publication:

> *Dogan, V., Prestwich, S. (2022). Bayesian Optimization with Multi-objective Acquisition Function for Bilevel Problems. In: Longo, L., O'Reilly, R. (eds) Artificial Intelligence and Cognitive Science. AICS 2022. Communications in Computer and Information Science, vol 1662. Springer, Cham. https://doi.org/10.1007/978-3-031-26438-2_32*

After that, we tackled solving multi-objective bilevel optimization problems that have multiple objectives at both levels. As the nature of the problem seeks the optimum for multiple objectives, generally there is no single optimum solution. Therefore, it is expected to have an optimum solution set instead of a single optimum solution. It is called the Pareto-optimum front. The Pareto-optimum front is the solution set which contains the non-dominated solutions for the problem. In Chapter 5, we proposed a Bayesian approach for solving multi-objective bilevel problems and attempted to approximate upper-level Pareto-optimal front by using specifically designed acquisition functions for multi-objective problems. Subsequently, we performed some experiments and investigated the algorithmic performance including the success of approximation to the Pareto-optimal front. We concluded our work by investigating the objective uncertainty in single- and multi-objective bilevel problems.

After each decision in an optimization problem, it is highly possible to have unexpected effects that might change our objective values. As bilevel optimization is a hierarchical decision-making process and the upper-level decision affects the lower-level optimization problem naturally, it is important to deal with the uncertainty during the optimization. We presented an algorithm called BamBiNo to tackle bilevel multi-objective optimization problems in such uncertain environments and presented the findings in Chapter 6. Also, the details can be found in the following workshop publications:

- *Dogan, V., Prestwich, S. (2023). A Batch Bayesian Approach for Bilevel Multi-Objective Decision Making Under Uncertainty. AAAI'23 Workshop on Uncertainty Reasoning and Quantification in Decision Making, Washington, USA*
- *Dogan, V., Prestwich, S. (2023). Multi-objective Bilevel Decision*

Making with Noisy Objectives: A Batch Bayesian Approach. Multi-objective Decision Making Workshop (MODeM2023), ECAI 2023. Krakow, Poland

1.4 Overview of the Dissertation

This dissertation consists of six main chapters. We present briefly the structure and the contents of each chapter below.

We presented the necessary background information in Chapter 2 to understand the concepts and correctly follow the notation used in this dissertation. The primary purpose of this chapter is to define all the concepts in detail. Also, we provided examples for the reader to help them understand the concepts before moving to the technical part of the dissertation. We started with an introduction to the mathematical structure of Bayesian optimization and Gaussian processes. We also presented the kernels and the acquisition functions used for this dissertation. Multi-objective Bayesian optimization is presented after with specifically designed acquisition functions. Then, we formally introduced the single objective bilevel optimization with general formulations and definitions. It also includes the definitions of optimistic and pessimistic approaches considering the sake of upper-level decision-makers. Then, we presented the multi-objective bilevel optimization problem without and with decision-maker uncertainty. It contains the definitions for uncertainty at both upper and lower-level decision-makers. We also introduced some of the popular techniques that can be used to solve single and multi-objective bilevel problems. Then we discussed the techniques that are limited to the ones we are using in this dissertation.

Then we focused on a conditional Bayesian approach to general single objective bilevel problems in Chapter 3 which we call ConBaBo. We first explained conditional Bayesian optimization and how conditioning the lower-level decision maker affects the upper-level optimization. Then, we presented the empirical results of our experiments with popular standard bilevel benchmarks. The work in this chapter is focused on improving the optimization performance and accuracy for single-objective bilevel problems with a black-box approach.

In Chapter 4, we focused on improving the search during bilevel optimization by using multiple acquisition functions at once. We applied the multi-objective acquisition function approach to bilevel problems. The upper-level decision has been made from the Pareto-optimal front at each iteration and empirical results have been shared.

In Chapter 5, we presented the multi-objective bilevel problems and the relationship between upper-level and lower-level decision-makers when they have multiple conflicting objectives. Generally, there is no single optimal solution for a multi-objective problem. So here, we discussed how the decision-making from the Pareto-optimal solution set of followers is affecting the upper-level optimization process. Then, we proposed a batch Bayesian optimization algorithm for multi-objective bilevel problems. We used and compared two acquisition functions which are specifically designed for multi-objective problems. The first one is an extension of the popular expected improvement acquisition function focused on hyper-volume improvement and called expected hyper-volume improvement (EHVI) [EGN06]. The second one is a hybrid algorithm with on-line landscape approximation for expensive multi-objective optimization problems, called ParEGO [Kno06a]. Decision-makers' uncertainty arises as a big problem when we consider practical problems. We also presented another model to deal with the upper-level objective uncertainty. The model is based on the acquisition functions we mentioned above but with noisy settings, called parallel noisy expected hypervolume improvement (qNEHVI) [DBB21a]. We concluded this section by presenting some experiments to observe the relative performance of the algorithm. We used hyper-volume improvement (HV) and inverted generational distance (IGD) [CCRS04] for performance metrics which are two popular metrics to measure how good the Pareto-optimal solution set is.

In Chapter 6, we tackled the uncertainty at the upper-level objective in bilevel problems by leveraging the BoMoBo algorithm presented in Chapter 5 and we call it BamBiNo. We researched to improve the algorithmic performance of solving multi-objective bilevel optimization problems with environmental uncertainty and experiments are conducted on both theoretical and practical problems.

In Chapter 7, we discussed some general concluding remarks, and open problems and identify some future directions related to the work presented in this dissertation.

Chapter 2

Background

Abstract. *In this chapter, we present the background and notation required to present our work. First, we introduce the mathematical structure of Bayesian optimization and Gaussian processes. After that, we define the multi-objective optimization problems and Pareto solutions terminology with some metrics to measure the performance. We mentioned hypervolume improvement and the Bayesian approach to multi-objective problems. Then, we define bilevel optimization with single- and multi-objective problems at both levels as well as review a number of techniques for solving those problems. Also, we presented optimistic and pessimistic approaches and discussed some existing works.*

2.1 Bayesian Optimization

Bayesian optimization (BO) [Fra18] is a method for performing global optimization of expensive-to-evaluate black-box functions. It processes iteratively and is guided by a statistical model of the objective function. It starts with a prior distribution over the objective functions and uses a likelihood function to encode the assumptions in the observations of the objective function. Computing posterior distribution at each iteration follows the process by conditioning the previous evaluations. These evaluations can be considered observations in Bayesian nonlinear regression.

BO uses a probabilistic surrogate model, typically Gaussian Process (GP) to model the objective function based on observations [RW06]. We describe it in Section 2.1.1. The GP is assisted by an acquisition function which is not expensive-to-evaluate to evaluate compared with the true objective function [Fra18]. Thus, it is easy and requires fewer function evaluations to optimize rather than a true objective function. It is used to map the beliefs about the objective function for measuring the potential of each location in the search space. The aim is to find an input that optimizes the acquisition function and evaluate the output. We discuss more about the acquisition functions in Section 2.1.3. It follows with updating the data set with the prior information for the next iteration. Usually, the termination criteria are selected as the maximum time for the optimization process or the number of maximum function evaluations. Optimizing acquisition functions is a global optimization problem itself. The acquisition function needs to be optimized many times to solve the original optimization problem. So, BO methods are convenient only if the task of optimizing the acquisition function is much easier than optimizing the true function. The standard procedure is explained in Algorithm 1 and more detailed material and perspectives on BO can be found in [SSW$^+$16a, RW06].

Algorithm 1 Bayesian Optimization (BO)

Input: f, α, $\mathcal{X}$, $\mathcal{M}$
Output: x^*

```
while i = 0 : Budget do
    Fit the model M to the data D
    Optimize the acquisition function x' = arg max_{x∈X} α(x, M)
    Evaluate the objective function y' = f(x')
    Update the data set with new observations: D = D ∪ {(x', y')}
return The best decision x* = arg min_{x∈X} E_M[f(x)]
```

2.1.1 Gaussian Process

The GP is used to model the unknown black-box objective function in BO [Fra18]. It is also called *kriging* in the literature in honour of the South African student who developed this technique at the University of the Witwatersrand [Kri51].

Let us assume that we have a set of collection points $\{x_1, \ldots, x_n\} \in \mathbb{R}^d$ and a *vector* of objective function values of these points $\{f(x_1), \ldots, f(x_n)\}$. We suppose that the vector is drawn randomly from some prior probability distribution, and GP takes the prior distribution to be multivariate normal with a mean function $\mu(x)$ and kernel function $\kappa(x, x')$, as in Equation 2.1.

$$f(x) \sim GP(\mu(x), \kappa(x, x')) \tag{2.1}$$

Figure 2.1 shows 5 different functions from GP distribution without any observed data. There are 30 correlated points sampled.

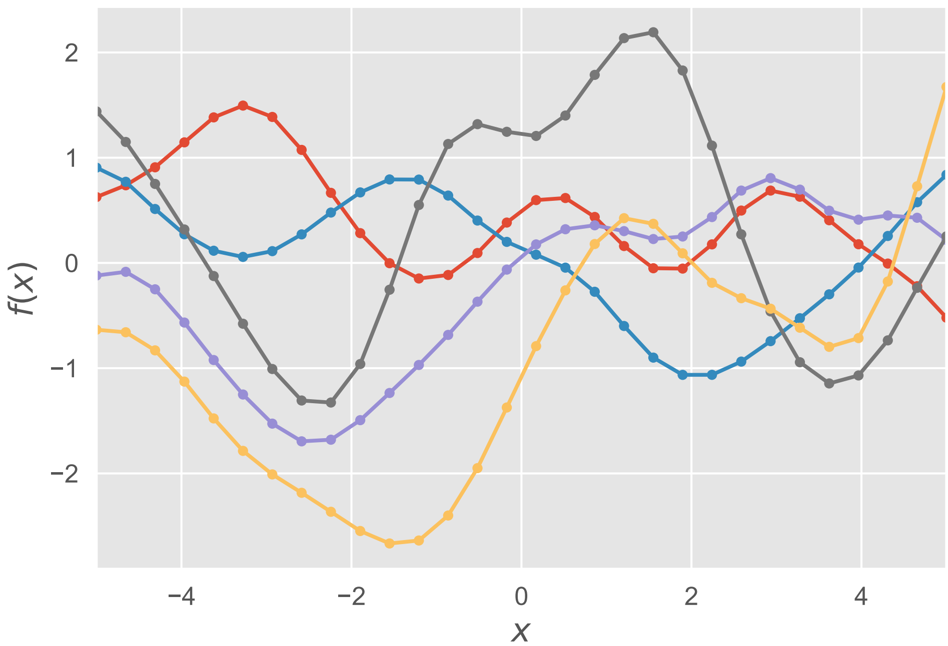


Figure 2.1: 5 different function realizations at 30 points sampled from a Gaussian process

We can compute the posterior distribution over unknown black-box objective functions with a prior and a likelihood function. For a finite number of query points, the GP makes it possible to condition observed data and compute the posterior distribution. After we observe n decision points, the mean vector is obtained by evaluating a mean function μ_0 at each decision point x_i and the covariance matrix by evaluating a covariance function or kernel κ at each pair of point x_i and x_j.

If we represent the design matrix with X with n observations, X' is a matrix containing the candidates, and $K(X, X')$ is a matrix containing the kernel function κ applied to the pair of X and X', then the predictive distribution of X' with a mean function μ and covariance matrix Σ are given by

$$\begin{aligned} &f(X) \sim \mathcal{N}(\mu(X), \Sigma(X)) \\ &\mu(X') = K(X, X')[K(X, X) + \nu^2 \mathbb{I}]^{-1} \mathbf{y} \\ &\Sigma(X') = K(X, X') - K(X', X)[K(X, X) + \nu^2 \mathbb{I}]^{-1} K(X, X') \end{aligned} \tag{2.2}$$

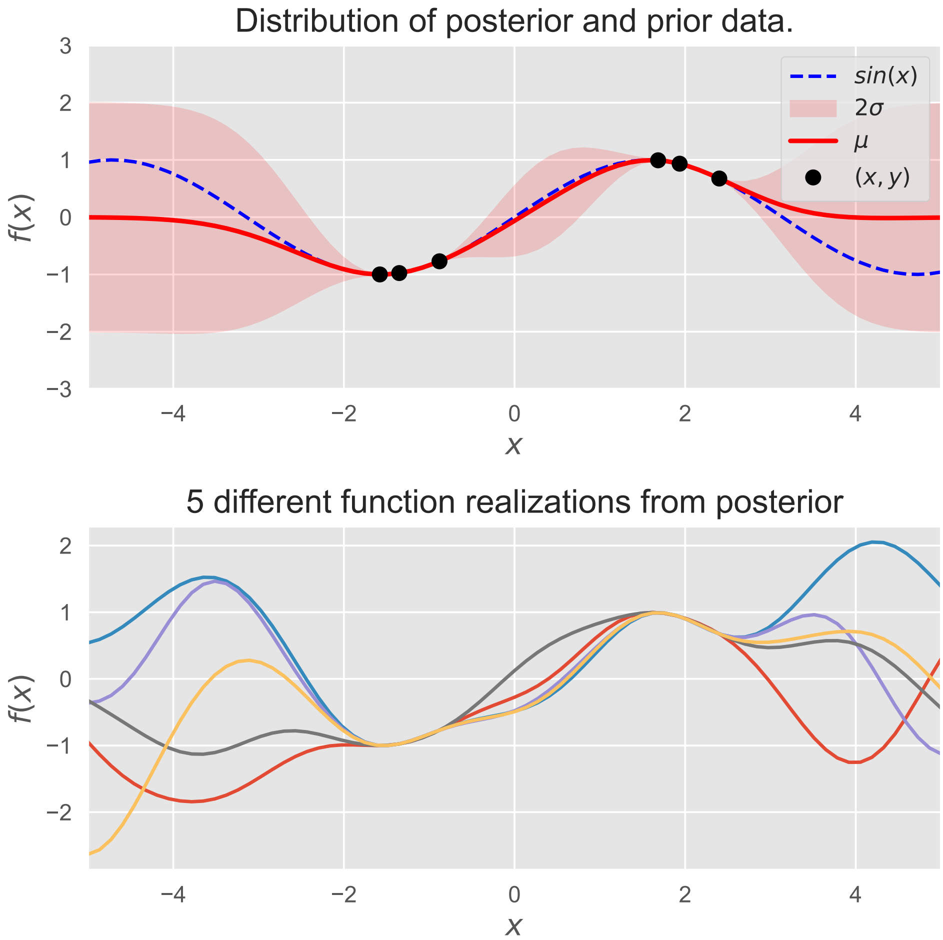


Figure 2.2: The posterior distribution after 6 observations of $\sin(e)$ function and sampled functions from distribution

where $\mathbf{y}$ is a vector of the corresponding observations X, and ν is the variance of i.i.d. Gaussian noise.

To sample from the GP we need to define the mean and covariance function that models the joint variability of the GP random variables. The covariance function is also known as the kernel function as we mentioned before and it makes it possible to set prior information on the distribution. The kernel function needs to be positive-definite to be a valid covariance function. We are going to talk about kernel functions more in Section 2.1.2.

For example, Figure 2.2 shows the posterior distribution based on 6 observations from a $\sin(e)$ function. In the top figure, the red line is the posterior mean and the shaded area is the 95% prediction interval. The black dots are observations. The prediction interval is calculated from standard deviation which is the square root of the covariance matrix. The figure on the bottom shows the 5

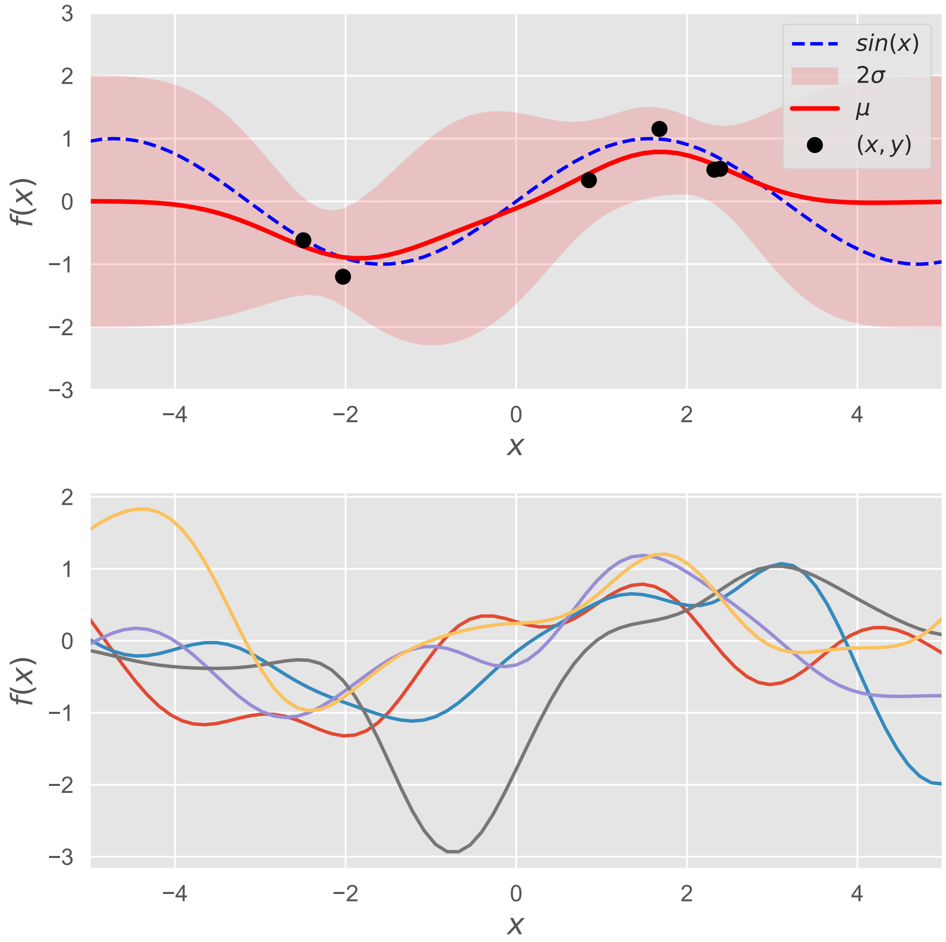


Figure 2.3: The posterior distribution after 6 *noisy* observations of sampled $\sin(e)$ functions from distribution

sampled functions from this distribution.

The observations in Figure 2.2 was noiseless. Thus, the distribution is noiseless. We can make predictions from noisy observations by modelling the noise ξ as Gaussian noise with variance σ_ξ^2. The noise can be modelled by adding it to the covariance kernel. Figure 2.3 shows the same posterior distribution based on 6 *noisy* observations from the $\sin(e)$ function. As the observations are noisy, the functions sampled do not have to go through observational points anymore.

Some properties come with the Eq. 2.2. One of them is the computational bottleneck of GP regression. The computation in Eq. 2.2 contains an inversion of $N \times N$ matrix $K(X, X) + \nu^2\mathbb{I}$ or the Cholesky decomposition which requires $\mathcal{O}(N^3)$ operations as N grows. Another one is the posterior predictive covariance matrix does not hinge on the observations. It means that it does not depend on the actual observation values, it does only the data that has been collected. The other one is a result of the multivariate normal distribution which is the marginal distribution at any decision point is univariate Gaussian. These can be calculated by using the Eq. 2.2. Thus, it leads to computing the acquisition functions efficiently in BO. For an in-depth treatment of GP, please see [RW06].

2.1.2 Kernels

A kernel or covariance function describes the covariance of the GP random variables. It defines a prior on the GP distribution. A kernel function should be *positive definite* which implies that the matrix should be symmetric, meaning *invertible*.

Definition 2.1.1. *[RW06] A real $n \times n$ matrix K which satisfies $Q(x) = x^\top Kx \geq 0$ for all vectors $x \in \mathbb{R}^n$ is called positive semidefinite. If $Q(x) = 0$ only when $x = 0$ the matrix is positive definite where $Q(x)$ is called a quadratic form.*

The two popular kernels we will focus on in this dissertation are *the squared exponential kernel* and *Matérn 5/2 kernel*. The squared exponential kernel follows the assumption that the objective function is being modelled as infinitely differentiable and formulated as follows:

$$\kappa_{SE}(x, x') = \sigma^2 \exp\Bigg(-\frac{1}{2} r^2(x, x') \Bigg) \tag{2.3}$$

where $\{l_d\}$ is length scale parameters and σ is an amplitude parameter. The Matérn 5/2 kernel is based on the assumption of the objective function being modelled twice differentiable and formulated as follows:

$$\kappa_{M52}(x, x') = \sigma^2 \Bigg(1 + \sqrt{5r^2(x, x')} + \frac{5}{3} r^2(x, x') \Bigg) \exp\Bigg(-\sqrt{5r^2(x, x')} \Bigg) \tag{2.4}$$

where $r^2(x, x') = \|M(x - x')\|_2^2$ and $M = diag(1/l_1, \ldots, 1/l_D)$ in both Eq. 2.3 and Eq. 2.4. More details can be found in [RW06] for exploring kernels in a categorised way.

There are two hyperparameters as we mentioned above: l_d and σ. It has $d + 1$ hyperparameters in d dimensions and the wrong hyperparameter selection may lead a poor overall performance. For instance, the low value of l_d may lead to a failure generalization while too large a value might lead to an overlook of the higher frequency in the true function. In Figure 2.4, we can see how

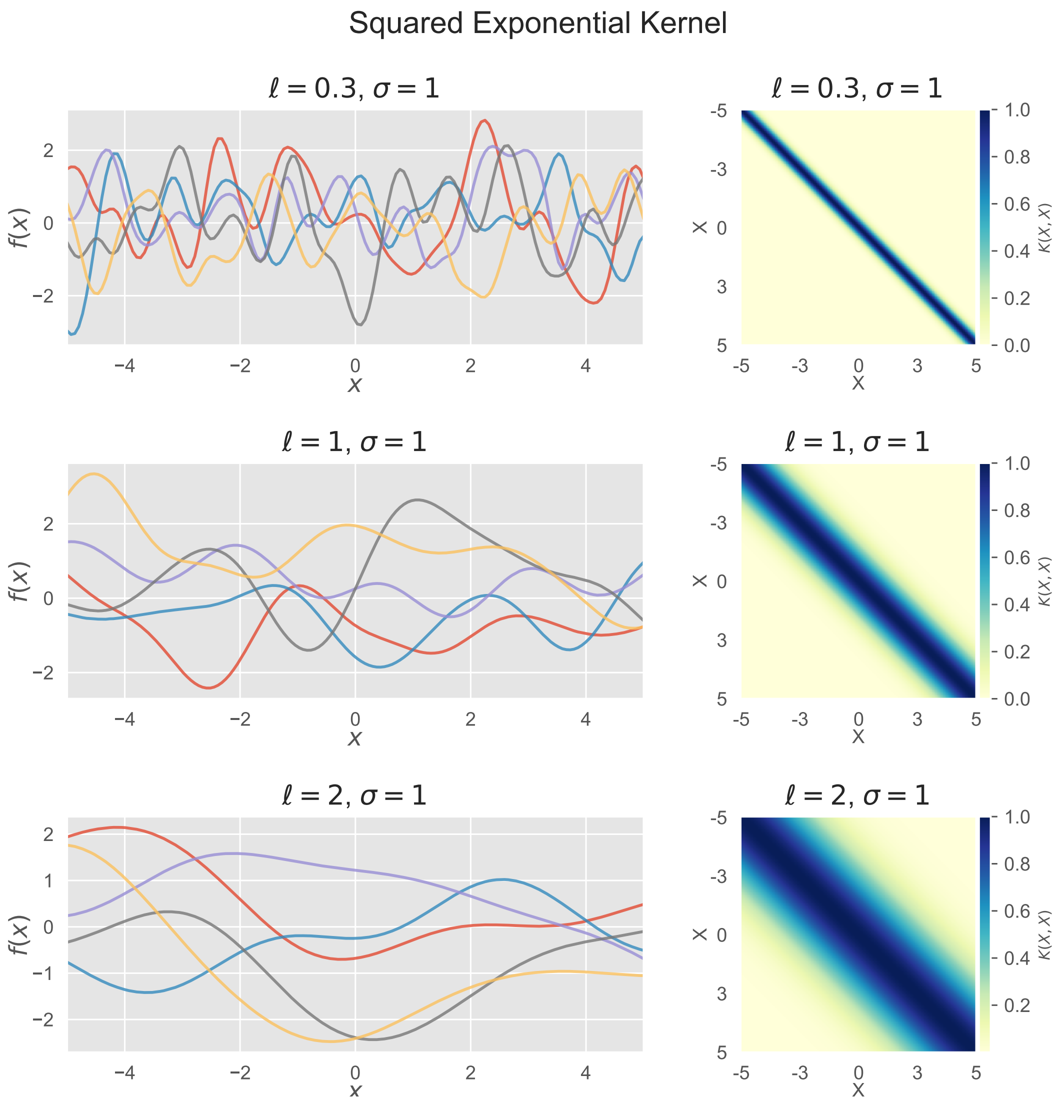


Figure 2.4: Different length scale comparison for the squared exponential kernel.

l_d parameter in squared exponential kernel changes the shape of the objective function and Figure 2.5 shows how l_d parameter affects the function shape. In both kernels, the low-length scale assumes the frequency is high, the high-length scale assumes the opposite. We can also observe the shape difference according to the kernel selection. While the function shape in the squared exponential kernel is smoother, in the Matérn $5/2$ kernel, it is rougher. More details can be found in [RW06].

2.1.3 Acquisition Functions

In this section, we describe the acquisition functions which are relevant to this thesis. Acquisition functions are designed to trade off exploration and of the

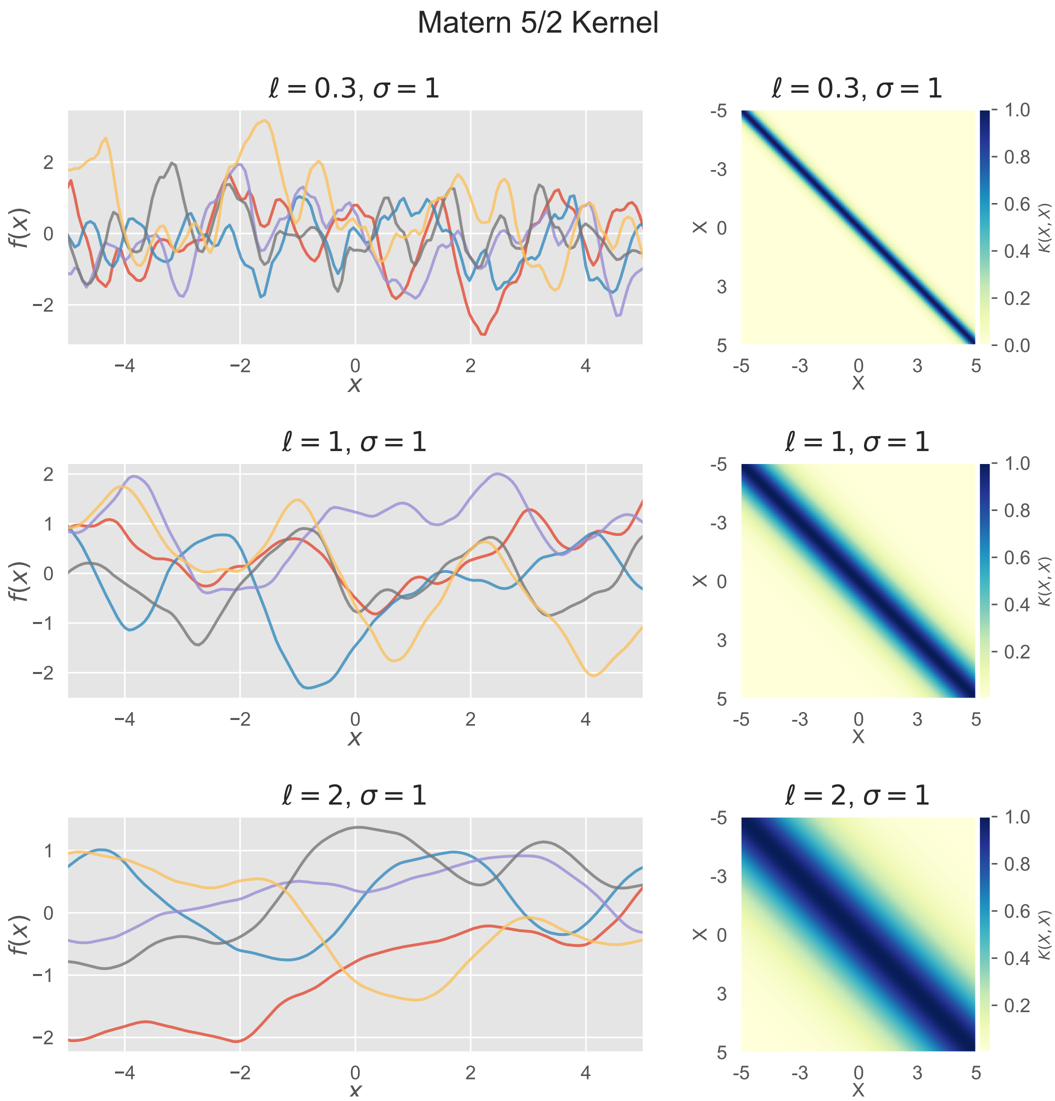


Figure 2.5: Different length scale comparison for the Matern $5/2$ kernel.

search space and exploitation of the current promising areas. We present the improvement-based and optimistic acquisition functions which are used in this dissertation.

Expected Improvement

The Expected Improvement (EI) [JSW98] acquisition function focuses on the expectation of the improvement in the function value at the current best solution. It observes the $f(x)$ for the improvement over some target τ:

$$\alpha_{EI}(x) = \mathbb{E}_y[\max(0, \tau - y)] = \int_{-\infty}^{\infty} \max(0, \tau - y)p(y|x)dy \tag{2.5}$$

When the predictive distribution under the model is Gaussian, the closed form of EI is defined as in Equation 2.6[Jon01]

$$\alpha_{EI}(x) = \sigma(x)(\lambda\Phi(\lambda) + \phi(\lambda)) \tag{2.6}$$

where $\phi(\cdot)$ is probability density function (PDF) of standard normal distribution and $\lambda = (\tau - \mu(x) - \xi)/(\sigma(x))$. $\mu(x)$ is the posterior predictive mean at x,$\sigma(x)^2$ is the posterior predictive variance at x while $\Phi(\cdot)$ is the standard normal cumulative distribution function (CDF) and $\phi(\cdot)$ is the standard normal PDF. EI is a popular acquisition function in BO literature. The theoretical properties can be found in [Bul11] and the author shows that the EI acquisition function may not converge when GP kernel hyperparameters are estimated from the observations via maximum likelihood. So, the kernel hyperparameters needed to be estimated for the convergence of EI, therefore he proposed a method for this purpose. Also, the practical effects in practice can be found in [SLA12].

Probability Improvement

When we generalize and extend the EI acquisition function in Equation 2.5 as in Equation 2.7 as follows:

$$\alpha_{EI-g}(x) = \mathbb{E}_y[\max(0, (\tau - y)^g)] = \int_{-\infty}^{\infty} \max(0, \tau - y)p(y|x)dy \tag{2.7}$$

with $g \in \mathbb{Z}^+$ which is a parameter that controls the trade-off between the global search and the local search. The larger g means the algorithm is more biased towards global search. For the consideration of the parameter $g = 0$, it returns the Probability of Improvement (POI) acquisition function, formulated in Equation 2.8 as follows:

$$\alpha_{PI}(x) = \Phi(\lambda) \tag{2.8}$$

where $\lambda = (\tau - \mu(x) - \xi)/(\sigma(x))$. The POI acquisition function tries to measure the probability that an arbitrary x exceeds the current best and it is a known fact that it tends towards local search, as can be found in more detail in [BCF10].

Upper Confidence Bound

Another popular acquisition function is the Gaussian Process Upper Confidence Bound (GP-UCB) acquisition function [SKKS10]. It balances the local and global search by constructing the regions with low posterior mean and high posterior variance. It is defined as in Equation 2.9

$$\alpha_{UCB}(x) = \mu(x) + \beta^{\frac{1}{2}}\sigma(x) \tag{2.9}$$

where $\mu(\cdot)$ is posterior predictive mean and $\sigma(\cdot)$ is variance, and β is a constant that balances exploration and exploitation.

Search Strategy Differences

The acquisition function largely determines the behaviour of space exploration for the BO algorithm. Each acquisition function has different selection principles and search strategies. Therefore, they tend to select different candidate points for the evaluation process. Each of them has its own strengths and weaknesses. While the EI acquisition function considers the magnitude of improvement and searches in a greedy manner, the POI acquisition function selects the points that maximize the probability of improvement. So, it is naturally biased towards exploitation and inefficient for exploring the design space. For a problem with multiple local optima, POI tends to be too conservative while EI considers the current best step, therefore it focuses on the improvement in each step. EI and PI acquisition functions are considered improvement-based policies in the literature [SSW$^+$16b]. On the other hand, the GP-UCP acquisition function, which can be considered an optimistic policy, guides the search by minimizing the cumulative regret bound. It also guarantees convergence after a relative number of iterations, however, the parameter β is hard to define as it controls the exploration and exploitation rate which differs according to the problem.

For instance, we illustrate the process in Figure 2.6. As we can see in Figure 2.6, the exploration-exploitation parameter for each acquisition function affects the search performance and strategy. While large values work well for EI and POI, the small value of parameters worked better for the GP-UCB acquisition function.

There are no acquisition function suits for every problem, therefore acquisition function selection is an important task in Bayesian optimization.

2.1.4 Batch Bayesian Optimization

The nature of the sequential decision-making process of acquisition functions blocks the parallelism and it limits the ability to utilize the computational re-

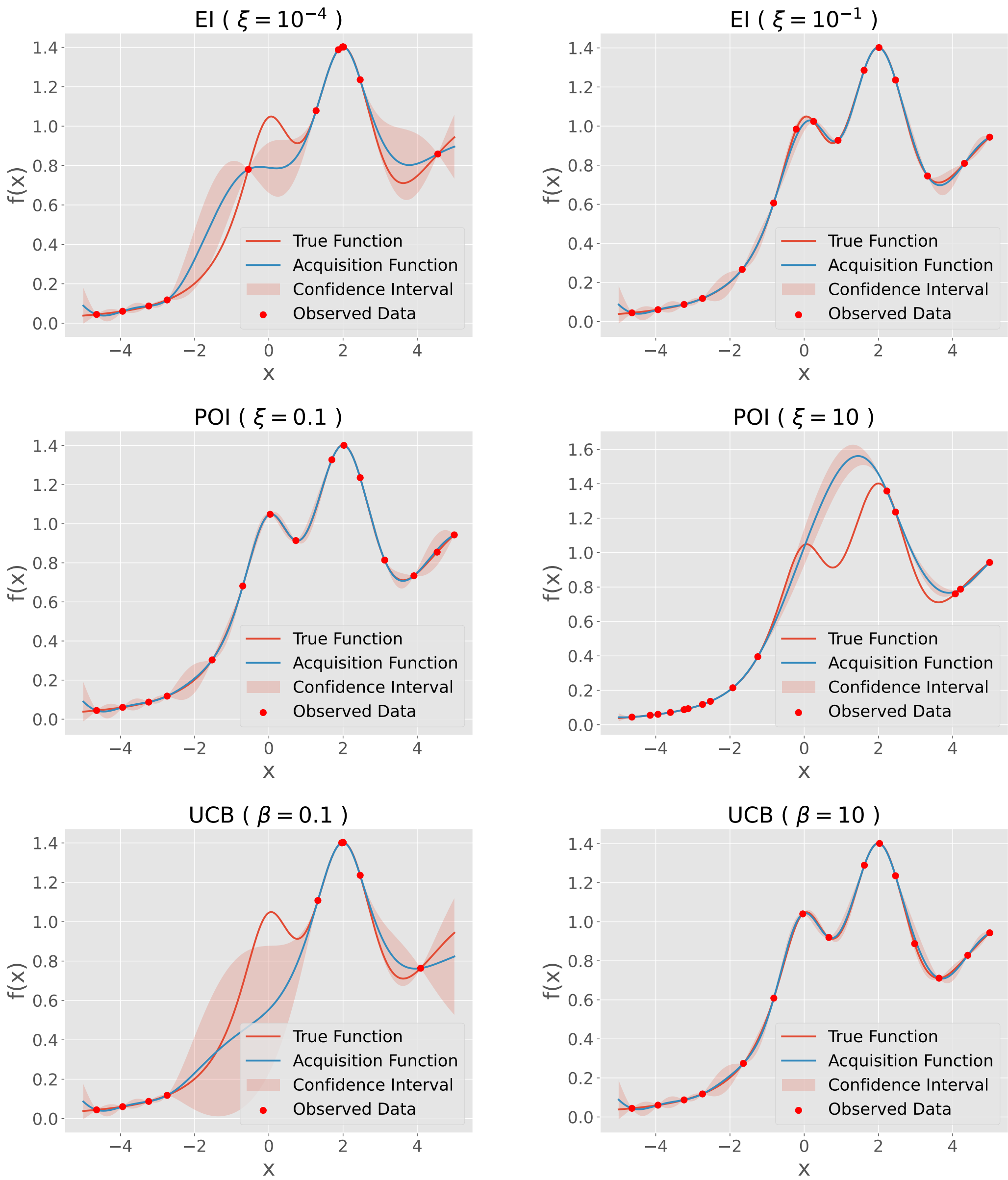


Figure 2.6: The illustration for the effect of the exploration-exploitation parameter on acquisition functions.

sources and optimization efficiency. Especially in the case of multi-core systems are available. There are mainly two challenges when we consider parallelism in BO. The first one is the point selection around the same region as maximizing the acquisition function naturally selects the points around the same region. The second challenge is maximizing the information gained at each batch point selection.

There are great works that have been made to make parallelism possible in the BO framework which means that selecting multiple points is possible during the optimization process. It also has been extended to enable batch selection and developed various approaches over the years, such as q-Expected Improvement (qEI) [CG13], q-Knowledge Gradient (qKG) [WF16], parallel predictive entropy search (PPES) [SG15a]. These methods scale poorly as the batch size increases cause they usually involve some approximations or Monte Carlo sampling during the optimization process. Also, there are other works, including the batch-UCB (BUCB) [DKB14], simulation matching (SM) [AFF10], parallel UCB with pure exploration (GP-UCBPE) [CBRV13] and local penalization (LP) [GDHL15] which adopted the greedy strategy selects next points until the batch is full.

2.2 Multi-objective Optimization

The optimization problem containing more than one objective is called the Multi-objective Optimization Problem (MOP). Inherently, it is hard and commonly impossible to find a single optimum solution as there may be conflicts between the objectives. The solution of MOP is a set of decision variables that optimizes a vector of objective functions within a feasible region of solutions. For a given vector of the objective functions $F(x)$, the MOP is defined by as in Equation 2.10,

$$
\begin{aligned}
\min_x F(x) &= \{f_1(x), \ldots, f_n(x)\} \\
\text{s.t. } \; h_i(x) &= 0, i = 1, \ldots, p \\
g_j(x) &\leq 0, j = 1, \ldots, q \\
x_{min} \leq x &\leq x_{max}
\end{aligned}
\tag{2.10}
$$

where $h_i(x)$ represents equality constraints and $g_j(x)$ represents inequality constraints. As there is no single solution that minimizes each objective, we must consider a set of solutions.

2.2.1 Pareto Solutions

Let us consider a solution S_1 to dominate a solution S_2 if all objective values of the solution S_1 are at least as good as the objective values of S_2 and at least one is better. In this case, we say S_2 is dominated by S_1. A solution S is not dominated by any other solution, also called a non-dominated solution, referred to as the Pareto solution. In Figure 2.7, we can observe the search space domination for two iterations. The red area in the figure represents the dominated space while the white area represents the non-dominated solutions in all graphs. The green and blue areas are improvements of the new points in the search space as the new solutions are dominating the existing ones. Because Pareto solutions are not finite, the goal is to find a finite number of solutions that approximates the Pareto front which is formed by a *set* of Pareto solutions.

2.2.2 Hypervolume Improvement

During the optimization process, a new observation point is added to the surface formed by the non-dominated solution set $\mathcal{D}_n^*$ defined by a set satisfying the following condition in Equation 2.11

$$\begin{aligned} &\forall x, (x, f(x)) \in \mathcal{D}_n^* \subset \mathcal{D}_n,\ \forall x',\ (x', f(x')) \in \mathcal{D}_n, \\ &\exists k \in \{1, \ldots, n\} \text{ such that } f_k(x) \leq f_k(x') \end{aligned} \tag{2.11}$$

where $\mathcal{D}_n = \{(x_1, F(x_1)), \ldots, (x_n, F(x_n))\}$, and the algorithm focused the improvement when adding a new observation for updating the observation data set as $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (x_{n+1}, F(x_{n+1}))\}$.

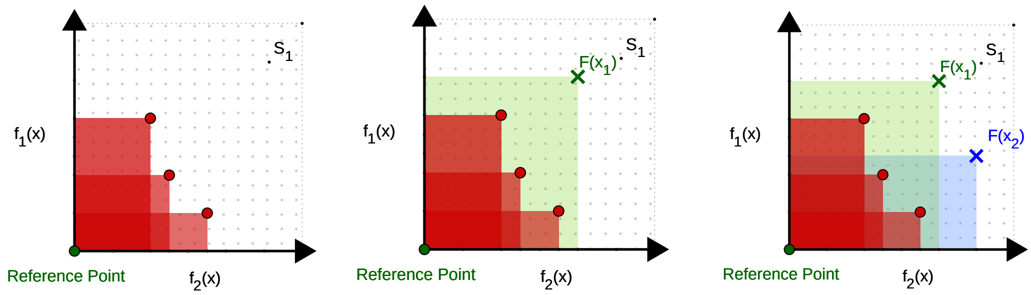


Figure 2.7: An illustration of the dominated (red area) and non-dominated (white area) space. The green and blue area on the graphs represents the HVI of the new points.

Hypervolume is a natural measure of the quality of a Pareto frontier in the outcome space. It is dominated by the Pareto frontier and bounded by a reference point. Hypervolume improvement (HVI) is a measure of improvement and it takes the volume into account that is dominated by the new point in the space corresponding to the candidate.

2.2.3 Bayesian Approach to MOPs

This chapter describes the BO dealing with multi-objective problems. Consider we have multiple *unknown* functions $f_1(x), \ldots, f_n(x)$ for the decision variables

x. We can present them collectively as $F(x) = [f_1(x), \ldots, f_n(x)]^\top$.

Suppose that $HV(\mathcal{D}_n^*)$ is the hypervolume of the region dominated by $\mathcal{D}_n^*$ with a reference point which is used defined. It specified the upper limit of the Pareto solution. Then HVI is defined as in Equation 2.12

$$HVI(F(x_{n+1})) = HV(\mathcal{D}_{n+1}^*) - HV(\mathcal{D}_n^*). \tag{2.12}$$

[EDK11] extended the EI acquisition function idea to HVI and proposed the Expected hypervolume improvement (EHVI) acquisition function. It is defined as in Equation 2.13

$$J(x) = \mathbb{E}_{p(F(x)|\mathcal{D}_n)}[HVI(F(x))] \tag{2.13}$$

where $p(F(x)|\mathcal{D}_n)$ presents the posterior distribution of $F(x)$ and it is often assumed to be independent. EHVI still suffers from some limitations, including the assumption that observations are noise-free, and the exponential scaling of its batch variant, called qEHVI with the batch size q, which precludes large-batch optimization. MOP can also be cast into a single-objective optimization problem by applying a random scalarization of the objectives. ParEGO maximizes the expected improvement using random augmented Chebyshev scalarizations [Kno06a]. MOEA/D-EGO [64] extends ParEGO to the batch setting using multiple random scalarizations and the genetic algorithm MOEA/D [ZZZ12] to optimize these scalarizations in parallel. Recently, qParEGO, another batch variant of ParEGO was proposed that uses compositional Monte Carlo objectives and sequential greedy candidate selection [DBB20a]. Additionally, the authors proposed a noisy variant, qNParEGO, but the empirical evaluation of that variant was limited. Recently, [DBB21b] proposed that the noisy observations improve the performance.

2.3 Bilevel Optimization

Many large-scale decision-making processes are *hierarchical* in terms of the obtained outcome of any decision taken by the upper-level authority (leader) to optimize their goals is affected by another decision considered as the response of lower-level entities (followers) who aim to optimize their own goals. For any given upper-level decision, there is a parametric lower-level optimization problem. These problems provide the rational response of the follower to the leader's decision. The structure of these problems are asymmetric: the leader has perfect knowledge about the followers' objective and constraints, while the follower must first observe the leader's decision before making their own decisions.

There are two roots in terms of the research on decision-making problems with a hierarchical structure. The first one is in the domain of mathematical programming and the second one is in the domain of game theory. In the context of game theory, von Stackelberg [Sta52] built a descriptive model of decision behaviour and provided game-theoretic equilibria. In the context of mathematical programming, an inner optimization problem appears as a constraint of an outer optimization problem which is called the bilevel optimization problem (BOP) [BM73]. They were introduced by J. Bracked and J. McGill, and a defence application was published by the same authors in the following year [BM74]. BOPs were modelled as mathematical programs at this time and are difficult to handle mathematically because of the hierarchical optimization structure. It may introduce difficulties such as non-convexity and disconnectedness between the upper-level and lower-level problems, even for simple instances. It has been shown that bilevel programming is strongly NP-hard [HJS92], and it has been proven that just evaluating a solution is also an NP-hard task [VSJ94a].

BOPs have been used to formulate various real-world hierarchical problems over the years. It has worked successfully in the field of traffic and transportation [Mar86, Mig95, CdLP21], production and capacity planning [GHMA$^+$15, MCVPL20], management science [Bar83a, DLM20], energy networks and market [GCF$^+$12, WPTA20] and defence industry [ADA14, FMVH19].

2.3.1 General Formulation and Definitions

In this section, we provide a general formulation of BOPs. Also, we discuss and formulate optimistic and pessimistic positions which are being assumed by the leader. The formal definition of BOPs was first introduced by Bracken and McGill [BM73], and the designation bilevel was introduced later by Candler and Norton [Can77]. For a general overview of BOPs, we refer to the books [DKPVK15a, DZ20].

We shall represent the leader decision by x_u and the follower response by x_l^*. A decision pair x_u, x_l^* represents a choice by the leader and an optimal feasible solution of the follower. Then the BOP can be defined as follows.

Definition 2.3.1. *For the upper-level objective function $F : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ and lower-level objective function $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$, bilevel optimization problem is given by as in Equation 2.14*

$$\begin{aligned} \min_{x_u} \quad & F(x_u, x_l) \\ \text{s.t.} \quad & x_l \in \underset{x_l}{\operatorname{argmin}}\{f(x_u, x_l) : g_j(x_u, x_l) \leq 0,\ j = 1, 2, \ldots, J\} \\ & G_k(x_u, x_l) \leq 0, \\ & k = 1, 2, \ldots, K \end{aligned} \tag{2.14}$$

where $x_u \in \mathcal{X}_U, x_l \in \mathcal{X}_L$ are vector-valued upper-level and lower-level decision variables, and $\mathcal{X}_U \subseteq \mathbb{R}^n$, $\mathcal{X}_L \subseteq \mathbb{R}^m$ decision spaces, G_k and g_j represent the constraints of the bilevel problem.

Because the lower-level decision maker depends on the upper-level variables, for every decision x_u there is a follower-optimal decision x_l^* [SMD18a]. The upper-level constraints $G(x_u, x_l) \leq 0$ are defined as *coupling constraints* if they depend on x_l. In bilevel optimization, a decision set $x^* = (x_u, x_l^*)$ is feasible for the upper-level only if it satisfies all the upper-level constraints and the vector x^* is an optimal solution to the lower-level problem with the upper-level decision as a parameter.

The feasible set of the BOP is called High Point Relaxation (HPR) if the decision set of (x_u, x_l) satisfies both upper and lower-level constraints.

Definition 2.3.2. *For the upper-level decision x_u and lower-level decision x_l, the High Point Relaxation is defined in Equation 2.15 as follows*

$$\mathit{HPR} = \{(x_u, x_l) \in \mathcal{X}_U \times \mathcal{X}_L \mid G(x_u, x_l) \leq 0 \wedge g(x_u, x_l) \leq 0\} \tag{2.15}$$

where $G(x_u, x_l)$ represents the upper level constraints and $g(x_u, x_l)$ represents lower level constraints.

Definition 2.3.3. *The inducible or induced region (IR) is the feasible set of the bilevel problem and is defined by in Equation 2.16*

$$\mathit{IR} = \{(x_u, x_l) \in \mathit{HPR} : x_l \textit{ satisfies } g(x_u, x_l) \leq 0\}. \tag{2.16}$$

The IR is usually nonconvex and can be disconnected with upper-level constraints. Considering the presence of multiple lower-level optimal solutions for some x_u values, there are two approaches have been proposed [Dem05]. Figure 2.8 illustrates the different Pareto-optimal fronts of two different approaches.

2.3.1.1 Optimistic Bilevel Problems

In this dissertation, we consider the *optimistic approach* which assumes that the follower selects the one that is the best for the leader in the case of the existence of the multiple-follower optimal solution.

Definition 2.3.4. *For the upper and lower-level decision x_u and x_l, optimistic bilevel problem defined by*

$$
\begin{aligned}
\underset{x_u,x_l}{\mathit{min}}\ \ & F(x_u, x_l) \\
\mathit{s.t.}\ \ & x_l \in \underset{x_l}{\mathit{argmin}}\{f(x_u, x_l) : g_j(x_u, x_l) \leq 0,\ j = 1, 2, \ldots, J\} \\
& G_k(x_u, x_l) \leq 0, \\
& k = 1, 2, \ldots, K.
\end{aligned}
\tag{2.17}
$$

This formulation considers that there is cooperation between the two decision-makers. The majority of the work that existed considers an optimistic approach. One of the reasons is it has an optimal solution under quite reasonable assumptions as can be seen in [Dem02].

2.3.1.2 Pessimistic Bilevel Problems

In the case of not allowing cooperation between the leader and the follower, *the pessimistic approach* is used which assumes that the follower selects the optimal solution corresponding to the worst outcome of the leader [WTKR13]. So, the leader optimizes his/her objective for the worst case. This approach is useful in the case of the leader who wants to limit the damage after an unexpected follower decision.

Definition 2.3.5. *The pessimistic bilevel problem defined by Equation 2.18*

$$
\begin{aligned}
\underset{x_u}{\mathit{min}}\ \underset{x_l}{\mathit{max}}\ \ & F(x_u, x_l) \\
\mathit{s.t.}\ \ & x_l \in \underset{x_l}{\mathit{argmin}}\{f(x_u, x_l) : g_j(x_u, x_l) \leq 0,\ j = 1, 2, \ldots, J\} \\
& G_k(x_u, x_l) \leq 0, \\
& k = 1, 2, \ldots, K.
\end{aligned}
\tag{2.18}
$$

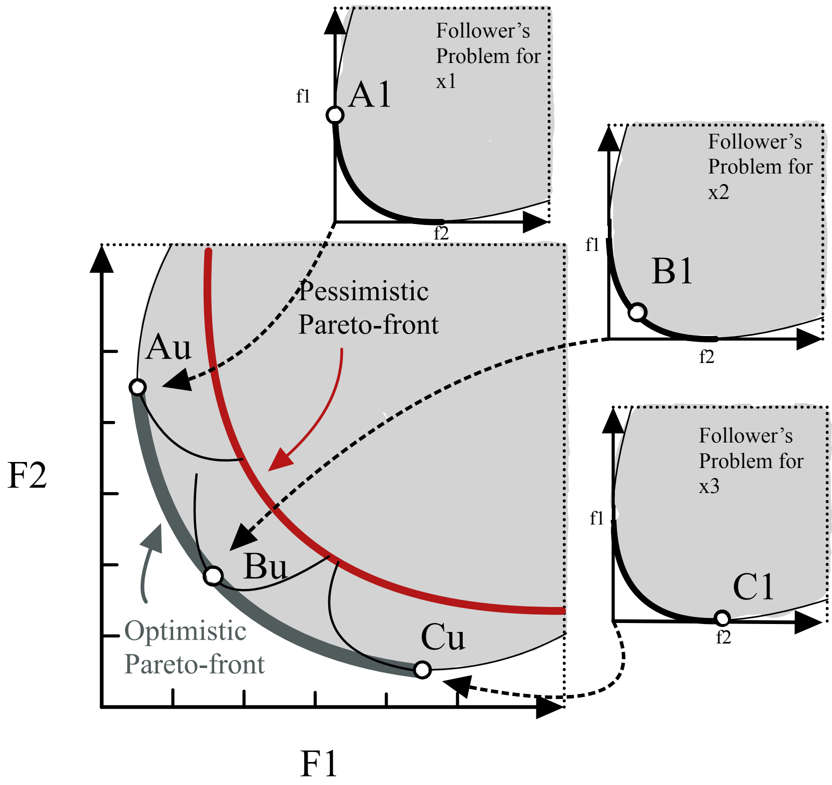


Figure 2.8: An Illustration presents the optimistic and pessimistic Pareto optimal fronts at both upper- and lower-level multi-objective bilevel problems.

As expected, this approach does not assume cooperation between the leader and the follower. The pessimistic approach is less tractable compared with the optimistic approach because the optimistic formulation can be reformulated to single-level problems in the case of a convex lower-level problem. However, the pessimistic approach can be considered a three-level task because every lower-level problem has to keep track of that lower-level optimal solution is the worst for the upper-level.

2.3.2 Multi-objective Bilevel Problems

Multi-objective bilevel optimization problems have two levels of multi-objective optimization, such that a feasible solution to the upper-level problem must be a member of the Pareto-optimal set of lower-level optimization problems. Each level has its variables, objectives, and constraints.

A general multi-objective bilevel optimization problem can be described as follows:

$$\begin{aligned}&\min_{x_u \in X_u, x_l \in X_l} \{F_1(x_u, x_l), ..., F_{M_u}(x_u, x_l)\}\\&\text{s.t. } x_l \in \underset{x_l}{\text{argmin}}\{f_1(x_u, x_l), ..., f_{M_l}(x_u, x_l);\ g_j(x_u, x_l) \leq 0,\ j = 1, 2, \ldots, J\}\\&\qquad\qquad G_k(x_u, x_l) \leq 0,\ k = 1, 2, \ldots, K\end{aligned} \tag{2.19}$$

In Eq. 2.19 the upper-level objective functions are $F_i(x_u, x_l),\ i = 1, 2, \ldots, M_u$ and the lower-level objective functions are $f_i(x_u, x_l),\ i = 1, 2, \ldots, M_l$ where $x_u \in \mathcal{X}_u$ and $x_l \in \mathcal{X}_l$. $G_k(x_u, x_l)$ and $g_j(x_u, x_l)$ represent upper and lower-level constraints respectively. j and K values represent the number of constraints at the upper- and lower-level. The lower-level optimization problem is optimized with respect to x_l considering x_u as a fixed parameter.

Multi-objective Bilevel Problems Under Uncertainty

Most studies in the MOBO literature focus on solving the optimization problem without addressing the impact of uncertainty. In practical problems, noise in the leader's objectives might represent environmental uncertainty, for example in a meta-learning regime [ABB+17] that can be mathematically formulated as BOP [FFS+18]. As another example, a government might need to prevent terrorist attacks using information from unreliable sources. Yet another example occurs in computing optimal recovery policies for financial markets [MBD12], using bilevel optimization with objective uncertainties caused by several uncontrollable parameters.

We consider the case that is crucial in practice where the leader needs to make his/her decision under uncertainty and has only noisy observations $F_i = f(\ x_{u_i},\ x_{l_i}) + \xi_i$ where $\xi_i \sim \mathcal{N}(0, \Sigma_i)$ and Σ_i is the noise covariance and decision variables for both upper and lower-level decision makers $x_u,\ x_l$. According to the consideration above, we reformulate the MOBO problems as in Equation

2.20 with noisy leader objectives by

$$
\begin{aligned}
&\min_{x_u x_l}\{F_1(x_u,x_l)+\xi_1,...,F_p(x_u,x_l)+\xi_p\}\\
&\text{s.t.}\\
&\ x_l\in \underset{x_l}{\text{argmin}}\{f_1(\ x_u,\ x_l),...,f_q(\ x_u,\ x_l);\\
&\qquad\qquad\qquad g_j(\ x_u,\ x_l)\leq 0,\ j=1,2,\ldots,J\}\\
&G_k(\ x_u,\ x_l)\leq 0,\ k=1,2,\ldots,K\\
&\xi_p\sim\mathcal{N}(0,\Sigma_p).
\end{aligned}
\tag{2.20}
$$

In this dissertation, we focus on optimizing the objective function in an uncertain environment.

2.3.3 Applications of Bilevel Optimization

One of the first applications of BOPs was in the Stackelberg game in the literature of Game Theory [Sta52]. It is a scenario in economics where two decision-makers, a leader and a follower, operate in the same market. Both aim to maximize their profit and mathematically it is represented as shown in Equation 2.21.

$$
\begin{aligned}
&\max_{x_u x_l}\ \ F(x_u,x_l)=P(x_u,x_l)-C(x_u)\\
&\text{s.t.}\quad x_l\in\underset{x_l}{\text{argmax}}\ \ f(x_u,x_l)=P(x_u x_l)-C(x_l)\\
&\qquad\ x_u+x_l\geq Q\\
&\text{where, } x_u,x_l,Q\geq 0.
\end{aligned}
\tag{2.21}
$$

The upper-level and lower-level variables, x_u and x_l, represent their respective production quantities while Q is the overall demand in the market to be fulfilled. $P(\cdot)$ and $C(\cdot)$ denote the total price obtained from the sales and the cost of the production. The problem reaches the optima when the leader and the follower are at Stackelberg equilibrium.

There are several applications of BOPs problems have been worked on over the years. They are briefly discussed in the following list.

- **Defense Industry**: BOPs have been applied to various kinds of defence applications over the years such as planning the prepositioning of defensive missile interceptors to prepare to counter an attack thread [BCD+05], homeland security [Low06], and attacker defender Stackelberg games [SC08, AOT+13, ZTM18]. The bilevel problem can be constructed such that the upper-level aims to maximize the damage caused to the opponent while considering the optimal reactions of the opponent at the lower-level.

- **Machine Learning**: BOPs mostly appear in hyperparameter tuning problems in machine learning. It allows for systematic and more efficient search when the number of parameters is large [MdDMMQC21].
- **Transportation**: A transportation network consists of an author at the upper-level who increases the revenue from the tolls on a road network and the users at the lower-level whose objective is finding the shortest or most effective path to minimize the cost for their travel [CKLK10, FM11, XTZ12, CVMOLR15].
- **Management**: While managing a network facility, it is important to coordinate with small businesses to increase the shared profits with them. That coordination appears as a BOP [Bar83b, DLM20]
- **Environmental Economics**: As a consequence of an operation of an organization, an authority may want to tax the organization considering the pollution that is caused by the operations. Finding an optimal level of tax considering the pollution and revenue appears as a BOP. Authority acts as a leader and the organization as a follower [SMFD13a, SMFD14].
- **Chemical Industry**: Several applications appear as a BOP in the chemical industry such as [HM06, CW90, SWI85]. For instance, an expert has to decide between conditions, such as states quantities or reactants, for the reaction to achieve the optimal result. Also, it needs to consider the lower-level equilibrium conditions for the process such as minimizing entropy functions.
- **Optimal Design**: In structural optimization, it is required to consider minimizing the weight or cost of a structure while deciding on the shape. On the other hand, while deciding the shape, the contact forces and stresses need to be determined to minimize the potential energy. It appears as a BOP for solving the optimal shape design, and different approaches can be found in [Ben95, CPW01, HLDS00].

2.4 Performance Comparison

2.4.1 Test Problems

The two particular test problems are used the most often in this dissertation. They are called SMD and TP problems. The rest of the problems are small in numbers such as multi-objective bilevel problems and applications. The details of these small number of problems are placed in the related chapter.

2.4.1.1 Benchmark 1: TP Problems

The first benchmark is standard test problems, called TP problems, and contains 10 bilevel problems. The problems are well known in the bilevel optimization literature and they are often used for evaluating the efficacy of different bilevel optimization algorithms. Each problem has a different interaction between upper-level and lower-level decision-makers. The benchmark includes 1 three-dimensional, 6 four-dimensional, 1 five-dimensional, and 2 ten-dimensional bilevel problems. Each problem has a different complexity in terms of interaction between decision-makers. The benchmark includes mostly constrained problems and is created for testing the efficiency of proposed methods by the researchers. Mostly the efficiency is determined by function evaluations and obtained fitnesses. We share the details about the TP problems in Table 2.1 and 2.2 including the formulations. More details can be found in [SMD13a].

2.4.1.2 Benchmark 2: SMD Problems

The second benchmark is called SMD problems and it has 6 unconstrained problems. These are unconstrained problems with controllable complexities. Each problem has a different difficulty level in terms of convergence, the complexity of interaction, and lower-level multimodality. They are also scalable in terms of dimensionality. The upper-level contains three components, such as F_1,F_2 and F_3. The sum of these functions is used to calculate the upper-level objective function. Similarly, the lower-level function is constructed by summation of f_1,f_2 and f_3. The generic form of the functions is shown in Equation 2.22.

$$\begin{aligned} \min_{x_u, x_l} \; & F(x_u, x_l) = F_1(x_{u_1}) + F_2(x_{l_1}) + F_3(x_{u_2}, x_{l_2}) \\ \text{s.t.} \; & \min_{x_l} \; f(x_u, x_l) = f_1(x_{u_1}, x_{u_2}) + f_2(x_{l_1}) + F_3(x_{u_2}, x_{l_2}) \end{aligned} \tag{2.22}$$

Table 2.1: Summary of the TP Problems (TP1-TP5)

Problem	Formulation	Best Known Solution
TP1 n = 2 m = 2	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l) = (x_{u_1}-30)^2 + (x_{u_1}-20)^2 - 20x_{l_1} + 20x_{l_2},$ s.t. $x_l \in \underset{x_l}{\text{argmin}} \left\{ \begin{array}{c} f(x_u,x_l) = (x_{u_1}-x_{l_1})^2 + (x_{u_2}-x_{l_2})^2, \\ 0 \le x_{l_i} \le 10,\ i=1,2 \end{array} \right\},$ $x_{u_1} + 2x_{u_2} \ge 30, x_{u_1} + x_{u_2} \le 25, x_{u_2} \le 15$	F = 225.0 f = 100.0
TP2 n = 2 m = 2	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l) = 2x_{u_1} + 2x_{u_2} - 3x_{l_1} - 3x_{l_2} - 60,$ s.t. $x_l \in \underset{x_l}{\text{argmin}} \left\{ \begin{array}{c} f(x_u,x_l) = (x_{l_1} - x_{u_1} + 20)^2 + x_{l_2} - x_{u_2} + 20)^2, \\ x_{u_1} - 2x_{l_1} \ge 10, x_{u_2} - 2x_{l_2} \ge 10,\ i=1,2 \\ -10 \ge x_{l_i} \ge 20, i=1,2 \end{array} \right\},$ $x_{u_1} + x_{u_2} + x_{l_1} - 2x_{l_2} \le 40, 0 \ge x_{u_i} \ge 50, i=1,2$	F = 0.0 f = 100.0
TP3 n = 2 m = 2	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l) = -(x_{u_1})^2 - 3(x_{u_2})^2 - 4(x_{l_1})^2 + (x_{l_2})^2,$ s.t. $x_l \in \underset{x_l}{\text{argmin}} \left\{ \begin{array}{c} f(x_u,x_l) = 2(x_{u_1})^2 + (x_{l_1})^2 - 5x_{l_2}, \\ (x_{u_1})^2 - 2x_{u_1} + (x_{u_2})^2 - 2x_{l_1} + x_{l_2} \ge -3, \\ x_{u_2} + 3x_{l_1} - 4x_{l_2} \ge 4, 0 \le x_{l_i}, i=1,2 \end{array} \right\},$ $(x_{u_1})^2 + 2x_{u_2} \le 4, 0 \le x_{u_i}, i=1,2$	F = -18.67 f = -1.0156
TP4 n = 2 m = 3	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l) = -8x_{u_1} - 4x_{u_2} + 4x_{l_1} - 40x_{l_2} - 4x_{l_3},$ s.t. $x_l \in \underset{x_l}{\text{argmin}} \left\{ \begin{array}{c} f(x_u,x_l) = x_{u_1} + 2x_{u_2} + x_{l_1} + x_{l_2} + 2x_{l_3}, \\ x_{l_2} + x_{l_3} - x_{l_1} \le 1, \\ 2x_{u_1} - x_{l_1} + 2x_{l_2} - 0.5x_{l_3} \le 1, \\ 2x_{u_2} + x_{l_1} - x_{l_2} - 0.5x_{l_3} \le 1, \\ 0 \le x_{l_i}, i=1,2,3 \end{array} \right\},$ $0 \le x_{u_i}, i=1,2$	F = -29.2 f = 3.2
TP5 n = 2 m = 2	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l) = rt(x_u)x_u - 3x_{l_1} - 4x_{l_2} + 0.5t(x_l)x_l,$ s.t. $x_l \in \underset{x_l}{\text{argmin}} \left\{ \begin{array}{c} f(x_u,x_l) = 0.5t(x_l)hx_l - t(b(x_u))x_l - \\ 0.333x_{l_1} + x_{l_2} - 2 \le 0, \\ x_{l_1} - 0.333x_{l_2} - 2 \le 0, \\ 0 \le x_{l_i}, i=1,2 \end{array} \right\},$ $h = \begin{pmatrix} 1 & 3 \\ 3 & 10 \end{pmatrix}, b(x) = \begin{pmatrix} -1 & 2 \\ 3 & -3 \end{pmatrix} x, r = 0.1$ $t(\cdot)$ denotes transpose of a vector	F = -3.6 f = -2.0

We set the dimensions of all problems as 4 for all experiments. The termination criteria are selected as the same as TP problems. The true optimum values and the details are reported in Table 2.3. More details about this benchmark can be found in [SMD12a].

2.4.2 Comparison Metrics

Single-objective Bilevel Problems

The following metrics are used for evaluating and comparing the performance of the proposed approaches.

1. **Fitness score**: For single-level BOPs, generally the performance is evaluated by obtaining the solutions over multiple runs by each algorithm. The obtained solutions in this dissertation are represented as *fitness score*

Table 2.2: Summary of the TP Problems (TP6-TP10)

Problem	Formulation	Best Known Sol.
TP6 n = 1 m = 2	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l) = (x_{u_1}-1)^2 + 2x_{l_1} + 2x_{u_1},$ $\text{s.t. } x_l \in \underset{x_l}{\text{argmin}} \left\{ \begin{array}{c} f(x_u,x_l) = (2x_{l_1}-4)^2 + (2x_{l_2}-1)^2 + x_{u_1}x_{l_1}, \\ 4x_{u_1} + 5x_{l_1} + 4x_{l_2} \leq 12, \\ 4x_{l_2} - 4x_{u_1} - 5x_{l_1} \leq -4, \\ 4x_{u_1} - 4x_{l_1} + 5x_{l_2} \leq 4, \\ 4x_{l_1} - 4x_{u_1} + 5x_{l_2} \leq 4, \\ 0 \leq x_{l_i}, i = 1,2 \end{array} \right\},$ $0 \leq x_{u_1}$	F = -1.2091 f = 7.6145
TP7 n = 2 m = 2	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l) = -\frac{(x_{u_1}+x_{l_1})(x_{u_2}+x_{l_2})}{1+x_{u_1}x_{l_1}+x_{u_2}x_{l_2}},$ $\text{s.t. } x_l \in \underset{x_l}{\text{argmin}} \left\{ \begin{array}{c} f(x_u,x_l) = \frac{(x_{u_1}+x_{l_1})(x_{u_2}+x_{l_2})}{1+x_{u_1}x_{l_1}+x_{u_2}x_{l_2}}, \\ 0 \leq x_{l_i} \leq x_{u_i}, i = 1,2 \end{array} \right\},$ $(x_{u_1})^2 + (x_{u_2})^2 \leq 100$ $x_{u_1} - x_{u_2} \leq 0$ $0 \leq x_{u_i}, i = 1,2$	F = -1.96 f = 1.96
TP8 n = 2 m = 2	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l) = \lvert 2x_{u_1} + 2x_{u_2} - 3x_{l_1} - 3x_{l_2} - 60 \rvert,$ $\text{s.t. } x_l \in \underset{x_l}{\text{argmin}} \left\{ \begin{array}{c} f(x_u,x_l) = (x_{l_1} - x_{u_1} + 20)^2 + (x_{l_2} - x_{u_2} + 20)^2, \\ 2x_{l_1} - x_{u_1} + 10 \leq 0, \\ 2x_{l_2} - x_{u_2} + 10 \leq 0, \\ -10 \leq x_{l_i} \leq 20, i = 1,2 \end{array} \right\},$ $x_{u_1} + x_{u_2} + x_{l_1} - 2x_{l_2} \leq 40$ $0 \leq x_{u_i} \leq 50, i = 1,2$	F = 0.0 f = 100.0
TP9 n = 10 m = 10	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l) = \sum_{i=1}^{10} (\lvert x_{u_i} - 1 \rvert + \lvert x_{l_i} \rvert),$ $\text{s.t. } x_l \in \underset{x_l}{\text{argmin}}$ $\left\{ \begin{array}{c} f(x_u,x_l) = e^{\left(1+\frac{1}{4000}\sum_{i=1}^{10}(x_{l_i})^2 - \prod_{i=1}^{10}\cos(\frac{x_{l_i}}{\sqrt{i}})\right)\sum_{i=1}^{10}(x_{u_i})^2}, \\ -\pi \leq x_{l_i} \leq \pi, i = 1,2,\ldots,10 \end{array} \right\}$	F = 0.0 f = 1.0
TP10 n = 10 m = 10	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l) = \sum_{i=1}^{10} (\lvert x_{u_i} - 1 \rvert + \lvert x_{l_i} \rvert),$ $\text{s.t. } x_l \in \underset{x_l}{\text{argmin}} \left\{ \begin{array}{c} f(x_u,x_l) = e^{\left(1+\frac{1}{4000}\sum_{i=1}^{10}(x_{l_i}x_{u_i})^2 - \prod_{i=1}^{10}\cos(\frac{x_{l_i}x_{u_i}}{\sqrt{i}})\right)}, \\ -\pi \leq x_{l_i} \leq \pi, i = 1,2,\ldots,10 \end{array} \right\}$	F = 0.0 f = 1.0

which represents the optimization results over multiple runs. The smaller value of the fitness score represents better results for minimization problems and the opposite for maximization problems.

2. **Computational effort**: The upper- and lower-level optimization problems have different numbers of function evaluations. In the numerical experiments in this dissertation, the function evaluations at both levels are presented and used for the computational effort comparison with selected algorithms.

Table 2.3: Summary of the SMD Problems

Problem	Formulation	Best Known Sol.
SMD1 n = 2 m = 2	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l)=(x_{u_1})^2+(x_{l_1})^2+(x_{u_2})^2+(x_{u_2}-\tan(x_{l_2}))^2,$ s.t. $x_l\in\underset{x_l}{\text{argmin}}\left\{\begin{matrix}f(x_u,x_l)=(x_{u_1})^2+(x_{l_1})^2+\\(x_{u_2}-\tan(x_{l_2}))^2\end{matrix}\right\},$ $-5\le x_{u_1}\le 10,-5\le x_{u_2}\le 10,-5\le x_{l_1}\le 10,-\frac{\pi}{2}\le x_{l_2}\le\frac{\pi}{2}$	F = 0.0 f = 0.0
SMD2 n = 2 m = 2	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l)=(x_{u_1})^2-(x_{l_1})^2+(x_{u_2})^2+(x_{u_2}-\log(x_{l_2}))^2,$ s.t. $x_l\in\underset{x_l}{\text{argmin}}\left\{\begin{matrix}f(x_u,x_l)=(x_{u_1})^2+(x_{l_1})^2+\\(x_{u_2}-\log(x_{l_2}))^2\end{matrix}\right\},$ $-5\le x_{u_1}\le 10,-5\le x_{u_2}\le 1,-5\le x_{l_1}\le 10,0\le x_{l_2}\le e$	F = 0.0 f = 0.0
SMD3 n = 2 m = 2	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l)=(x_{u_1})^2-(x_{l_1})^2+(x_{u_2})^2+((x_{u_2})^2-\tan(x_{l_2}))^2,$ s.t. $x_l\in\underset{x_l}{\text{argmin}}\left\{\begin{matrix}f(x_u,x_l)=(x_{u_1})^2+1+((x_{l_1})^2-\cos(2\pi x_{l_1}))+\\((x_{u_2})^2-\tan(x_{l_2}))^2,\end{matrix}\right\},$ $-5\le x_{u_1}\le 10,-5\le x_{u_2}\le 10,-5\le x_{l_1}\le 10,-\frac{\pi}{2}\le x_{l_2}\le\frac{\pi}{2}$	F = 0.0 f = 0.0
SMD4 n = 2 m = 2	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l)=(x_{u_1})^2-(x_{l_1})^2+(x_{u_2})^2-(\lvert x_{u_2}\rvert-\log(1+x_{l_2}))^2,$ s.t. $x_l\in\underset{x_l}{\text{argmin}}\left\{\begin{matrix}f(x_u,x_l)=(x_{u_1})^2+1+((x_{l_1})^2-\cos(2\pi x_{l_1}))+\\(\lvert x_{u_2}\rvert-\log(1+x_{l_2}))^2,\end{matrix}\right\},$ $-5\le x_{u_1}\le 10,-1\le x_{u_2}\le 1,-5\le x_{l_1}\le 10,0\le x_{l_2}\le e$	F = 0.0 f = 0.0
SMD5 n = 2 m = 2	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l)=(x_{u_1})^2-((x_{l_1})^2-(x_{l_2})^2)+(x_{l_1}-1)^2+x_{u_2}^2$ $-(\lvert x_{u_2}\rvert-(x_{l_2})^2)^2,$ s.t. $x_l\in\underset{x_l}{\text{argmin}}\left\{\begin{matrix}f(x_u,x_l)=(x_{u_1})^2+((x_{l_1})^2-(x_{l_2})^2)+(x_{l_1}-1)^2+\\(\lvert x_{u_2}\rvert-(x_{l_2})^2)^2\end{matrix}\right\},$ $-5\le x_{u_1}\le 10,-5\le x_{u_2}\le 10,-5\le x_{l_1}\le 10,-5\le x_{l_2}\le 10$	F = 0.0 f = 0.0
SMD6 n = 2 m = 2	$\underset{x_u,x_l}{\text{Minimize}}\ F(x_u,x_l)=(x_{u_1})^2-(x_{l_1})^2,((x_{l_1})^2)^2+(x_{u_2})^2-(x_{u_2}-x_{l_2})^2$ s.t. $x_l\in\underset{x_l}{\text{argmin}}\left\{\begin{matrix}f(x_u,x_l)=(x_{u_1})^2+(x_{l_1})^2,((x_{l_1})^3-x_{l_1})^2)^2+\\(x_{u_2}-x_{l_2})^2,\end{matrix}\right\},$ $-5\le x_{u_1}\le 10,-5\le x_{u_2}\le 10,-5\le x_{l_1}\le 10,-5\le x_{l_2}\le 10$	F = 0.0 f = 0.0

Multi-objective Bilevel Problems

For multi-objective problems, hypervolume (HV) and inverted generational distance (IGD) metrics are used for performance comparison.

1. HV is commonly used to provide a measure of convergence and diversity of a given non-dominated set of solutions. It is calculated as the union of the dominated regions by each of the solutions concerning a reference point in the objective space. Thus, the larger hypervolume value is preferred. For minimization problems, HV can be computed using Equation 2.23.

$$HV = \cup a_{i,R} \vert v_i \in Q \tag{2.23}$$

   where $a_{i,R}$ is the dominated space with respect to reference point R for the solution v_i in the non-dominated set of Q.

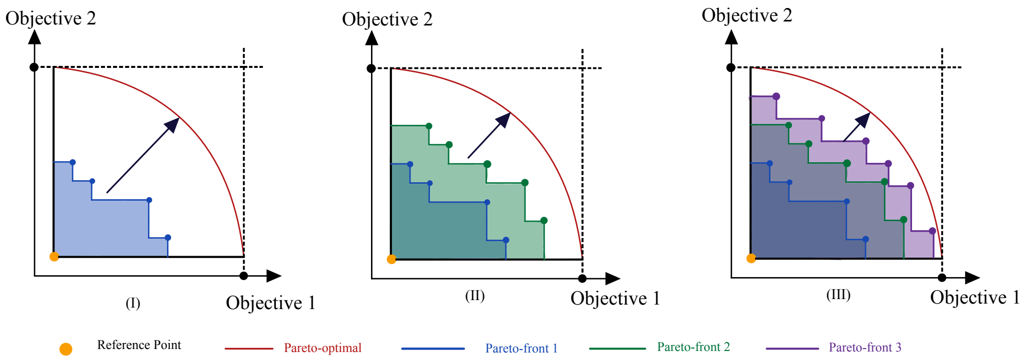


Figure 2.9: An illustration about HV indicator.

Each graph in Figure 2.9 represents different non-dominated solutions and how improvement is being measured. The red line represents the Pareto-optimal front and the algorithm tries to find a set of solutions that approximates the Pareto-optimal solution. It is expected from Figure 2.9 that the HV value of Pareto-front 3 is higher than the others.

2. The generational distance (GD) of a set A is defined as the distance between each point $a \in A$ and closest point r in a reference set R, averaged over the size of A. GD is calculated as in Equation 2.24.

$$GD_p(A, R) = \left(\frac{1}{|A|} \sum_{a \in A} \min_{e \in R} d(a, r)^p \right)^{\frac{1}{p}} \tag{2.24}$$

where $d(a, r)$ is a distance function. Regarding Equation 2.24, IGD is computed using Equation 2.25.

$$IGD_p(A, R) = GD_p(R, A) \tag{2.25}$$

Figure 2.10 shows the objective space of the third graph in Figure 2.9. Grey lines show the distance between Pareto-front 3 and a set of Pareto-optimal. As we can see from the figure, the expected IGD value of the Pareto-front 3 should be smaller than the others. It means that it has a better approximation to the Pareto-optimal solution set.

2.5 Literature Review

In this section, we present some existing solution methods for BOPs. We mention a wide range of existing works from classical approaches to evolutionary

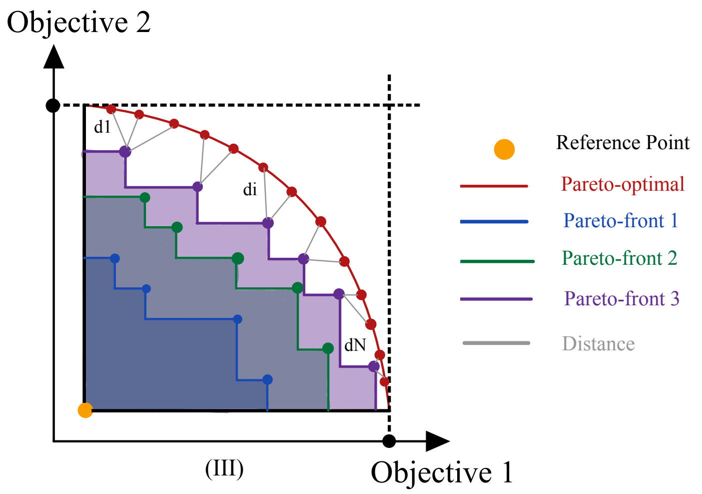


Figure 2.10: An illustration about IGD indicator.

algorithms including some special cases and reformulations.

2.5.1 Classical Approaches

There are several *classical approaches* developed over the years in the literature for solving BOPs. A serious amount of works considered linear bilevel problems. Such as [BA93, WH91] that focused on linear bilevel problems with continuous variables and [VSJ96] for combinatorial variables. BOPs can be reduced to single-level problems under some conditions. If the lower-level problem is convex and sufficiently regular, it is possible the replace the lower-level problem with its Karush-Kuhn-Tucker (KKT) conditions as they appear as Lagrangian and complementary constraints. For instance, the Eq. 2.17 can be reduced to the following form as in Equation 2.26:

$$
\begin{aligned}
\min_{x_u, x_l} \; & F(x_u, x_l) \\
\text{s.t. } & G_k(x_u, x_l) \leq 0, \quad k = 1, 2, \ldots, K \\
& \Delta_{x_l} L(x_u, x_l, \lambda) = 0, \\
& g_j(x_u, x_l) \leq 0, \quad j = 1, 2, \ldots, J \\
& \lambda_j g_j(x_u, x_l) = 0 \\
& \lambda_j \geq 0
\end{aligned}
\tag{2.26}
$$

where Lagrangian constraints is $L(x_u, x_l, \lambda) = f(x_u, x_l) + \sum_{j=1}^{J} \lambda_j g_j(x_u, x_l)$. For linear bilevel problems, the Lagrangian constraint is also linear so the single-level problem is a mixed integer linear problem. At this point, we would like to share a recent survey on mixed integer programming techniques in BOPs [KLLS21]. Considering this, some approaches based on vertex enumeration [BK84, TMV93] and branch-and-bound [FAM81, BF82] have been used to solve these problems. The branch-and-bound approach is also successfully applied to the single-level reduction of linear-quadratic [BM90] and quadratic-quadratic [EB91] bilevel problems as the upper- and lower-level.

There are also a quite number of *descent methods* that have been proposed for solving BOPs. In bilevel problems, the descent directions advantage is decreasing the upper-level function value while the new point is still feasible which is only if it is lower-level optimal. For example, [KL90] approximates the gradient of the upper-level objective and [VSJ94b] works on determining the direction of descent.

There are also some *penalty function methods* that have been proposed over the years. The first one was made by Aiyoshi and Shimizu in [ADP20, AW90] and they replaced the lower-level problem with a penalized problem. Then [IA92] proposed a double penalty method where both upper and lower levels were penalized. [LHWW07] reduced the linear bilevel problem into a single level using KKT conditions and the problem is handled using a series of linear problems.

There are some approximate algorithms developed under the name of *trust-region methods* for bilevel problems such as [CMS05] that solve nonlinear BOPs. It deals with convex lower-level objectives and with linear constraints. Also, Colson et al. [CMS02] approximated the bilevel problem around each iteration with a model that is a linear-quadratic bilevel problem. The authors used a mixed integer solver after applying KKT conditions.

2.5.2 Evolutionary Approaches

Evolutionary algorithms are popular approaches to solving BOPs. Even though it is effective, these strategies are computationally expensive and not successful in handling large-scale bilevel problems. It is applied in two different strategies: the first is solving upper-level problems with an evolutionary algorithm while the lower-level is being solved with classical approaches and the second one is solving both levels with evolutionary algorithms. The complexity of lower-level problems is the main factor in deciding which strategy is the best. One of the first attempts to solve bilevel problems with evolutionary algorithms was in the early 1990s by Mathieu et al. [MPA94]. The authors used a genetic algorithm to solve upper-level problems and linear programming methods at the lower-level. [LTM06] used particle swarm optimization to solve BOPs. [LW07] proposed a hybrid approach and they used a simplex-based crossover strategy to solve upper-level problems and a classical approach for solving lower-level problems.

Another popular approach for solving BOPs is differential evolution-based approaches. For example, [ZYW06] solved upper-level problems while [AKB13] used a differential evolution-based approach for both levels. The authors also improved the algorithm by using different evolutionary approaches and combining them to handle both levels. [AB15] used an ant colony optimization for solving upper-level problems and differential evolution for solving the lower-level problem for a transportation problem. [CGO11] attempt to solve the production-distribution planning problem in a bilevel model by utilizing an ant colony algorithm.

Several works have been presented and used KKT conditions to reduce the bilevel problem to a single level and apply evolutionary algorithms. For example, [HMJS02] reduced a linear bilevel problem to a single level and used a genetic algorithm to solve the problem. [WJL05] proposed an algorithm by utilizing a constraint-handling process to solve the problem. The algorithm was successful in handling non-differentiable upper-level objective functions and they improved the algorithm in [WLD11] and it was able to handle non-convex lower-level objectives. [JLHW13] used a particle swarm optimization algorithm to solve the bilevel problem after reducing it into a nonlinear optimization problem with complementarity constraints. Also, [WWS13] improved particle swarm optimization by embedding the chaos search technique to solve bilevel problems after reducing a single-level problem.

2.5.3 Meta-modelling Approaches

The number of function evaluations to tune even small variable BOPs is quite high in these methods. The *meta-modelling approaches* are developed over the years to handle this problem. It is a commonly used method for expensive-to-evaluate optimization problems [WS07]. It uses a surrogate model to approximate the actual model to reduce the function evaluations required to solve the actual problem. The complexity of BOPs leads to a requirement for a large number of evaluations. Population-based algorithms in meta-modelling techniques improve the optimization process of BOPs. Reaction set mapping is one of the meta-modelling-based techniques to solve the problem. It is based on the idea of approximation to the lower-level problem during the bilevel optimization process. Let us assume that an approximate mapping, Φ, can be generated and then the bilevel problem is reduced to the single-level problem as in Equation 2.27, as follows:

$$\begin{aligned} \min_{x_u, x_l} \;& F(x_u, x_l) \\ \text{s.t. } & x_l \in \Phi(x_u) \\ & G_k(x_u, x_l) \leq 0, \quad k = 1, 2, \ldots, K. \end{aligned} \tag{2.27}$$

Some works used evolutionary algorithms based on this approach, such as [AKB14, SMD17c]. Also [SMD13b, SMD14a] proposed a nested approach which based on quadratic approximations to the local reaction set. Another meta-modelling-based method is approximate of the optimal value function. If the Φ mapping is given, the bilevel problem can be reduced to single level problem [YZ10] and defined by

$$\begin{aligned} \min_{x_u, x_l} \;& F(x_u, x_l) \\ \text{s.t. } & f(x_u, x_l) \leq \Phi(x_u) \\ & G_k(x_u, x_l) \leq 0, \quad k = 1, 2, \ldots, K \\ & g_j(x_u, x_l) \leq 0, \quad j = 1, 2, \ldots, J \end{aligned} \tag{2.28}$$

In Eq. 2.28, $\Phi(\cdot)$ is unknown so approximating this function can be done by meta-modelling techniques. As it is a single-valued function, approximating avoids the complexities associated with the set-valued mapping. Following this idea, if Φ' represents the *approximation mapping*, the new single-level optimization problems defined as follows:

$$\begin{aligned} \min_{x_u, x_l} \;& F(x_u, x_l) \\ \text{s.t. } & f(x_u, x_l) \leq \Phi'(x_u) \\ & G_k(x_u, x_l) \leq 0, \quad k = 1, 2, \ldots, K \\ & g_j(x_u, x_l) \leq 0, \quad j = 1, 2, \ldots, J. \end{aligned} \tag{2.29}$$

In Eq. 2.29, the $\Phi'(\cdot)$ can be generated with the population members of an evolutionary algorithm. The related works are in [SLDM17, SMD16a].

We would like to refer to some review papers in general bilevel optimization [KDPV$^+$15, SMD17a, SMD18b, QWP21] and books [Tal13, DKPVK15b, Dem20] for more in-depth information. Also [JYL21] worked recently on convergence analysis on bilevel optimization.

2.5.4 Multi-objective Bilevel Approaches

There are quite a lot of works have been presented for single-objective bilevel problems but little has been done to solve multi-objective bilevel problems. One of the main reasons behind it is computational complexity and hierarchical decision-making process with conflicting objectives at both levels.

In [GD12] and [Ye11] present some results on optimality conditions in multi-objective bilevel problems. Eichfelder used classical approaches to solve in [Eic10] while Shi and Xia [SX01] used ϵ-constraint method at both upper and lower-levels to reshape the problem into ϵ-constraint bilevel problem. In this way, the authors reduced the problem to a single-level problem using KKT conditions. One of the first studies was proposed in [Yin00] and they studied nested genetic algorithms for solving a transportation planning and management problem. Particle swarm optimization based nested strategy used to solve the multicomponent chemical system in [HM06] for solving the problem with linear at the lower-level. A hybrid evolutionary algorithm proposed in [DS10c] for solving multi-objective bilevel problems coupled with local search. The authors worked on both nonlinear and discrete bilevel problems. In multi-objective problems, selection in the Pareto-frontier solutions is important, and there has been some work done on the decision-making perspective in both lower and upper-levels. For instance, [DS09b] the optimistic version of multi-objective bilevel problems are solved and the most preferred point is considered at the upper-level instead of all Pareto-frontier solutions. Similarly but at the lower-level, some works have been done in [SMD15a, SMD$^+$15b]. There are also several applications of multi-objective bilevel problems. For example, [BEA21] worked on solving the urban transit network design problem by multi-objective bilevel optimization lately. Also, [dVWR20] presented a model multi-objective bilevel model for the investment in gas infrastructures. We would like to refer to a good recent survey at this point for a more in-depth look for multi-objective bilevel optimization [MdDRMMM23]. Also, several works attempt to deal with uncertain environments in bilevel programming and we refer to a recent survey [BLS23] for more details.

2.6 Chapter Summary

This chapter reviews the definitions and notations relevant to this dissertation. Mainly, two mathematical structures: Bayesian optimization, and single- and multi-objective bilevel optimization have been introduced as they are being widely used in the technical chapters. A formal definition for the optimization problems has been given and detailed information on the field including modelling and search techniques has been presented. Subsequently, bilevel optimization problems with a single objective and with Pareto-optimal solutions in the case of multi-objective problems have been presented in detail. The chapter concludes by presenting the existing works that have been used widely to solve those problems. Subsequent chapters will introduce new advanced techniques in Bayesian optimization and some approaches that are developed to deal with single and multi-objective bilevel problems.

Chapter 3

Conditional Bayesian Algorithm for Bilevel Problems

The works presented in this chapter have been published in the following article:

- Vedat Dogan, Steven Prestwich. 2023. Bilevel Optimization by Conditional Bayesian Optimization. In Machine Learning, Optimization, and Data Science: 8th International Conference, LOD 2023, Grasmere, Lake District, UK, September 22–26, 2023,

Abstract. *In this chapter, we propose a method for solving bilevel optimization problems based on conditional Bayesian optimization. We first formulate the bilevel optimization problem. Then, we present a formulation based on Bayesian optimization for solving upper-level optimization by conditioning the lower-level decisions. We present empirical results on popular test benchmarks and compare our methods with the state-of-the-art. We conclude the chapter by identifying challenges and some future work.*

3.1 Introduction

In this chapter, we consider non-cooperative games called Stackelberg Games [Sta52], which are sequential non-zero-sum games with two players. In Section 2.3, we provided necessary definitions and a generalization of the bilevel problems. In this chapter, we focused on constrained and unconstrained standard bilevel problems for experiments.

A decision pair x_u, x_l^* represents a choice by the leader and an optimal feasible solution of the follower. The structure of Stackelberg games is asymmetric: the leader has perfect knowledge about the follower's objective and constraints, while the follower must first observe the leader's decisions before making its own optimal decisions. Any feasible bilevel solution should contain an optimal solution for the follower problem. In these problems, the lower-level (follower) optimization problem is a constraint of the upper-level (leader) problem. Because the lower-level solution is expected to be optimal given the upper-level decisions, these optimization problems are hard to solve. Bilevel optimization is known to be strongly NP-hard [HJS92] and it is known that even evaluating a solution for optimality is NP-hard [VSJ94a].

According to the observations from existing literature (Section 2), we can conclude the following issues:

- Classical approaches are limited in terms of the functions that they can handle even though they are relatively fast. Also, the problem can be converted to a single-level using complementary constraints but it needs some assumptions and is defined only as necessary conditions for optimality. Moreover, it often converges to the local optimum for non-convex problems.
- Evolutionary approaches are relatively better than classical approaches but they require a high number of function evaluations. Also, the optimality cannot be guaranteed.
- Hybrid approaches are popular in solving bilevel problems and efficient in solving constrained and unconstrained problems for single objective optimization problems. The majority of the works focused on hybrid evolutionary approaches as again a high number of function evaluations is required for convergence.

The primary motivation for the research presented in this chapter comes from the limitations of existing works described above. This work aims to be able to solve constrained and unconstrained, linear and non-linear problems while reducing the function evaluations as many as possible. The objective functions are treated as black-box at the upper-level, meaning we did not use the function characteristics such as linearity, convexity or derivatives as input for the algorithm. Moreover, we attempt the extract information from lower-level decisions to improve the upper-level decision-making process by looking at the correlation between them at each iteration. Towards this goal, we present a **Con**ditional **Ba**yesian Algorithm for **B**ilevel **P**roblems (ConBaBo) in this chapter. An extensive set of sixteen benchmark problems has been used to evaluate the performance and the performance compared with state-of-the-art in the literature.

3.2 Notation and Definitions

In this section, we formally define the notation of the BOP and formulate our notation of conditional BO and multi-objective acquisition approach to solve BOPs.

BO [Fra18] is a method used to optimize black-box functions that are expensive to evaluate. BO uses a probabilistic surrogate model, commonly BP [RW06], to obtain a posterior distribution $\mathbb{P}(\mathbf{f}|\mathcal{D})$ over the objective function $\mathbf{f}$ given the observed data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ where n represents the number of observed data. The surrogate model is assisted by an acquisition function to choose the next candidate or set of candidates $\mathbf{X} = \{x_j\}_{j=1}^{q}$ where q is the batch size of promising points found during optimization. The objective function is expensive but the surrogate-based acquisition function is not, so it can be optimized much more quickly than the true function to yield $\mathbf{X}$. It is discussed in more detail in Section 2. When we adapt the terminology to BOPs using the conditional optimization [PKG20], and assume that we have an expensive-to-evaluate black-box function that takes a state $s \in S$ and an action as an input $x \in X$, then the function returns a reward, as formulated in Equation 3.1,

$$f(s, x) : S \times X \rightarrow \mathbb{R}. \tag{3.1}$$

There is a distribution over states in conditional optimization $\mathbb{P}[x]$ which encodes the priority of states. The objective is to learn the best action for every state in general and by maximizing the expected output for all states, as in Equation 3.2,

$$\textit{Total Reward} = \int_S \mathbb{E}[f(s, x)]\mathbb{P}[s]ds. \tag{3.2}$$

Conditional algorithms are those that consider a set of functions, each corresponding to a state and for each iteration, the algorithm is willing to choose the actions according to the state. At a stage after obverving n data points, we have the data $\{(s_i, x_i, y_i)\}_{i=1}^{n}$ where $y_i = f(s_i, x_i)$. After we fit the observations to the BP from joint space $S \times X$ to the outputs $y \in \mathbb{R}$, we define the prior mean $\mu_0(s,x)$ and covariance function $k_0((s,x),(s',x'))$ as follows in Equation 3.3:

$$\begin{aligned} \mu_n(s,x) &= \mu_0(s,x) + k_0(s,x,\tilde{X})(K+\sigma_0^2 I)^{-1}(Y_n - \mu^0(\tilde{X}_n)) \\ k_n(s,x,s',x') &= k_0(s,x,s',x') + k_0(s,x,\tilde{X}_n)(K+\sigma_0^2 I)^{-1}k_0(\tilde{X}_n,s',x') \end{aligned} \tag{3.3}$$

where $\tilde{X}_n = ((s,x)_1,\ldots,(s,x)_n,)$, $Y_n = (y_1,\ldots,y_n)$ and $K = k_0(\tilde{X}_n,\tilde{X}_n) \in \mathbb{R}^{n\times n}$.

As [PKG20] declared conditional optimization is only necessary *with interaction*. There are several existing works focused on conditional optimization. For instance, the SCoT algorithm [BBKS13] works on discrete states and it visits each state in a round-robin function. After it chooses the actions using expected improvement. They applied their approach to hyperparameter optimization for machine learning algorithms for multiple datasets. The Profile Expected Improvement (PEI) [GBC+13] considers continuous states and they propose an acquisition function for selecting the next state-action pair by using expected improvement. The Profile q-Expected Improvement (PEQI) method [SRBG19] extends the PEI approach for noisy problems by using the SKO algorithm [HANM06]. We adapted the approach to the nested structure of BOPs which contains the interaction between decision-makers. Informally, a BOP is a nested optimization problem in which the lower-level appears as a constraint of the upper-level. In this work, we are going to use *leader* for the upper-level decision maker and *follower* for the lower-level decision maker.

Note that we use men-oriented language in this work (for instance, we say "his reaction" for the follower's reaction to the leader's decision), there are no sexist meanings to it. We also are going to consider dealing with the *optimistic* approach for solving BOPs. More detail is discussed in Section 2.3.

Let us first define the terminology to be used throughout this dissertation. For a given leader objective function $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and follower objective function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, bilevel optimization problem can be defined as in Equation 3.4;

$$\begin{aligned} &\underset{x_u,x_l}{\text{Minimize}} \;\; F(x_u,x_l) \\ &\quad \text{s.t.} \;\; x_l \in \underset{x_l}{\text{argmin}}\{f(x_u,x_l) : g_j(x_u,x_l) \le 0,\; j=1,2,\ldots,J\} \\ &\qquad G_k(x_u,x_l) \le 0,\; k=1,2,\ldots,K \end{aligned} \tag{3.4}$$

where $x_u \in \mathcal{X}_U, x_l \in \mathcal{X}_L$ are the vector-valued leader and follower decision variables in decision spaces. G_k represents the leader constraints while g_j represents the followers constraints. Because the follower's decision depends on the leader's variables, for every leader's decision x_u there is a follower-optimal decision x_l^* [SMD18a]. In BOPs, a decision set $x^* = (x_u, x_l^*)$ is feasible for the leader *only if* it satisfies all the leader's constraints. The vector x^* is an optimal solution to the follower's problem with the leader's decision as a parameter.

3.3 Proposed Approach

As declared in [SMD18a], for every leader's decision, there is an optimal follower decision, such that those members are considered feasible and satisfy the upper-level constraints. Because the follower problem is a parametric optimization problem that depends on the leader decision x_u, it is very time-consuming to adopt a nested strategy approach that sequentially solves both levels. In the continuous domain, the computational cost is very high. During the bilevel optimization process, it is important to choose wisely the next leader decision x_u, to speed up the search. So the question is: *How to choose the next leader decision x_u^*?* Treating the leader problem as a black box, and fitting the previous leader's decisions and best follower reactions to the BP model makes it possible to estimate the next point to evaluate wisely.

Recently an algorithm has been proposed based on BO [KDBN17], using differential evolution to optimize the acquisition function. It has been shown to perform better than an evolutionary algorithm based on quadratic approximations (BLEAQ) [SMD13a]. In this section, we proposed an algorithm by using the previous iterations, including the follower's decisions as priors and updated the acquisition function with *conditioning* on the follower's decisions to improve performance. Moreover, we investigated the contribution of follower decisions to optimizing the overall BOPs.

The follower can observe the leader's decisions, but if we treat leader fitness as a black box, then has been proposed then we also have available the best follower reactions to the leader's decisions during the optimization process. Using this data and extracting knowledge from previous iterations of the optimization process is the main idea of the proposed approach. Another question is: *Can the leader learn from the follower's decisions and use this knowledge for its own decision-making process?* Because of the structure of BOPs, the leader has no idea about the potential follower's reaction at the beginning of the decision-making. But in the following iterations, it can observe the feasible reaction of the follower without knowing the follower's objective, even if he is aware in some cases.

Following the idea above, let us assume that we have an expensive black-box function that takes leader decisions in leader decision space $x_u \in \mathcal{X}_u$ and the optimal follower's response $x_l^* \in \mathcal{X}_l$ as input. The upper-level function returns a scalar fitness score, $F(x_u, x_l^*) : \mathcal{X}_u \times \mathcal{X}_l \rightarrow \mathbb{R}$. Given a budget of N, the leader makes a decision and the follower responds to the leader accordingly. The leader can observe this information during the optimization process, learn how the follower reacts to the leader's decisions in every iteration, and choose the next leader's decision to optimize the fitness score.

First, we discussed fitting the decision data to the BP model, then the motivation behind conditioning. After observing n data points, that is leader decisions, follower responses, and leader's fitness score, $\{(x_{u_i}, x_{l_i}^*, y_i)\}_{i=1}^n$ where $y_i = F(x_{u_i}, x_{l_i}^*)$ is leader's objective. We fit the BP from $\mathcal{X}_u \times \mathcal{X}_l$ to $y \in \mathbb{R}$. After we have the vector-valued dataset $\tilde{\mathbf{X}}_n = \{(x_{u_1}, x_{l_1}^*), ..., (x_{u_n}, x_{l_n}^*)\}$ and $Y_n = \{y_1, ..., y_n\}$, then we construct the GP by a prior mean μ_0 and prior covariance function k_0. Let us suppose to find a value of $F(x)$ at some vector-valued decision x. So, the prior mean and covariance function are defined as follows as in Equation 3.5:

$$
\begin{aligned}
\mu_n(x_l^*, x_u) &= \mu_0(x_l^*, x_u) + k_0(x_l^*, x_u, \tilde{\mathbf{X}})(K + \sigma_0^2 I)^{-1}(Y_n - \mu^0(\tilde{\mathbf{X}}_n)) \\
k_n(x_l^*, x_u, (x_l^*)', x_u') &= k_0(x_l^*, x_u, (x_l^*)', x_u') + \\
&\qquad k_0(x_l^*, x_u, \tilde{\mathbf{X}}_n)(K + \sigma_0^2 I)^{-1} k_0(\tilde{\mathbf{X}}_n, (x_l^*)', x_u')
\end{aligned}
\tag{3.5}
$$

Note that there is an interaction between decision-makers in BOPs because of their hierarchical structure. Thus, it is important not to violate the leader's constraints during lower-level optimization. In this dissertation, we follow the bilevel definition in [SMD18a] to avoid violating the leader's constraints. As declared in the paper, the optimal follower's decision satisfies the leader's constraints. Every optimal follower's decision is a reaction to the leader's decision, which makes possible the use of conditional optimization.

After fitting the data to the model, we choose the next leader decision $x_{u_{n+1}}$. Then, we find the feasible reaction $x_{l_{n+1}}^*$ and the fitness score of the leader function $y_{n+1} = F(x_{u_{n+1}}, x_{l_{n+1}}^*)$. We update the GP model with new data. Then,

Algorithm 2 ConBaBo: Conditional Bayesian Optimization for Bilevel Optimization

Input: Random starting upper-level decisions with size of n
$\tilde{X}_{un} : \{x_{u_1}, \ldots, x_{u_n}\}$

1: Find the best *lower-level* responses $\tilde{X}_{ln} : \{x^*_{l_1}, \ldots, x^*_{l_n}\}$ regarding $\tilde{X}_{un}$ with *SLSQP* algorithm [Kra88]
2: Evaluate *upper-level* fitness values : $y_{1:n} = \{F(x_{u_1}, x^*_{l_1}), \ldots, F(x_{u_n}, x^*_{l_n})\}$
3: Initialize Gaussian process ($\mathcal{GP}$) model and fit the observed data :
4: $\mathcal{GP}(x_{u_{1:n}}, x^*_{l_{1:n}}, y_{1:n})$
5: Update the $\mathcal{GP}$ model
6: **for** i = 0 : *Budget* **do**
7: Evaluate the *conditional LCB acquisition function* value
8: Suggest the next candidate $x_{u_{n+i+1}}$ by optimizing *acquisition function*
9: Evaluate the best *lower-level* response $(x^*_{l_{n+i+1}})$ regarding $x_{u_{n+i+1}}$ using *SLSQP*
10: Calculate the *upper-level* fitness: $F(x_{u_{n+i+1}}, x^*_{l_{n+i+1}})$
11: Update the $\mathcal{GP}$ model with new observations
12: **return** the best decision values (x_u, x^*_l) with *upper-level* fitness value $F(x_u, x^*_l)$

we update the predicted optimal decisions for every iteration. After updating the model with the new follower feasible decision point at each iteration, we calculate the *conditional acquisition values*. For each follower optimal point x^*_l there is a global optimization problem over $x_u \in \mathcal{X}_u$. The lower confidence bound (LCB) acquisition function by [CJ97] is used to choose the next point. We choose the exploration weight $\beta = 2$ for the LCB acquisition function, as it is the default setting. Acquisition function by conditioning on the follower's decision is defined as in Equation 3.6:

$$\alpha_{LCB}(x^*_l, x_u) = \mu_n(x^*_l, x_u) - \beta^{\frac{1}{2}}\sigma_n(x^*_l, x_u) \tag{3.6}$$

where β is a parameter that balances exploration and exploitation during optimization. μ represents the conditional mean and σ represents the standard deviation. The conditional mean is calculated as follows:

$$\mu_n(x^*_l, x_u) = \mu_0(x^*_l, x_u) + k_0(x^*_l, x_u, \tilde{\mathbf{X}})(K + \sigma_0^2 I)^{-1}(Y_n - \mu^0(\tilde{\mathbf{X}}_n)) \tag{3.7}$$

where $\mu(\cdot)$ is mean function and $k(\cdot)$ is covariance function.

In Equation 3.7, we can see that the correlation between the conditioned data set and the leader decision data set affects the mean function representing the prediction. This change in posterior distribution is vital for the prediction of

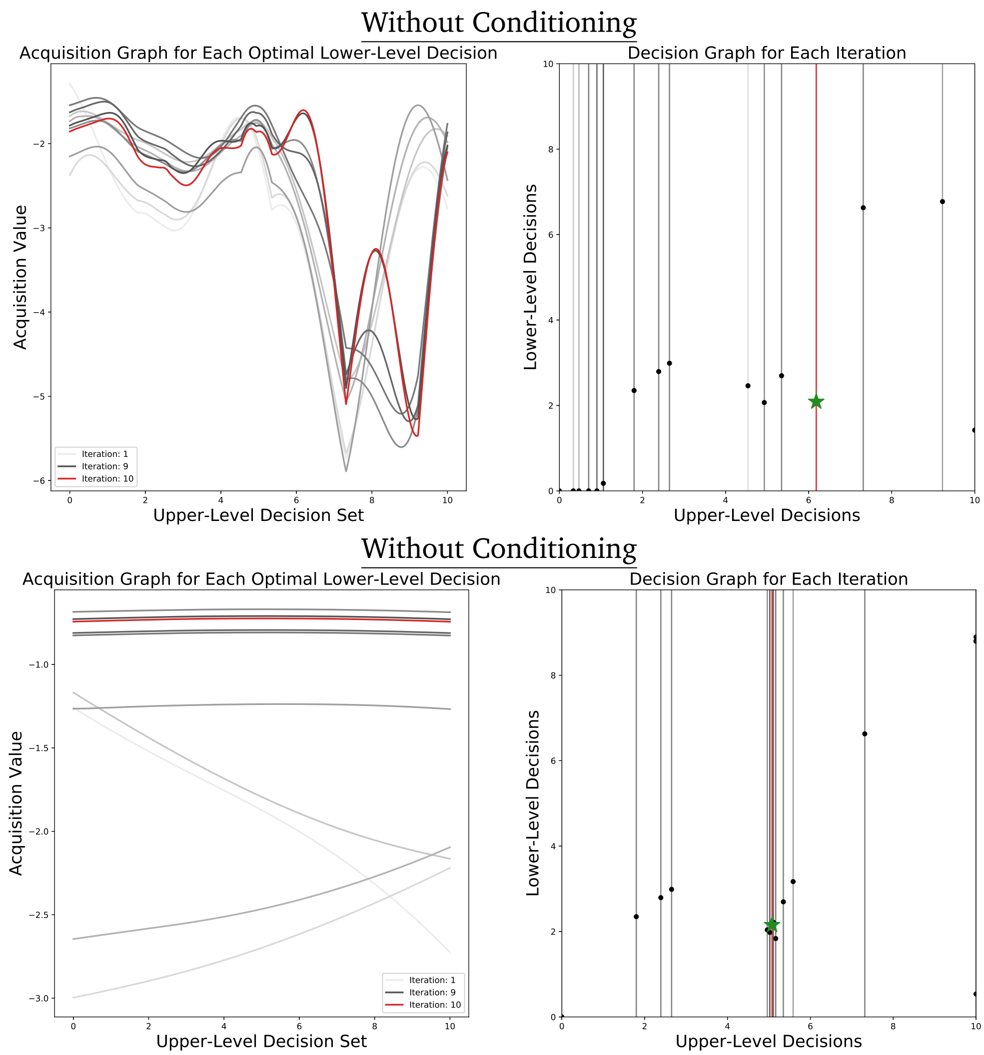


Figure 3.1: Two graphs on the right show the results for the CONBABO algorithm, and the two graphs on the right show the results without conditioning the lower-level decisions. The line's colour changes from light grey to dark grey in each iteration, and the red line is the shape of the acquisition function at the end. Vertical lines show the upper-level decisions and black dots show the decision pair (x_u, x_l^*). The red line shows the last decision pair after 10 iterations with 5 training points.

the GP. In this way, we used conditional setting on BOPs by improving the optimization with the follower's decisions.

We can see the observation for the first 10 iterations for the acquisition graph and how conditioning affects the optimization process in Figure 3.1. The left graph shows the acquisition function shape over 10 iterations without conditioning the follower's decision. The light grey colour represents the previous acquisition graph and the red one represents the 10^{th} iteration. During the GP, the acquisition function is the key point to select the next candidate and it is

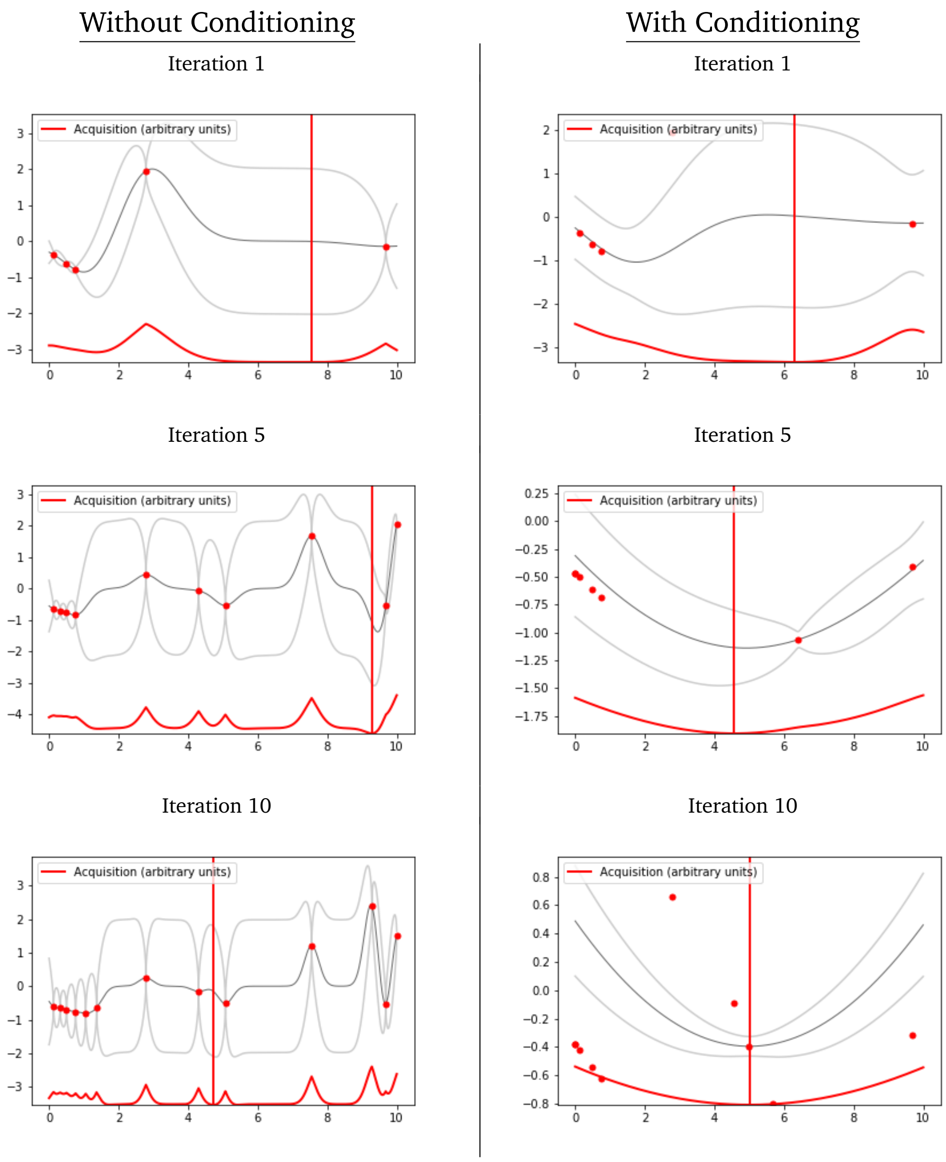


Figure 3.2: 1^{st},5^{th} and 10^{th} iterations of the acquisition function graph with and without conditioning the follower's decision.

done by optimizing the acquisition function. Thus, the acquisition function is essential for BO. As we can observe from the 3.1, the conditional acquisition function shape (on the right in Figure 3.1) is becoming convex even after 8 iterations while the standard acquisition function shape (on the left in Figure 3.1) is maintaining non-convexity. It shows the importance of gaining the lower-level optimal decision information for the upper-level decision makers. The graphs on the right on each side show the upper and lower-level decision space. The vertical lines show the upper-level decisions and the stars show the lower-level responses to the upper-level decision-makers. We shared these graphs to illustrate search behaviour during the acquisition optimization process. We can see clearly that conditioning the lower-level decisions affects the choice of decisions at both levels. It converges at global optima even in 8 iterations while the standard acquisition function tries to minimize the uncertainty at different points in the search space. Practical problems in the bilevel structure are extremely hard, and need time and consume computational power in most cases. So, we believe the improvement of CONBABO algorithm and the technique proposed provides a good technique for applying it to several practical problems.

We shared more acquisition graphs in a detailed way in Figure 3.2. It shows the acquisition graph for each iteration for 9 iteration. As we can see from Figure 3.2, the standard acquisition function behaves more explorative and tries different points at each step while trying to obtain a global optimal decision. As we can see from the graphs, the conditional acquisition function reached the near global optima even in 6 iterations. This fact makes us save lots of function evaluation during the bilevel optimization process. The acquisition function shape with conditioning becomes convex around global optima over the iterations quicker compared with the non-conditional setting. The CONBABO algorithm details are shown in Algorithm 2.

3.4 Numerical Experiments

In this section, the numerical experiments conducted using ConBabo on sixteen benchmark problems are discussed. The performance of ConBaBo is reported on these problems and compared with various state-of-the-art algorithms. This section first introduces these algorithms used for comparison and then follows the details of the experiments. Finally, we discussed the obtained results.

3.4.1 Parameters

All experiments were conducted on a single core of 1.4 GHz Quad Core i5, 8Gb 2133 Mhz LPDDR3 RAM. The algorithm is executed 31 times for each test function and the results shown are medians. The termination criterion is selected as $\epsilon = 10^{-6}$ representing the difference between the obtained result and the best result of compared algorithms. BO and GP are implemented in Python via the GPyOpt library [aut16] and *optimize_restarts* selected as 10 with the parameters of *verbose=False*, and *exact_feval=True*. The method initialized 5 Sobol points to construct the initial GP model. The LCB acquisition function has been selected for GP and the Radial Basis Function (RBF) kernel is used.

For optimizing the acquisition function, the L-BFGS method [LN89] has been used. The exploration-exploitation parameter for LCB, w is set to 2. After we set up the upper-level optimization method by conditional acquisition function, we used Sequential Least Squares Programming (SLSQP) [Kra88] for solving the lower-level problem in Python programming language. SLSQP is a gradient-based algorithm that can handle both bounds and constraints. For evaluating the CONBABO performance and making the comparison with the other selected algorithms, we set the termination criteria considering the best-obtained result of the other algorithms. The number of function evaluations is shared considering the Bayesian optimization iteration and compared with the selected algorithms median of 31 runs. We run the algorithm 31 with different random seeds to make the comparison fair.

3.4.2 Algorithms Used for Comparison

In this section, we present the selected algorithms from the literature for comparison which are briefly described below along with their references.

- **NBLE** [ISRS17]: This is a nested bilevel evolutionary algorithm that uses different evolutionary algorithms at both levels.
- **BLEAQ** [SMD13a]: Bilevel Evolutionary Algorithm with Quadratic Approximation (BLEAQ) focuses on reducing the computational cost by approximating to the lower-level optimum variables instead of using an explicit search. The lower-level optimum variables are being approximated as a quadratic function of the upper-level variables based on the truly evaluated solutions on previous iterations.
- **BOBP** [KDBN17]: The BOBP algorithm is the first attempt in the literature to apply BO to solve BOPs. It approximates upper-level decisions in a standard BO setting. It also has some similarities with the ConBaBo algorithm, however, ConBaBo uses followers' decisions in a conditional setting and improves the search technique compared with BOBP.
- **CGA-BS** [AEN20]: The algorithm is based on a genetic algorithm with new selection techniques based on chaos searching techniques. The paper attempts to solve BOPs more efficiently and faster than other algorithms by using the advantages of chaos theory.
- **BLMA** [ISRS17]: BLMA is a nested bilevel memetic algorithm that attempts to reduce the computational cost by combining the advantages of

global (Differential Evolution) and local search (Interior Point Algorithm) strategies to identify optimum solutions.

- **BIDE** [ISRS17]: This is a nested approach and uses a differential evolution based algorithm to optimize both levels in bilevel optimization problems. The needed function evolutions are very high because the exhaustive search is being used at both levels to locate the optimum. The reported results are being used in [ISRS17].

3.4.3 Numerical Experiments on TP Problems

We compared the experimental results with some state-of-art algorithms in the literature. To show the effectiveness of ConBaBo we compare it with the results in [SMD13a, KDBN17]. Sinha et al. present an improved evolutionary algorithm in [SMD14b] based on quadratic approximations of the lower-level problem, called BLEAQ. [KDBN17] propose an algorithm called BOBP with a Bayesian approach.

Table 3.1 shows the median fitness values at both upper- and lower-level for CONBABO and other algorithms. We also show the median function evaluations for upper- and lower-level optimization in Table 3.2. We did not share the standard deviation over multiple runs because we could not compare it with the other algorithms as there is no information presented about the confidence interval. The primary purpose of bilevel optimization is to solve the leader problem, so in comparing the algorithms we focus on leader fitness and function evaluations at each level.

As can be seen in Table 3.1, ConBaBo achieved better upper-level results for 7 problems TP2, TP4, TP5, TP6, TP7, TP8, and TP9 in terms of fitness scores obtained. A Wilcoxon rank-sum test has been realized to determine if the function evaluation differences are statistically significant. We compared the obtained result with the best of the compared algorithms. One of the test sets was constructed with the obtained results from the proposed algorithm and the other one was constructed with the best algorithm from the others. The p-value obtained from the test should be less than the significance level α, which is generally $\%0.05$. The obtained p-value for upper-level function evaluations is 0.0019. So as it is $\leq \alpha$, the results show the results are statistically significant compared with the best obtained results of the other algorithms. The number of function calls by the leader decreases significantly as we encounter near-optimal solutions at both levels as we can see in Table 3.2. The test benchmark has constraint problems with various dimensional decision variables. The CONBABO results are promising when we consider the complexity of the problems in the benchmark. We can see that ConBaBo performs well on this standard test set.

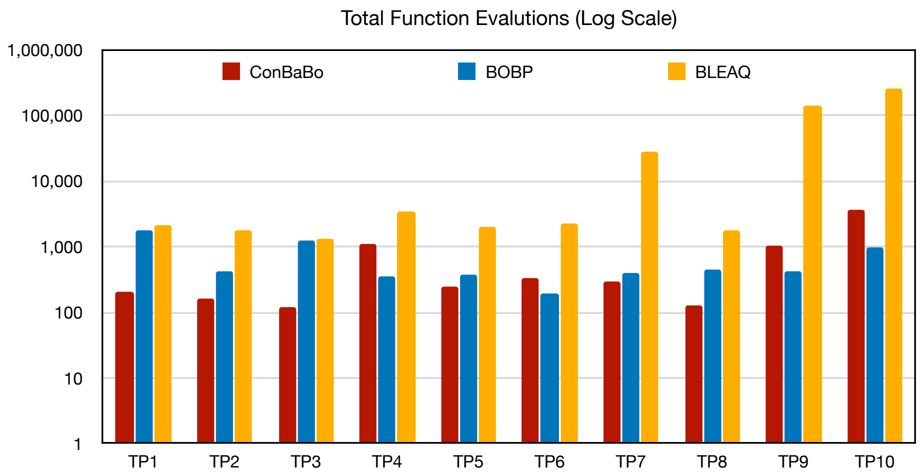


Figure 3.3: Left: Median of total function evaluations for TP problems in log scale.

Table 3.1: Median Upper level and lower level fitness scores of ConBaBo, BOBP, BLEAQ algorithms for TP1 to TP10.

	Median Upper-level (UL) and Lower-level (LL) Fitness Scores					
	ConBaBo		BOBP		BLEAQ	
	UL	LL	UL	LL	UL	LL
TP1	226.0223	98.0075	253.6155	70.3817	224.9989	99.9994
TP2	0.0000	200.0000	0.0007	183.871	2.4352	93.5484
TP3	-18.0155	-1.0155	-18.5579	-0.9493	-18.6787	-1.0156
TP4	-30.9680	3.1472	-27.6225	3.3012	-29.2	3.2
TP5	-4.3105	-1.5547	-3.8516	-2.2314	-3.4861	-2.569
TP6	-1.2223	7.6751	-1.2097	7.6168	-1.2099	7.6173
TP7	-1.9001	1.9001	-1.6747	1.6747	-1.9538	1.9538
TP8	0.0000	200.0000	0.0008	180.6452	1.1463	132.5594
TP9	0.00003	1.0000	0.0012	1.0000	1.2642	1.0000
TP10	0.000285	1.0000	0.0049	1.000	0.0001	1.0000

Table 3.2: Median function upper level and lower level function evaluations for ConBaBo and other known algorithms for TP1 to TP10

	Median Upper-level (UL) and Lower-level (LL) Function Evaluations					
	ConBaBo		BOBP		BLEAQ	
	UL	LL	UL	LL	UL	LL
TP1	18.1333	190.6646	211.1333	1,558.8667	588.6129	1543.6129
TP2	17.6772	148.2746	35.2581	383.0645	366.8387	1,396.1935
TP3	10.1836	111.1374	89.6774	1,128.7097	290.6452	973.0000
TP4	14.6129	1,063.901	16.9677	334.6774	560.6452	2,937.3871
TP5	23.5164	222.2308	57.2258	319.7742	403.6452	1,605.9355
TP6	8.3333	323.1333	12.1935	182.3871	555.3226	1,689.5484
TP7	13.7092	274.9229	72.9615	320.2308	494.6129	26,682.4194
TP8	14.6066	114.9942	37.7097	413.7742	372.3226	1,418.1935
TP9	8.9215	1,020.8618	16.6875	396.3125	1,512.5161	141,303.7097
TP10	15.1935	3,689.6452	21.3226	974.0000	1,847.1000	245,157.9000

3.4.4 Numerical Experiments on SMD Problems

SMD problems are scalable in terms of the number of decision variables. We compared our results with CGA-BS, BLMA [ISRS17], NBLE [ISRS17], BIDE [ISRS17], and BLEAQ algorithms. Table 3.3 shows the upper-level fitness comparison with the other algorithms.

We shared the upper-level and lower-level median function evaluations at Table 3.4 and Table 3.5 to show the effectiveness of the CONBABO algorithm. Also for SMD problems, we did not share the standard deviation over multiple runs because of the same reason with TP problems. We can observe that from Tables 3.4 and 3.5 that the upper- and lower-level function evaluations decrease significantly. We run the Wilcoxon rank-sum test to determine if the function evaluation difference is significant or not. One of the test sets was constructed with the obtained results from the proposed algorithm and the other one was constructed with the best algorithm from the others. The obtained p-value is 0.0031. These results show that ConBaBo can manage the difficulties resulting from the proposed SMD benchmark by [SMD12a]. Moreover, it converges faster to optimal solutions compared to other algorithms in the literature. The problems in the benchmark are scalable and have different difficulties in terms of the interaction between decision-makers. Considering this, CONBABO performs well on this unconstrained bilevel benchmark problem.

Table 3.3: ConBaBo, CGA-BS, BLMA, NBLE, BIDE and BLEAQ upper level fitness for SMD1 to SMD6.

	Median Upper-level Fitness Scores					
	ConBaBo	CGA-BS	BLMA	NBLE	BIDE	BLEAQ
SMD1	1.8×10^{-6}	0	1.0×10^{-6}	5.03×10^{-6}	3.41×10^{-6}	1.0×10^{-6}
SMD2	9.09×10^{-6}	2.22×10^{-6}	1.0×10^{-6}	3.17×10^{-6}	1.29×10^{-6}	5.44×10^{-6}
SMD3	1.6×10^{-6}	0	1.0×10^{-6}	1.37×10^{-6}	4.1×10^{-6}	7.55×10^{-6}
SMD4	6.2×10^{-6}	3.41×10^{-11}	1.0×10^{-6}	9.29×10^{-6}	2.3×10^{-6}	1.0×10^{-6}
SMD5	1.49×10^{-6}	1.13×10^{-9}	1.0×10^{-6}	1.0×10^{-6}	1.58×10^{-6}	1.0×10^{-6}
SMD6	2.36×10^{-6}	9.34×10^{-11}	1.0×10^{-6}	1.0×10^{-6}	3.47×10^{-6}	1.0×10^{-6}

Table 3.4: Median function upper level function evaluations for ConBaBo and other known algorithms for SMD1 to SMD6

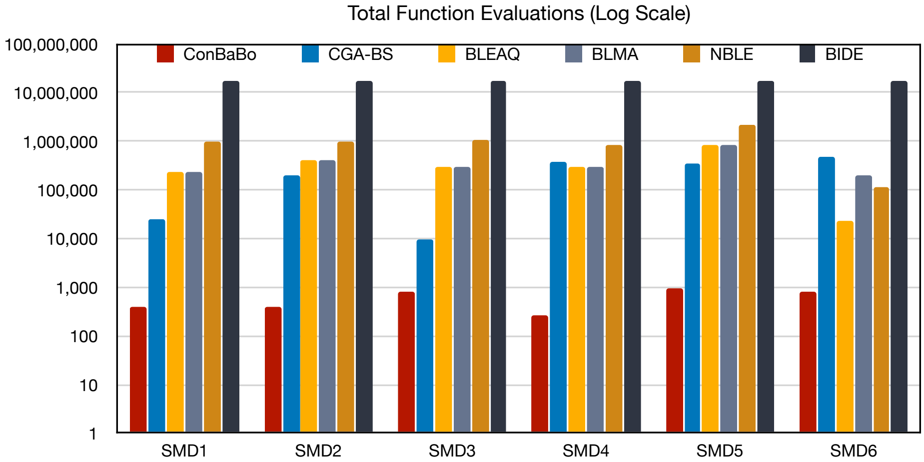
	Median Upper-level Function Evaluations					
	ConBaBo	CGA-BS	BLMA	NBLE	BIDE	BLEAQ
SMD1	1.1×10^{1}	1.01×10^{4}	1.19×10^{3}	1.52×10^{3}	6.0×10^{3}	1.19×10^{3}
SMD2	1.4×10^{1}	5.0×10^{4}	1.20×10^{3}	1.56×10^{3}	6.0×10^{3}	1.20×10^{3}
SMD3	2.3×10^{1}	1.0×10^{4}	1.29×10^{3}	1.56×10^{3}	6.0×10^{3}	1.29×10^{3}
SMD4	1.01×10^{1}	1.25×10^{5}	1.31×10^{3}	1.53×10^{3}	6.0×10^{3}	1.31×10^{3}
SMD5	2.37×10^{1}	1.0×10^{5}	2.06×10^{3}	3.40×10^{3}	6.0×10^{3}	2.06×10^{3}
SMD6	2.41×10^{1}	1.37×10^{5}	4.08×10^{3}	4.06×10^{3}	6.0×10^{3}	4.08×10^{3}

Table 3.5: Median function lower level function evaluations for ConBaBo and other known algorithms for SMD1 to SMD6

	Median Lower-level Function Evaluations					
	ConBaBo	CGA-BS	BLMA	NBLE	BIDE	BLEAQ
SMD1	3.8×10^{2}	1.5×10^{4}	2.37×10^{5}	9.520×10^{5}	1.8×10^{7}	2.37×10^{5}
SMD2	3.81×10^{2}	1.5×10^{5}	4.08×10^{5}	9.63×10^{5}	1.8×10^{7}	4.08×10^{5}
SMD3	7.7×10^{2}	2.0×10^{4}	3.02×10^{5}	1.04×10^{6}	1.8×10^{7}	3.02×10^{5}
SMD4	2.58×10^{2}	2.58×10^{5}	3.07×10^{5}	8.33×10^{5}	1.8×10^{7}	3.07×10^{5}
SMD5	9.05×10^{2}	2.58×10^{5}	8.42×10^{5}	2.22×10^{6}	1.8×10^{7}	8.42×10^{5}
SMD6	7.69×10^{2}	3.39×10^{5}	1.98×10^{4}	1.11×10^{5}	1.8×10^{7}	1.98×10^{4}

3.5 Conclusion

BOPs are a specific kind of problem that feature two decision makers, with a mathematical representation called “Stackelberg Games” in Game Theory. There are two optimization problems in these problems called the “upper-level” and “lower-level”. Decisions in each level are affected by those made in the other, so the bilevel optimization scheme is time-consuming in terms of performance. We propose CONBABO, a conditional Bayesian optimization algorithm



'

Figure 3.4: Left: Median of total function evaluations for SMD problems in log scale.

for bilevel optimization problems. The conditional Bayesian approach allows us to extract knowledge from previous upper- and lower-level decisions, leading to smarter choices and therefore fewer function evaluations. ConBaBo increases algorithm performance quality and dramatically accelerates the search for an optimal solution. We evaluate our method on two common benchmark sets from the bilevel literature, comparing results with those for top-performing algorithms in the literature. The ConBaBo algorithm can be considered a powerful global technique for solving bilevel problems, which can handle the difficulties of non-linearity and conflict between decision-makers. Moreover, it can deal with constrained and unconstrained problems with multi-dimensional decision variables.

Chapter 4

Bayesian Algorithm with Multi-objective Acquisition Function for Bilevel Problems

The works presented in this chapter have been published in the following article:

- Dogan, V., Prestwich, S. (2022). Bayesian Optimization with Multi-objective Acquisition Function for Bilevel Problems. In: Longo, L., O'Reilly, R. (eds) Artificial Intelligence and Cognitive Science. AICS 2022. Communications in Computer and Information Science, vol 1662. Springer, Cham. https://doi.org/10.1007/978-3-031-26438-2_32

Abstract. *In this chapter, we propose a method based on multi-objective acquisition optimization during Bayesian optimization for solving bilevel problems. First, we present the necessary background and define multi-objective optimization problems. After we present the Pareto optimality, we propose a hybrid approach with the improvement of Bayesian optimization with multi-objective acquisition function optimization processes, called MACBP. Subsequently, we perform experiments on a popular bilevel problem benchmark. We first report the performance and compare it with existing state-of-the-art algorithms. Finally, we report how the approach can be performed on many.*

4.1 Introduction

BO has been applied so far successfully to single-level optimization problems to solve with a lower number of function evaluations. It is preferred generally when the true function evaluations are computationally expensive. In the domain of bilevel optimization, few works have been done that applied BO to solve BOPs as we discussed in Chapter 2. From the existing literature, we can conclude that the benefit of black-box approaches cannot be underestimated as they improve the function evaluations during optimization significantly. BOBP [KDBN17] and ConBaBo, which is presented in Chapter 3 successfully applied to improve algorithmic performance for solving BOPs. Lately, [LYY+18a] presented the benefits of using multiple acquisitions during BO.

In this chapter, we present an algorithm to solve BOPs with BO via multiple acquisition functions, which we call MACBP. The details of MACBP are discussed in Section 4.2. Numerical experiments are conducted on a set of six linear and nonlinear unconstrained problems. The performance is compared with existing strategies (CGA-BS, BLMA, NBLE, BIDE, and BLEAQ) to show the benefits of the proposed approach in Section 4.3.

4.2 Proposed Approach

In this section, the details of the proposed approach are discussed. The description of the MACBP algorithm will be divided into three parts. Firstly, we explain the notation used in this section for BOPs and their structure. Secondly, we discuss BO and GP. Finally, we propose the MACBP algorithm for solving BOPs.

4.2.1 Overall Framework

For the upper-level objective function $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and lower-level objective function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, the bilevel optimization problem can be defined as in Equation 4.1

$$\begin{aligned} &\min_{\mathbf{x}_u \in X_u, \mathbf{x}_l \in X_l} F(\mathbf{x}_u, \mathbf{x}_l) \\ &\quad \text{s.t.} \;\; \mathbf{x}_l \in \underset{\mathbf{x}_l}{\operatorname{argmin}} \{ f(\mathbf{x}_u, \mathbf{x}_l) : g_j(\mathbf{x}_u, \mathbf{x}_l) \leq 0, j = 1, 2, \ldots, J \} \\ &\qquad\quad G_k(\mathbf{x}_u, \mathbf{x}_l) \leq 0, \;\; k = 1, 2, \ldots, K \end{aligned} \tag{4.1}$$

where $\mathbf{x}_u \in X_U, \mathbf{x}_l \in X_L$ are upper-level and lower-level decision variables and decision spaces, G_k, g_j are constraints. Because the lower-level decision maker depends on the upper-level variables, for every decision x_u, there is a follower-optimal decision x_l^*. In bilevel optimization, the decision set $\mathbf{x}^* = (\mathbf{x}_u^*, \mathbf{x}_l^*)$ is a feasible member for the upper-level *only if* it satisfies all the upper-level constraints and vector $\mathbf{x}_x^*$ is an optimal solution to the lower-level problem with the upper-level decision as *parameter*.

BO is a method to optimize expensive-to-evaluate black-box functions. The probabilistic surrogate model and acquisition functions is important for BO. Predictions and uncertainties are provided by the surrogate model. It uses commonly GP [RW09] as a surrogate model, to obtain a posterior distribution $\mathbb{P}(\mathbf{f}|D)$ over the objective function $\mathbf{f}$ given the observed data $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$. An acquisition function uses the posterior distribution to explore the search space. So the surrogate model is assisted by an *acquisition function* to choose the next candidate or a set of candidates $X_{cand} = \{\mathbf{x}_i\}_{i=1}^q$. Though the objective function is expensive to evaluate, the surrogate-based acquisition function is not, so it can be optimized much more easily than the true function to yield X_{cand}.

Let us assume that we have a set of collection points $\{x_1, \ldots, x_n\} \in \mathbb{R}^d$ and an objective function values of these points $\{f(x_1), \ldots, f(x_n)\}$. After we observe n points, the mean vector is obtained by evaluating a *mean function* μ_0 at each point x_i and the covariance matrix by evaluating a *covariance function* or *kernel* Σ_0 at each pair of x_i, x_j. The resulting prior distribution on $\{f(x_1), \ldots, f(x_n)\}$ is defined by

$$f(x_{1:n}) \sim N(\mu_0(x_{1:n}), \Sigma_0(x_{1:n}, x_{1:n})) \tag{4.2}$$

Let us suppose we wish to find a value of $f(X_{cand})$ at some new candidate point X_{cand}. For this purpose, the prior over $\{f(x_{1:n}), f(X_{cand})\}$ is given by (4.2). Then we can compute the distribution of $f(X_{cand})$ given the observations in Equations 4.3,4.4, and 4.5.

$$f(X_{cand})|f(x_{1:n}) \sim N(\mu_0(X_{cand}), \sigma_0^2(X_{cand})) \tag{4.3}$$

$$\mu_0(X_{cand}) = \Sigma_0(X_{cand}, x_{1:n})\Sigma_0(x_{1:n}, x_{1:n})^{-1}(f(x_{1:n}) - \mu_0(x_{1:n})) + \mu_0(X_{cand}) \tag{4.4}$$

$$\sigma_n^2(X_{cand}) = \Sigma_0(X_{cand}, X_{cand}) - \Sigma_0(X_{cand}, x_{1:n})(\Sigma_0(x_{1:n}, x_{1:n})^{-1}\Sigma_0(x_{1:n}, X_{cand}) \tag{4.5}$$

The distribution is called the *posterior probability distribution* in Bayesian statistics. So it is very important during the BO and GP to choose the next point to evaluate.

Acquisition functions are used to guide the search to a promising next point during the likelihood optimization, and it balances exploration and exploitation. Several acquisition functions have been developed over the years, such as the probability of improvement (PI), expected improvement (EI) and upper confidence bound (UCB).

- **Probability of Improvement** The PI acquisition function tries to measure the probability that an arbitrary x exceeds the current best. Given the minimum objective function value τ in the data set, the formulation is as follows [Kus63]:
$$PI(x) = \Phi(\lambda) \tag{4.6}$$
where $\Phi(\lambda)$ is the cumulative distribution function of standard normal distribution and $\lambda = (\tau - \mu(x))/(\sigma(x))$.

- **Expected Improvement** We can expect that the observation x will not only reach the current best but also, reach the current best value at the highest magnitude. The corresponding formulation can be expressed as [SKKS09]:
$$EI(x) = \sigma(x)(\lambda\Phi(\lambda) + \phi(\lambda)) \tag{4.7}$$
where $\phi(\cdot)$ is probability density function of standard normal distribution and $\lambda = (\tau - \mu(x))/(\sigma(x))$.

- **Upper Confidence Bound** This is not an improvement-based strategy like EI and PI. It tries to guide the search from an optimistic perspective. The formulation is:
$$UCB(x) = \mu(x) + \beta\sigma(x) \tag{4.8}$$
where β is a parameter that represents the exploration-exploitation trade-off. We fix $\beta = 0.1$.

BOPs have two levels of optimization tasks, such that the lower-level problem is a constraint of the upper-level problem. In general BOPs, the follower depends on the leader's decisions x_u. The leader has no control over the follower's decision x_l. For every leader's decision, there is an optimal follower decision, which can be called the reaction. Because the follower problem is a parametric optimization problem that depends on the leader decision x_u, it is very time-

Algorithm 3 The MACBP Algorithm for Upper-level Optimization

Inputs: $\mathbf{F}_u(\mathbf{x}_u, \mathbf{x}_l) : \mathbf{x_u} \in \mathbb{X}_u, \mathbf{x_l} \in \mathbb{X}_l$,
Total epoch *N*,
Initial decision data set $D = (\mathbf{x}_{u_i}, F_u(\mathbf{x}_{u_i}, \mathbf{x}^*_{l_i}))_{i=1:n}$ with size of n,

$\mathbf{x_l^*}$: Find the best lower-level decisions and set as upper-level parameters,
Initialize *Gaussian model* with Observations $\{\mathbf{x}_u, F_u(\mathbf{x}_u, \mathbf{x}^*_l)\}$
for i = 0 : *N* **do**
    Construct the acquisition functions according to the equations (4.8), (4.7) and (4.6)
    Find Pareto-front of multi-objective acquisition problem using the NSGA-II algorithm
    For the chosen upper-level decision $\mathbf{x}_u$, find optimal $\mathbf{x}^*_l$ by using SLSQP
    Calculate fitness scores $\mathbf{F}^*_u$ and $\mathbf{f}^*_u$
    Update the data set $D = (\mathbf{x}_{u_i}, \mathbf{F}_u(\mathbf{x}_{u_i}, \mathbf{x}^*_{l_i}))_{i=1:q}$
    Update Multi-objective Gaussian Process Model with new observations

Return Best F^*_u and corresponding optimum variables $\mathbf{x}^*_u, \mathbf{x}^*_l$

consuming to adopt a nested strategy approach that sequentially solves both levels. In the continuous domain, the computational cost is very high. During the optimization process, it is important to choose wisely the next leader decision x_u to make the process faster. For this purpose, we will present the proposed algorithm, which we call MACBP, for solving bilevel problems by BO via multiple acquisition functions.

Let us assume that we have an expensive black-box function that takes leader decisions in leader decision space $x_u \in X_u$ and follower decisions coming from the follower decision maker $x_l \in X_l$ as input. The function returns a scalar fitness score calculated as in Equation 4.9:

$$F(x_u, x_l) : X_u \times X_l \to \mathbb{R} \tag{4.9}$$

Given a budget of N, the leader makes a decision and the follower makes its decisions accordingly. The leader can observe this information during the optimization process, and how follower decision-makers react to leader decisions in every iteration and choose the next leader's decision to optimize the fitness score.

First, we discuss fitting the decision data to the GP model. After observing n decision data $\{(x_u^i, y^i)\}_{i=1}^n$ where $y_i = F(x_u^i, x_l^i)$, we fit the data set to the Gaussian process model. After we have the data set let $\hat{X}^n = ((x_u)^1, ..., (x_u)^n)$

and $Y^n = (y^1, ..., y^n)$, then we define the Gaussian process by a prior mean $\mu(x_u)$ and prior covariance function $k((x_u), (x_u^{'}))$. After observing n data points, let $K = k(\hat{X}^n, \hat{X}^n) \in \mathbb{R}^{n \times n}$. So the posterior mean and covariance are given by as in Equations 4.10 and 4.11:

$$\mu(x_u)^n = \mu(x_u) + k(x_u, \hat{X}^n)(K + \sigma_0^2 I)^{-1}(Y^n - \mu(\hat{X}^n)) \tag{4.10}$$

$$k^n(x_u, x_u^{'})^n = k^n(x_u, x_u^{'}) - k^n(x_u, \hat{X}^n)(K + \sigma_0^2 I)^{-1} k(\hat{X}^n, x_u^{'}) \tag{4.11}$$

After fitting the data to the model, we choose the next leader decision. After we find the optimal reaction (x_u^{n+1}, x_l^{n+1}) and the fitness score of leader function $F(x_u^{n+1}, x_l^{n+1}) = y^{n+1}$, we update the Gaussian process model with new decision data (x_u^{n+1}, y^{n+1}). We shared the details of the MACBP algorithm on Algorithm 3 for upper-level optimization.

There are multiple objectives to optimize when we consider the multi-objective optimization problems. It is formulated as in Equation 4.12

$$\underset{\mathbf{x} \in X}{\text{minimize}} \;\; \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_d(\mathbf{x})) \tag{4.12}$$

for a vector-valued function $\mathbf{f}(\mathbf{x}) : \mathbb{R} \to \mathbb{R}^d$ and $X \in \mathbb{R}$. So it is hard and commonly impossible to find a single optimum solution as there may be conflicts between the objectives. Therefore the main goal for these problems is to approximate the Pareto-front. Let us say that $\mathbf{f}(\mathbf{x})$ *dominates* another solution $\mathbf{f}(\mathbf{x}')$ if $\mathbf{f}^{(i)}(x) \succ \mathbf{f}^{(i)}(x')$ for all $i = 1, 2, \ldots, M$ and there exists $i' \in \{1, 2, \ldots, M\}$ such that $f^{i'}(x) \succ f^{i'}(x')$. So we can express the *Pareto-optimal* by $P^* = \{\mathbf{f}(\mathbf{x})$ *s.t.* $\nexists \mathbf{x}' \in X : \mathbf{f}(\mathbf{x}') \succ \mathbf{f}(\mathbf{x})\}$ and $X^* = \{\mathbf{x} \in \mathbf{X}$ *s.t.* $\mathbf{f}(\mathbf{x}) \in P^*\}$. A solution set is Pareto-optimal if it is not dominated by any other point and dominates at least one point. The Pareto-set is the set of all Pareto-optimal points, and a set of Pareto-optimal points is called a Pareto-front. There are many multi-objective optimization algorithms such as non-dominated sorting-based genetic algorithm (NSGA-II), multi-objective evolutionary algorithm based on decomposition (MOEA/D) [ZL07] and multi-objective optimization based on differential evolution (DEMO) [TF05].

Different acquisition functions have different characteristics according to their structure and point selection strategy. Improvement-based strategies rely on the best selection so far at each iteration. For example, the PI function value decreases when the difference between the mean function and the best objective value so far below zero, $\mu(x) - F^*(x) < 0$. The EI function value at sampled points would always be worse than the EI values at pending decision points. Uncertainty-based acquisition functions, for instance, UCB, increase as $\sigma(x)$ increases.

According to the different selection strategies explained above, we use the multi-objective optimization method NSGA-II in this work, to find the best trade-off between acquisition functions. It is a variant of the genetic algorithm and because there are various kinds of studies and analyses available on this method [DPAM02], it is not discussed further in this chapter. Then we select the next point in the leader's decision during the bilevel optimization process from the best trade-off between acquisition functions. This is called the Pareto-front of acquisition functions. So in every iteration, the multi-objective optimization problem constructed is:

$$\underset{\mathbf{x}\in X}{\text{minimize}} \left\{ -UCB(x), -PI(x), -EI(x) \right\} \tag{4.13}$$

After we find the Pareto-front from the multi-objective optimization Problem 4.13 we make the random selection from the Pareto-optimal decision set.

4.3 Numerical Experiments

4.3.1 Parameters

We evaluate the MACBP algorithm using two experiments. First, we run the experiments by choosing a single point at each iteration for the setting of $N_{iter} = 50$. We set the number of initial random sampling to $N_{init} = 20$. Then, we compare the results with those for the three single acquisition functions EI, PI and UCB performances. Second, we run the experiment with stopping criteria of $d < 10^{-5}$ where d represents the difference between the results and the optimum value of functions. We compare the performance of our proposed method in terms of function evaluations in Table 4.3 and in terms of accuracy in Table 4.2. We run the algorithm in sequential mode and the Matern52 kernel is used for GP for both experiments. The parameters for acquisition functions are declared in Section 3.2. For the first experiment, the experiments are repeated 31 times to average the random fluctuations and the optimality gap in the log scale presented in Figure 4.1.

The optimization is completed in a single core of 1.4 GHz Quad Core i5, 8Gb

2133 MHz LPDDR3 RAM. Bayesian optimization is implemented in the Python language and uses BoTorch [PGM+19], the SLSQP algorithm [Kra88] is used for lower-level optimization, and the NSGA-II algorithm for multi-objective optimization by using PyMOO [BD20] library.

4.3.2 Algorithms Used for Comparison

The comparison of the proposed approach with the existing algorithms in the literature is done based on objective values and computational effort on 6 unconstrained BOPs which are discussed in Chapter 2, Section 2.4.1.2. The selected algorithms for comparison are discussed below:

- **NBLE** [ISRS17]: This is a nested bilevel evolutionary algorithm that uses different evolutionary algorithms at both levels.
- **BLEAQ** [SMD13a]: Bilevel Evolutionary Algorithm with Quadratic Approximation (BLEAQ) focuses on reducing the computational cost by approximating to the lower-level optimum variables instead of using an explicit search. The lower-level optimum variables are being approximated as a quadratic function of the upper-level variables based on the truly evaluated solutions on previous iterations.
- **CGA-BS** [AEN20]: The algorithm is based on a genetic algorithm with new selection techniques based on chaos searching techniques. The algorithm attempts to solve bilevel optimization problems more efficiently and faster than other algorithms by using the advantages of chaos theory.
- **BLMA** [ISRS17]: BLMA is a nested bilevel memetic algorithm that attempts to reduce the computational cost by combining the advantages of global (Differential Evolution) and local search (Interior Point Algorithm) strategies to identify optimum solutions.
- **BIDE** [ISRS17]: This is a nested approach and uses a differential evolution based algorithm to optimize both levels in bilevel optimization problems. The needed function evolutions are very high because the exhaustive search is being used at both levels to locate the optimum. The reported results are being used as in the reference.

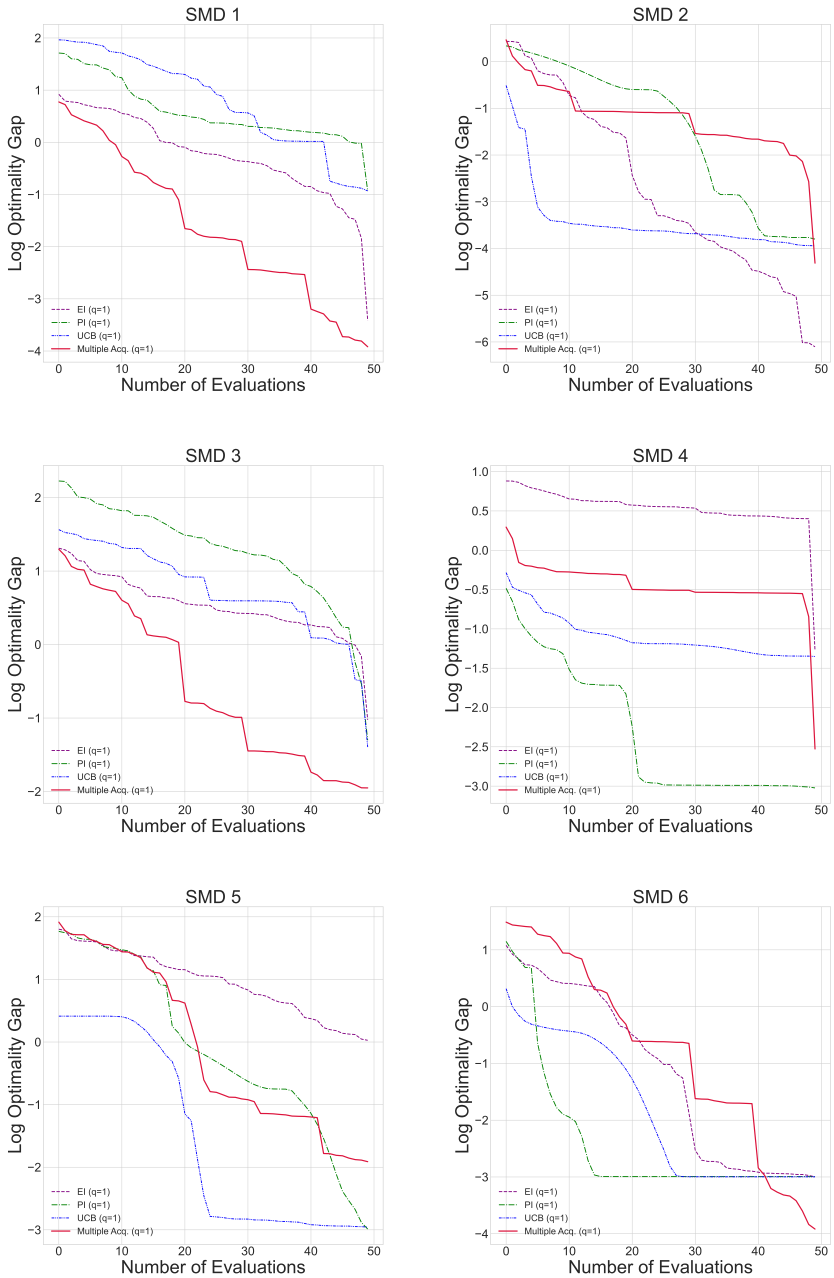


Figure 4.1: The log optimality gap for the leader's objective in SMD benchmark problems.

4.3.3 Numerical Experiments

We evaluated the MACBP algorithm on six standard benchmark problems proposed in [SMD12b]. It is called SMD test problems and the problems are unconstrained and high-dimensional with controllable complexities. They are scalable in terms of the number of decision variables. Each problem in the benchmark represents a different difficulty level in terms of convergence, the complexity of

Table 4.1: The summary of SMD benchmark problems

	Dimension		Search Domain		Optimal Value	
	UL	LL	UL	LL	UL	LL
SMD1	2	2	[-5,10] × [-5,10]	[-5,10] × [-$\frac{\pi}{2}$,$\frac{\pi}{2}$]	0.0	0.0
SMD2	2	2	[-5,10] × [-5,1]	[-5,10] × (0, e]	0.0	0.0
SMD3	2	2	[-5,10] × [-5,10]	[-5,10] × [-$\frac{\pi}{2}$,$\frac{\pi}{2}$]	0.0	0.0
SMD4	2	2	[-5,10] × [-1,1]	[-5,10] × (0, e]	0.0	0.0
SMD5	2	2	[-5,10] × [-5,10]	[-5,10] × [-5,10]	0.0	0.0
SMD6	2	2	[-5,10] × [-5,10]	[-5,10] × [-5,10]	0.0	0.0

Table 4.2: Upper-level accuracy for the proposed MACBP algorithm and other known algorithms for SMD1-SMD6.

	Upper-Level Accuracy					
	MACBP	CGA-BS [AEN20]	BLMA [ISRS17]	NBLE [ISRS17]	BIDE [ISRS17]	BLEAQ [SMD14b]
SMD1	2.51×10^{-6}	0	1.00×10^{-6}	5.03×10^{-6}	3.41×10^{-6}	1.00×10^{-6}
SMD2	1.33×10^{-6}	2.22×10^{-6}	1.00×10^{-6}	3.17×10^{-6}	1.29×10^{-6}	5.44×10^{-6}
SMD3	3.82×10^{-6}	0	1.00×10^{-6}	1.37×10^{-6}	4.10×10^{-6}	7.55×10^{-6}
SMD4	6.27×10^{-7}	3.41×10^{-11}	1.00×10^{-6}	9.29×10^{-6}	2.30×10^{-6}	1.00×10^{-6}
SMD5	7.44×10^{-7}	1.13×10^{-9}	1.00×10^{-6}	1.00×10^{-6}	1.58×10^{-6}	1.00×10^{-6}
SMD6	1.09×10^{-7}	9.34×10^{-11}	1.00×10^{-6}	1.00×10^{-6}	3.47×10^{-6}	1.00×10^{-6}

interaction and lower-level multi-modality as declared in [SMD12b]. Table 4.1 provides details on the problems. For all functions, we used 2D decision variables. The total function evaluations for the leader's objective can be calculated by $N_{iter} + N_{init}$.

Although BOPs deal with the leader's and follower's optimization problems, we shall consider only the leader's performance as it is the only one we model as an expensive black-box function. The optimality gap plots between true optimal points and approximated points in 50 iterations *in log scale* are given in Figure 4.1. As can be seen in Figure 4.1, the proposed algorithm for bilevel optimization is competitive with the sequential Bayesian method at upper-level optimization with the UCB, EI and PI acquisition functions. We fixed the iteration number for the first experiment to see how using multiple acquisitions affects the performance when we compare it with single acquisition functions. As we can see in Figure 4.1, the multi-objective acquisition approach gave better results than EI, PI and UCB alone for SMD1, SMD3 and SMD6. We can see that the proposed algorithm reaches a closer point to the optimal value for these problems compared with the single acquisition approach. The proposed

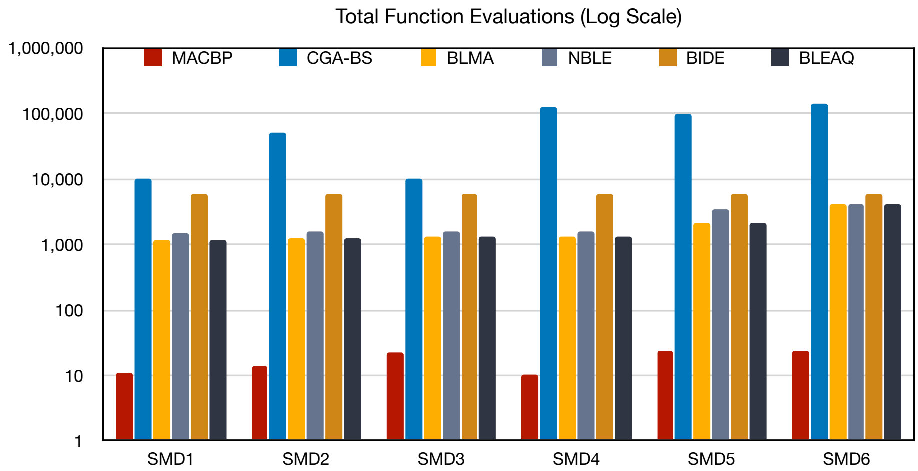


Figure 4.2: Left: Median of total function evaluations for TP problems in log scale.

algorithm gave better performance than UCB and PI for the SMD2 problem but EI eventually reached a point that was closer to optimality. PI reached the best point at the end of iterations for SMD 4 and they are so close as it is the second-best one.

In the second experiment, we can see from Table 4.2 that the MACBP algorithm reached better results for SMD4, SMD5 and SMD7 than the compared algorithms. For SMD1, we get closer to the optimal solution than NBLE and BIDE algorithms. We reached better results for SMD2 when we compared the results with NBLE and BLEAQ algorithms. Compared with BIDE and BLEAQ, the proposed algorithm gets better results for SMD3. We run the Wilcoxon rank-sum test to determine if the difference in function evaluations is significant or not. One of the test sets was constructed with the obtained results from the proposed algorithm and the other one was constructed with the best algorithm from the others. The obtained p-value for upper-level function evaluations is 0.0051. In terms of function evaluations, as the p-value is smaller than the significance level, we can conclude that our MACBP algorithm decreased significantly the function evaluations as we can see in Table 4.3 when we compare the other state-of-art algorithm in the literature.

4.4 Conclusion

In this chapter, we proposed the MACBP algorithm, a Bayesian approach via multi-objective acquisition functions for bilevel optimization problems. We ap-

Table 4.3: Upper-level function evaluations for the proposed MACBP algorithm and other known algorithms for SMD1-SMD6

	Upper-Level Function Evaluations					
	MACBP	CGA-BS [AEN20]	BLMA [ISRS17]	NBLE [ISRS17]	BIDE [ISRS17]	BLEAQ [SMD14b]
SMD1	9.70×10^1	1.01×10^4	1.19×10^3	1.52×10^3	6.00×10^3	1.19×10^3
SMD2	19.90×10^1	5.00×10^4	1.20×10^3	1.56×10^3	6.00×10^3	1.20×10^3
SMD3	10.30×10^1	1.00×10^4	1.29×10^3	1.56×10^3	6.00×10^3	1.29×10^3
SMD4	26.90×10^1	1.25×10^5	1.31×10^3	1.53×10^3	6.00×10^3	1.31×10^3
SMD5	17.00×10^1	1.00×10^5	2.06×10^3	3.40×10^3	6.00×10^3	2.06×10^3
SMD6	14.70×10^1	1.37×10^5	4.08×10^3	4.06×10^3	6.00×10^3	4.08×10^3

proached the leader's objective as an expensive black-box function. We used multiple acquisition functions during the bilevel optimization process and made our selection from a Pareto-front solution set in each iteration. We selected six popular SMD benchmark problems for the experiments. We compared our experimental results with a classic sequential setting of Bayesian optimization with each acquisition function performance individually. We also compare our results in terms of required function evaluations at the upper-level. It is shown that the proposed MACBP algorithm is competitive with existing well-known algorithms to solve bilevel optimization problems.

Chapter 5

Batch Bayesian Optimization for Multi-objective Bilevel Problems

Abstract. *Bilevel multi-objective programming problems have multiple objectives in their upper- and/or lower-levels. These problems inherit the computational complexities of both hierarchical structure and multi-objective optimization. They are strongly NP-hard and require significant computation to obtain non-dominated solutions at both levels. Since every solution affects the lower-level optimality, efficient algorithms at the upper-level are mandatory. In this study, we propose a hybrid algorithm called* BoMoBo*, based on batch BO to improve efficiency. We treat the upper problem as a black-box function, approximate its Pareto-front by considering hypervolume improvement, and perform parallel evaluations during the Bayesian optimization process. We use two different acquisition functions that are popular in GP, specifically designed for multi-objective problems. We evaluate the effect of batch size on the optimization process and compare our results with state-of-the-art algorithms. On three artificial problems and one practical problem, the proposed* BoMoBo *algorithm approximates the Pareto-optimal front using significantly fewer function evaluations.*

5.1 Introduction

Bilevel problems can have single or multiple objectives at either or both levels. In the case of multiple objectives at the lower-level, a set of good solutions exists for each upper-level variable x_u. So the upper-level decision maker observes the lower-level decision maker's decisions via the x_u decision variable.

Standard BO is a sequential experimental design framework for efficient global optimization of expensive-to-evaluate black-box functions. Multi-objective BO (MOBO) combines a Bayesian surrogate model with an acquisition function specifically designed for multi-objective problems, as it is discussed in Section 2.2.3. Hypervolume-based acquisition functions seek to maximize the volume of objective space dominated by Pareto-front solutions. Expected hypervolume improvement (EHVI) [EGN06] is a natural extension of the EI acquisition function to multi-objective optimization. Single-objective problems can be derived from MOPs by scalarizing objectives. For instance, ParEGO [Kno06b] maximizes the EI using random augmented Chebyshev scalarizations [Kno06b]. MOEA/E-EGO [ZLTV10] extends ParEGO to a batch setting using multiple random scalarizations and a genetic algorithm to optimize them in parallel. Recently in [DBB20b] another batch variant of ParEGO was proposed. Called qParEGO, it uses compositional Monte Carlo objectives and a sequential greedy candidate selection process. In this chapter, we focus on solving multi-objective bilevel problems with a GP and batch Bayesian approach with a *noise-free setting*. We treat upper-level multi-objective problems as black-box functions and optimize them using batch BO with a multi-output GP as a surrogate model.

Batch BO aims to query multiple locations at once. The benefit of this approach makes parallel evaluations possible at the same time. It is a vital part of various kinds of practical applications such as product design in the food industry which can be modelled as BOPs with multiple objectives [CPW01]. One can only produce a relatively small batch of different products at one time. Product quality, especially for foods, is usually highly dependent on the processing time since production. The batch is usually best to be evaluated at once. So, the next batch of products should be designed based on the feedback so far. We use tailored acquisition functions to optimize the design space for multi-objective problems with fewer expensive function calls. To the best of our knowledge, there is no previous work on a batch BO approach to multi-objective bilevel problems. The closest work is that of [KPH21] which uses multiple Gaussian models for multiple objectives to solve robot and behaviour co-design, but not with batch selection during BO or with hypervolume improvement.

In this chapter, we propose an algorithm to solve multi-objective bilevel problems by a black-box approach to upper-level multi-objective problems. The main aim is to improve computational efficiency while using hypervolume improvement to approximate the Pareto-optimal solution set. We compare the BOMOBO algorithm with some existing algorithms in terms of function evaluations and hypervolume improvement. Moreover, we show how batch selection affects performance in batch BO. We propose a hybrid algorithm with a multi-objective BO approach, using two different acquisition optimizers specifically designed for multi-objective optimization problems. We use a Non-dominated Sorting Genetic Algorithm (NSGA-II) to solve the lower-level multi-objective problems, with a crossover operator to decrease lower-level function evaluations. Using batch BO via a multi-output GP surrogate model in multi-objective BOPs allows us to measure uncertainties in the multi-objective upper-level design space. Selecting multiple points rather than one point at a time makes the optimization process more efficient. Moreover, the multi-output GP allows us to work with several kernels that worked well in the literature while optimizing the multi-objective upper-level design space.

The contribution of this chapter can be explained in three parts. First, we present a new hybrid algorithm for solving multi-objective BOPs using batch BO in a multi-objective setting. Different modified kernels and surrogate surfaces are described in the multi-output GP literature, and we can apply them to multi-objective BOPs. Second, batch BO allows us to use batch selection during optimization rather than just once per iteration. It provides information on how batch selection affects the whole bilevel optimization process at both levels. Last and third we use the qEHVI acquisition function and consider hypervolume improvement for solving the following q-batch decision points at the upper-level. This provides an approximation to the Pareto-optimal solutions with less expensive function evaluations. Moreover, we use the qParEGO acquisition optimizer in GP during upper-level decision-making and show how parallel evaluations affect the whole multi-objective bilevel optimization process.

5.2 Proposed Approach

BO is a method to optimize expensive-to-evaluate black-box functions. BO uses a probabilistic surrogate model, typically GP [RW09], $p(f|\mathcal{D})$ to model the objective function f based on previously observed data points, that can be declared as $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$. GPs are models that are specified by a mean function $\mu(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta) : \mathbb{R}^d \to \mathbb{R}$ and predictive variance function

$\sigma(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$. Surrogate model $p(f|\mathcal{D})$ is assisted by an acquisition function $\alpha : \mathcal{X} \rightarrow \mathbb{R}$. We represent acquisition functions depending on the previous observations as $\alpha(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)$ where θ is Gaussian parameters such as a kernel for the model. Because the objective function is expensive to evaluate and the surrogate-based acquisition function is not, it can be optimized more easily than the true function to yield $\mathbf{x}_{new}$. The acquisition function selects the point $\mathbf{x}_{new}$ that maximizes the acquisition function $\mathbf{x}_{new} = argmax_{\mathbf{x}\in\mathcal{X}}\alpha(\mathbf{x})$. Then, it evaluates the objective function $y_{new} = f(\mathbf{x}_{new})$ and updates the data set with new observations $\mathcal{D} \leftarrow \mathcal{D} \cup (\mathbf{x}_{new}, y_{new})$.

In the GP, $\mu(x)$ can be viewed as the prediction of the function value, and $\sigma(x)$ is a measure of the uncertainty of the prediction. Multi-objective BO tackles the problem of optimizing a vector-valued objective $\mathbf{f}(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ with $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_d(\mathbf{x}))$ for a vector-valued decision variable $\mathbf{x} \in \mathbb{R}^d$. Because of the nature of the multi-objective black-box problems, we assume that there is no known analytical expression. Multi-objective optimization problems generally do not have a single best solution, so we must find a solution set instead of a single solution: the set of Pareto-optimal solutions. We say that $\mathbf{f}(\mathbf{x})$ dominates another solution $\mathbf{f}(\mathbf{x}')$ if $f^{(i)}(\mathbf{x}) \succ f^{(i)}(\mathbf{x}')$ for all $i = 1, 2, \ldots, M$ and there exists $i' \in \{1, 2, \ldots, M\}$ such that $f^{i'}(\mathbf{x}) \succ f^{i'}(\mathbf{x}')$. So we can express the Pareto-optimal solution set by $P^* = \{\mathbf{f}(\mathbf{x}) \text{ s.t. } \nexists \mathbf{x}' \in X : \mathbf{f}(\mathbf{x}') \succ \mathbf{f}(\mathbf{x})\}$ and $X^* = \{\mathbf{x} \in X \text{ s.t. } \mathbf{f}(\mathbf{x}) \in P^*\}$. After obtaining the Pareto-front, the decision maker can make decisions using the trade-off between objectives, or any preferences. In general, multi-objective optimization algorithms try to find a set of distributed solutions that approximate the Pareto-front.

Hypervolume improvement (HVI) is often used as a measure of improvement in multi-objective problems [WH19]. Figure 2.7 illustrates the hypervolume improvement for 2 dimensional setting. The red area on the left graph represents the dominant solutions while the white area is non-dominated solutions, and the green area on the middle graph represents the hypervolume improvement after the first point selection. The right graph illustrates the HVI in the $q = 2$ setting where q represents that batch selection in batch BO. Several methods have been proposed. Expected Hypervolume Improvement (EHVI) is an updated version of Expected Improvement (EI) to HVI, and determined by $J(\mathbf{x}) = \mathbb{E}_{p(f(x)|D_n)}[HVI(f(\mathbf{x}))]$. Previous work has considered unconstrained sequential optimization with ParEGO [Kno06b]. ParEGO is often optimized with gradient-free methods. qParEGO supports parallel and constrained optimization [DBB20b], and exact gradients are computed via auto-differentiation for acquisition optimization. This approach enables the use of sequential greedy optimization of q candidates with proper integration over the posterior at the pending points.

Multi-objective bilevel optimization problems have two levels of multi-objective optimization, such that a feasible solution to the upper-level problem must be a member of the Pareto-optimal set of lower-level optimization problems. Each level has its variables, objectives, and constraints. For the given upper-level vector $\mathbf{x}_u$ the evaluation of the upper-level function is valid *only if* the $\mathbf{x}_l$ is the optimum of the lower-level problem. A general multi-objective BOP can be described as follows:

$$\begin{aligned}
&\underset{\mathbf{x}_u \in X_u, \mathbf{x}_l \in X_l}{\text{minimize}} \{F_1(\mathbf{x}_u, \mathbf{x}_l), ..., F_{M_u}(\mathbf{x}_u, \mathbf{x}_l)\} \\
&\text{subject to } \mathbf{x}_l \in \underset{\mathbf{x}_l}{\text{argmin}}\{f_1(\mathbf{x}_u, \mathbf{x}_l), ..., f_{M_l}(\mathbf{x}_u, \mathbf{x}_l);\ g_j(\mathbf{x}_u, \mathbf{x}_l) \leq 0,\ j = 1, 2, \ldots, J\} \\
&\qquad\qquad G_k(\mathbf{x}_u, \mathbf{x}_l) \leq 0,\ k = 1, 2, \ldots, K
\end{aligned} \tag{5.1}$$

In Eq. 5.1 the upper-level objective functions are $F_i(\mathbf{x}_u, \mathbf{x}_l),\ i = 1, 2, \ldots, M_u$ and the lower-level objective functions are $f_i(\mathbf{x}_u, \mathbf{x}_l),\ i = 1, 2, \ldots, M_l$ where $\mathbf{x}_u \in \mathbb{X}_u$ and $\mathbf{x}_l \in \mathbb{X}_l$. $G_k(\mathbf{x}_u, \mathbf{x}_l)$ and $g_j(\mathbf{x}_u, \mathbf{x}_l)$ represent upper and lower-level constraints respectively. j and K values represent the number of constraints at the upper- and lower-level. The lower-level optimization problem is optimized for $\mathbf{x}_l$ considering $\mathbf{x}_u$ as a fixed parameter.

In this chapter, we solve a multi-objective upper-level problem in a bilevel structure by treating it as a black box. We solve a vector-valued upper-level problem, $\mathbf{F}(\mathbf{x_u}, \mathbf{x_l}) : \mathbb{R}^d \times \mathbb{R}^v \rightarrow \mathbb{R}^{M_u}$ where d and v are number of upper-level and lower-level decision variable dimensions.

An acquisition function for MOBO is EHVI and maximizing HV is a procedure for finding the maximum coverage with Pareto fronts [YEDB19]. We use the q-expected hypervolume improvement (qEHVI) acquisition function for a MOBO procedure at the upper level. qEHVI computes the exact gradient of the Monte-Carlo estimator using auto-differentiation, allowing it to employ efficient and effective gradient-based optimization methods. More details about the qEHVI can be found in [DBB20b]. Another acquisition optimizer we use is an extended version of a hybrid algorithm with on-line landscape approximation for expensive multi-objective optimization problems (ParEGO) [Kno06b]. It is extended to the constrained setting by weighting the expected improvement by the probability of feasibility which is called qParEGO [GKX+14]. qParEGO uses

Algorithm 4 BoMoBo Algorithm - Upper-level Optimization

Inputs: $\mathbf{F}_u(\mathbf{x}_u, \mathbf{x}_l) : \mathbf{x_u} \in \mathbb{X}_u, \mathbf{x_l} \in \mathbb{X}_l$,
Batch points per epoch *q*,
Number of iteration *n*,
Reference point

1: Initial decision data set $D = \{\mathbf{x}_{u_i}, \mathbf{F}_u(\mathbf{x}_{u_i}, \mathbf{x}^*_{l_i})\}_{i=1}^n$ with size of n,
2: $\mathbf{x_l^*}$: Initialize Best Lower-Level Decisions as parameters from NSGA-2 Algorithm,
3: Initialize *Multi-objective Gaussian Model* with Observations $\{\mathbf{x}_u, \mathbf{F}_u(\mathbf{x}_u, \mathbf{x}^*_l)\}$
4: **for** i = 0 : *N* **do**
5: Suggest new q-batch points by optimizing *q-NEHVI* and *qParEGO* acquisition functions
6: **for** j = 0 : q **do**
7: For each upper-level decision $\mathbf{x}_u$, find optimal $\mathbf{x}^*_l$ by applying NSGA-2 Algorithm
8: Calculate fitness scores $\mathbf{F}^*_u$ and $\mathbf{f}^*_u$
9: Update the data set $D = (\mathbf{x}_{u_i}, \mathbf{F}_u(\mathbf{x}_{u_i}, \mathbf{x}^*_{l_i}))_{i=1}^n$
10: **Return** Pareto-front $\mathbf{F}^*_u$ and corresponding optimum variables $\mathbf{x}^*_u, \mathbf{x}^*_l$

a Monte Carlo-based expected improvement acquisition function, where the objectives are modelled independently and the augmented Chebyshev scalarization [Kno06b] is applied to the posterior samples as a composite objective. More details can be found in [DBB20b].

The proposed BoMoBo algorithm is a hybrid method for solving multi-objective BOPs. Briefly, it works as follows. A population of the size of N_u initial decisions, $\mathbf{x_u}$, is randomly selected from upper-level decision set $\mathbb{X}_u$. We used Sobol sampling for the initial random selection. For each upper-level decision, the lower-level problem is optimized using a Non-dominating Sorting Genetic Algorithm (NSGA-II) with population size N_l. For a given $\mathbf{x}_u$ a Pareto-optimal solution set is obtained by the NSGA-II algorithm. Lower-level decision values are selected randomly from the Pareto-front obtained by the NSGA-II algorithm. We decided to use the random selection from the Pareto-optimal solution set considering the work [LYY+18b]. The solution set obtained after lower-level optimization iteration $(\mathbf{x}_u, \mathbf{x}^*_l)$ is used to find upper-level fitnesses $F_i(\mathbf{x}_u, \mathbf{x}^*_l)$ for $i = 1, 2, \ldots, M_u$. We train the GP model with the data set $(\mathbf{x}_u, y_i)$ where $y_i = F_i(\mathbf{x}_u, \mathbf{x}^*_l)$. We use batch BO to choose the next candidates with the n-batch which declares the number of the batch. In each BO iteration, the qEHVI and qParEGO acquisition optimizers suggest q-batch (q) candidates.

The lower-level optimization process is repeated for each candidate of the upper level batch decision candidate $\mathbf{x}_u$. It is important to note that dealing with constraints is the most challenging aspect of BOPs. There are various approaches developed over the years, such as optimistic and pessimistic approaches regarding lower-level decision-making strategies; please see [AAC21] for more details. While in the optimistic approach, the lower-level decision maker deals with the constraints and makes decisions considering the most favourable decision for the upper-level decision maker, the pessimistic approach assumes the least favourable decision for the upper-level. In this work, we considered the optimistic multi-objective bilevel problems and dealt with the constraints regarding the best interest of the leader. To avoid upper-level constraint violation, we made the random selection from lower-level Pareto-front considering the upper-level constraints. The algorithm runs for 50 iterations for the whole multi-objective bilevel optimization process. The details of the algorithm can be found in Algorithm 4.

5.3 Numerical Experiments

We share the details of the experiments for the proposed approach in this section. We discuss the numerical experiments using the proposed BOMOBO algorithm on three different benchmark problems from [CAA15], including quadratic unconstrained and linear constrained problems. Also, we applied the proposed algorithm to the popular CEO problem from [ZHG+13].

- **Problem 1.** The problem is linear and constrained at both levels. The formulation of the problem is given in Equation 5.2.

$$\begin{aligned}
&\underset{(x_u,\mathbf{x_l})}{\text{Minimize}}\ \mathbf{F}(x_u,\mathbf{x_l}) = \begin{pmatrix} x_{l_1} - x_u \\ x_{l_2} \end{pmatrix} \\
&\text{subject to}\ \ \mathbf{x}_l \in \begin{pmatrix} \underset{\mathbf{x_l}}{\text{argmin}}\ \mathbf{f}(x_u,\mathbf{x_l}) = \begin{pmatrix} x_{l_1} \\ x_{l_2} \end{pmatrix} \\ g(x_u,\mathbf{x_l}) = x_u^2 - x_{l_1}^2 - x_{l_2}^2 \geq 0 \end{pmatrix} \\
&G(x_u,\mathbf{x_l}) = 1 + x_{l_1} + x_{l_2} \geq 0 \\
&0 \leq x_u \leq 1, x_{l_1} \geq -1, x_{l_2} \leq 1
\end{aligned} \tag{5.2}$$

Both upper- and lower-level optimization tasks have two objectives and the reference point for the problem is $x_{ref} = (-1, 0)$. The linear constraint in the upper-level optimization task does not allow the entire quarter circle to be feasible for some y. Thus, at most, a couple of points from the quarter circle belong to the Pareto-optimal set of the overall problem. Reported Pareto-optimal solutions for the problem are as in Equation 5.3 as follows:

$$\left\{ \mathbf{x} \in \mathbb{R}^3 |\ x_{l_1} = -1 - x_{l_2},\ x_{l_2} = -\frac{1}{2} \mp \frac{1}{4}\sqrt{8x_u^2 - 4},\ x_u \in \left[\frac{1}{\sqrt{2}}, 1\right] \right\}. \tag{5.3}$$

Figure 5.1 on the left graph shows the Pareto-front solutions for Problem 1. Dashed lines represent the lower-level Pareto-fronts and B and C points as the two most Pareto-optimal solutions for the upper-level decision $y = 0.9$.

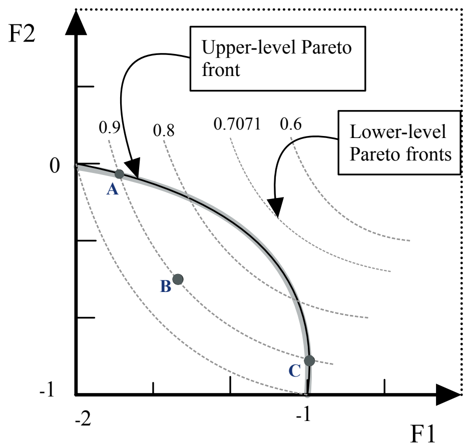


Figure 5.1: Pareto-optimal fronts of the upper-level problem and some representative lower-level optimization tasks are shown for Problem 1 with upper-level decision variable y and lower-level decision x_i for $i = 1, 2$.

- **Problem 2.** The problem is a quadratic and unconstrained problem at both levels. The formulation of the problem is given in Equation 5.4.

$$\begin{aligned} &\underset{(x_u,\mathbf{x_l})}{\text{Minimize}}\ \mathbf{F}(x_u, \mathbf{x_l}) = \begin{pmatrix} (x_{l_1} - 1)^2 + x_{l_2}^2 + (x_u)^2 \\ (x_{l_1} - 1)^2 + x_{l_2}^2 + (x_u - 1)^2 \end{pmatrix} \\ &\text{subject to } \ \mathbf{x}_l \in \left(\underset{\mathbf{x_l}}{\text{argmin}}\ \mathbf{f}(x_u, \mathbf{x_l}) = \begin{pmatrix} x_{l_1}^2 + x_{l_2}^2 \\ (x_{l_1} - x_u)^2 + x_{l_2}^2 \end{pmatrix} \right) \\ &-1 \leq (x_u, x_{l_1}, x_{l_2}) \leq 2 \end{aligned} \tag{5.4}$$

The reference point for the problem is $x_{ref} = (1, 0.5)$. The upper-level problem has one variable and the lower-level has two variables. For a fixed of x_u, the Pareto-optimal solutions of the lower-level optimization problem are given as $\{x_l \in \mathbb{R} | x_u \in [0, y], x_{l_1} = x_{l_2} = 0\}$. For the upper-level problem, the Pareto-optimal solutions are as follows:

$$\left\{ \mathbf{x} \in \mathbb{R}^3 | \; x_u = y, x_{l_1} = x_{l_2} = 0, \; x_u \in [0.5, 1.0] \right\}. \tag{5.5}$$

As we can see in Figure 5.2 on the right, if an algorithm fails to find true Pareto-optimal solutions of the lower-level problem and finds a solution such as a point C, then it might dominate a true Pareto-optimal point such as point A. So, finding true Pareto-optimal solutions is a difficult task in this problem.

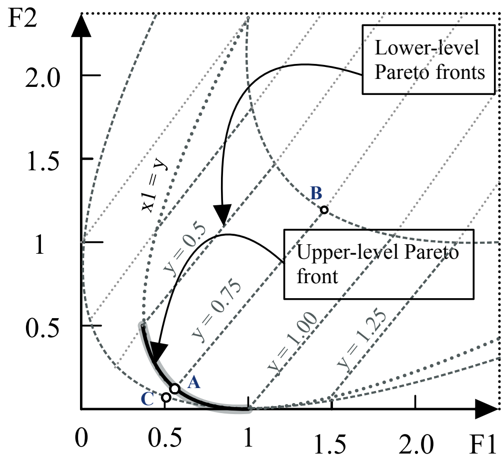


Figure 5.2: Pareto-optimal fronts of the upper-level problem and some representative lower-level optimization tasks are shown for Problem 2 with upper-level decision variable y and lower-level decision x_i for $i = 1, \ldots, K$.

- **Problem 3.** We increase the dimension of the variable vector of Problem 2. The formulation of the problem is given in Equation 5.6.

$$\begin{aligned}
&\underset{(x_u, \mathbf{x_l})}{\text{Minimize}}\; \mathbf{F}(x_u, \mathbf{x_l}) = \begin{pmatrix} (x_{l_1} - 1)^2 + \sum_{i=2}^{K} x_{l_i} + (x_u)^2 \\ (x_{l_1} - 1)^2 + \sum_{i=2}^{K} x_{l_i} + (x_u - 1)^2 \end{pmatrix} \\
&\text{subject to } \; \mathbf{x}_l \in \left(\underset{\mathbf{x_l}}{\text{argmin}}\; \mathbf{f}(x_u, \mathbf{x_l}) = \begin{pmatrix} x_{l_1} + \sum_{i=2}^{K} x_{l_i}^2 \\ x_{l_1} - x_u + \sum_{i=2}^{K} x_{l_i}^2 \end{pmatrix} \right) \\
&-1 \leq (x_u, x_{l_1}, x_{l_2}, ..., x_{l_K}) \leq 2
\end{aligned} \tag{5.6}$$

The reference point for the problem is $x_{ref} = (1, 0.5)$. The upper-level problem has one variable and the lower-level is scalable with K variables. For this work, K is set to 14 and the total number of variables is 15 in this problem. For a fixed x_u the Pareto-optimal solutions of the lower-level optimization problem are given as follows: $\{x_l \in \mathbb{R}^K | x_u \in [0, y], x_{l_i} = 0,$ for $i = 2, ..., K\}$. For the upper-level problem, the Pareto-optimal solutions are defined in Equation 5.7 as follows:

$$\left\{ \mathbf{x} \in \mathbb{R}^{K+1} | x_u = y,\ x_{l_i} = 0, x_u \in [0.5, 1.0],\ i = 2, \ldots, K \right\}. \tag{5.7}$$

- **Practical Case.** This case is selected from [ZHG+13]. In a company, the CEO is as the leader in the problem and optimizing net profit and the quality of products. The branch head is the follower in the problem and optimises its own profit and worker satisfaction. According to the scenario above, the problem itself can be modelled as a multi-objective BOP. The deterministic way of the problem can be formulated in Equation 5.8. Let the CEO's decision variable by $\mathbf{x_u} = (x_{u_1}, x_{u_2})$ and the branch head's decision variable $\mathbf{x_l} = (x_{l_1}, x_{l_2}, x_{l_3})$. The constraints model requirements such as material, marking cost, labour cost, and working hours.

$$\begin{aligned}
&\underset{(\mathbf{x_u},\mathbf{x_l})}{\text{Maximize}}\ \mathbf{F}(\mathbf{x_u}, \mathbf{x_l}) = \begin{pmatrix} x_{u_1} + 9x_{u_2} + 10x_{l_1} + x_{l_2} + 3x_{l_3} \\ 9x_{u_1} + 2x_{u_2} + 2x_{l_1} + 7x_{l_2} + 4x_{l_3} \end{pmatrix} \\
&\text{subject to}\ \ \mathbf{x}_l \in \begin{pmatrix} \underset{\mathbf{x_l}}{\text{argmax}}\ \mathbf{f}(\mathbf{x_u}, \mathbf{x_l}) = \begin{pmatrix} 4x_{u_1} + 6x_{u_2} + 7x_{l_1} + 4x_{l_2} + 8x_{l_3} \\ 6x_{u_1} + 4x_{u_2} + 8x_{l_1} + 7x_{l_2} + 4x_{l_3} \end{pmatrix} \\ g_1(\mathbf{x_u}, \mathbf{x_l}) = 3x_{u_1} - 9x_{u_2} - 9x_{l_1} - 4x_{l_2} - 61 \leq 0 \\ g_2(\mathbf{x_u}, \mathbf{x_l}) = 5x_{u_1} + 9x_{u_2} + 10x_{l_1} - -x_{l_2} - 2x_{l_3} - 924 \leq 0 \\ g_3(\mathbf{x_u}, \mathbf{x_l}) = 3x_{u_1} + 3x_{u_2} + 1x_{l_2} - 5x_{l_3} - 420 \leq 0 \end{pmatrix} \\
&G_1(\mathbf{x_u}, \mathbf{x_l}) = 3x_{u_1} + 9x_{u_2} + 9x_{l_1} + 5x_{l_2} + 3x_{l_3} - 1039 \\
&G_2(\mathbf{x_u}, \mathbf{x_l}) = -4x_{u_1} - 2x_{u_2} + 3x_{l_1} - 3x_{l_2} + 2x_{l_3} - 94 \\
&(x_{u_1}, x_{u_2}, x_{l_1}, x_{l_2}, x_{l_3}) \geq 0
\end{aligned} \tag{5.8}$$

While the CEO's objective is to maximize the quality of the product and the net profit, the objective of the branch heads is to maximize worker satisfaction and branch profits.

5.3.1 Algorithms Used for Comparison

- **DBMA**: This is a popular nested algorithm that uses differential evolutionary methods at both levels of multi-objective BOPs, proposed in [ISR16].
- **OMOPSO-BL**: This algorithm uses a multi-objective particle swarm optimization based algorithm for BOPs and is presented in [CAA15]. It uses

a lower-level archive to save the non-dominated solutions obtained by lower-level iterations and it guides the lower-level search.

- **H-BLEMO**: This is a hybrid evolutionary-cum-local-search algorithm presented in [DS10a]. The evolutionary part of the algorithm contributes by handling the multi-solution aspect of the problem and the local search part ensures the lower level of optimality. It is also declared that the algorithm is self-adaptive in terms of the key hyperparameters during the optimization.

5.3.2 Parameters

We use the Python programming language to implement the problems. BoTorch library [BKJ+19] is used for the upper-level optimization problem. qEHVI and qParEgo acquisition functions for multiobjective optimization problems can be found in the library. *SobolQMCNormalSampler* is used for random sampling for starting the optimization. $2(d+1)$ random upper-level decision point selected where d represents the dimensions of the problem for constructing the Gaussian process (GP) model. *MaternKernel* is used as it is the default selection. During acquisition optimization, we selected the number of starting points $num\_restarts = 10$ and $raw\_samples = 512$ as it is recommended in [EGN06] for both qEHVI and qParEGO acquisition functions. Multi-start optimization of the acquisition function is performed using *LBFGS-B* with exact gradients computed via auto-differentiation. Other parameters are set as default for BO. Lower-level optimization problems are solved using a genetic algorithm. For the implementation of the problems, the pymoo library [BD20] is selected. We use the Non-dominated Sorting Genetic Algorithm, called NSGA-II with the following parameters. We selected *pop_size* = 50 to represent the population size and the number of generations is selected 50 as termination criteria. *real_random* sampling method is used for sampling. After we obtained the Pareto-optimal solution set, we used a uniform random selection from the Pareto-optimal solution set. The algorithm is executed 10 times for each test function with different batch sizes.

5.3.3 Performance Measures

The multi-objective algorithms are assessed in terms of convergence to the Pareto-optimal front and concerning the diversity of the obtained solutions.

Table 5.1: Function Evaluation Comparison Table for different batch sizes (*q = 1, q = 2, q = 4, q = 8*)

	(q)	BOMOBO UL	BOMOBO LL	DBMA UL	DBMA LL	H-BLEMO UL	H-BLEMO LL
Problem 1	1	4,500	27,000	22,229	312,500	14,163	629,590
	2	6,500	54,000				
	4	11,500	108,000				
	8	18,500	216,000				
Problem 2 (K = 3)	1	2,700	14,580	12,028	312,500	16,146	700,634
	2	4,300	28,620				
	4	7,200	38,880				
	8	9,900	48,060				
Problem 3 (K = 14)	1	2,675	57,780	N/A	N/A	18,733	319,499
	2	1,575	34,020				
	4	6,575	142,020				
	8	3,125	67,500				

To examine the BOMOBO algorithm's performance, we compare it with the OMOPSO-BL and H-BLEMO algorithms in terms of the HV indicator and the IGD metrics. We present our results in terms of function evaluations and compare them with DBMA and H-BLEMO. We use the same number of final sample sizes of the Pareto-optimal solution set to compare HV and IGD values with the other algorithms. HV measures the volume of the space between the non-dominated front obtained and a reference point. It is a common metric for comparing the performance of the obtained solutions with the true Pareto-front published for the problems. IGD calculates the sum of the distances from each point of the Pareto-front to the nearest point of the non-dominated set found by the algorithm. More details can be found for the metric in [CCRS04]. The IGD values of the DBMA algorithm are not reported [ISR16] but we compare the results with the HV metric. Both HV and IGD measure the convergence and the spread of the obtained set of solutions.

5.3.4 Results and Observations

The statistics on function evaluations, HV difference, and IGD from Pareto-optimal are given in Table 5.1 and Table 5.2. The HV and IGD values are shared with the standard deviations of 21 runs. The standard deviations come from BO. It is clear from Table 5.1 that most computational effort is spent on the lower-level problem. Because of the nested structure of BOPs, the lower-level problem must be solved for every upper-level decision, so it is vital to improve upper-level performance for efficiency. We can observe that the upper-level function evaluations decreased significantly compared to DBMA and H-BLEMO. We could not compare the function evaluations with the OMOPSO-BL algorithm because of unavailability in [CAA15]. We compare HV and the IGD values with DBMA, OMOPSO-BL, and H-BLEMO algorithms in Table 5.2. It also includes the HV difference graph for different batch sizes. Figures 5.3,5.4 and 5.5 reflect that the proposed algorithm can converge to Pareto-fronts with good diversity and uniform distribution. Although BO is not generally effective on high-dimensional problems, we nevertheless obtain results that are better than the other tested methods.

5.3.4.0.1 Problem 1. As shown in Table 5.1, function evaluations decreased significantly compared with DBMA and H-BLEMO algorithms for Problem 1. We observe that in Table 5.2 the HV obtained by the proposed BOMOBO algorithm is better than DBMA, H-BLEMO, and OMOPSO-BL. We also show hypervolume values for different batch sizes of $2, 4, 8$. Because of a lack of observations for a batch size of $q = 1$ as we ran the algorithm for 50 iterations, the algorithm could not find better HV values. The IGD values of observed values are not better than OMOPSO-BL and H-BLEMO algorithms for this specific problem with batch sizes $2, 4, 8$ as we can observe from Table 5.2. We think that as the problem is linear, and because of limited iteration numbers, the algorithm could not get as close as the others. There were no reported IGD values for the DBMA algorithm on this comparison.

5.3.4.0.2 Problem 2. We can observe from Table 5.1 for Problem 2 that the BOMOBO algorithm improves computational performance and decreases the function evaluations significantly compared to DBMA and H-BLEMO. The HV values are much better than DBMA, OMOPSO-BL, and H-BLEMO algorithms as we can see from Table 5.2. Note that because of the unavailability of OMOPSO-BL and H-BLEMO experiment data, the results shown for these algorithms are taken from [CAA15]. The IGD results in Table 5.2 show that the proposed algorithm approximates successfully to the Pareto-optimal solutions.

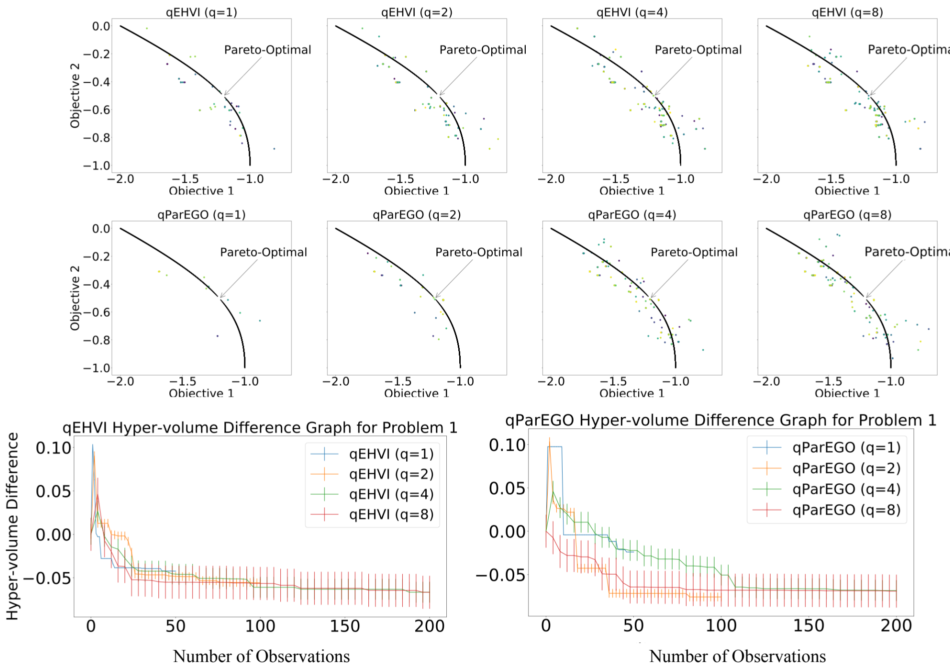


Figure 5.3: Final achieved solutions for Problem 1 for the whole batch BO process with *qEHVI* and *qParEGO* acquisition optimizers. Also, the hypervolume difference values for different batch sizes (q = 1, q = 2, q = 4, q = 8).

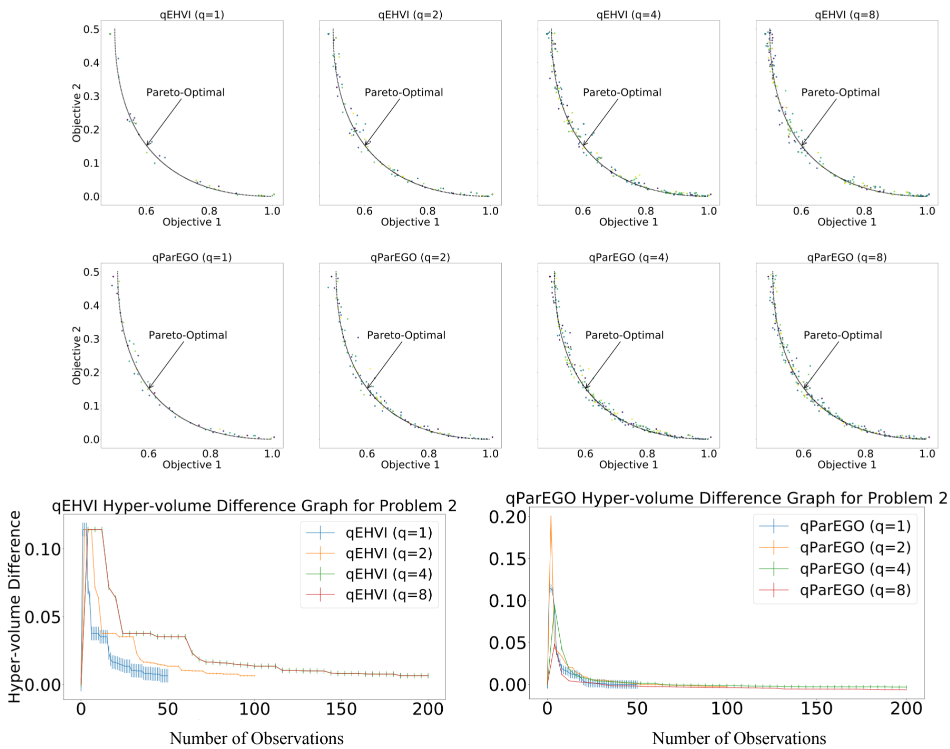


Figure 5.4: Final achieved solutions for Problem 2 (K=2) for the whole batch Bayesian optimization process via *qEHVI* and *qParEGO* acquisition optimizers, and hypervolume difference values for different batch sizes (*q = 1, q = 2, q = 4, q = 8*).

Table 5.2: Medium Hypervolume(HV) and Inverted Generational Distance (IGD) Comparison Table for different batch sizes (*q = 1, q = 2, q = 4, q = 8*) with the standard deviation

		BoMoBo		DBMA		OMOPSO-BL		H-BLEMO	
	(q)	HV	IGD	HV	IGD	HV	IGD	HV	IGD
Problem 1	1	0.3524±0.0003	0.0143±0.0005	0.3075	N/A	0.3068	0.0156	0.3024	0.0113
	2	0.3632±0.005	0.0138±0.002						
	4	0.3621±0.012	0.0126±0.005						
	8	0.3773±0.019	0.0119±0.009						
Problem 2 (K = 3)	1	0.2067±0.005	0.0105± 0.006	0.2069	N/A	0.2074	0.0102	0.2067	0.0063
	2	0.2097±0.0005	0.0068±0.002						
	4	0.2125±0.002	0.0045±0.0005						
	8	0.2144±0.0004	0.0032±0.0002						
Problem 3 (K = 14)	1	0.2026±0.004	0.0105±0.006	N/A	N/A	0.2018	0.0312	0.2059	0.0105
	2	0.1915± 0.023	0.0147± 0.002						
	4	0.2089±0.009	0.0141± 0.001						
	8	0.2241±0.042	0.0138± 0.0087						

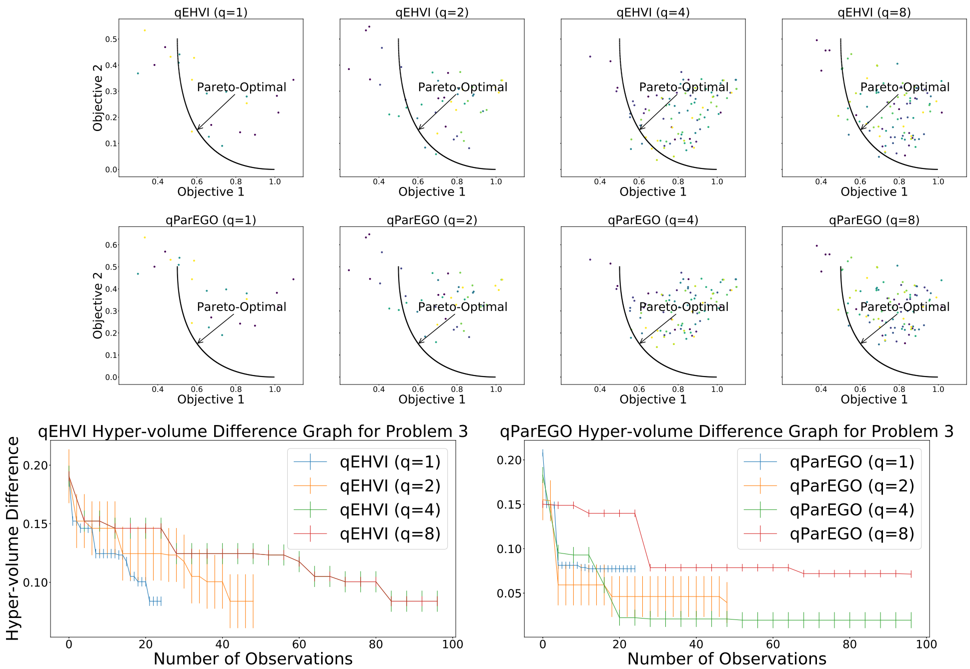


Figure 5.5: Final achieved solutions for Problem 3 (K=13) for the whole batch Bayesian optimization process via *qEHVI* and *qParEGO* acquisition optimizers, and hypervolume difference values for different batch sizes (*q = 1, q = 2, q = 4, q = 8*).

5.3.4.0.3 Problem 3. Table 5.1 shows that the proposed algorithm significantly decreases function evaluations during the optimization while approximating the Pareto-optimum solutions for Problem 3. We compare our results with DBMA and H-BLEMO algorithms, but not OMOPSO-BL because they did not share this information (as mentioned above). The HV value for batch size 1 is better than OMOPSO-BL but not H-BLEMO, as we can see in Table 5.2. However, the proposed BoMoBo algorithm reached better HV for batch size 8. IGD values are shown in Table 5.2 and we can observe that the approximation solution sets are not as close as the other algorithms. However, we think that considering the significant decrease in function evaluations, it shows good performance.

5.3.4.0.4 Practical Case. For the CEO problem, they use a weighted sum method to obtain a single optimal solution $\mathbf{x_u} = (146.2955, 28.9394)^\top$ and $\mathbf{x_l} = (0, 67.9318, 0)^\top$ in [ZHG+13]. They use a hybrid particle swarm optimization algorithm with a crossover operator in [ZHG+13]. [LZ21b] present Pareto-front solutions and extreme points using CMODE/D algorithm. We present the extreme points of our results in Table 5.3. The proposed algorithm can obtain the Pareto-front containing the single optimal solution obtained in [LZ21b].

We focused on the extreme points on the Pareto-front solution set to make the comparison with the solutions in [LZ21b]. As we can observe, the obtained optimal solutions in Table 5.3 cover more broadly in terms of the objectives. We also share the optimal decisions for the leader and the follower for better comparison. The experiments use both qNEHVI and qParEGO and all batch sizes. Table 5.3 presents the best results obtained.

Table 5.3: Extreme points (EP) in obtained Pareto front for the practical case by the proposed BOMOBO algorithm (1) and comparison with CMODE/D (2)

Exteme Point	Leader's Objective (F_1, F_2)	Follower's Objective (f_1, f_2)	Leader's Optimal decision $(\mathbf{x_u})$	Follower's Optimal decision $(\mathbf{x_l})$
$(1)EP_1$	$(529.86, 1480.09)$	$(3554.98, 5266.08)$	$(46.82, 25.02)^\top$	$(0, 24.83, 0)^\top$
$(1)EP_2$	$(964.41, 264.74)$	$(533.02, 646.19)$	$(53.61, 66.81)^\top$	$(4.72, 0, 0)^\top$
$(2)EP_1$	$(474.48, 1849.36)$	$(1030.12, 1468.39)$	$(146.30, 28.92)^\top$	$(0, 67.84, 0)^\top$
$(2)EP_2$	$(1038.90, 307.90)$	$(830.79, 523.31)$	$(0.03, 107.75)^\top$	$(0, 0, 23.01)^\top$

5.4 Conclusion

Multi-objective BOPs constitute one of the hardest classes of optimization models known, as they inherit the computational complexity of the hierarchical structure and optimization. We propose a hybrid algorithm, called BOMOBO, based on batch BO to reduce the computational cost of the problem. We embed two specifically designed acquisition optimizers for multi-objective problems based on hypervolume improvement and parallel evaluations during optimization to approximate the leader's Pareto-front solutions. Moreover, we explore how batch size selection affects the optimization process. We evaluate our proposed algorithm on three numerical and one practical problem selected from the literature and compare our results with state-of-the-art algorithms in the literature. The results show that the proposed algorithm improves efficiency by significantly reducing the computational cost.

Chapter 6

Multi-objective Bilevel Decision Making with Noisy Objectives: A Batch Bayesian Approach

The works presented in this chapter have been published in the following article:

- Dogan, V., Prestwich S., (2023). Multi-objective Bilevel Decision Making with Noisy Objectives: A Batch Bayesian Approach. ECAI'2023 Workshop on Multi-objective Decision Making Workshop (MODeM2023), Krakow, Poland.
- Dogan, V., Prestwich, S. (2023). A Batch Bayesian Approach for Bilevel Multi-Objective Decision Making Under Uncertainty. AAAI'23 Workshop on Uncertainty Reasoning and Quantification in Decision Making, Washington, USA

Abstract. *Bilevel multi-objective optimization is a field of mathematical programming representing a nested hierarchical decision-making process, with one or more decision-makers at each level. These problems appear in many practical applications, solving tasks such as optimal control, process optimization, development of government and game strategy, and transportation. Uncertainty cannot be ignored in these practical problems. We present a hybrid algorithm called* BamBiNo*, based on the batch Bayesian approach via expected hypervolume improvement, that can handle uncertainty at the upper level. Three popular modified benchmark problems with multiple dimensions,* *and one real-world example from the field of environmental economics are used to evaluate the performance. The real-world example is a decision-making problem between a governmental authority and a gold mining company. The experimental results under the objective noise compared with two popular algorithms in the literature. The results show that* BAMBINO *is computationally efficient and capable of handling upper-level objective uncertainty while approximating the Pareto-optimal front. We also evaluate the effect of batch size on performance.*

6.1 Introduction

Our work focuses on the special case of multi-objective BOPs in which the leader has noisy objectives. We assumed that the follower is free to choose any feasible solution from a Pareto-optimal set. We used batch BO to improve efficiency, approximating the leader's Pareto-optimal front using fewer function evaluations than existing works. We also presented a black-box approach to the noisy leader's objectives for handling the uncertainty during decision-making.

Hierarchical decision-making under uncertainty with noisy objectives becomes more interesting in a bilevel structure. The follower can observe the leader's decisions, but the leader may have no idea how the follower is going to respond. Previously observed decisions are therefore important. Uncertainty in the objective also complicates the leader's decision-making, and our algorithm uses a specifically designed acquisition function called qNEHVI to maximize EHVI *under noisy objectives*. We call our algorithm BAMBINO (**B**ayesian **A**pproach for **M**ultiobjective **Bi**level Problems with **N**oisy **O**bjectives). To evaluate its performance, we considered three multi-dimensional test problems from two different suites of multi-objective BOPs. Also, one environmental economics problem is a decision-making problem between an authority and a gold mining company. All examples illustrate the importance of taking uncertainty into account.

Most studies in the multi-objective bilevel optimization literature focus on solving the optimization problem without addressing the impact of uncertainty. In practical problems, noise in the leader's objectives might represent environmental uncertainty, for example, in a meta-learning regime [ABB+17] that can be mathematically formulated as BOP [FFS+18]. As another example, a government might need to prevent terrorist attacks using information from unreliable sources. Yet another example occurs in computing optimal recovery policies

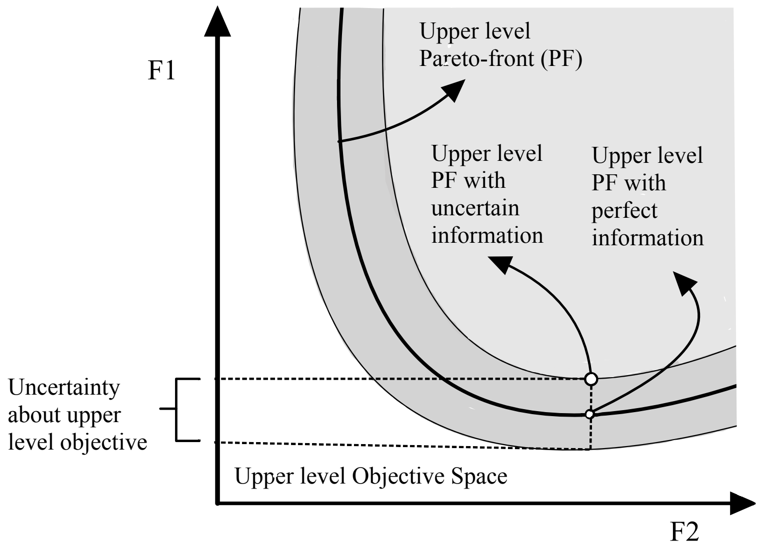


Figure 6.1: Decision making under upper-level uncertainty.

for financial markets [MBD12], using bilevel optimization with objective uncertainties caused by several uncontrollable parameters.

BOPs are computationally expensive to solve because of their nested structure, and they become even more complex when there are multiple objectives and uncertainty (possibly at both levels). The main purpose of our work is to improve the efficiency of solving multi-objective bilevel optimization while handling leader objective uncertainties.

6.2 Proposed Approach

We consider a case that is crucial in practice, in which the leader must make decisions under uncertainty based on noisy observations $\mathbf{F}_i = \mathbf{f}(\mathbf{x}_{u_i}, \mathbf{x}_{l_i}) + \xi_i$ where $\xi_i \sim \mathcal{N}(0, \Sigma_i)$ and Σ_i is the noise covariance and $\mathbf{x}_u, \mathbf{x}_l$ are upper and lower decision variables respectively. We reformulate the leader's objective with noisy observations as in Equation 6.1.

$$\underset{\mathbf{x}_u,\mathbf{x}_l}{\text{minimize}}\{F_1(\mathbf{x}_u,\mathbf{x}_l)+\xi_1,...,F_p(\mathbf{x}_u,\mathbf{x}_l)+\xi_p\} \tag{6.1}$$

where $\xi_p \sim \mathcal{N}(0, \Sigma_p)$. The HV indicator measures the volume of space between the non-dominated front and a reference point, which we assume is known by the upper-level decision-maker. The selection of reference points is tricky. In this work, the reference point is chosen to be an extreme points of the Pareto-optimal front, because reference points should be dominated by all

Algorithm 5 BAMBINO

Inputs: $\mathbf{F}_u(\mathbf{x}_u, \mathbf{x}_l) : \mathbf{x_u} \in \mathbb{X}_u, \mathbf{x_l} \in \mathbb{X}_l$,
Batch points in each iteration *Q*,
The number of iterations for BO: *N*,
Reference point

1: $\mathbf{x_l}$: Find the Best Lower-level response as parameters with NSGA-II algorithm,
2: Initial decision data set with the objective noise
3: $D = \mathbf{x}_{u_i}, \mathbf{F}_u(\mathbf{x}_{u_i}, \mathbf{x}_{l_i}), (\Sigma_i)_{i=1}^n$ with size of n,
4: Initialize the $\mathcal{GP}$ model with the observations and the objective noise
5: **for** i = 0 : N **do**
6:     Suggest new q-batch points by optimizing *qNEHVI*
7:     **for** j = 0 : q **do**
8:         For each upper-level decision $\mathbf{x}_u$, find optimal $\mathbf{x}_l^*$ by applying the NSGA-II
9:         Calculate fitness scores with noise $\mathbf{F}_u(\mathbf{x}_u, \mathbf{x}_l^*) + \xi$
10:     Update the data set D with new observations
11: **Return** Pareto-optimal front $\mathbf{F}_u^*$ and $(\mathbf{x}_u, \mathbf{x}_l^*)$

Pareto-optimal solutions.

HVI of a set of points $\mathcal{P}'$ is defined as $HVI(\mathcal{P}'|\mathcal{P}, \mathbf{r}) = HV(\mathcal{P} \cup \mathcal{P}'|\mathbf{r}) - HV(\mathcal{P}|\mathbf{r})$ where $\mathcal{P}$ represents the Pareto front and $\mathbf{r}$ the reference point. Given observations of the upper-level decision-making process, the $\mathcal{GP}$ surrogate model provides us with a posterior distribution over the upper-level function values for each observation. These values can be used to compute the EHVI acquisition function, formulated in Equation 6.2 defined by

$$\alpha_{ehvi}(\mathbf{x}_u|\mathcal{P}) = \mathbb{E}[HVI(\mathbf{F}_u|\mathcal{P})] \tag{6.2}$$

So the EHVI iterates over the posterior distribution, an approach that worked well in [DBB21a]. After n observations of the leader's decisions and the follower's response, the posterior distribution can be defined by the conditional probability $p(\mathbf{F}(\mathbf{x}_{u_n}, \mathbf{x}_{l_n})|\mathcal{D}_n)$ of the leader's objective values given decision variables $(\mathbf{x}_{u_n}, \mathbf{x}_{l_n})$ based on *noisy* observations $\mathcal{D}_n = \mathbf{x}_{u_i}, \mathbf{F}_i(\mathbf{x}_{u_i}, \mathbf{x}_{l_i}), (\Sigma_i)_{i=1}^n$. NEHVI is defined as in Equation 6.3

$$\alpha_{\mathit{NEHVI}}(\mathbf{x}_u) = \int \alpha_{ehvi}(\mathbf{x}_u|P_n)p(\mathbf{F}|\mathcal{D}_n)d\mathbf{F} \tag{6.3}$$

where P_n denotes the Pareto-optimal front, the optimal decision set over the leader's objectives $\mathbf{F}_n$. The aim is to improve the efficiency of the optimization, and the handling of noise in the leader's objective, by using the approach above and reformulating the multi-objective BOP. The algorithm details can be found in Algorithm 5.

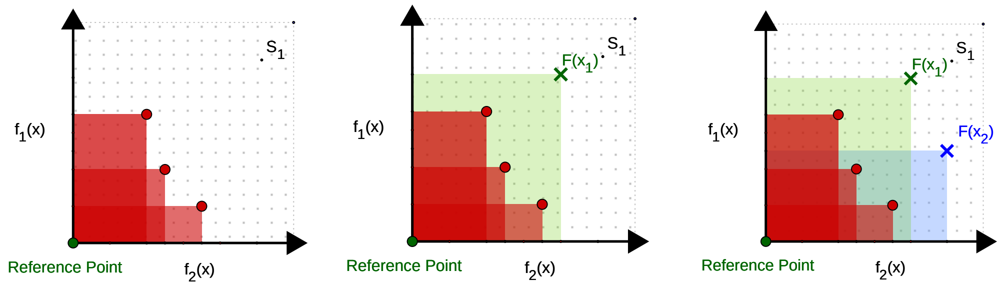


Figure 6.2: An illustration of the dominated (red area) and non dominated (white area) space. The green and blue area on the graphs represents the hypervolume improvement of the new points.

6.3 Numerical Experiments

The test problems are selected from the literature [DS09c], to test scalability in terms of decision variable dimensionality. The results are compared with state-of-art evolutionary algorithms m-BLEAQ [SMD+16b] and H-BLEMO [DS10b]. The Pareto-optimal front is independent of the parameters. Also, we use a real-world problem from environmental economics literature which considers a hierarchical decision-making problem between an authority and a gold mining company [SMFD13b].

- **Example 1.** The first example is formulated in Equation 6.4 and is a bi-objective problem that is scalable in terms of the number of follower decision variables. We choose $K = 14$ and $K = 19$, giving 15 and 20 follower variables respectively, with 1 leader decision variable. We choose the reference point required to measure hypervolume improvement to be

$(1.0, 0.5)$.

$$\underset{(x_u,\mathbf{x_l})}{\text{Min}}\ \mathbf{F}(x_u,\mathbf{x_l}) = \begin{pmatrix} (x_{l_1}-1)^2 + \sum_{i=2}^{K} x_{l_i} + \\ (x_u)^2 + \xi \\ (x_{l_1}-1)^2 + \sum_{i=2}^{K} x_{l_i} + \\ (x_u - 1)^2 + \xi \end{pmatrix}$$

$$\text{subject to} \quad \mathbf{x}_l \in \underset{\mathbf{x_l}}{\text{argmin}}\ \mathbf{f}(x_u,\mathbf{x_l}) = \begin{pmatrix} x_{l_1} + \sum_{i=2}^{K} x_{l_i}^2 \\ x_{l_1} - x_u + \sum_{i=2}^{K} x_{l_i}^2 \end{pmatrix} \tag{6.4}$$

$$-1 \leq (x_u, x_{l_1}, x_{l_2}, ..., x_{l_K}) \leq 2$$

$$\xi \sim \mathcal{N}(0,\Sigma_\xi),\ \Sigma_\xi = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$$

The Pareto-optimal decision set of the lower-level optimization problems for any given x_u is given by Equation 5.5 in Chapter 5.

- **Example 2.** The second test problem is formulated in Equation 6.5 and is the modified test problem with 10 and 20 variable instances. We choose the required reference point to be $(1.1, 1.1)$.

$$\underset{(\mathbf{x_u},\mathbf{x_l})}{\text{Min}}\ \mathbf{F}(\mathbf{x_u},\mathbf{x_l}) =$$

$$\begin{pmatrix} (1 + r - \cos(\alpha\pi x_{u_1})) + \sum_{j=2}^{K}(x_{u_j} - \frac{j-1}{2})^2 + \\ \tau\sum_{i=2}^{K}(x_{l_i} - x_{u_i})^2 - r\cos(\gamma\frac{\pi}{2}\frac{x_{l_1}}{x_{u_1}}) + \xi \\ (1 + r - \sin(\alpha\pi x_{u_1})) + \sum_{j=2}^{K}(x_{u_j} - \frac{j-1}{2})^2 + \\ \tau\sum_{i=2} K(x_{l_i} - x_{u_i})^2 - r\sin(\gamma\frac{\pi}{2}\frac{x_{l_1}}{x_{u_1}} + \xi) \end{pmatrix}$$

subject to

$$\mathbf{x}_l \in \underset{\mathbf{x_l}}{\text{argmin}}\ \ \mathbf{f}(\mathbf{x_u},\mathbf{x_l}) = \begin{pmatrix} x_{l_1}^2 + \sum_{i=2}^{K}(x_{l_i} - x_{u_i})^2 + \\ \sum_{i=2}^{K} 10(1 - \cos(\frac{\pi}{K}(x_{l_i} - x_{u_i})) \\ \sum_{i=2}^{K}(x_{l_i} - x_{u_i})^2 + \\ \sum_{i=2}^{K} 10|\sin(\frac{\pi}{K}(x_{l_i} - x_{u_i}))| \end{pmatrix} \tag{6.5}$$

$$x_{l_i} \in [-K, K], i = 1, \dots, K$$

$$x_{u_1} \in [1, 4], x_{u_j} \in [-K, K], j = 2, \dots, K$$

$$\xi \sim \mathcal{N}(0,\Sigma_\xi),\ \Sigma_\xi = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.16 \end{bmatrix}$$

The Pareto-optimal front for a given leader is defined as a circle of radius $(1+r)$ with centre $((1+r),(1+r))$. We choose $K = 5$ for our experiments with parameters $r = 0.1, \tau = 1$ and $\alpha = 1$, following [SMD+16b] so that our results can be compared with those for m-BLEAQ and H-BLEMO.

- **Example 3.** The third test problem is formulated in Equation 6.6 and is the modified test problem with 10 and 20 variable instances. We choose the required reference point $(0.8, 0.0)$ for measuring the HVI during the optimization. More details can be found about the Pareto optimal solutions in [DS09a].

$$
\begin{aligned}
&\underset{(\mathbf{x_u},\mathbf{x_l})}{\text{Min}}\ \mathbf{F}(\mathbf{x_u},\mathbf{x_l}) = \begin{pmatrix} v_1(x_{u_1}) + \sum_{j=2}^{K}(x_{u_j}^2 + 10(1-\cos(\frac{\pi}{K}x_{u_i}))+ \\ \tau\sum_{i=2}^{K}(x_{l_i}-x_{u_i})^2 - r\cos(\gamma\frac{\pi}{2}\frac{x_{l_1}}{x_{u_1}}) + \xi \\ v_2(x_{u_1}) + \sum_{j=2}^{K}(x_{u_j}^2 + 10(1-\cos(\frac{\pi}{K}x_{u_i}))+ \\ \tau\sum_{i=2}^{K}(x_{l_i}-x_{u_i})^2 - r\sin(\gamma\frac{\pi}{2}\frac{x_{l_1}}{x_{u_1}}) + \xi \end{pmatrix} \\
&\text{subject to }\ \mathbf{x}_l \in \underset{\mathbf{x_l}}{\text{argmin}}\ \mathbf{f}(\mathbf{x_u},\mathbf{x_l}) = \begin{pmatrix} x_{l_1}^2 + \sum_{i=2}^{K}(x_{l_i}-x_{u_i})^2 \\ \sum_{i=2}^{K} i(x_{l_i}-x_{u_i})^2 \end{pmatrix} \\
&x_{l_i} \in [-K, K], i = 1, \ldots, K \\
&x_{u_1} \in [0.001, K], x_{u_j} \in [-K, K], j = 2, \ldots, K \\
&\xi \sim \mathcal{N}(0, \Sigma_\xi),\ \Sigma_\xi = \begin{bmatrix} 0.09 & 0 \\ 0 & 0.09 \end{bmatrix} \\
&\text{where }\ v_1(x_{u_1}) = \begin{cases} \cos(0.2\pi)x_{u_1} + \\ \quad \sin(0.2\pi)\sqrt{|0.02\sin(5\pi x_{u_1})|} \\ \text{for } 0 \le x_{u_1} \le 1, \\ x_{u_1} - (1-\cos(0.2\pi)) \ \text{ for } x_{u_1} \ge 1. \end{cases} \\
&\qquad\quad v_2(x_{u_1}) = \begin{cases} -\sin(0.2\pi)x_{u_1} + \\ \quad \cos(0.2\pi)\sqrt{|0.02\sin(5\pi x_{u_1})|}, \\ \text{for } 0 \le x_{u_1} \le 1. \\ 0.01(x_{u_1}-1) - \sin(0.2\pi), \ \text{ for } x_{u_1} \ge 1. \end{cases}
\end{aligned} \tag{6.6}
$$

6.3.1 Algorithms Used for Comparison

- **m-BLEAQ**: [SMD+16b] proposed an algorithm for multi-objective BOPs which is an evolutionary algorithm based on quadratic approximations for the lower-level decisions. It is an extended version of the BLEAQ [SMD13a] algorithm for multi-objective problems.

- **H-BLEMO**: Another algorithm is selected to compare the results and presented in Section 5.3.1.

6.3.2 Parameters

We fixed the number of BO iterations to $N = 50$ and repeated our experiments 21 times to obtain median results for making the comparison fair. We use the independent $\mathcal{GP}$ model with *Matern52* kernel and fit the $\mathcal{GP}$ by maximizing the marginal log-likelihood. The method is initialized with $2 \times (d + 1)$ Sobol points where d represents the dimension of the problem to construct the initial $\mathcal{GP}$ model. All experiments are conducted using the BoTorch [BKJ$^+$19] library. We solved the follower's problem with the popular *non-dominated sorted genetic algorithm* (NSGA-II) and chose the population size 100 and number of generations 200. We choose the follower's decisions from the obtained Pareto-optimal front at random, as all solutions in the Pareto-optimal front are feasible.

6.3.3 Performance Metrics

We compare our results in terms of upper-level function evaluations (FE) to determine the efficiency of the algorithm as BO aims to minimize the function evaluations while optimizing the expensive black-box functions. HV [FPLI06] and IGD [CCRS04] are also used to evaluate the success of approximation to Pareto-optimal fronts, in terms of convergence and diversity, discussed in more detail in Section 2.4. HV measures the volume of the space between the non-dominated front obtained and a reference point. IGD calculates the sum of the distances from each point of the true Pareto-optimal front to the nearest point of the non-dominated set found by the algorithm. Therefore, a smaller IGD value means approximated points are closer to the Pareto-optimal front of the problem.

6.3.4 Results of Test Problems

The performance of BAMBINO is compared with m-BLEAQ and H-BLEMO in Table 6.1, showing computational expense and convergence. The FE is calculated by $N_{initial} + (N_{batch} \times N_{iter} \times N_{restarts})$ for the leader problem, where $N_{initial}$ is the number of initial decisions for starting the algorithm and $N_{restarts}$ is the parameter for Gaussian process declares the number of restart to avoid to stuck at local optima. We choose it to be $2 \times (d + 1)$ where d is the dimensions of the decision variable. We run the experiment for different batch numbers $q = 1, q = 2, q = 4, q = 8$ to test the effect on performance. The HV difference is shown in Figure 6.3 for 15 and 20 variables, and 10 variables for Exam-

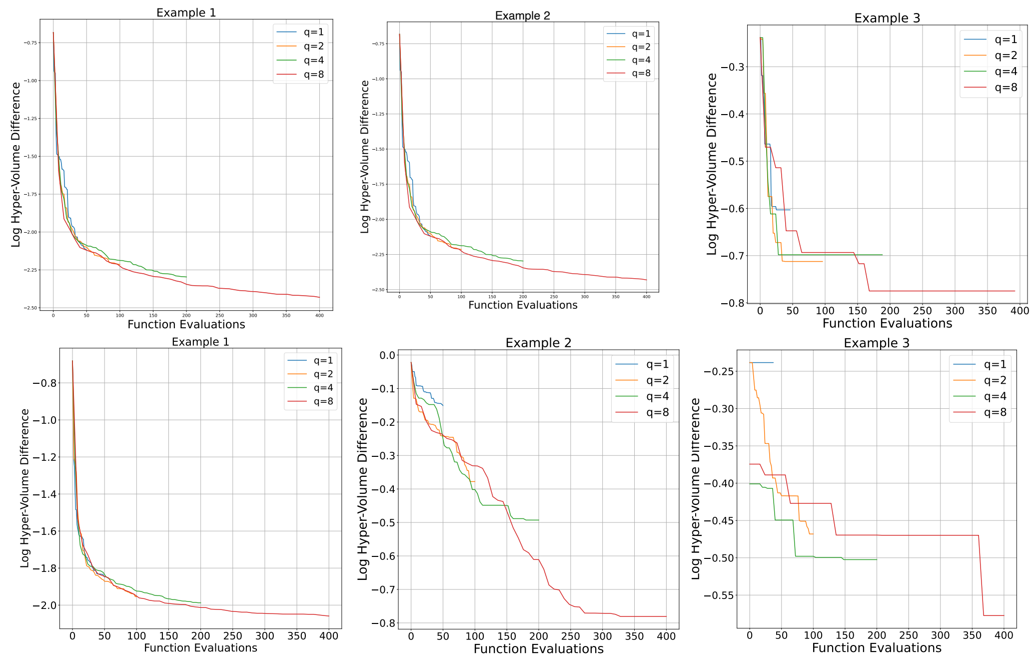


Figure 6.3: Hypervolume difference graph (log scale) with different batch sizes ($q = 1, q = 2, q = 4, q = 8$) for Example 1 with 15 (*top-left*) and 20 (*bottom-left*) dimensions, for Example 2 with 10 (*top-middle*) and 20 (*bottom-middle*) dimensions, for Example 3 with 10 (*top-right*) and 20 (*bottom-right*) dimensions.

ples 1 and 2 respectively. While increasing the batch size, decreasing the HV difference as presented in Figure 2.7 shows the convergence of the proposed algorithm. Because of a lack of information in the reference paper, we could not obtain the FE results for Example 1 with 20 variables.

- **Example 1.** We can see from Table 6.1 that the required upper-level FE is significantly lower, while the algorithm approximates successfully to the Pareto-optimal front while handling the uncertainty at the leader's objective. For 15 variables BAMBINO achieves $\approx \%38$ improvement in terms of FE compared to m-BLEAQ and $\approx \%89$ compared to H-BLEMO. The IGD values in Table 6.1 for 15 and 20 variables show that BAMBINO successfully approximates the Pareto-optimal front of the problem while handling the uncertainty of the leader's objective for both. We show the HV difference between Pareto-optimal front solutions and approximated BAMBINO decisions algorithm in Figure 6.3. Again we tried different batch sizes for the experiment, and it can be seen that a batch number of 8 is best for this specific example at both dimensions. We could not compare the 20 dimensional version of the problem with the selected algorithms because of the lack of information in [SMD$^+$16b].

Table 6.1: FE and IGD values for the examples with the number of variable dimensions for the batch size $q = 8$.

	Number of Variables	BAMBINO		m-BLEAQ		H-BLEMO	
		IGD	FE	IGD	FE	IGD	FE
Example 1	15	0.0044	4032	0.0013	6,464	0.0046	39,818
Example 2	10	0.0051	4022	0.0069	22,223	0.0134	106,003
Example 3	10	0.0076	4022	0.0079	25,364	0.0134	132,907
Example 1	20	0.0105	4042	-	-	-	-
Example 2	20	0.1032	4042	0.0435	34,110	0.1106	191,357
Example 3	20	0.0924	4042	0.0623	36,439	0.1321	216,083

- **Example 2.** Table 6.1 shows that BAMBINO obtains the best IGD results compared to the other algorithms. In terms of FE, it significantly improves state of the art, with $\approx \%81$ improvement for 10 variables and $\approx \%88$ improvement for 20 variables compared to m-BLEAQ. We also show the HV difference in Figure 6.3 and we can observe that, for this specific example, the batch number of 8 is the best selection for both 10 and 20 dimensional versions.

- **Example 3.** Table 6.1 shows that BAMBINO obtains the best IGD value compared to m-BLEAQ and H-BLEMO while improving efficiency in terms of FE: $\approx 84\%$ and $\approx 97\%$ with 10 variables, and $\approx 89\%$ and $\approx 98\%$ with 20 variables. Figure 6.3shows that batch size $q = 8$ gives the best results.

In summary, BAMBINO is successful on the selected test problems while handling noisy objectives with less computational cost. Noise in the leader's objective makes the problem harder to solve but more realistic for modelling practical problems, because of real-world uncertainty. We show the proposed BAMBINO algorithm works well on these test benchmark problems.

6.3.5 Practical Case: Gold Mining in Kuusamo

Kuusamo region is a popular tourist destination known for its natural beauty. There is a lot of interest in this region cause of contains a huge amount of gold deposits, considered to be a "highly prospective Paleoproterozoic Kuusamo Schist Belt" [Wol20]. The expected gold amount in the ore is around 4.9 g per ton according to an Australia-based gold mining company. Even though there is a big potential in terms of providing lots of jobs in the region and leading to a great amount of gold resources, there are some concerns about harming the environment. The first of them is that mining operations may cause pollution of the river water in the region. It causes fear for environmentalists. Second, the ore in the region contains uranium and if it is mined, it might decrease the reputation of the tourist resorts around. Another one is the open pit mines around the area called Ruka will cause a turn-off for skiing resorts and hiking routes around and it will decrease tourist interest.

The regulating authority, which is the government, acts as a leader and the mining company is the follower which reacts rationally to the decisions of the leader to maximize its profit. The leader should find an optimal strategy assuming that he holds the necessary information about the follower. In the situation explained above, the government has a decision-making problem which is whether to allow mining and to what extent. In the problem above, the leader has two objectives while the follower has one. The first objective is maximization of revenues coming from the project and the second objective is to minimize the environmental harm which is a result of the mining. The mining company also aims to maximize its profit. While the government is optimizing its taxation strategy, it needs to model how the mining company reacts to any given tax structure. Therefore, the authority makes an environmental regulatory decision instead of solving the problem to optimality. The objectives are conflicting such as large profits may affect the environment by increasing the damage which follows with a bad public image. The mathematical formulation of this hierarchical decision-making problem can be found in Table 6.2. More

Table 6.2: Gold Mining in Kuusamo

Category	Level	Formulation
Variables	Upper-level	τ [per unit tax imposed on the mine]
	Lower-level	q [amount of metal extracted by the mine]
Objectives	Upper-level	$F_1(\tau, q) = E(\tau, q) = \tau q$ [tax revenue] $F_2(\tau, q) = -D(q) = -kq$ [environmental damage]
	Lower-level	$f_1(\tau, q) = \pi(\tau, q) = (\alpha - \beta q)q - (\delta q^2 + \gamma q + \phi) - \tau q$ [profit]
Constraints	Upper-level	$q \geq 0, \tau \geq 0$
	Lower-level	$\pi(\tau, q) \geq 0$
Uncertainty	Upper-level	$\xi \sim N(\mu_\xi, \Sigma_\xi), \mu_\xi = (1, 1), \Sigma_\xi = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}$
Parameters		$k = 1, \eta = 1$ $\alpha = 100, \beta = 1, \delta = 1, \gamma = 1, \phi = 0$

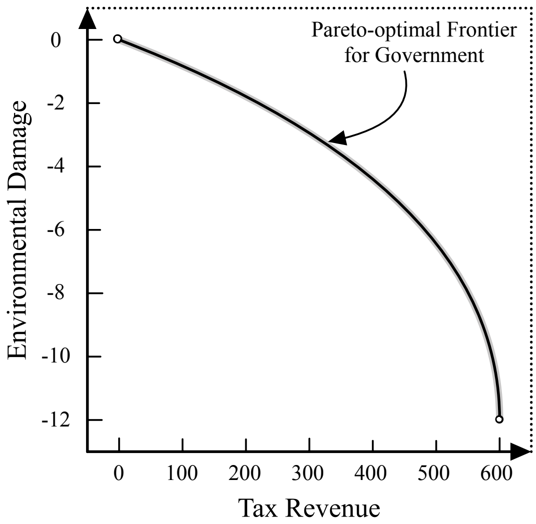


Figure 6.4: The true Pareto optimal frontier for the government represents the trade off between tax revenues and environmental pollution.

details can be found about the bilevel modelling of the problem in [SMFD13b] and [SMD+16c]. Figure 6.4 presents the Pareto-optimal frontier of the given problem for the government according to the formulation in Table 6.2. We can observe that increasing the tax revenue decreases environmental damage.

6.3.5.1 Results of Practical Case

In this section, we present the results obtained using the BAMBINO algorithm on the analytical model of the problem proposed in Table 6.2. Figure 6.5 shows the Pareto-optimal front obtained using the BAMBINO approach. The plot gives the idea to the authority how to consider the trade-off between its objectives. We used 50 iterations for upper-level optimization using a batch Bayesian approach with a batch size of 4. For lower-level optimization, the NSGA-II algorithm is implemented with the same parameters specified in Section 5.3.2. We can observe that with the proposed method, the obtained results are distributed around the true Pareto-optimal frontier and are approximated to it successfully. We also run the experiments for the batch size $q = 1$,$q = 2$ and $q = 4$ for testing if the algorithm converges to the Pareto-optimal front. The IGD value is 0.0494 for $q = 1$, 0.0033 for $q = 2$ and 0.0021 for $q = 4$. We can see from the decreasing IGD values, that as we increase the batch number, the selected points are more approximated to the true Pareto-optimal front. The increasing batch size provides us with parallel evaluations during upper-level optimization which decreases the needed execution time. It is also good to handle environmental uncertainty, represented by ξ in Table 6.2, which may

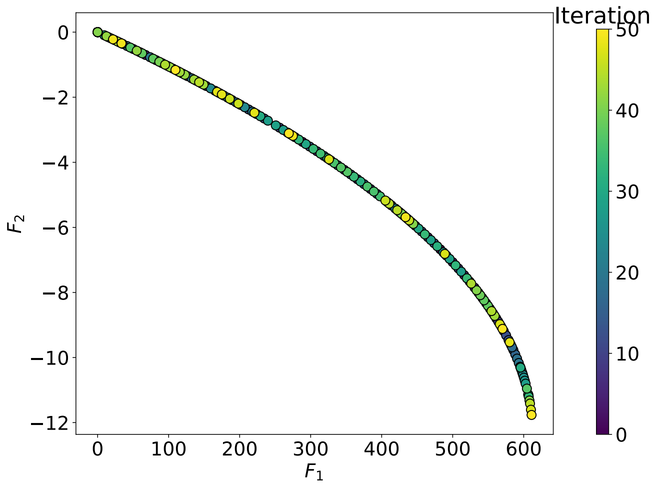


Figure 6.5: The obtained Pareto-optimal frontier for the government represents the trade off between tax revenues and environmental pollution.

be an uncontrollable parameter such as inflation during the time of taxation or unexpected environmental damage during the mining process. We can observe that the proposed method is successfully approximated even handling the uncertainty of objectives.

We believe that BAMBINO can be applied to several practical bilevel problems successfully applied in the machine learning community, such as image classification [MLA+20], deep learning [HLX+22], neural networks [LZ21a], and hyperparameter optimization [LGZ+21].

6.4 Conclusion

In this chapter, we discussed bilevel multi-objective optimization under upper-level uncertainty and presented a hybrid algorithm called BAMBINO, based on batch Bayesian optimization with hypervolume improvement. We ran experiments using three benchmark problems and one real-world problem from environmental economics literature which is a decision-making problem between an authority and a gold mining company. BAMBINO performed very competitively in terms of computational efficiency and convergence. We also showed how batch size selection affects performance in terms of hypervolume improvement.

Chapter 7

Conclusion and Future Work

We proposed and investigated several different approaches to handle single- and multi-objective generic bilevel optimization problems in this dissertation. Toward this goal, we demonstrated the benefits of these approaches through various numerical experiments and real-world problems. Also, we provided a comparison with the established techniques developed in the literature on bilevel optimization.

7.1 Conclusions

Bayesian approach for single-objective BPs

In Chapter 3, we presented the **Con**ditional **Ba**yesian optimization for **B**ilevel **O**ptimization (ConBaBo) algorithm. In this algorithm, a hybrid approach was developed where the goal is to search the global optimum at the upper-level via the surrogate-assisted model and the SLSQP algorithm is used at the lower-level. The main difference of the proposed approach is conditioning the lower-level decisions to improve the search at the upper-level. The approach attempts to improve the upper-level search by conditioning the previous lower-level decisions and the significant reduction in the number of function evaluations while obtaining competitive results with other state-of-the-art algorithms.

In Chapter 4, we presented a model of the multi objective acquisition function approach, called MACBP. We applied the multi-objective acquisition function idea to improve the search at the upper-level to reduce the required function evaluations while reaching competitive results. Upper-level problems are considered as black-box and solved by Bayesian optimization via a multi-objective acquisition function at each iteration. Then, the lower-level problem is optimized by using the SLSQP algorithm at each iteration and results are compared with other successful algorithms proposed in the literature. The results show that the approach improved the computational performance of bilevel optimization while reaching the competitive objective results.

Bayesian approach for multi-objective BPs

In Chapter 5, we tackled to solve multi-objective bilevel optimization problems and developed an algorithm, called BoMoBo. The algorithm considers the upper-level multi-objective problem as a black-box. It uses a specifically designed HVI-based acquisition function for multi-objective problems during the BO process. The lower-level problem is solved by an evolutionary algorithm, called NSGA-2 to obtain a Pareto-optimal front. The main goal of this approach was to decrease the computational cost during the optimization process while tackling the optima for multiple conflicting objectives. The experimental results show that the proposed approach approximates the known Pareto-optimal front while decreasing the necessary number of function evaluations. The approach got competitive results compared with other algorithms.

The BamBiNo algorithm is presented in Chapter 6 to tackle the multi-objective bilevel problems with upper-level objective uncertainty. It is an extended version of the BoMoBo algorithm presented in Chapter 5. When we consider black-box upper-level objective functions, it is highly possible and various kinds of real-world applications with the uncertain environment and these uncertainties affect the objective results. The BamBiNo algorithm attempted to deal with these problems by multi-objective Bayesian optimization and the presented results show that the approach is successfully approximated to the Pareto-optimal front while dealing with certainty. Moreover, it requires fewer function evaluations compared with other algorithms.

7.2 Future Work

We identified some future research directions that arise from developing approaches in this dissertation to deal with single- and multi-objective bilevel optimization problems. As the concept of the black-box approach to upper-level decision makers, many interesting different directions can be identified.

Fully approximating to both upper- and lower-levels

In Section 3, a surrogate-assisted model is used to approximate the upper-level objective. There is also a considerable amount of work approximating the lower-level problem in the literature. In such hybrid approaches, generally, evolutionary methods are used at the other level. Fully approximated approaches at both upper- and lower-level are still an open question in this literature and these strategies can be combined as future work. While optimizing the objectives at both levels, the black-box approach can make the full simulation optimization possible during the decision-making process. Several real-world applications are solved as bilevel problems with expensive approaches. Using the full approximation methods in these practical problems might be useful.

Probabilistic approaches to pessimistic BPs

To examine the performance of the proposed approaches in this dissertation, only an optimistic approach has been considered for selected bilevel problems. Pessimistic approaches need more attention with approximating at each level as several real-world scenarios exist with conflicting multi-level decision-making systems. The majority of the works presented in this dissertation were examined with generic bilevel benchmarks and several real-world applications in the literature can be solved by the presented approaches. Also, the uncertainty information during the proposed approach, such as the acquisition optimization step in Bayesian optimization, can be used for explainability during the decision-making process.

Game-theoretic decision-making model and simulation optimization

It is well known that bilevel optimization with both single- and multi-objective problems is widely used for decision-making systems [MPM22]. The approaches developed and presented in Chapter 3 and Chapter 5 can be applied to several practical problems, such as negotiations [ZQYS23] in diplomacy or optimizing the tax policy of authority while optimizing the specific objectives of a mining company as discussed in Section 6.3.5. Another interesting application to work on is the defence industry in terms of attacker-defender Stackelberg games. For instance, the positioning of the missile interceptors to counter an attack threat or interdicting nuclear weapons are some of them. The black-box approach at the upper level as presented in this dissertation is not dependent on the specifications of the problems. In this way, the simulation-optimization approach [SMdCMdSJ22] can be applied to multiple problems.

Acronyms

BO Bayesian Optimization. 8

BOP Bilevel Optimization Problem. 22

CDF Cumulative Distribution Function. 16

EHVI Expected Hypervolume Improvement. 21

EI Expected Improvement. 15

GD Generational Distance. 33

GP Gaussian Process. 8

GP-UCB Upper Confidence Bound. 16

HPR High Point Relaxation. 23

HV Hypervolume. 32

HVI Hypervolume Improvement. 20

IGD Inverted Generational Distance. 32

IR Inducible Region. 23

LCB Lower Confidence Bound. 46

MOP Multi-objective Optimization Problem. 19

PDF Probability Density Function. 16

PEI Profile Expected Improvement. 43

POI Probability of Improvement. 16

qEHVI q-Expected Hypervolume Improvement. 72

RBF Radial Basis Function. 49

References

[AAC21] Maria Alves, Carlos Antunes, and João Costa. New concepts and an algorithm for multiobjective bilevel programming: optimistic, pessimistic and moderate solutions. *Operational Research*, 21, 12 2021.

[AB15] Jaqueline S. Angelo and Helio Barbosa. A study on the use of heuristics to solve a bilevel programming problem. *International Transactions in Operational Research*, 22, 09 2015.

[ABB+17] Maruan Al-Shedivat, Trapit Bansal, Yuri Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. *CoRR*, abs/1710.03641, 2017.

[ADA14] Natalia Alguacil, Andrés Delgadillo, and José M. Arroyo. A trilevel programming approach for electric grid defense planning. *Computers & Operations Research*, 41:282–290, 2014.

[ADP20] Shahram Aghaei, Abolghasem Daeichian, and Vicenç Puig. Hierarchical decentralized reference governor using dynamic constraint tightening for constrained cascade systems. *Journal of the Franklin Institute*, 357(17):12495–12517, 2020.

[AEN20] Yousria Abo-Elnaga and Sarah Nasr. Modified evolutionary algorithm and chaotic search for bilevel programming problems. *Symmetry*, 12, 5 2020.

[AFF10] Javad Azimi, Alan Fern, and Xiaoli Fern. Batch bayesian optimization via simulation matching. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances* *in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[AKB13] Jaqueline S. Angelo, Eduardo Krempser, and Helio J.C. Barbosa. Differential evolution for bilevel programming. In *2013 IEEE Congress on Evolutionary Computation*, pages 470–477, 2013.

[AKB14] Jaqueline S. Angelo, Eduardo Krempser, and Helio J.C. Barbosa. Differential evolution assisted by a surrogate model for bilevel programming problems. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1784–1791, 2014.

[AOT+13] Bo An, Fernando Ordóñez, Milind Tambe, Eric Shieh, Rong Yang, Craig Baldwin, Joseph DiRenzo, Kathryn Moretti, Ben Maule, and Garrett Meyer. A deployed quantal response-based patrol planning system for the u.s. coast guard. *Interfaces*, 43(5):400–420, 2013.

[aut16] The GPyOpt authors. GPyOpt: A bayesian optimization framework in python. http://github.com/SheffieldML/GPyOpt, 2016.

[AW90] G. Anandalingam and D.J. White. A solution method for the linear static stackelberg problem using penalty functions. *IEEE Transactions on Automatic Control*, 35(10):1170–1173, 1990.

[BA93] Omar Ben-Ayed. Bilevel linear programming. *Computers & Operations Research*, 20(5):485–501, 1993.

[Bar83a] Jonathan F Bard. Coordination of a multidivisional organization through two levels of management. *Omega*, 11(5):457–468, 1983.

[Bar83b] Jonathan F Bard. Coordination of a multidivisional organization through two levels of management. *Omega*, 11(5):457–468, 1983.

[BB12] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(null):281–305, feb 2012.

[BBBK11] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[BBKS13] R. Bardenet, Mátyás Brendel, Balázs Kégl, and M. Sebag. Collaborative hyperparameter tuning. *30th International Conference on Machine Learning, ICML 2013*, 28, 06 2013.

[BCD+05] Gerald Brown, Matt Carlyle, Douglas Diehl, Jeffrey Kline, and R. Wood. A two-sided optimization for theater ballistic missile defense. *Operations Research*, 53:745–763, 10 2005.

[BCF10] Eric Brochu, Vlad Cora, and Nando Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *CoRR*, abs/1012.2599, 12 2010.

[BD20] J. Blank and K. Deb. pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020.

[BEA21] Amir Hossein Barahimi, Alireza Eydi, and Abdolah Aghaie. Multi-modal urban transit network design considering reliability: multi-objective bi-level optimization. *Reliability Engineering & System Safety*, 216:107922, 2021.

[Ben95] Martin P. Bendsøe. Optimization of structural topology, shape, and material. In *Optimization of Structural Topology, Shape, and Material*, 1995.

[BF82] Jonathan F. Bard and James E. Falk. An explicit solution to the multi-level programming problem. *Computers & Operations Research*, 9(1):77–100, 1982.

[BK84] Wayne F. Bialas and Mark H. Karwan. Two-level linear programming. *Management Science*, 30(8):1004–1020, 1984.

[BKJ+19] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. Botorch: Programmable bayesian optimization in pytorch. *CoRR*, abs/1910.06403, 2019.

[BLS23] Yasmine Beck, Ivana Ljubić, and Martin Schmidt. A survey on bilevel optimization under uncertainty. *European Journal of Operational Research*, 2023.

[BM73] Jerome Bracken and James T. McGill. Mathematical programs with optimization problems in the constraints. *Operations Research*, 21(1):37–44, 1973.

[BM74] Jerome Bracken and James T. McGill. Defense applications of mathematical programs with optimization problems in the constraints. *Oper. Res.*, 22(5):1086–1096, oct 1974.

[BM90] Jonathan F. Bard and James T. Moore. A branch and bound algorithm for the bilevel programming problem. *SIAM Journal on Scientific and Statistical Computing*, 11(2):281–292, 1990.

[Büh12] Peter Bühlmann. *Bagging, Boosting and Ensemble Methods*, pages 985–1022. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[Bul11] Adam D. Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(88):2879–2904, 2011.

[CAA15] Pedro Carrasqueira, Maria Alves, and Carlos Antunes. A bilevel multiobjective pso algorithm. In *EMO*, volume 9018, pages 263–276, 03 2015.

[Can77] W. Norton R. Candler. *Multi-level programming and development policy*. The World Bank, 1977.

[CBRV13] Emile Contal, David Buffoni, Alexandre Robicquet, and Nicolas Vayatis. Parallel gaussian process optimization with upper confidence bound and pure exploration. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 225–240, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[CCRS04] Carlos A. Coello Coello and Margarita Reyes Sierra. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In Raúl Monroy, Gustavo Arroyo-

Figueroa, Luis Enrique Sucar, and Humberto Sossa, editors, *MICAI 2004: Advances in Artificial Intelligence*, pages 688–697, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[CdLP21] Martina Cerulli, Claudia d'Ambrosio, Leo Liberti, and Mercedes Pelegrín. Detecting and solving aircraft conflicts using bilevel programming. *Journal of Global Optimization*, 2021.

[CG13] Clément Chevalier and David Ginsbourger. Fast computation of the multi-points expected improvement with applications in batch selection. In Giuseppe Nicosia and Panos Pardalos, editors, *Learning and Intelligent Optimization*, pages 59–69, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[CGO11] Herminia I. Calvete, Carmen Galé, and María-José Oliveros. Bilevel model for production–distribution planning solved by using ant colony optimization. *Computers & Operations Research*, 38(1):320–327, 2011. Project Management and Scheduling.

[CJ97] Dennis D. Cox and Susan John. Sdo: A statistical method for global optimization. In *in Multidisciplinary Design Optimization: State-of-the-Art*, pages 315–329, 1997.

[CKLK10] Anthony Chen, Juyoung Kim, Seungjae Lee, and Youngchan Kim. Stochastic multi-objective models for network design problem. *Expert Systems with Applications*, 37(2):1608–1619, 2010.

[CMS02] Benoit Colson, Patrice MARCOTTE, and Gilles SAVARD. *A trust-region method for nonlinear bilevel programming: algorithm and computational experience*. FUNDP, Faculté des Sciences. Département de Mathématique., 2002.

[CMS05] Benoît Colson, Patrice Marcotte, and Gilles Savard. A trust-region method for nonlinear bilevel programming: Algorithm and computational experience. *Computational Optimization and Applications*, 30:211–227, 03 2005.

[CPW01] Snorre Christiansen, Michael Patriksson, and L. Wynter. Stochastic bilevel programming in structural optimization.

Structural and Multidisciplinary Optimization, 21:361–371, 07 2001.

[CVMOLR15] José-Fernando Camacho-Vallejo, Julio Mar-Ortiz, Francesc Lopez, and Ricardo Rodríguez. A genetic algorithm for the bi-level topological design of local area networks. *PLoS ONE*, 10:1–21, 06 2015.

[CW90] P.A. Clark and A.W. Westerberg. Bilevel programming for steady-state chemical process design—i. fundamentals and algorithms. *Computers & Chemical Engineering*, 14(1):87–97, 1990.

[DBB20a] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization, 2020.

[DBB20b] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NeurIPS, Red Hook, NY, USA, 2020. Curran Associates Inc.

[DBB21a] Sam Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In *Neural Information Processing Systems*, 2021.

[DBB21b] Sam Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In *Neural Information Processing Systems*, 2021.

[Dem02] Stephan Dempe. *Foundations of Bi-Level Programming*. Springer, 01 2002.

[Dem05] Stephan Dempe. *Bilevel Programming*, pages 165–193. Springer US, Boston, MA, 2005.

[Dem20] Stephan Dempe. *Bilevel Optimization: Theory, Algorithms, Applications and a Bibliography*, pages 581–672. Springer, 11 2020.

[DGMnCnEG+19] Noemí DeCastro-García, Ángel Luis Muñoz Castañeda, David Escudero García, Miguel V. Carriegos, and Fernando Sánchez Lasheras. Effect of the sampling of a dataset in the hyperparameter optimization phase over the efficiency of a machine learning algorithm. *Complex.*, 2019, jan 2019.

[DKB14] Thomas Desautels, Andreas Krause, and Joel W. Burdick. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *Journal of Machine Learning Research*, 15(119):4053–4103, 2014.

[DKPVK15a] Stephan Dempe, Vyacheslav Kalashnikov, Gerardo A. Prez-Valds, and Nataliya Kalashnykova. *Bilevel Programming Problems: Theory, Algorithms and Applications to Energy Networks*. Springer Publishing Company, Incorporated, 2015.

[DKPVK15b] Stephan Dempe, Vyacheslav Kalashnikov, Gerardo A. Prez-Valds, and Nataliya Kalashnykova. *Bilevel Programming Problems: Theory, Algorithms and Applications to Energy Networks*. Springer Publishing Company, Incorporated, 2015.

[DLM20] Teodora Dan, Andrea Lodi, and Patrice Marcotte. Joint location and pricing within a user-optimized environment. *EURO Journal on Computational Optimization*, 8(1):61–84, 2020.

[DP23] Vedat Dogan and Steven Prestwich. Bayesian optimization with multi-objective acquisition function for bilevel problems. In Luca Longo and Ruairi O'Reilly, editors, *Artificial Intelligence and Cognitive Science*, pages 409–422, Cham, 2023. Springer Nature Switzerland.

[DPAM02] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[DS09a] Kalyanmoy Deb and Ankur Sinha. Constructing test problems for bilevel evolutionary multi-objective optimization. In *2009 IEEE Congress on Evolutionary Computation*, pages 1153–1160, 2009.

[DS09b] Kalyanmoy Deb and Ankur Sinha. Solving bilevel multi-objective optimization problems using evolutionary algorithms. In *International Conference on Evolutionary Multi-Criterion Optimization*, 2009.

[DS09c] Kalyanmoy Deb and Ankur Sinha. Solving bilevel multi-objective optimization problems using evolutionary algorithms. In *EMO*, 2009.

[DS10a] Kalyan Deb and Ankur Sinha. An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary-local-search algorithm. *Evolutionary computation*, 18:403–49, 09 2010.

[DS10b] Kalyan Deb and Ankur Sinha. An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary-local-search algorithm. *Evolutionary computation*, 18:403–49, 09 2010.

[DS10c] Kalyanmoy Deb and Ankur Sinha. An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary-local-search algorithm. *Evolutionary Computation*, 18(3):403–449, 2010.

[dVWR20] Aurora del Valle, Sonja Wogrin, and Javier Reneses. Multi-objective bi-level optimization model for the investment in gas infrastructures. *Energy Strategy Reviews*, 30:100492, 2020.

[DZ20] Stephan Dempe and Alain Zemkoho. *Bilevel Optimization: Advances and Next Challenges*. Springer, 11 2020.

[EB91] T.A. Edmunds and J.F. Bard. Algorithms for nonlinear bilevel mathematical programs. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(1):83–89, 1991.

[EDK11] Michael T. M. Emmerich, André H. Deutz, and Jan Willem Klinkenberg. Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 2147–2154, 2011.

[EGN06] M.T.M. Emmerich, K.C. Giannakoglou, and B. Naujoks. Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

[EHHLB15] Katharina Eggensperger, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Efficient benchmarking of hyperparameter optimizers via surrogates. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 1114–1120. AAAI Press, 2015.

[Eic10] Gabriele Eichfelder. Multiobjective bilevel optimization. *Math. Program.*, 123:419–449, 06 2010.

[FAM81] José Fortuny-Amat and Bruce McCarl. A representation and economic interpretation of a two-level programming problem. *The Journal of the Operational Research Society*, 32(9):783–792, 1981.

[FFS+18] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1568–1577. PMLR, 10–15 Jul 2018.

[FKH18] Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: robust and efficient hyperparameter optimization at scale. *CoRR*, abs/1807.01774, 2018.

[FM11] Wei Fan and Randy Machemehl. Bi-level optimization model for public transportation network redesign problem. *Transportation Research Record: Journal of the Transportation Research Board*, 2263:151–162, 12 2011.

[FMVH19] Ferdinando Fioretto, Terrence W.K. Mak, and Pascal Van Hentenryck. Privacy-preserving obfuscation of critical infrastructure networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-*

19, pages 1086–1092. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[FPLI06] C.M. Fonseca, L. Paquete, and M. Lopez-Ibanez. An improved dimension-sweep algorithm for the hypervolume indicator. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1157–1163, 2006.

[Fra18] P. Frazier. A tutorial on bayesian optimization. *ArXiv*, abs/1807.02811, 2018.

[GBC+13] David Ginsbourger, Jean Baccou, Clément Chevalier, Frederic Perales, Nicolas Garland, and Yann Monerie. Bayesian adaptive reconstruction of profile optima and optimizers. *SIAM/ASA Journal on Uncertainty Quantification*, 2, 12 2013.

[GCF+12] Steven A. Gabriel, Antonio J. Conejo, J. David Fuller, Benjamin F. Hobbs, and Carlos Ruiz. *Complementarity Modeling in Energy Markets*. Springer Publishing Company, Incorporated, 2012.

[GD12] Nazih Gadhi and S. Dempe. Necessary optimality conditions and a new approach to multiobjective bilevel optimization problems. *Journal of Optimization Theory and Applications*, 155, 10 2012.

[GDHL15] Javier I. González, Zhenwen Dai, Philipp Hennig, and Neil D. Lawrence. Batch bayesian optimization via local penalization. In *International Conference on Artificial Intelligence and Statistics*, 2015.

[GHMA+15] Pablo Garcia-Herreros, Pratik Misra, Erdem Arslan, Sanjay Mehta, and Ignacio Grossmann. Mixed-integer bilevel optimization for capacity planning with rational markets. *Computers & Chemical Engineering*, 86, 12 2015.

[GKX+14] Jacob R. Gardner, Matt J. Kusner, Zhixiang Xu, Kilian Q. Weinberger, and John P. Cunningham. Bayesian optimization with inequality constraints. In *Proceedings of the 31st International Conference on International Conference on Ma-*

chine Learning - Volume 32, ICML'14, page II–937–II–945. JMLR.org, 2014.

[HANM06] D. Huang, Theodore Allen, William Notz, and R.A. Miller. Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32:369–382, 11 2006.

[HBdF11] Matthew Hoffman, Eric Brochu, and Nando de Freitas. Portfolio allocation for bayesian optimization. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, page 327–336, Arlington, Virginia, USA, 2011. AUAI Press.

[HJS92] Pierre Hansen, Brigitte Jaumard, and Gilles Savard. New branch-and-bound rules for linear bilevel programming. *SIAM J. Sci. Stat. Comput.*, 13(5):1194–1217, sep 1992.

[HKV19] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors. *Automated Machine Learning - Methods, Systems, Challenges*. Springer, 2019.

[HLDS00] Jose Herskovits, Anatoli Leontiev, Guanacy Dias, and Gabel Santos. Contact shape optimization: A bilevel programming approach. *Structural and Multidisciplinary Optimization*, 20:214–221, 11 2000.

[HLX+22] Ya-nan Han, Jianwei Liu, Bing-biao Xiao, Xin-Tan Wang, and Xionglin Luo. Bilevel online deep learning in non-stationary environment. *CoRR*, abs/2201.10236, 2022.

[HM06] W. Halter and S. Mostaghim. Bilevel optimization of multi-component chemical systems using particle swarm optimization. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1240–1247, 2006.

[HMJS02] S.R. Hejazi, A. Memariani, G. Jahanshahloo, and M.M. Sepehri. Linear bilevel programming solution by genetic algorithm. *Computers & Operations Research*, 29(13):1913–1925, 2002.

[IA92] Yo Ishizuka and Eitaro Aiyoshi. Double penalty method for bilevel optimization problems. *Annals of Operations Research*, 34(1):73–88, December 1992.

[ISN+18] MohammadNoor Injadat, Fadi Salo, Ali Bou Nassif, Aleksander Essex, and Abdallah Shami. Bayesian optimization with machine learning algorithms towards anomaly detection. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2018.

[ISR16] Md Monjurul Islam, Hemant Kumar Singh, and Tapabrata Ray. A nested differential evolution based algorithm for solving multi-objective bilevel optimization problems. In *ACALCI*, 2016.

[ISRS17] Md Monjurul Islam, Hemant Kumar Singh, Tapabrata Ray, and Ankur Sinha. An enhanced memetic algorithm for single-objective bilevel optimization problems. *Evolutionary Computation*, 25:607–642, 12 2017.

[JLHW13] Yan Jiang, Xuyong Li, Chongchao Huang, and Xianing Wu. Application of particle swarm optimization based on chks smoothing function for solving nonlinear bilevel programming problem. *Applied Mathematics and Computation*, 219(9):4332–4339, 2013.

[JM15] M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[Jon01] Donald Jones. A taxonomy of global optimization methods based on response surfaces. *J. of Global Optimization*, 21:345–383, 12 2001.

[JSW98] Donald Jones, Matthias Schonlau, and William Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 12 1998.

[JYL21] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of

Proceedings of Machine Learning Research, pages 4882–4892. PMLR, 18–24 Jul 2021.

[KDBN17] Emmanuel Kieffer, Grégoire Danoy, Pascal Bouvry, and Anass Nagih. Bayesian optimization approach of general bi-level problems. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '17, page 1614–1621, New York, NY, USA, 2017. Association for Computing Machinery.

[KDPV+15] Dr. Vyacheslav Kalashnikov, Stephan Dempe, Gerardo Perez-Valdes, Nataliya Kalashnykova, and José-Fernando Camacho-Vallejo. Bilevel programming and applications. *Mathematical Problems in Engineering*, 2015:1–16, 03 2015.

[KL90] C. D. Kolstad and L. S. Lasdon. Derivative evaluation and computational experience with large bilevel mathematical programs. *Journal of Optimization Theory and Applications*, 65(3):485–499, jun 1990.

[KLLS21] Thomas Kleinert, Martine Labbé, Ivana Ljubić, and Martin Schmidt. A survey on mixed-integer programming techniques in bilevel optimization. *EURO Journal on Computational Optimization*, 9:100007, 2021.

[Kno06a] J. Knowles. Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

[Kno06b] J. Knowles. Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

[KPH21] Yeonju Kim, Zherong Pan, and Kris Hauser. Mo-bbo: Multi-objective bilevel bayesian optimization for robot and behavior co-design. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9877–9883, 2021.

[Kra88] D. Kraft. *A software package for sequential quadratic programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988.

[Kri51] D.G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6):119–139, 1951.

[Kus63] Harold J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86:97–106, 1963.

[LGZ+21] Risheng Liu, Jiaxin Gao, Jin Zhang, Deyu Meng, and Zhouchen Lin. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *CoRR*, abs/2101.11517, 2021.

[LHWW07] Yibing Lv, Tiesong Hu, Guangmin Wang, and Zhongping Wan. A penalty function method based on kuhn–tucker condition for solving linear bilevel programming. *Applied Mathematics and Computation*, 188(1):808–813, 2007.

[LJD+16] Lisha Li, Kevin G. Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Efficient hyperparameter optimization and infinitely many armed bandits. *CoRR*, abs/1603.06560, 2016.

[LN89] Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming 1989 45:1*, 45:503–528, 8 1989.

[LNK+17] Pablo Ribalta Lorenzo, Jakub Nalepa, Michal Kawulok, Luciano Sanchez Ramos, and José Ranilla Pastor. Particle swarm optimization for hyper-parameter selection in deep neural networks. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, page 481–488, New York, NY, USA, 2017. Association for Computing Machinery.

[Low06] James K. Lowe. Homeland security: Operations research initiatives and applications. *Interfaces*, 36(6):483–485, 2006.

[LTM06] Xiangyong Li, Peng Tian, and Xia Min. A hierarchical particle swarm optimization for solving bilevel programming problems. In *International Conference on Artificial Intelligence and Soft Computing*, 2006.

[LW07] Hecheng Li and Yuping Wang. A hybrid genetic algorithm for solving nonlinear bilevel programming problems based on the simplex method. In *Third International Conference on Natural Computation (ICNC 2007)*, volume 4, pages 91–95, 2007.

[LYY+18a] Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. Batch Bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3306–3314. PMLR, 10–15 Jul 2018.

[LYY+18b] Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. Batch bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design. In *International Conference on Machine Learning*, 2018.

[LZ21a] Hong Li and Li Zhang. A bilevel learning model and algorithm for self-organizing feed-forward neural networks for pattern classification. *IEEE Transactions on Neural Networks and Learning Systems*, 32:4901–4915, 2021.

[LZ21b] Hong Li and Li Zhang. An efficient solution strategy for bilevel multiobjective optimization problems using multiobjective evolutionary algorithm. *Soft Computing*, 25:8241–8261, 2021.

[Mar86] P Marcotte. Network design problem with congestion effects: A case of bilevel programming. *Math. Program.*, 34(2):142–162, mar 1986.

[MBD12] Carlo Mannino, Fred Bernt, and Geir Dahl. Computing optimal recovery policies for financial markets. *Operations Research*, 60, 12 2012.

[MCVPL20] Héctor Maravillo, José-Fernando Camacho-Vallejo, Justo Puerto, and Martine Labbé. A market regulation bilevel problem: A case study of the mexican petrochemical industry. *Omega*, 97:102105, 2020.

[MdDMMQC21] Jesús-Adolfo Mejía-de Dios, Efrén Mezura-Montes, and Marcela Quiroz-Castellanos. Automated parameter tuning as a bilevel optimization problem solved by a surrogate-assisted population-based approach. *Applied Intelligence*, 51(8):5978–6000, aug 2021.

[MdDRMMM23] Jesús-Adolfo Mejía-de Dios, Alejandro Rodríguez-Molina, and Efrén Mezura-Montes. Multiobjective bilevel optimization: A survey of the state-of-the-art. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–0, 2023.

[Mig95] Athanasios Migdalas. Bilevel programming in traffic planning: Models, methods and challenge. *Journal of Global Optimization*, 7:381–405, 01 1995.

[MLA+20] Saypraseuth Mounsaveng, Issam H. Laradji, Ismail Ben Ayed, David Vázquez, and Marco Pedersoli. Learning data augmentation with online bilevel optimization for image classification. *CoRR*, abs/2006.14699, 2020.

[MPA94] Richard Mathieu, L. Pittard, and G. Anandalingam. Genetic algorithm based approach to bi-level linear programming. *RAIRO. Recherche Opérationnelle*, 28, 01 1994.

[MPM22] M.A. Muñoz, S. Pineda, and J.M. Morales. A bilevel framework for decision-making under uncertainty with contextual information. *Omega*, 108:102575, 2022.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019.

[PKG20] Michael Pearce, Janis Klaise, and Matthew J. Groves. Practical bayesian optimization of objectives with conditioning variables. *arXiv: Machine Learning*, 2020.

[PVG+11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12(null):2825–2830, nov 2011.

[QWP21] Jingwen Qi, Shuaian Wang, and Harilaos Psaraftis. Bi-level optimization model applications in managing air emissions from ships: A review. *Communications in Transportation Research*, 1:100020, 2021.

[RW06] Carl Edward. Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[RW09] Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian processes for machine learning. In *Adaptive computation and machine learning*, 2009.

[SC08] Maria P. Scaparra and Richard L. Church. A bilevel mixed-integer program for critical infrastructure protection planning. *Computers & Operations Research*, 35(6):1905–1923, 2008. Part Special Issue: OR Applications in the Military and in Counter-Terrorism.

[SG15a] Amar Shah and Zoubin Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *NIPS*, 2015.

[SG15b] Amar Shah and Zoubin Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[SKKS09] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process bandits without regret: An experimental design approach. *CoRR*, abs/0912.3995, 2009.

[SKKS10] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 1015–1022, Madison, WI, USA, 2010. Omnipress.

[SLA12] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[SLDM17] Ankur Sinha, Zhichao Lu, Kalyanmoy Deb, and Pekka Malo. Bilevel optimization based on iterative approximation of multiple mappings, 2017.

[SMD12a] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. Unconstrained scalable test problems for single-objective bilevel optimization. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, 2012.

[SMD12b] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. Unconstrained scalable test problems for single-objective bilevel optimization. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, 2012.

[SMD13a] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. Efficient evolutionary algorithm for single-objective bilevel optimization, 2013.

[SMD13b] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. Efficient evolutionary algorithm for single-objective bilevel optimization, 2013.

[SMD14a] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. An improved bilevel evolutionary algorithm based on quadratic approximations. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1870–1877, 2014.

[SMD14b] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. An improved bilevel evolutionary algorithm based on quadratic approximations. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1870–1877, 2014.

[SMD15a] Ankur Sinha, Pekka Malo, and Kalyan Deb. Towards understanding bilevel multi-objective optimization with deterministic lower level decisions. In *Evolutionary Multi-Criterion Optimization*, volume 9018, pages 426–443, 03 2015.

[SMD+15b] Ankur Sinha, Pekka Malo, Kalyan Deb, Pekka Korhonen, and Jyrki Wallenius. Solving bilevel multicriterion optimization problems with lower level decision uncertainty. *IEEE Transactions on Evolutionary Computation*, 20:1–1, 01 2015.

[SMD16a] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. Solving optimistic bilevel programs by iteratively approximating lower level optimal value function. *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 1877–1884, 2016.

[SMD+16b] Ankur Sinha, Pekka Malo, Kalyanmoy Deb, Pekka Korhonen, and Jyrki Wallenius. Solving bilevel multicriterion optimization problems with lower level decision uncertainty. *IEEE Transactions on Evolutionary Computation*, 20(2):199–217, 2016.

[SMD+16c] Ankur Sinha, Pekka Malo, Kalyanmoy Deb, Pekka Korhonen, and Jyrki Wallenius. Solving bilevel multicriterion optimization problems with lower level decision uncertainty. *IEEE Transactions on Evolutionary Computation*, 20:199–217, 4 2016.

[SMD17a] Ankur Sinha, Pekka Malo, and Kalyan Deb. *Evolutionary Bilevel Optimization: An Introduction and Recent Advances*, volume 20, pages 71–103. Springer, 01 2017.

[SMD17b] Ankur Sinha, Pekka Malo, and Kalyan Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, PP, 05 2017.

[SMD17c] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. Evolutionary algorithm for bilevel optimization using approximations of the lower level optimal solution mapping. *European Journal of Operational Research*, 257(2):395–411, 2017.

[SMD18a] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295, 2018.

[SMD18b] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295, 2018.

[SMdCMdSJ22] João Victor Soares do Amaral, José Arnaldo Barra Montevechi, Rafael de Carvalho Miranda, and Wilson Trigueiro de Sousa Junior. Metamodel-based simulation optimization: A systematic literature review. *Simulation Modelling Practice and Theory*, 114:102403, 2022.

[SMFD13a] Ankur Sinha, Pekka Malo, Anton Frantsev, and Kalyanmoy Deb. Multi-objective stackelberg game between a regulating authority and a mining company: A case study in environmental economics. In *2013 IEEE Congress on Evolutionary Computation*, pages 478–485, 2013.

[SMFD13b] Ankur Sinha, Pekka Malo, Anton Frantsev, and Kalyanmoy Deb. Multi-objective stackelberg game between a regulating authority and a mining company: A case study in environmental economics. In *2013 IEEE Congress on Evolutionary Computation*, pages 478–485, 2013.

[SMFD14] Ankur Sinha, Pekka Malo, Anton Frantsev, and Kalyanmoy Deb. Finding optimal strategies in a multi-period multi-leader–follower stackelberg game using an evolutionary algorithm. *Computers Operations Research*, 41:374–385, 2014.

[SRBG19] Diariétou Sambakhé, Lauriane Rouan, Jean-Noël Bacro, and Eric Goze. Conditional optimization of a noisy function using a kriging metamodel. *Journal of Global Optimization*, 73:1–22, 03 2019.

[SSW+16a] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104:148–175, 2016.

[SSW+16b] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104:148–175, 2016.

[Sta52] Heinrich von. Stackelberg. *The theory of the market economy*. William Hodge, London, 1952.

[SWI85] Warren D. Seider and Charles W. White III. Chemical reaction equilibrium analysis: Theory and algorithms by william r. smith and ronald w. missen, 364 pp., john wiley, 1983, $42.95. *AIChE Journal*, 31(1):176–176, 1985.

[SX01] Xinping Shi and Hong Sheng Xia. Model and interactive algorithm of bi-level multi-objective decision-making with multiple interconnected decision makers. *Journal of Multi-Criteria Decision Analysis*, 10(1):27–34, 2001.

[Tal13] El-Ghazali Talbi. *Metaheuristics for Bi-level Optimization*, volume 482. Springer, 05 2013.

[TF05] Tea Tusar and Bogdan Filipic. Demo: Differential evolution for multiobjective optimization. *Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization, Guanajuato, Mexico*, pages 520–533, 03 2005.

[TMV93] Hoang Tuy, Athanasios Migdalas, and Peter Värbrand. A global optimization approach for the linear two-level program. *Journal of Global Optimization*, 3:1–23, 01 1993.

[VSJ94a] L. Vicente, G. Savard, and J. Júdice. Descent approaches for quadratic bilevel programming. *J. Optim. Theory Appl.*, 81(2):379–399, may 1994.

[VSJ94b] Luís Nunes Vicente, Gilles Savard, and Joaquim Júdice. Descent approaches for quadratic bilevel programming. *Journal of Optimization Theory and Applications*, 81:379–399, 1994.

[VSJ96] L. Vicente, G. Savard, and J. Júdice. Discrete linear bilevel programming problem. *J. Optim. Theory Appl.*, 89(3):597–614, jun 1996.

[WCZ+19] Jia Wu, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. Hyperparameter optimization for machine learning models based on bayesian optimizationb. *Journal of Electronic Science and Technology*, 17(1):26–40, 2019.

[WF16] Jian Wu and Peter Frazier. The parallel knowledge gradient method for batch bayesian optimization. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[WH91] Ue-Pyng Wen and Shuh-Tzy Hsu. Linear bi-level programming problems – a review. *The Journal of the Operational Research Society*, 42(2):125–133, 1991.

[WH19] Takashi Wada and Hideitsu Hino. Bayesian optimization for multi-objective optimization and multi-point search, 2019.

[WJL05] Yuping Wang, Yong-Chang Jiao, and Hong Li. An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(2):221–232, 2005.

[WLD11] Yuping Wang, Hong Li, and Chuangyin Dang. A new evolutionary algorithm for a class of nonlinear bilevel programming problems and its global convergence. *INFORMS Journal on Computing*, 23:618–629, 10 2011.

[Wol20] Brian Wolfe. Kuusamo project mineral resource estimate k1 gold and cobalt deposit latitude 66 cobalt ltd. Technical report, Latitude 66 Cobalt Ltd, 2020.

[WPTA20] Sonja Wogrin, Salvador Pineda, and Diego Tejada-Arango. *Applications of Bilevel Optimization in Energy and Electricity Markets*, pages 139–168. Springer, 11 2020.

[WS07] Gary Wang and S. Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design - J MECH DESIGN*, 129, 04 2007.

[WTKR13] Wolfram Wiesemann, Angelos Tsoukalas, Polyxeni-Margarita Kleniati, and Berç Rustem. Pessimistic bilevel optimization. *SIAM Journal on Optimization*, 23(1):353–380, 2013.

[WWS13] Zhongping Wan, Guangmin Wang, and Bin Sun. A hybrid intelligent algorithm by combining particle swarm optimization with chaos searching technique for solving nonlinear bilevel programming problems. *Swarm and Evolutionary Computation*, 8:26–32, 2013.

[XTZ12] Jiuping Xu, Yan Tu, and Ziqiang Zeng. A nonlinear multiobjective bilevel model for minimum cost network flow problem in a large-scale construction project. *Mathematical Problems in Engineering*, 2012, 04 2012.

[Ye11] Jane Ye. Necessary optimality conditions for multiobjective bilevel programs. *Math. Oper. Res.*, 36:165–184, 02 2011.

[YEDB19] Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. Multi-objective bayesian global optimization using expected hypervolume improvement gradient. *Swarm and Evolutionary Computation*, 44:945–956, 2019.

[Yin00] Yafeng Yin. Genetic-algorithms-based approach for bilevel programming models. *Journal of Transportation Engineering-asce - J TRANSP ENG-ASCE*, 126, 03 2000.

[YS20] Li Yang and Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316, 11 2020.

[YZ10] Jane J. Ye and Daoli Zhu. New necessary optimality conditions for bilevel programs by combining the mpec and value function approaches. *SIAM Journal on Optimization*, 20(4):1885–1905, 2010.

[ZHG+13] Tao Zhang, Tiesong Hu, Xuning Guo, Zhong Chen, and Yue Zheng. Solving high dimensional bilevel multiobjective programming problem using a hybrid particle swarm optimization algorithm with crossover operator. *Knowledge-Based Systems*, 53:13–19, 11 2013.

[ZL07] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[ZLTV10] Qingfu Zhang, Wudong Liu, Edward Tsang, and Botond Virginas. Expensive multiobjective optimization by moea/d with gaussian process model. *IEEE Trans. Evolutionary Computation*, 14:456–474, 06 2010.

[ZQYS23] Kaifang Zhao, Kai Qiu, Jian Yan, and Mir Pasha Shaker. Technical and economic operation of vpps based on competitive bi–level negotiations. *Energy*, 282:128698, 2023.

[ZTM18] Mohammad Zhalechian, S.A. Torabi, and Mahyar Mohammadi. Hub-and-spoke network design under operational and disruption risks. *Transportation Research Part E: Logistics and Transportation Review*, 109:20–43, 01 2018.

[ZYW06] Xiaobo Zhu, Qian Yu, and Xianjia Wang. A hybrid differential evolution algorithm for solving nonlinear bilevel programming with linear constraints. In *2006 5th IEEE International Conference on Cognitive Informatics*, volume 1, pages 126–131, 2006.

[ZZZ12] Aimin Zhou, Qingfu Zhang, and Guixu Zhang. A multiobjective evolutionary algorithm based on decomposition and probability model. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, 2012.