PH.D. SCHOOL

UNIVERSITY OF MILANO-BICOCCA

Department of Economics, Management and Statistics

PhD program **Economics, Statistics and Data Science** Cycle **XXXVII**
Curriculum in **Data Science**

MULTI-TASK LEARNING IN BLACK-BOX OPTIMIZATION

Surname: **Ponti**
Name: **Andrea**
Registration number: **816311**

Supervisor: **Prof. Antonio Candelieri**
Tutor: **Prof. Fabio Mercorio**
Coordinator: **Prof. Matteo Manera**

Academic Year 2024/2025

Abstract

The increasing complexity of modern scientific and machine learning applications has brought renewed attention to *black-box optimization*, where objective functions are expensive to evaluate and lack analytical form. In many real-world scenarios, optimization must simultaneously handle multiple competing objectives, heterogeneous information sources with different fidelities, and even combinatorial design spaces. This thesis addresses these challenges by developing novel methods and frameworks that advance the state of the art in *multi-objective* and *multi-fidelity* black-box optimization.

The first contribution of this work concerns the design of Wasserstein-enabled Multi-Objective Evolutionary Algorithms. By incorporating the Wasserstein distance into key components of NSGA-II and MOEA/D, the proposed methods improve the balance between convergence and diversity in Pareto front approximations. A novel binary crossover operator is also introduced. All algorithms were implemented in the `pymoo` framework and empirically validated on standard benchmarks and real-world problems, showing improved performance over classical counterparts.

The second main contribution extends the Augmented Gaussian Process (AGP) framework to new settings within Bayesian optimization. A combinatorial variant, based on genetic algorithms for acquisition function optimization, enables AGP to efficiently handle discrete and constrained spaces. Furthermore, AGP is extended to the multi-objective and multi-information source case, resulting in the MISO-AGP model, capable of jointly learning from multiple fidelities and objectives. The AGP has been contributed to the open-source `BoTorch` library, extending its usability to a broader class of problems.

A third contribution of this thesis lies in the formulation of several real-world problems under the proposed optimization paradigms. These include multi-objective formulations of Optimal Sensor Placement (OSP) and Recommender Systems, multi-fidelity extensions of Risk-Averse OSP and Binary Quadratic Programming, and a multi-objective multi-fidelity formulation of Hyperparameter Optimization in machine learning. Such reformulations demonstrate the flexibility and applicability of the proposed methods across diverse domains.

Comprehensive experimental analyses confirm the effectiveness of the proposed approaches in terms of convergence, cost-efficiency, and ecological impact. In particular,

Wasserstein-based MOEAs achieve improved coverage of the Pareto front, while the MISO-AGP algorithm efficiently leverages multiple information sources to reduce computational cost. All code and experimental material have been made publicly available, ensuring transparency and reproducibility.

Overall, this thesis contributes both methodological and practical advances to the fields of evolutionary and Bayesian optimization. By integrating Wasserstein geometry, Gaussian process modeling, and multi-fidelity reasoning, it provides a unified and extensible framework for efficient and sustainable optimization of complex black-box systems.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Motivations

In many scientific, engineering, and applied machine learning domains, decision-makers must optimize functions whose internal structure is unknown or extremely complex. Such functions are often called **black-box** functions: one can evaluate them at chosen inputs, but analytic gradients or closed-form expressions are unavailable, evaluations are expensive, noisy, or both. Examples include hyperparameter tuning in deep neural networks, simulation-based design in engineering, environmental models, and control tasks. Because each evaluation may incur a large computational cost (e.g., hours of training, long simulators, real experiments), it is crucial to make every query count.

Compounding this difficulty, real-world problems rarely involve only a single criterion. Often there are multiple conflicting objectives, such as accuracy vs. latency, performance vs. resource usage, fairness vs. utility, each of which matters. Multi-objective optimization provides a principled framework for finding trade-offs, represented as Pareto fronts, rather than collapsing all objectives into one via scalarization, which can hide important trade-offs or lead to suboptimal choices. There is an extensive literature showing the benefits and challenges of multi-objective Bayesian optimization, especially where black-box evaluations are expensive or high-dimensional. For example, researchers in [49] propose methods for multi-objective BO in high-dimensional spaces, showing both the promise and the limitations of current approaches in real applications.

Yet another dimension of structure is often available: multiple fidelity levels or information sources. In many contexts, one has access to cheaper approximate evaluations (lower fidelity) that may be faster, coarser, or less accurate, along with high-fidelity, expensive ones. Multi-fidelity optimization methods exploit these cheaper sources to speed up optimization, by trading off cost vs. accuracy. For instance, low fidelity may mean fewer training epochs,

coarser simulation grids, or downsampling data. Using low fidelity sources can dramatically reduce total cost when done appropriately. Research such as [1] highlights how switching adaptively among fidelity levels improves efficiency.

The combination of these axes, black-box functions, multiple objectives, and multiple fidelities, naturally leads to what may be called multi-objective multi-fidelity optimization (MOMFO). This setting is very relevant in modern machine learning, AutoML, fairness-aware learning, resource-aware design, and sustainability (where each evaluation may carry environmental cost). However, existing methods often struggle in this setting: either they handle only single objectives, or they rely on expensive fidelity models, or they poorly scale to combinatorial or discrete domains, or they do not properly balance cost vs. information gain when fidelities are not well correlated with true objectives.

This thesis is motivated by the need to fill these gaps. The goal is to develop algorithms that are (i) **sample-efficient**, making sure each expensive evaluation is used wisely, (ii) **multi-objective**, capable of constructing rich Pareto fronts in trade-off scenarios, (iii) **multi-fidelity / multi-information source** aware, so as to exploit cheaper approximations without compromising reliability, and (iv) **practical and scalable**, including for combinatorial spaces, discrete decision variables, and real applications like sensor placement, recommender systems, and hyperparameter optimization.

In the following chapters, these motivations guide the methodological contributions.

1.2 Scientific Contributions

The research presented in this thesis advances the field of optimization under multiple objectives and information sources, with contributions that span both theoretical development and practical implementation. The main contributions can be summarized as follows.

Wasserstein-enabled evolutionary algorithms and operators. The thesis introduces new mechanisms to enhance multi-objective evolutionary algorithms through the use of the Wasserstein distance. Specifically, NSGA-II has been extended with a Wasserstein-based selection strategy, while MOEA/D has been adapted to exploit the Wasserstein distance for ordering decomposition weights. In addition, a novel binary combinatorial crossover operator has been proposed, designed to improve search efficiency in discrete and constrained domains.

Augmented Gaussian Processes for complex optimization tasks. Building on the Augmented Gaussian Process (AGP) framework, this work extends the methodology in two

directions. First, a combinatorial variant is developed, employing genetic algorithms to optimize discrete and constrained acquisition functions. Second, a multi-objective formulation of AGP is proposed, enabling the model to directly address problems involving multiple, potentially conflicting objectives.

Novel formulations of real-world optimization problems. The thesis contributes to the literature by reformulating several relevant problems within advanced optimization frameworks. A multi-objective perspective is adopted for both optimal sensor placement and recommender systems. Multi-fidelity formulations are proposed for risk-averse sensor placement and binary quadratic programming. Moreover, a multi-objective multi-fidelity formulation is introduced for the hyperparameter optimization of machine learning algorithms, bridging theoretical innovation with pressing applications.

Code contributions and open-source dissemination. All methodological developments have been supported by software contributions to widely used optimization libraries. In particular, NSGA-II/W, MOEA/D/W, and the proposed binary crossover operator have been implemented in the `pymoo` framework. Furthermore, the MISO-AGP algorithm has been contributed to `BoTorch`, ensuring accessibility to the research community and enabling future developments.

Together, these contributions provide new methods, problem formulations, and tools that advance the state-of-the-art in multi-objective, multi-fidelity, and multi-information source optimization, while also ensuring that the results are reproducible and usable by the broader scientific community.

1.3 Publications

The present dissertation is primarily based on the following three journal papers, which constitute its main scientific contributions:

- [116] Ponti, A., Candelieri, A., Giordani, I., and Archetti, F. (2023a). Intrusion detection in networks by wasserstein enabled many-objective evolutionary algorithms. *Mathematics*, 11(10):2342

- [129] Sabbatella, A., Ponti, A., Candelieri, A., and Archetti, F. (2024b). Bayesian optimization using simulation-based multiple information sources over combinatorial structures. *Machine Learning and Knowledge Extraction*, 6(4):2232–2247

- [29] Candelieri, A., Ponti, A., and Archetti, F. (2024b). Fair and green hyperparameter optimization via multi-objective and multiple information source bayesian optimization. *Machine Learning*, 113(5):2701–2731

In addition, several other works closely related to the topics addressed in this thesis have been published and are cited throughout the subsequent chapters:

- [113] Ponti, A., Candelieri, A., and Archetti, F. (2021a). A new evolutionary approach to optimal sensor placement in water distribution networks. *Water*, 13(12):1625
- [115] Ponti, A., Candelieri, A., and Archetti, F. (2021c). A wasserstein distance based multiobjective evolutionary algorithm for the risk aware optimization of sensor placement. *Intelligent Systems with Applications*, 10:200047
- [34] Candelieri, A., Ponti, A., Giordani, I., and Archetti, F. (2022d). Lost in optimization of water distribution systems: better call bayes. *Water*, 14(5):800
- [36] Candelieri, A., Ponti, A., Giordani, I., Bosio, A., and Archetti, F. (2023g). Distributional learning in multi-objective optimization of recommender systems. *Journal of Ambient Intelligence and Humanized Computing*, 14(8):10849–10865

Furthermore, preliminary results and ongoing research were presented at international conferences, contributing to the dissemination and discussion of the work within the scientific community:

- [114] Ponti, A., Candelieri, A., and Archetti, F. (2021b). Optimal sensor placement by distribution based multiobjective evolutionary optimization. In *International Conference on Learning and Intelligent Optimization*, pages 315–332. Springer
- [20] Candelieri, A., Ponti, A., and Archetti, F. (2021b). Risk aware optimization of water sensor placement. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 295–296
- [112] Ponti, A. and Archetti, F. (2023). The unreasonable effectiveness of optimal transport distance in the design of multi-objective evolutionary optimization algorithms. In *International Conference on Numerical Computations: Theory and Algorithms*, pages 151–164. Springer
- [25] Candelieri, A., Ponti, A., and Archetti, F. (2023b). Multi-objective and multiple information source optimization for fair & green machine learning. In *International Conference on Numerical Computations: Theory and Algorithms*, pages 49–63. Springer

Part of this research has also been published as a Springer Briefs[1]:

- [32] Candelieri, A., Ponti, A., and Archetti, F. (2025c). Multiple information source bayesian optimization

Beyond the contributions directly connected to the thesis, the doctoral research activity led to a number of additional publications in journals and conference proceedings, covering a wider range of topics in artificial intelligence, machine learning, and optimization:

- [119] Ponti, A., Giordani, I., Mistri, M., Candelieri, A., and Archetti, F. (2022a). The "unreasonable" effectiveness of the wasserstein distance in analyzing key performance indicators of a network of stores. *Big Data and Cognitive Computing*, 6(4):138

- [22] Candelieri, A., Ponti, A., and Archetti, F. (2022b). Explaining exploration–exploitation in humans. *Big Data and Cognitive Computing*, 6(4):155

- [26] Candelieri, A., Ponti, A., and Archetti, F. (2023c). Uncertainty quantification and exploration–exploitation trade-off in humans. *Journal of Ambient Intelligence and Humanized Computing*, 14(6):6843–6876

- [35] Candelieri, A., Ponti, A., Giordani, I., and Archetti, F. (2023f). On the use of wasserstein distance in the distributional analysis of human decision making under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 91(2):217–238

- [27] Candelieri, A., Ponti, A., and Archetti, F. (2023d). Wasserstein enabled bayesian optimization of composite functions. *Journal of Ambient Intelligence and Humanized Computing*, 14(8):11263–11271

- [33] Candelieri, A., Ponti, A., Fersini, E., Messina, E., and Archetti, F. (2023e). Safe optimal control of dynamic systems: Learning from experts and safely exploring new policies. *Mathematics*, 11(20):4347

- [118] Ponti, A., Giordani, I., Candelieri, A., and Archetti, F. (2024). Wasserstein-enabled leaks localization in water distribution networks. *Water*, 16(3):412

- [131] Sabbatella, A., Ponti, A., Giordani, I., Candelieri, A., and Archetti, F. (2024d). Prompt optimization in large language models. *Mathematics*, 12(6):929

[1] Springer Briefs are concise summaries of cutting-edge research and practical applications across a wide spectrum of fields. Featuring compact volumes of 55 to 125 pages, each series covers a range of professional and academic topics.

- [128] Sabbatella, A., Archetti, F., Ponti, A., Giordani, I., and Candelieri, A. (2024a). Bayesian optimization for instruction generation. *Applied Sciences*, 14(24):11865

- [31] Candelieri, A., Ponti, A., and Archetti, F. (2025b). Gaussian process regression over discrete probability measures: on the non-stationarity relation between euclidean and wasserstein squared exponential kernels. *Journal of Global Optimization*, pages 1–26

- [30] Candelieri, A., Ponti, A., and Archetti, F. (2025a). Bayesian optimization over the probability simplex. *Annals of Mathematics and Artificial Intelligence*, 93(1):77–91

- [21] Candelieri, A., Ponti, A., and Archetti, F. (2022a). Bayesian optimization in wasserstein spaces. In *International Conference on Learning and Intelligent Optimization*, pages 248–262. Springer

- [120] Ponti, A., Irpino, A., Candelieri, A., Bosio, A., Giordani, I., and Archetti, F. (2022b). Network vulnerability analysis in wasserstein spaces. In *International Conference on Learning and Intelligent Optimization*, pages 263–277. Springer

- [23] Candelieri, A., Ponti, A., and Archetti, F. (2022c). Safe-exploration of control policies from safe-experience via gaussian processes. In *International Conference on Learning and Intelligent Optimization*, pages 232–247. Springer

- [117] Ponti, A., Giordani, I., Candelieri, A., and Archetti, F. (2023b). A leak localization algorithm in water distribution networks using probabilistic leak representation and optimal transport distance. In *International Conference on Learning and Intelligent Optimization*, pages 31–45. Springer

- [24] Candelieri, A., Ponti, A., and Archetti, F. (2023a). Generative models via optimal transport and gaussian processes. In *International Conference on Learning and Intelligent Optimization*, pages 135–149. Springer

- [17] Candelieri, A. (2023). Resource allocation via bayesian optimization: an efficient alternative to semi-bandit feedback. In *International Conference on Numerical Computations: Theory and Algorithms*, pages 34–48. Springer

- [130] Sabbatella, A., Ponti, A., Giordani, I., and Archetti, F. (2024c). A bayesian approach for prompt optimization in llms. In *International Conference on Learning and Intelligent Optimization*, pages 348–360. Springer

- [28] Candelieri, A., Ponti, A., and Archetti, F. (2024a). A constrained-jko scheme for effective and efficient wasserstein gradient flows. In *International Conference on Learning and Intelligent Optimization*, pages 66–80. Springer

- [3] Archetti, F., Ponti, A., Candelieri, A., and Sabbatella, A. (2025). Bayesian optimization, machine learning, and probabilistic numerics. In *AIP Conference Proceedings*, volume 3315, page 400049. AIP Publishing LLC

1.4 Organization of the Thesis

The thesis is organized into five chapters, in addition to the introduction. Chapter 2 provides the necessary theoretical background to contextualize the research contributions. In particular, it introduces the main concepts of multi-task learning and illustrates how this framework naturally connects to multi-objective optimization and multi-fidelity optimization. This chapter establishes the conceptual foundations upon which the work is built.

The core contributions of the thesis are presented in Chapters 3, 4, and 5. Each of these chapters originates from a peer-reviewed publication and follows a common structure. They begin with a background section, reviewing the relevant literature and theoretical aspects; this is followed by the presentation of the proposed methodological contribution; and finally, a section devoted to experimental validation is provided, including benchmark studies and real-world applications.

- Chapter 3 focuses on multi-objective optimization, with particular attention to the integration of Wasserstein-based metrics within evolutionary algorithms.

- Chapter 4 addresses multi-information source optimization, introducing a Gaussian process model augmented to handle combinatorial structures and discussing its implementation and empirical assessment.

- Chapter 5 extends the investigation to the multi-objective multi-fidelity setting, exploring fairness- and sustainability-oriented hyperparameter optimization algorithms.

Finally, the thesis concludes with a dedicated chapter summarizing the main findings, highlighting the scientific contributions, and outlining possible directions for future research. This closing discussion emphasizes both the theoretical significance of the proposed methods and their potential practical impact.

Chapter 2

Multi-Task Learning

This chapter provides a concise overview of Multi-Task Learning (MTL) and its connections to optimization. It begins by introducing the fundamental concepts of MTL, highlighting how learning multiple related tasks simultaneously can improve predictive performance and generalization. The chapter then explores the relationship between MTL and multi-objective optimization, illustrating how the simultaneous optimization of multiple tasks can be framed as a multi-objective problem. Finally, it discusses the extension to multi-fidelity optimization, showing how task evaluations at varying levels of cost or accuracy can be integrated into the learning process. These perspectives set the stage for the optimization-focused methods presented in subsequent chapters.

2.1 Multi-Task Learning

The paradigm of Multi-Task Learning (MTL) can be traced back to the seminal work of Caruana [37], in which it was argued that data from multiple tasks can be jointly exploited to improve performance over learning each task independently. The fundamental intuition is that apparently different tasks may share hidden dependencies due to a common data-generating process. By designing a family of parameterized hypotheses that share part of their parameters across tasks, MTL enables the transfer of knowledge among tasks while minimizing a weighted sum of the empirical risks. This mirrors human learning, where skills acquired in one domain (e.g., tennis) can facilitate learning in another (e.g., padel).

The primary motivation behind MTL is to alleviate the challenge of limited labeled data, a bottleneck in many machine learning scenarios. By aggregating knowledge across related tasks, MTL reduces the reliance on extensive manual labeling and improves model robustness against overfitting. The rise of deep learning has further highlighted the advantages of MTL, as shared representations have led to significant improvements in large-scale applications.

MTL is closely related to other learning paradigms: transfer learning focuses on improving a single target task using auxiliary tasks, whereas MTL treats all tasks equally; continual learning addresses tasks that arrive sequentially; *multi-label learning* and multi-output regression are special cases of MTL where all tasks share the same dataset; and multi-view learning can be interpreted as a single-task scenario with multiple feature sets.

Zhang and Yang [169, 170] provide a comprehensive categorization of algorithmic strategies for MTL, which can be broadly divided into five families:

1. **Feature learning approaches**, which identify common representations through feature transformation or selection.

2. **Low-rank approaches**, which enforce low-dimensional structures on the parameter space to capture shared information.

3. **Task clustering approaches**, which group related tasks to promote intra-cluster knowledge sharing.

4. **Task relation learning approaches**, which explicitly model the relationships among tasks during training.

5. **Decomposition approaches**, which decompose model parameters into shared and task-specific components.

MTL can also be extended by combining it with other paradigms such as semi-supervised learning, active learning, reinforcement learning, and graphical models, further broadening its applicability.

Depending on the type of task and data availability, several settings of MTL have been investigated. These include multi-task supervised learning (classification and regression with labeled data), multi-task unsupervised learning (joint clustering or representation learning), multi-task semi-supervised learning (leveraging both labeled and unlabeled data), multi-task active learning (optimizing label acquisition across tasks), and multi-task reinforcement learning (policy learning across related environments). Online, parallel, and distributed MTL have been introduced to deal with sequential data streams and large-scale scenarios where computational efficiency and storage constraints are critical.

MTL has been successfully applied across a wide range of domains. In computer vision, it has improved tasks such as face recognition and object detection; in natural language processing, it has been applied to translation, sentiment analysis, and question answering; in speech processing, it has facilitated joint modeling of recognition and speaker adaptation; in bioinformatics and health informatics, it has enhanced disease prediction and biomarker

discovery; in web applications, it underlies modern recommender systems and search engines. These diverse applications demonstrate the ability of MTL to exploit task relatedness for better generalization in real-world settings.

The theoretical foundations of MTL have focused on generalization bounds, optimization guarantees, and conditions under which knowledge sharing leads to improvements. Open challenges include the principled modeling of task relatedness, the integration of heterogeneous tasks involving multiple modalities, and the design of scalable algorithms suitable for distributed and high-dimensional data. Moreover, the increasing role of deep learning has raised questions about how to most effectively incorporate MTL into complex neural architectures. Finally, ethical and fairness concerns in multi-task scenarios are emerging as relevant research directions, particularly when shared representations may propagate biases across tasks.

In essence, MTL represents a powerful framework for leveraging commonalities among related tasks. By sharing knowledge, it reduces overfitting, improves generalization, and allows efficient use of limited labeled data. Its broad applicability and close connections with other paradigms make it a central concept in modern machine learning, and a natural starting point for extending the idea of joint learning into optimization contexts, such as multi-objective and multi-fidelity optimization.

2.2 From MTL to Multi-Objective Optimization

A typical MTL system is provided with a collection of input points and sets of targets corresponding to multiple tasks. A standard way to impose an inductive bias across tasks is to define a parameterized hypothesis class that shares part of its parameters among them. These shared parameters are then estimated by solving an optimization problem that minimizes a weighted sum of the empirical risk for each task. While straightforward, this linear combination is only meaningful when there exists a common parameter configuration that is simultaneously effective across all tasks. In practice, tasks are often conflicting, which makes such a formulation inadequate. Minimization of a weighted sum of empirical risks implicitly assumes task compatibility and neglects the inherent trade-offs among objectives.

When tasks are in conflict, the problem is more naturally formulated in terms of Multi-Objective Optimization (MOO) [139]. In this setting, the goal is not to minimize a single aggregate loss, but rather to identify solutions that represent desirable compromises among tasks. A solution is considered desirable if it is not dominated by any other, in the sense of Pareto optimality: a point is Pareto optimal if no other solution improves one objective without deteriorating at least one of the others. Casting MTL as MOO therefore shifts the

focus from optimizing a scalarized loss to exploring the Pareto front of trade-offs among tasks. This perspective has been advocated in recent works that highlight the limitations of linear scalarization and emphasize the need for principled approaches to balance competing objectives.

The transition from MTL to MOO opens the door to a wide spectrum of algorithmic strategies. Gradient-based approaches, such as the multiple-gradient descent algorithm (MGDA) [53], exploit task-specific gradients to compute descent directions that decrease all objectives simultaneously and converge to Pareto stationary points. While theoretically well-grounded, such methods face challenges when applied to large-scale models, as they require the explicit computation of gradients for each task, leading to significant computational overhead in high-dimensional settings.

An alternative family of methods relies on Evolutionary Algorithms (EAs). Multi-objective Evolutionary Algorithms (MOEAs) incorporate Pareto dominance directly into their selection mechanisms [50]. These approaches are simple to implement and do not require derivative information, making them flexible tools for complex tasks. However, their main drawback lies in poor sample efficiency, since they typically do not exploit surrogate models to guide the search, making them unsuitable for problems where evaluations are computationally expensive.

A third promising direction is represented by Bayesian optimization (BO), which leverages probabilistic surrogate models, most commonly Gaussian processes, to model objectives and guide the search through an explicit exploration-exploitation trade-off. In the multi-objective setting, surrogate models can be trained either independently for each task or jointly, in order to exploit potential correlations among objectives. Methods such as ParEGO [85] illustrate how uncertainty estimates from Gaussian processes can be used to balance the discovery of new Pareto-optimal regions with the refinement of already promising ones. Furthermore, hybrid approaches that integrate surrogate modeling within MOEAs have recently been proposed to address the efficiency limitations of purely evolutionary methods.

In summary, while traditional MTL relies on scalarized formulations that implicitly assume non-conflicting objectives, casting MTL as a multi-objective optimization problem provides a more general and principled framework. This perspective highlights the central role of Pareto analysis in handling competing tasks and motivates the adoption of algorithmic strategies that can efficiently approximate the Pareto front. The study of such methods, their limitations, and their application to learning scenarios constitutes an active and growing research area.

2.3 From MTL to Multi-Fidelity Optimization

The analogy between MTL and Multi-Fidelity Optimization (MFO) emerges naturally when the notion of related tasks is extended to evaluations of the same underlying function at different levels of fidelity. In MTL, tasks correspond to distinct but related prediction problems, and the goal is to leverage inter-task correlations to improve generalization. In MFO, the "tasks" correspond to function evaluations at varying degrees of accuracy and computational cost. Low-fidelity evaluations are cheaper but biased, whereas high-fidelity evaluations are more accurate yet costly. The central challenge is to exploit correlations across fidelities to guide the optimization efficiently.

A foundational approach to modeling correlations across outputs is the Multi-Task Gaussian Process (MT-GP), introduced by [13]. MT-GPs allow the covariance structure to capture dependencies between tasks, enabling information transfer from related tasks. In the multi-fidelity setting, this framework can be interpreted as a Multi-Fidelity Gaussian Process (MF-GP), where fidelities are treated as correlated outputs. Similarly, the classical co-kriging model of [82] provides an autoregressive formulation for low- and high-fidelity outputs, forming the basis for many MF-GP constructions.

Bayesian Optimization (BO) can exploit these surrogate models to decide where and at which fidelity to evaluate the objective function. Multi-Task GPs have been successfully applied to BO to transfer information between related tasks [145], a perspective that directly motivates the use of Multi-Fidelity GPs. Modern algorithms, such as MF-GP-UCB and BOCA [77, 78], extend classical BO to handle fidelity hierarchies, providing theoretical guarantees while trading off cost and information gain. Information-based approaches such as MF-PES [168] and deep surrogate models [122, 94] further enhance the flexibility of MF-BO by using entropy or neural-network-based surrogates to capture complex correlations across fidelities.

The connection between MTL and MFO clarifies key algorithmic insights. Just as linear-scalarization methods in MTL fail when tasks are conflicting, naive fidelity-agnostic optimization may neglect important fidelity-specific biases. Conversely, joint modeling through MF-GPs enables efficient transfer of information from cheap, low-fidelity evaluations to high-fidelity objectives, dramatically improving sample efficiency. Kernel structures, such as linear models of coregionalization or hierarchical autoregressive kernels, explicitly encode these correlations and provide a flexible modeling framework for heterogeneous fidelities.

In summary, multi-fidelity optimization can be viewed as a specialization of the multi-task learning paradigm, where the "tasks" correspond to different fidelity levels rather than independent learning problems. Multi-task Gaussian processes provide a unifying framework, allowing Bayesian optimization methods to exploit low-cost approximations while

maintaining accuracy at high-fidelity evaluations. This conceptual bridge motivates further research into acquisition function design, surrogate modeling, and algorithmic strategies for efficiently navigating hierarchical or heterogeneous sources of information.

Chapter 3

Multi-Objective Optimization

This chapter provides a comprehensive overview of Multi-Objective Optimization (MOO) and its application within the scope of this thesis. It begins with a theoretical background, covering the formulation of multi-objective optimization problems, common solution strategies, and the use of the Wasserstein distance as a metric for comparing solutions. The chapter then presents the methodological contributions of the thesis, including the integration of the Wasserstein distance into evolutionary algorithms and the development of a combinatorial binary crossover operator. Finally, the chapter illustrates the effectiveness of the proposed methods through experiments on both benchmark functions and real-world applications, such as optimal sensor placement and recommendation systems.

3.1 Background

This section provides the theoretical background for multi-objective optimization. Section 3.1.1 introduces the fundamental concepts of multi-objective optimization, followed in Section 3.1.2 by an overview of the main strategies to address such problems, with particular emphasis on multi-objective evolutionary algorithms. Finally, Section 3.1.3 presents the Wasserstein distance, which plays a central role in the algorithms proposed in Section 3.2.

3.1.1 Multi-Objective Optimization Problems

In multi-objective optimization problems, m objective functions, denoted as $f_1(x), \ldots, f_m(x)$, must be simultaneously optimized over a search space $\Omega \subseteq \mathbb{R}^d$. A general multi-objective optimization problem is formulated as:

$$\min_{x \in \Omega} F(x) = (f_1(x), \ldots, f_m(x)) \tag{3.1}$$

where $F(x)$ is a vector-valued function representing multiple conflicting objectives. In addition, some multi-objective problems may include equality and inequality constraints that define a feasible region within the search space. The presence of multiple objectives introduces a fundamental challenge: in most cases, no single solution simultaneously optimizes all objectives, making a trade-off analysis among competing criteria necessary.

Unlike single-objective optimization, where solutions can be directly ranked based on a single numerical value, multi-objective optimization requires a different approach to assess solution quality. The concept of dominance is used to establish a preference order among solutions.

Pareto Optimality and Dominance

Pareto rationality provides the theoretical foundation for analyzing multi-objective optimization problems. Given two solutions $x^{(k)}, x^{(h)} \in \Omega$, the solution $x^{(k)}$ is said to *dominate* $x^{(h)}$ (denoted as $F(x^{(k)}) \prec F(x^{(h)})$) if and only if:

- $f_i(x^{(k)}) \leq f_i(x^{(h)})$ for all $i \in \{1, \ldots, m\}$, and
- $f_j(x^{(k)}) < f_j(x^{(h)})$ for at least one index j.

A solution that is not dominated by any other solution in the feasible space is said to be Pareto optimal. The set of all such non-dominated solutions forms the Pareto set, and its image in the objective space defines the Pareto front. Figure 3.1 shows an example of Pareto set and its associated Pareto front.

The primary objectives of multi-objective optimization are:

- To find a set of solutions as close as possible to the true Pareto front (convergence).
- To obtain a diverse set of solutions well-distributed along the Pareto front (diversity).

Multi-Objective vs. Many-Objective Optimization

Multi-objective optimization typically refers to problems with a relatively small number of objectives, usually between two and four. In such cases, Pareto dominance remains an effective criterion for solution comparison, and visual representations of the Pareto front are feasible.

However, as the number of objectives increases beyond four, the problem enters the domain of many-objective optimization. This transition introduces additional challenges:

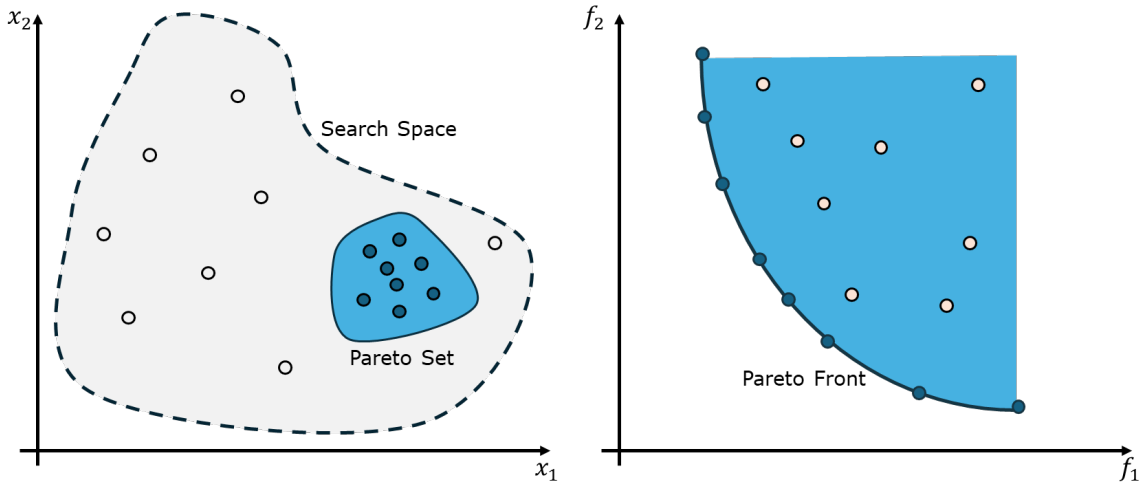

Fig. 3.1 An example of Pareto Set (left) and its associated Pareto Front (right) considering a minimization problem.

- **Dominance weakening**: In high-dimensional objective spaces, the probability of one solution dominating another decreases significantly, leading to a large proportion of solutions being non-dominated. This makes Pareto-based selection less effective.
- **Computational complexity**: Many quality indicators become computationally infeasible as the number of objectives grows.
- **Visualization challenges**: The Pareto front can no longer be effectively visualized, making the interpretation and analysis of solutions more difficult.

To address these issues, many-objective optimization methods often employ alternative strategies, such as decomposition-based approaches, indicator-based selection, and dimensionality reduction techniques to facilitate solution evaluation and selection.

The distinction between multi-objective and many-objective optimization is crucial, as the choice of algorithms and performance metrics must be adapted accordingly to ensure effective optimization in high-dimensional objective spaces.

Metrics

Evaluating the quality of solutions in multi-objective optimization is inherently more complex than in single-objective optimization, where a straightforward comparison of objective function values suffices. In multi-objective problems, performance assessment requires specialized metrics that quantify how well the obtained set of solutions approximates the true Pareto front. These metrics typically focus on aspects such as convergence to the true Pareto

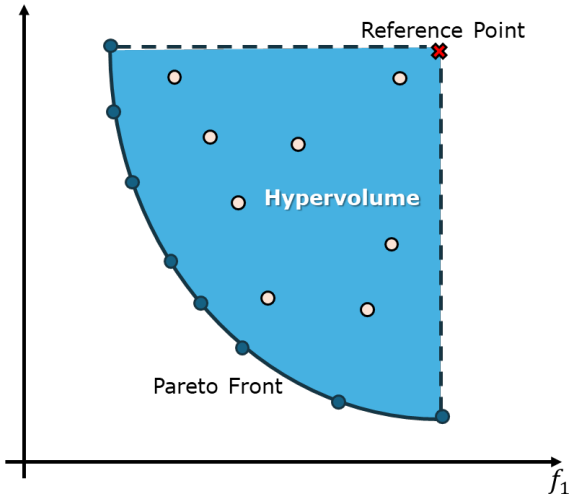

Fig. 3.2 An example of Hypervolume of a Pareto Front.

front, diversity of the solutions, and uniformity of their distribution. For a complete review of metrics for multi-objective optimization refer to [124].

One of the most widely used quality indicators is the Hypervolume (HV) metric [5], which measures the volume of the objective space dominated by the obtained Pareto front while being bounded by a reference point z^*. Formally, given a set of solutions S in the objective space, the hypervolume is defined as:

$$HV(S,z^*) = \lambda\left(\bigcup_{s\in S}\{y \mid s \preceq y \preceq z^*\}\right), \tag{3.2}$$

where $\lambda(\cdot)$ denotes the Lebesgue measure (volume) in the objective space, and $s \preceq y$ indicates that solution s dominates or is equal to point y (Figure 3.2). A higher hypervolume value indicates a better approximation of the Pareto front, as it suggests that the solutions cover a larger region of the objective space. However, the main drawback of the hypervolume indicator is its computational complexity [11], which grows exponentially with the number of objectives. This makes it impractical for many-objective optimization problems (i.e., problems with more than three or four objectives).

To overcome this limitation, an alternative metric commonly used in many-objective optimization is the Inverse Generational Distance (IGD) [75, 39]. The IGD measures the distance between the obtained solution set and a set of reference points that approximate the true Pareto front. Let P^* be a set of uniformly sampled points from the true Pareto front, and let S be the set of solutions found by the algorithm. The IGD is defined as:

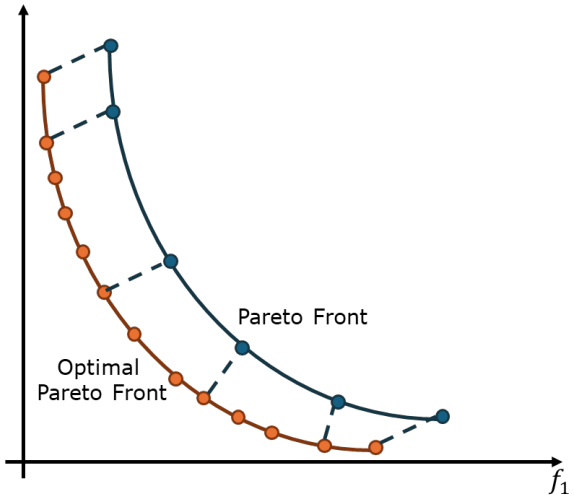

Fig. 3.3 An example of IGD computation.

$$IGD(S,P^*) = \frac{1}{|P^*|} \sum_{y^* \in P^*} d(y^*,S), \tag{3.3}$$

where $d(y^*,S)$ is the Euclidean distance between a reference point y^* and its nearest neighbor in S (Figure 3.3). A lower IGD value indicates better quality, as it implies that the obtained solutions are close to the true Pareto front and well-distributed. The key advantage of the IGD over the hypervolume metric is its computational efficiency, making it feasible for problems with a high number of objectives. However, it is important to note that IGD can only be used when a reliable reference set P^* is available, which may not always be the case in practical scenarios.

Another commonly used metric is the Generational Distance (GD), that is similar to IGD, but computes the average distance from the obtained solutions to the true Pareto front instead of the other way around [148].

Selecting an appropriate metric depends on the characteristics of the problem, the number of objectives, and the availability of the true Pareto front for reference. In many-objective optimization, computationally efficient metrics such as IGD are preferred, while for problems with fewer objectives, hypervolume remains a robust measure of solution quality.

3.1.2 Strategies

Solving multi-objective optimization problems requires specialized strategies capable of addressing the simultaneous optimization of conflicting objectives. Unlike single-objective optimization, which seeks a single optimal solution, the goal in the multi-objective setting

is to approximate the Pareto front by identifying a set of trade-off solutions that balance convergence toward the front and diversity across it.

A classical approach to multi-objective optimization involves scalarization techniques, which convert the vector-valued objective into a scalar function that can be minimized using standard optimization methods. The weighted sum method [55] is the most basic scalarization technique, where each objective is assigned a weight and the scalar objective is defined as a convex combination of the original functions. Although simple and computationally efficient, this method can only identify solutions lying on the convex regions of the Pareto front. Other scalarization techniques, such as the Tchebycheff method [99] and ε-constraint method [90], offer improved flexibility and can explore non-convex regions by transforming objectives into constraints or minimizing the worst-case deviation from a reference point. While scalarization approaches are effective when user preferences are known, they typically require multiple runs with varying configurations to produce a diverse Pareto front approximation.

Population-based methods, particularly evolutionary algorithms, have emerged as some of the most popular and effective strategies for multi-objective optimization [173, 171]. Their inherent ability to maintain and evolve a population of solutions makes them well-suited to explore a wide range of trade-offs in a single optimization run. Among these, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [50] is widely adopted due to its use of fast non-dominated sorting and crowding distance mechanisms to guide selection. The Strength Pareto Evolutionary Algorithm 2 (SPEA2) [174] extends this idea by incorporating an external archive and fitness assignment that combines dominance ranking and density estimation. Another influential algorithm is MOEA/D (Multi-Objective Evolutionary Algorithm based on Decomposition) [166], which decomposes the problem into a set of scalar subproblems optimized in parallel, promoting solution diversity across the objective space. These evolutionary strategies are particularly robust when handling noisy, discontinuous, or multi-modal objective functions. However, they can be computationally expensive and require careful parameter tuning.

Surrogate-based optimization, and in particular Bayesian optimization (BO), provides an attractive alternative for problems where objective function evaluations are costly. BO relies on probabilistic surrogate models—typically Gaussian Processes—to approximate the objective functions and uses acquisition functions to sequentially select new evaluation points [61, 2, 63]. In the multi-objective setting, several extensions of BO have been developed [83, 89, 155]. One of the most widely used is the Expected Hypervolume Improvement (EHVI), which quantifies the expected increase in the volume dominated by the Pareto front upon evaluating a new candidate solution [47]. Other strategies include ParEGO [85], which repeatedly scalarizes the multi-objective problem using randomly sampled weights

and applies standard BO on the resulting surrogate, and methods that model correlations among multiple objectives using multi-output Gaussian Processes. Bayesian multi-objective optimization offers a compelling trade-off between sample efficiency and global exploration, making it particularly suitable for expensive black-box functions. Nevertheless, its applicability becomes limited in the many-objective setting due to the challenges in accurately modeling and selecting among high-dimensional trade-offs.

Other optimization paradigms also exist. Gradient-based methods can be applied when the objective functions are smooth and differentiable, using multi-gradient descent or scalarization-based gradients. Preference-based and interactive optimization methods are suitable when explicit objective functions are not available or when decision makers' preferences evolve over time. These methods leverage reinforcement learning or user feedback to guide the search process.

The choice of optimization strategy depends on various factors, including the characteristics of the objective functions, the dimensionality of the decision and objective spaces, and the computational budget. In practice, hybrid approaches that combine elements from different strategies—such as evolutionary algorithms with surrogate modeling or scalarization with preference learning—are increasingly adopted to balance exploration, efficiency, and adaptability across diverse multi-objective optimization scenarios.

In this chapter, the focus is on two of the most known Multi-Objective Evolutionary Algorithms, i.e., NSGA-II and MOEA/D.

Non-dominated Sorting Genetic Algorithm II (NSGA-II)

NSGA-II [50] is a widely used multi-objective evolutionary algorithm that relies on Pareto dominance and promotes both convergence and diversity in the population through two core mechanisms: an elitist strategy and a crowding distance-based selection. At each generation, a population of candidate solutions is ranked according to non-dominated sorting: solutions are grouped into different non-dominated fronts $F_1, F_2, \ldots$, where F_1 contains all non-dominated individuals, F_2 includes those dominated only by individuals in F_1, and so on.

To ensure diversity, NSGA-II uses the concept of crowding distance. For solutions within the same front, the algorithm calculates the crowding distance of each individual, which reflects the density of solutions surrounding it in the objective space. The crowding distance for an individual i is computed as:

$$CD(i) = \sum_{j=1}^{m} \frac{f_j(i+1) - f_j(i-1)}{\max_{x \in F} f_j(x)} \tag{3.4}$$

Solutions with a higher crowding distance are preferred during selection, encouraging the maintenance of a well-spread set of solutions. An illustration of the non-dominated sorting and the crowding distance mechanism is shown in Figure 3.4.

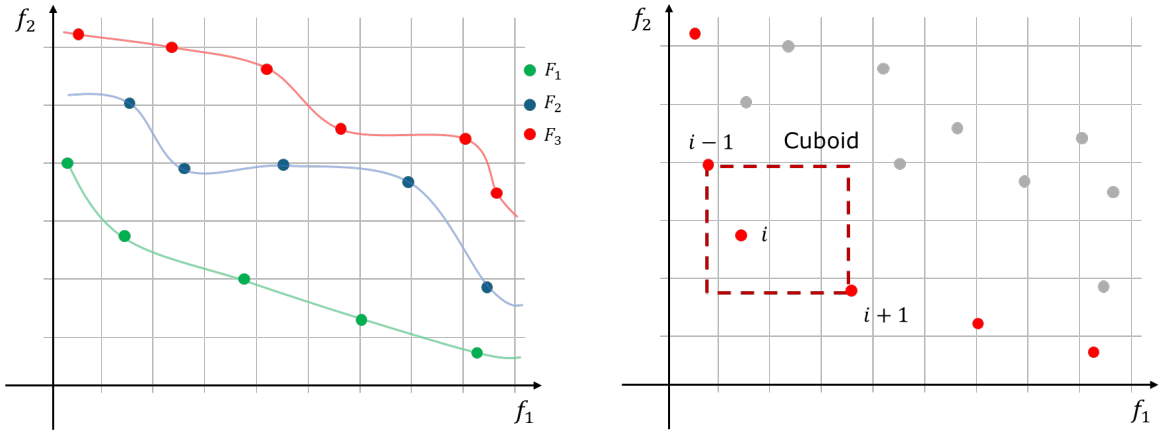

Fig. 3.4 The mechanism of non-dominated sorting (left) and crowding distance (right).

The general workflow of NSGA-II includes non-dominated sorting, selection based on rank and crowding distance, and the application of genetic operators such as crossover and mutation. This process is illustrated in Figure 3.5.

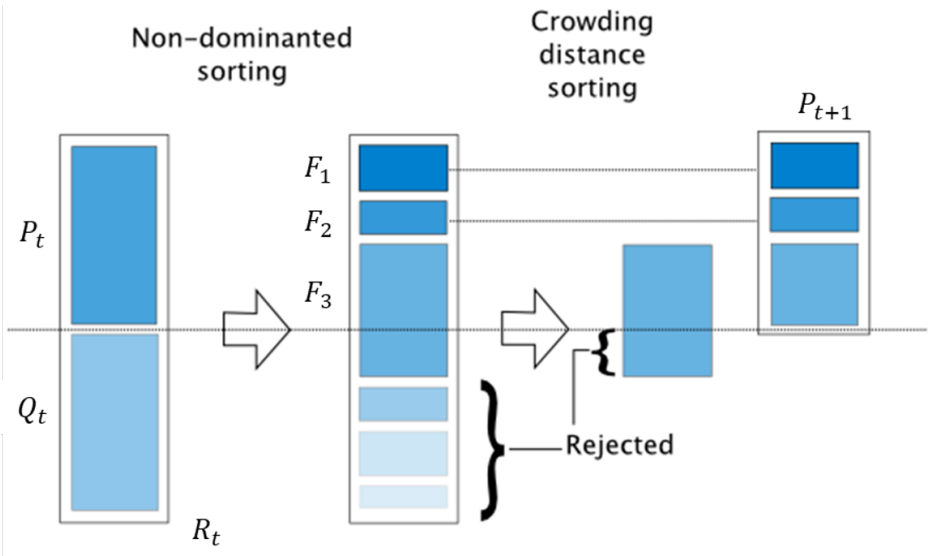

Fig. 3.5 General framework of NSGA-II.

Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D)

MOEA/D [166] approaches multi-objective optimization by decomposing the original problem into a set of N scalar optimization sub-problems. Each sub-problem is defined using a unique aggregation weight vector λ^j and is associated with a specific region of the Pareto front. The optimization of all sub-problems is carried out simultaneously, and neighborhood relationships among them are exploited during the variation and selection phases.

A common scalarization technique used in MOEA/D is the weighted Chebyshev approach:

$$\min g^{te}(x|\lambda^j, z^) = \max_{1\leq i\leq m} \lambda_i^j |f_i(x) - z_i| \tag{3.5}$$

where z^* is the reference point in the objective space, and $\lambda^j = (\lambda_1^j, \ldots, \lambda_m^j)^T$ is the aggregation vector for the j-th sub-problem (Figure 3.6).

During initialization, a set of evenly distributed weight vectors $\lambda^1, \ldots, \lambda^N$ is generated. For each weight vector, the T closest vectors (based on Euclidean distance) are identified, defining the neighborhood $B(i)$. The evolutionary process proceeds by selecting parents from within the neighborhood, applying crossover and mutation operators, and updating solutions if improvement is observed.

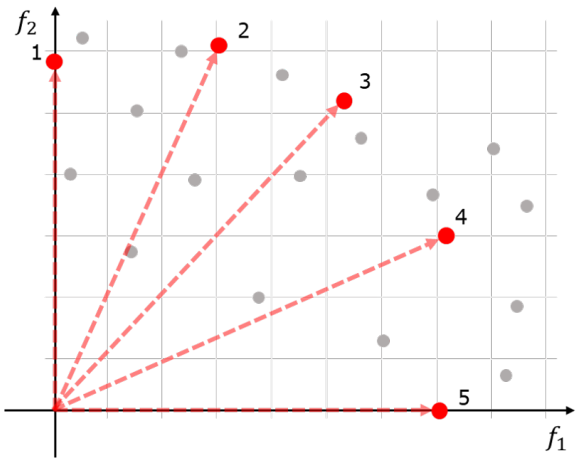

Fig. 3.6 An example of problem decomposition.

Unlike NSGA-II, MOEA/D promotes diversity through the decomposition mechanism itself. Each sub-problem corresponds to a specific region of the Pareto front, and maintaining one solution per sub-problem naturally leads to a diverse set of solutions. This intrinsic decomposition provides a structured way to explore and cover the Pareto front, making MOEA/D particularly effective in problems with a large number of objectives.

A key advantage of MOEA/D is its scalability, as the computational cost does not depend on dominance-based comparisons across the entire population. This makes it suitable for many-objective problems where Pareto-based methods may struggle due to loss of selection pressure.

3.1.3 Wasserstein Distance

Many evolutionary algorithms rely on distance measures to compare individuals, either to assess their similarity to other individuals or to evaluate their proximity to reference points in the objective space. Traditionally, the Euclidean distance has been the most commonly used metric due to its simplicity and computational efficiency. However, the Euclidean distance - and more generally, pointwise metrics - have notable limitations, particularly when comparing sets of points or distributions that are structurally different or exhibit non-overlapping supports. This raises a natural question: *what if a more expressive distance metric were employed?*
Several measures have been proposed in the literature to compare probability distributions. Information-theoretic metrics such as Kullback–Leibler divergence and Jensen–Shannon divergence are among the most widely used, but they can become undefined when the compared distributions do not share identical support. Other measures, such as the total variation distance or the Hellinger distance, are better behaved but still fail to provide meaningful values when distributions have little or no overlap.
In contrast, Wasserstein distances possess a solid mathematical foundation, are generally well defined, and provide an interpretable metric between distributions. Moreover, under most conditions they are differentiable, which makes them particularly suitable for learning and optimization. The notion of optimal transport underlying Wasserstein distances can be traced back to the pioneering works of Gaspard Monge [101] and Lev Kantorovich [80]. In recent years, the Wasserstein distance [149] - also known as the Earth Mover's Distance [127] - has gained increasing attention in the machine learning and optimization communities. Its strength lies in its ability to measure the discrepancy between probability distributions or point clouds in a way that accounts for their geometric structure. This property makes it especially appealing in contexts where solutions are naturally represented as distributions or sets of points, such as in multi-objective evolutionary optimization.

Definition

Let μ and ν be two probability measures defined on a metric space $(\mathscr{X}, d)$, where d is a ground distance (often the Euclidean distance). The p-Wasserstein distance between μ and ν

(Figure 3.7) is defined as [110]:

$$W_p(\mu,\nu) = \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathscr{X}\times\mathscr{X}} d(x,y)^p \, d\gamma(x,y) \right)^{1/p} \tag{3.6}$$

where $\Gamma(\mu,\nu)$ is the set of all couplings (joint distributions) with marginals μ and ν. Intuitively, $\gamma(x,y)$ describes how much "mass" is transported from point x to point y, and the Wasserstein distance quantifies the minimal cost of transporting μ into ν.

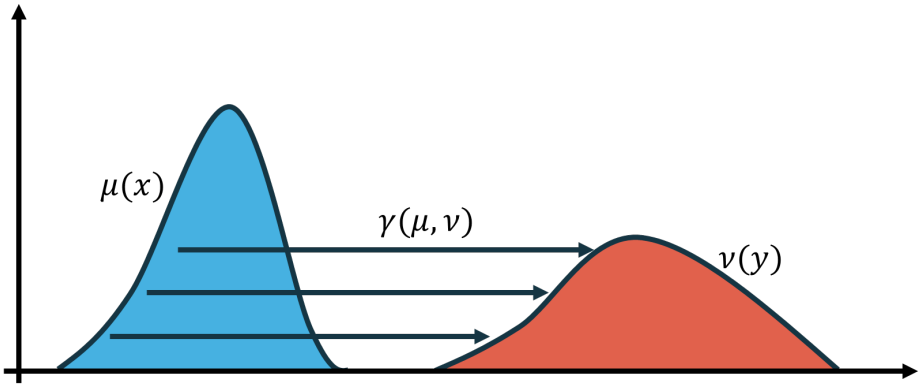

Fig. 3.7 A visual representation of the Wasserstein distance between two continuous probability distributions

When μ and ν are discrete probability distributions supported on finite sets $\{x_1,\ldots,x_n\}$ and $\{y_1,\ldots,y_m\}$, with associated weights $a = (a_1,\ldots,a_n)$ and $b = (b_1,\ldots,b_m)$ such that $\sum_{i=1}^n a_i = \sum_{j=1}^m b_j = 1$, the p-Wasserstein distance reduces to the solution of the following optimal transport problem:

$$W^p(\mu,\nu) = \min_{T \in \Pi(a,b)} \sum_{i=1}^{n} \sum_{j=1}^{m} T_{ij} \, d(x_i,y_j)^p \tag{3.7}$$

where $T \in \mathbb{R}_+^{n\times m}$ is a transport plan such that:

$$\sum_{j=1}^{m} T_{ij} = a_i \quad \forall i = 1,\ldots,n \tag{3.8}$$

$$\sum_{i=1}^{n} T_{ij} = b_j \quad \forall j = 1,\ldots,m \tag{3.9}$$

That is, T specifies how much mass is transported from x_i to y_j, while satisfying the marginal constraints. This problem can be cast as a linear program and solved with standard solvers, though it becomes computationally expensive for large n and m. Despite its desirable theoret-

ical properties, the computation of the Wasserstein distance is significantly more demanding than that of simpler metrics like Euclidean distance. For two empirical distributions with n points each, solving the optimal transport problem naively has a computational complexity of $\mathcal{O}(n^3 \log n)$. Recent advances, such as the Sinkhorn algorithm and entropic regularization techniques, have mitigated this to some extent, making Wasserstein distance more practical for larger datasets.

When the support is one-dimensional and both μ and ν are discrete distributions with sorted values and equal total mass, the 1-Wasserstein distance admits a closed-form expression. Let:

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}, \quad \nu = \sum_{i=1}^{n} b_i \delta_{y_i}$$

where δ_{x_i} denotes a Dirac delta located at x_i, and both (x_i) and (y_i) are assumed to be sorted in increasing order. Then, the 1-Wasserstein distance is given by:

$$W_1(\mu, \nu) = \sum_{i=1}^{n} |F_\mu^{-1}(i/n) - F_\nu^{-1}(i/n)| \tag{3.10}$$

where F_μ^{-1} and F_ν^{-1} are the quantile functions (inverse cumulative distribution functions) of μ and ν, respectively. In the case of uniform weights ($a_i = b_i = 1/n$), this reduces to:

$$W_1(\mu, \nu) = \frac{1}{n} \sum_{i=1}^{n} |x_i - y_i| \tag{3.11}$$

This closed-form formula makes the Wasserstein distance particularly attractive for comparing sorted one-dimensional point sets or distributions, such as individual objectives or scalarized fitness values in evolutionary algorithms.

Applications

The Wasserstein distance has been successfully applied across a wide range of machine learning and optimization tasks. In generative modeling, it has played a central role in the development of Wasserstein GANs (WGANs) [4], where it stabilizes training and improves the quality of generated samples by providing a more meaningful notion of distance between distributions. It has also been employed in clustering [69, 172] and domain adaptation [42], where comparing data distributions across domains or clusters enables more robust alignment and transfer of knowledge. Beyond these areas, the Wasserstein distance has gained increasing prominence in several other domains such as imaging [15, 14] and natural

language processing [87]. More recent applications include recommender systems [98, 6] and neural architecture search [79], where its modeling flexibility and computational tractability have proven advantageous.

In the context of this thesis, the Wasserstein distance is leveraged to enhance both the diversity and the quality of solutions produced by evolutionary algorithms. By interpreting sets of objective vectors as point clouds and evaluating the distance between them in a geometrically meaningful manner, the Wasserstein distance enables a more effective exploration of the solution space and mitigates premature convergence to suboptimal regions of the Pareto front. Overall, incorporating the Wasserstein distance into evolutionary algorithms represents a promising direction for advancing their ability to generate diverse and high-quality solutions, particularly in complex multi-objective optimization settings.

3.2 Multi-Objective Evolutionary Algorithms with Wasserstein

In this section, the main contributions of the thesis to the field of Multi-Objective Optimization are presented. The first is the `NSGA-II/W` algorithm (Section 3.2.1), originally introduced in [113] and then applied to a variety of real-world problems [115, 114, 20, 34, 112, 36]. The second contribution is the `MOEA/D/W` algorithm (Section 3.2.2), proposed in [116] and further investigated in [112]. In addition to these two algorithms, a binary combinatorial crossover operator (Section 3.2.3) was also developed and first presented in [113].

3.2.1 NSGA-II with Wasserstein

One of the contributions of this thesis is a novel variant of the widely adopted `NSGA-II` algorithm, referred to as `NSGA-II/W` [113, 115], which incorporates the Wasserstein distance into the selection mechanism. This variant is developed on top of the `Pymoo` implementation of `NSGA-II` [12], and introduces a modified parent selection strategy that operates in the space of objective vectors, viewed as probability distributions.

In `NSGA-II/W`, the traditional selection operator is replaced by a mechanism that leverages the Wasserstein distance to promote diversity in the objective space. Specifically, the objective values of candidate solutions are treated as point clouds (Figure 3.8), enabling the use of the Wasserstein distance to quantify their dissimilarity. The core idea is to guide mating selection toward individuals that are more diverse in terms of their positions in the objective space, potentially improving the exploration of the Pareto front.

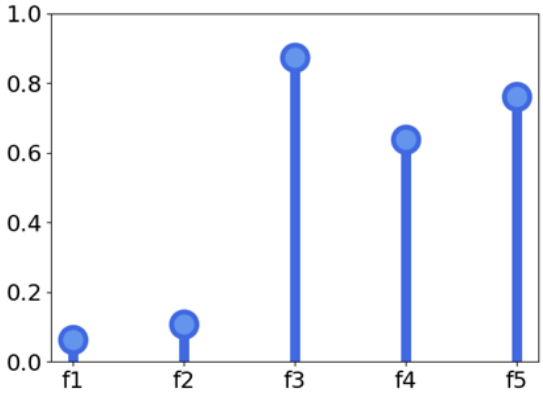

Fig. 3.8 The objective values represented as point clouds.

The mating selection is performed via a binary tournament between two pairs of candidate parents, denoted as $(x^{(1)}, y^{(1)})$ and $(x^{(2)}, y^{(2)})$. These individuals are randomly sampled from the set of non-dominated solutions in the current population (i.e., the current approximation of the Pareto set). The pair selected for reproduction is the one that maximizes the Wasserstein distance between the corresponding objective vectors:

$$(x^{(*)}, y^{(*)}) = \underset{i \in \{1,2\}}{\operatorname{argmax}} \mathscr{W}\left(F(x^{(i)}), F(y^{(i)})\right) \tag{3.12}$$

where $\mathscr{W}(\cdot,\cdot)$ denotes the Wasserstein distance between the objective vectors of two individuals. This selection criterion favors the recombination of parents that are widely separated in the objective space, encouraging the generation of offspring in under-explored regions of the Pareto front.

Once the parents have been selected, crossover and mutation operators are applied as in the standard `NSGA-II` framework to produce offspring and update the population. All other components of the algorithm, including non-dominated sorting and crowding distance, remain unchanged.

By incorporating the Wasserstein distance into the selection process, `NSGA-II/W` aims to enhance the diversity of the evolving population without compromising convergence. Empirical results presented later in this thesis demonstrate that this modification leads to improved performance, particularly in problems characterized by complex and irregular Pareto fronts.

3.2.2 MOEA/D with Wasserstein

The proposed variant of `MOEA/D`, namely `MOEA/D/W` [112], is built upon the Pymoo implementation of `MOEA/D` [12]. The key difference is to consider the weight vectors as discrete probability measures and in particular as points in the unite simplex (Figure 3.9). This is possible due to their nature: the components of each weight vector are non-negative, and they all sum to one. This enables considering the Wasserstein distance, instead of the Euclidean distance, to compare the weight vectors.

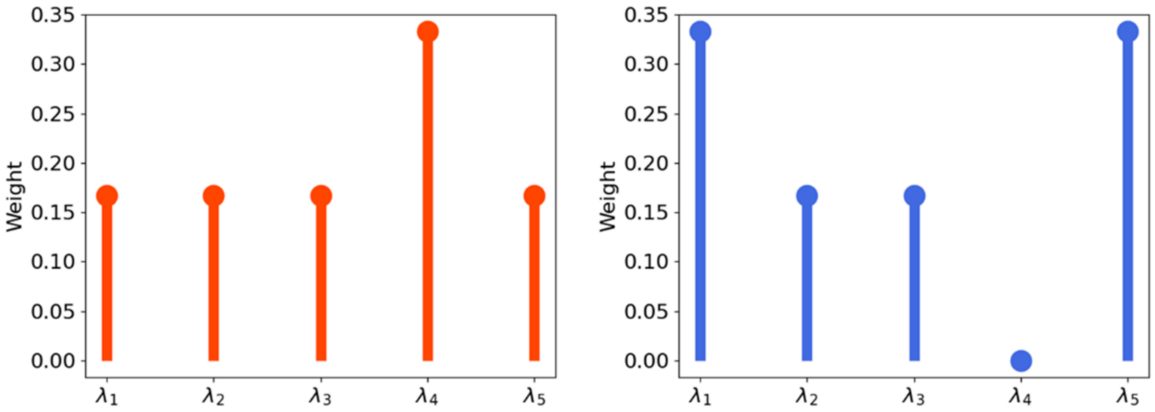

Fig. 3.9 The weight vectors represented as discrete probability distributions.

Take for example the Tchebycheff decomposition. The original multi-objective problems is decomposed in multiple subproblems as

$$\max_{i=1,\dots,m} \lambda_i |f_i(x) - z_i^*| \tag{3.13}$$

where z^* is a reference point and λ is a weight vector such that $\sum_{i=1}^{n} \lambda_i = 1$. In `MOEA/D/W`, for each weight vector $\lambda^{(i)}$, the T farthest weight vectors are chosen as its neighbors $B(\lambda^{(i)}) = \left\{\lambda^{(i)_1}, \dots, \lambda^{(i)_T}\right\}$. As suggested in [12], in all experiments, for problems with $m = 2$ objectives the Tchebycheff decomposition has been used, while for problems with $m > 2$ the Penalty-based Boundary Intersection (PBI) scalarization has been used.
It is important to note that in the standard implementation of `MOEA/D` the T closest (in terms of Euclidean distance) weight vectors are chosen as neighbors instead. Then, two indexes k, l are randomly sampled from $B(\lambda^{(i)})$, and a new solution y is generated from $x^{(k)}$ and $x^{(l)}$ by using genetic operators (crossover and mutation). This process is repeated until a termination criterion is satisfied, such as the number of generations or the number of function evaluations.

3.2.3 Combinatorial Binary Crossover Operator

When decision variables are encoded as binary vectors subject to a maximum number of active components (i.e., a fixed budget on the number of ones), standard crossover operators such as single-point or uniform crossover may produce infeasible offspring that violate the cardinality constraint. This situation arises in various combinatorial optimization problems, such as sensor placement, where each one in the binary vector represents the placement of a sensor, and the total number of sensors cannot exceed a given budget.

To address this issue, a problem-specific crossover operator is proposed that directly operates in the input space and guarantees that the offspring are *feasible by design*, i.e., they respect the maximum allowed number of active components [115]. Such tailored crossover operators have been shown to substantially improve the performance of evolutionary algorithms by preserving feasibility while maintaining diversity.

Let $x, x' \in \{0,1\}^d$ be two feasible parents such that $\|x\|_1 \leq p$ and $\|x'\|_1 \leq p$, where p denotes the maximum allowed number of active components. Define the support sets of the parents as $J = \{i \mid x_i = 1\}$ and $J' = \{i \mid x'_i = 1\}$. The goal is to generate two children $c, c' \in \{0,1\}^d$, each satisfying $\|c\|_1 \leq p$ and $\|c'\|_1 \leq p$.

The proposed crossover works as follows:

1. Initialize two empty support sets for the children: K and K'.
2. While both K and K' have cardinality less than p and at least one of J or J' is non-empty, alternate assigning elements to K and K' as follows:
   (a) Sample one index from J, if available, and add it to K; then remove it from J.
   (b) Sample one index from J', if available, and add it to K; then remove it from J'.
   (c) Sample one index from J, if available, and add it to K'; then remove it from J.
   (d) Sample one index from J', if available, and add it to K'; then remove it from J'.
3. Continue alternating between K and K' until both reach p or no more indices are available.
4. Finally, set the children vectors by defining $c_i = 1$ if $i \in K$, and $c'_i = 1$ if $i \in K'$, and zero otherwise.

This procedure ensures that both offspring remain feasible by construction, with at most p components set to one. The alternating sampling strategy ensures that both children inherit components from both parents in a balanced and diverse way, while preserving feasibility. Figure 3.10 illustrates an example of this problem-specific crossover operator. It can be

observed that the tailored operator guarantees feasibility of the offspring while maintaining diversity between the solutions.

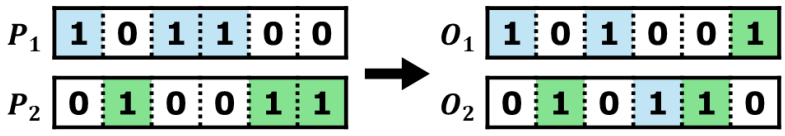

Fig. 3.10 The framework of the proposed crossover operator. The two parents are shown on the left, while the resulting offspring is shown on the right.

This operator can be applied to any binary-encoded problem with a cardinality (budget) constraint, not limited to sensor placement. It is particularly useful in contexts where the feasible set constitutes only a small fraction of the entire search space, and where standard operators would otherwise generate many infeasible solutions, thus wasting computational resources and slowing convergence.

3.2.4 Implementation Details

All algorithms presented in this chapter have been implemented using the `Pymoo` library [12], an open-source Python framework for single- and multi-objective optimization. `Pymoo` provides a comprehensive suite of evolutionary algorithms, performance metrics, visualization tools, and a flexible interface for customizing operators and workflows. Its modular design facilitates rapid experimentation with custom variations of existing algorithms as well as the development of new ones.

NSGA-II/W. The `NSGA-II/W` algorithm builds upon the standard `NSGA-II` implementation in `Pymoo`, with a key modification to the mating selection process. Specifically, the standard binary tournament selection is replaced with a Wasserstein-based selection operator that encourages diversity in the objective space. In this operator, the objective vectors of individuals are interpreted as empirical distributions (i.e., point clouds), allowing the use of the Wasserstein distance to quantify the dissimilarity between candidate solutions. By default, the objective values are used directly as point clouds. However, the selection operator is implemented to be flexible: users can specify alternative representations for individuals, enabling better performance in structured real-world problems. Details on these application-specific adaptations are provided in the subsequent sections.

MOEA/D/W. Similarly, the `MOEA/D/W` algorithm is implemented by modifying the neighborhood structure of the base `MOEA/D` algorithm in `Pymoo`. In the standard formulation, the neighborhood of a subproblem is defined based on the Euclidean distances between weight vectors. In contrast, `MOEA/D/W` treats weight vectors as discrete probability measures (points in the unit simplex) and defines neighborhoods using the Wasserstein distance. As in `NSGA-II/W`, offspring generation proceeds via crossover and mutation applied to parents sampled from the Wasserstein-based neighborhood. All other components of the `MOEA/D` algorithm, including decomposition strategy and update mechanisms, remain unchanged.

Wasserstein Distance. The Wasserstein distance is computed using the `scipy` library [150], which provides an efficient implementation for 1-dimensional discrete distributions. In this setting, the Wasserstein distance quantifies the minimum cost of transporting probability mass to transform one distribution into another, where the cost is defined as the product of mass moved and the ground distance.

3.3 Experiments

The proposed algorithms have been evaluated on a set of widely used benchmark functions (Section 3.3.1) as well as on two real-world applications: optimal sensor placement (Section 3.3.2) and top-k recommendations (Section 3.3.3). This section reports the experimental setup and the corresponding results. All code and data used in the experiments are publicly available on GitHub[1].

3.3.1 Benchmark Functions

The results in this section were originally presented at the NUMTA 2023 conference[2] and published in its proceedings [112].

Problem Description

The experimental campaign considers three widely adopted families of benchmark problems in multi-objective optimization: DTLZ, WFG, and DAS-CMOP. These test suites have been extensively used to assess the performance of multi-objective optimization algorithms due to their scalability, diversity of Pareto front shapes, and ability to incorporate specific challenges such as non-separability, deception, and constraints.

[1] https://github.com/andreaponti5/moeaw
[2] https://www.numta.org/numta2023/

DTLZ Test Suite. The DTLZ test suite, introduced by Deb, Thiele, Laumanns, and Zitzler [51], consists of box-constrained continuous problems that are scalable in both the number of objectives and decision variables. Each problem is characterized by a parameter k that determines the dimensionality of the decision space, and by M, the number of objectives. This suite includes a variety of Pareto front shapes and difficulty levels (Figure 3.11):

- DTLZ1: Linear Pareto front.
- DTLZ2: Continuous, unimodal, and non-deceptive.
- DTLZ3: Similar to DTLZ2 but more challenging due to a more complex distance function.
- DTLZ4: Emphasizes solutions near the boundaries of the objective space.
- DTLZ5-6: Designed to evaluate the ability of algorithms to converge towards a Pareto-optimal curve; DTLZ6 is harder due to a non-linear distance function.
- DTLZ7: Features disconnected Pareto-optimal regions.

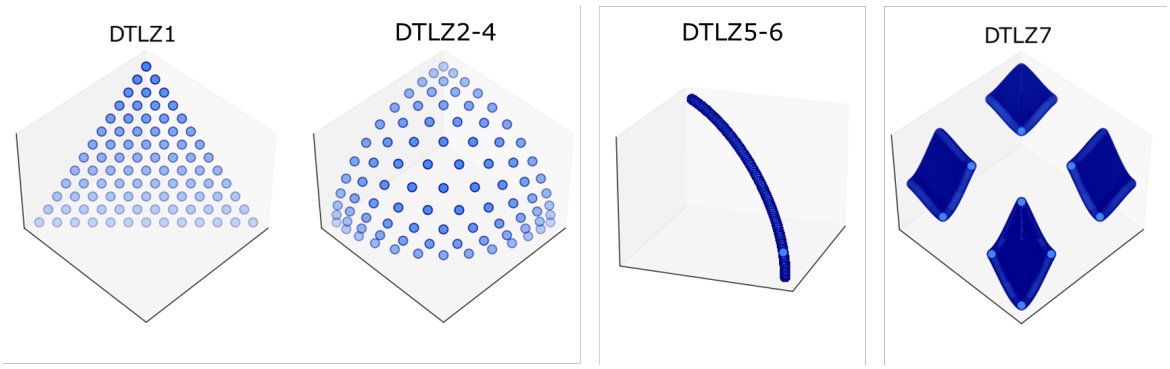

Fig. 3.11 The Pareto Fronts of the DTLZ test functions.

WFG Test Suite. The WFG test suite [71] was designed to address several limitations of earlier benchmarks, providing problems that are scalable in both the number of objectives and decision variables, and that include a wide variety of characteristics: non-separability, multimodality, deception, and mixed Pareto front geometries (Figure 3.12). Nine problems are defined (WFG1-9), covering convex, concave, disconnected, degenerate, and biased PF shapes, with varying degrees of separability and modality. For example, WFG1 introduces weighted parameter significance with a convex mixed PF, WFG5 is a deceptive separable problem, WFG6 is non-separable and unimodal with a concave PF, and WFG9 combines multimodality, deception, and non-separability.

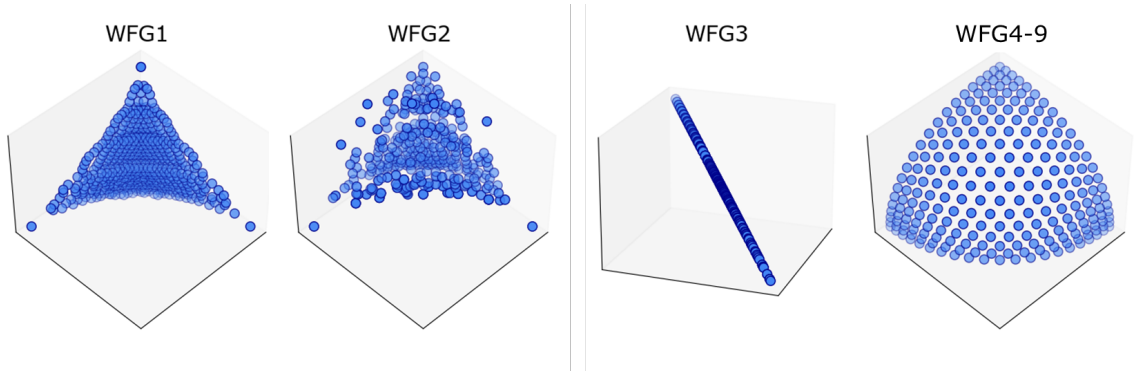

Fig. 3.12 The Pareto Fronts of the WFG test functions.

DAS-CMOP Test Suite. The DAS-CMOP suite (Difficulty Adjustable and Scalable Constrained Multi-Objective Problems) suite [58] extends benchmark design to constrained multi-objective problems, allowing for tunable difficulty through a triplet $(\eta, \zeta, \gamma) \in [0,1]^3$, which controls diversity, feasibility, and convergence hardness, respectively (Figure 3.13). The suite comprises six bi-objective problems (DAS-CMOP1-6) and three three-objective problems (DAS-CMOP7-9), each defined by a combination of scalable objective functions and constraint functions. The difficulty triplet allows one to systematically adjust the problem characteristics, making it suitable for testing algorithms under different constraint levels and PF shapes (convex, concave, or discrete). The construction of higher-dimensional constrained problems follows a methodology inspired by the WFG toolkit, ensuring scalability to any number of objectives.

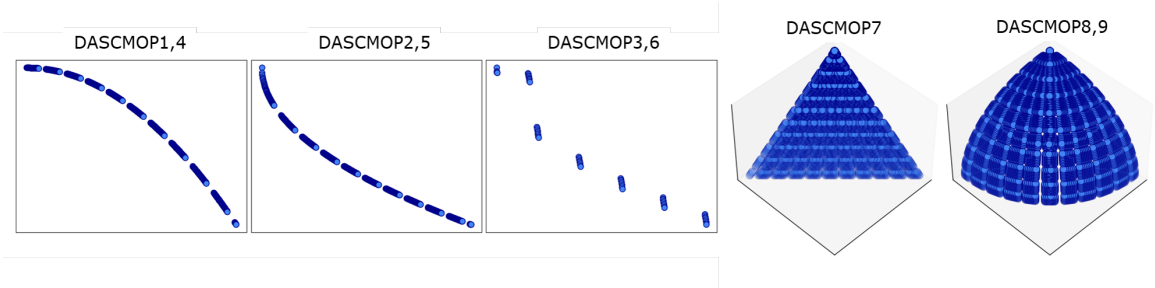

Fig. 3.13 The Pareto Fronts of the DASCMOP test functions.

Overall, these three benchmark families provide a comprehensive set of challenges for evaluating multi-objective optimization algorithms, including unconstrained and constrained settings, different Pareto front geometries, and varying levels of problem complexity.

Experimental Settings

The proposed algorithms, `NSGA-II/W` and `MOEA/D/W`, have been compared against their standard counterparts (`NSGA-II` and `MOEA/D`) to assess the performance improvements introduced by the Wasserstein-enabled selection operator. All algorithms used simulated binary crossover and polynomial mutation, with parameter configurations suggested in [12].

For the DTLZ and WFG benchmark families, the number of objectives has been set to $m \in \{3, 5, 10, 15\}$. The number of decision variables followed the common settings: $d = m + 10 - 1$ for the DTLZ family, and $d = 2 \times (m - 1) + 20$ for the WFG family, as recommended in [92].

The DASCMOP test functions were considered with fixed numbers of decision variables and objectives: $d = 30$, $m = 2$ for DASCMOP1–6 and $d = 30$, $m = 3$ for the remaining instances. These problems also include a difficulty parameter ranging from 1 to 16. In the experiments, five configurations of this parameter have been used: $\{1, 2, 3, 4, 13\}$.

The configuration of generations and population sizes for the DTLZ and WFG families is summarized in Table 3.1. For the DASCMOP problems, a total of 1000 generations was performed, using a population of 51 individuals for the bi-objective instances and 136 individuals for the tri-objective instances.

Table 3.1 Configuration of generations and population sizes for the DTLZ and WFG test families.

m	Generations	Population Size
3	300	10
5	500	35
8	800	36
10	1000	55
15	1500	120

Each algorithm was independently executed 10 times for every problem configuration to account for stochastic variability.

Experimental Results

The experimental results are reported and discussed in this section, focusing on the Inverted Generational Distance (IGD) metric. For each benchmark family, the IGD values obtained by the original algorithms (`MOEA/D` and `NSGA-II`) are compared with those achieved by their Wasserstein-enabled counterparts (`MOEA/D/W` and `NSGA-II/W`). A Wilcoxon signed-rank test with a significance level of 0.05 was performed to assess whether the observed differences

are statistically significant. The null hypothesis assumes no significant difference between the compared algorithms.

Results on DTLZ Test Problems Table 3.2 reports the IGD values achieved by MOEA/D, MOEA/D/W, NSGA-II, and NSGA-II/W on the DTLZ benchmark problems. On average, the proposed MOEA/D/W algorithm outperformed its standard counterpart in 12 out of 23 cases, corresponding to approximately half of the evaluated problems. Notably, MOEA/D consistently achieved superior performance on all DTLZ2 instances, suggesting that the structural characteristics of DTLZ2 favor the original decomposition-based approach. In contrast to MOEA/D, the Wasserstein-enabled selection operator led to limited improvements for NSGA-II: better IGD values were obtained in only 6 out of 23 cases (approximately 26%). This indicates that Pareto-based approaches, such as NSGA-II, are already well-suited to handling the regular Pareto front structures of DTLZ problems, and therefore benefit less from the introduction of Wasserstein selection. Overall, MOEA/D and MOEA/D/W generally outperformed NSGA-II and NSGA-II/W, with the exception of DTLZ4, DTLZ5, and DTLZ7, where the Pareto-based approaches performed competitively. As shown in Figure 3.14, in some instances the Wasserstein distance also promoted faster convergence.

Results on WFG Test Problems The WFG problems are characterized by complex transformations of decision variables, leading to challenging Pareto front geometries that may be disconnected, degenerate, or non-separable. As reported in Table 3.3, decomposition-based approaches benefited more consistently from the Wasserstein-enabled selection. In particular, MOEA/D/W exhibited improved performance on WFG2 and WFG3, with the advantage becoming more pronounced as the number of objectives increased. These results suggest that the Wasserstein distance is especially valuable in situations where decomposition-based algorithms typically struggle, namely when confronted with highly irregular or complex Pareto front geometries.

Figure 3.15 shows that the Wasserstein-enabled variants also converge faster, similar to the observations on the DTLZ benchmarks. In contrast, the performance difference between NSGA-II and NSGA-II/W remained marginal, further supporting the interpretation that Pareto-based methods are naturally better equipped to handle complex Pareto fronts. Consequently, the additional contribution of the Wasserstein selection is limited in this case. Indeed, the results show that both NSGA-II and NSGA-II/W handle the WFG frontiers robustly, often outperforming decomposition-based approaches despite their additional modeling complexity.

Table 3.2 Average IGD values over 10 independent runs, with standard deviations shown in parentheses, for the WFG problems. For each problem, the best-performing algorithm within each category (NSGA-based and decomposition-based) is highlighted in bold, while the overall best result is underlined. p-values are reported for pairwise comparisons, with * and ** indicating statistical significance at the 5% and 1% levels, respectively.

DTLZ	d	m	MOEA/D	MOEA/D/W	p-value	NSGA-II	NSGA-II/W	p-value
1	12	3	14.84 (5.67)	**12.34 (4.89)**	0.610	**23.20 (7.83)**	23.84 (7.12)	1.000
1	14	5	2.87 (2.68)	**1.93 (1.54)**	0.221	**38.28 (14.18)**	55.15 (10.24)	0.019 *
1	17	8	0.29 (0.50)	**0.28 (0.24)**	0.308	**60.66 (17.39)**	86.60 (36.67)	0.126
1	19	10	**0.08 (0.04)**	0.13 (0.02)	0.011 *	**60.96 (26.21)**	74.63 (23.16)	0.683 *
1	24	15	**0.06 (0.01)**	0.21 (0.01)	0.006 **	**51.44 (13.38)**	97.47 (25.33)	0.011 *
2	12	3	**0.00 (0.00)**	0.00 (0.00)	0.683	0.18 (0.02)	**0.16 (0.01)**	0.019
2	14	5	**0.00 (0.00)**	0.01 (0.00)	0.006 **	**0.35 (0.02)**	0.37 (0.01)	0.008 **
2	17	8	**0.00 (0.00)**	0.02 (0.00)	0.006 **	**1.02 (0.05)**	1.09 (0.06)	0.014 *
2	19	10	**0.00 (0.00)**	0.19 (0.00)	0.006 **	1.14 (0.03)	**1.13 (0.05)**	0.610 ***
2	24	15	**0.00 (0.00)**	0.64 (0.17)	0.006 **	1.22 (0.04)	**1.21 (0.04)**	0.083
3	12	3	**27.41 (11.57)**	27.47 (15.34)	0.919	**55.75 (18.94)**	57.90 (14.85)	0.919
3	14	5	5.98 (6.32)	**2.49 (1.94)**	0.308	**87.89 (24.76)**	140.68 (29.77)	0.011 *
3	17	8	1.47 (0.35)	**0.93 (0.42)**	0.011 *	**126.91 (30.20)**	220.05 (60.81)	0.006 **
3	19	10	1.14 (0.40)	**0.64 (0.36)**	0.103	**123.16 (42.40)**	260.34 (72.85)	0.008 **
3	24	15	0.91 (0.61)	**0.81 (0.02)**	1.000	**158.55 (33.47)**	250.64 (88.61)	0.053
4	12	3	0.91 (0.14)	**0.87 (0.19)**	0.103	0.80 (0.34)	**0.80 (0.35)**	0.683
4	14	5	0.87 (0.31)	**0.68 (0.39)**	0.308	**0.48 (0.06)**	0.50 (0.05)	0.476
4	17	8	**0.61 (0.28)**	0.79 (0.35)	0.103	**1.01 (0.08)**	1.03 (0.05)	0.610
4	19	10	0.84 (0.30)	**0.33 (0.11)**	0.006 *	**1.05 (0.07)**	1.08 (0.08)	0.683
4	24	15	**0.45 (0.19)**	0.48 (0.02)	0.610	**1.07 (0.05)**	1.09 (0.06)	0.154
5	12	3	0.12 (0.04)	**0.12 (0.05)**	0.476	0.05 (0.00)	**0.05 (0.00)**	0.262
6	12	3	4.21 (0.80)	**3.90 (0.62)**	0.359	5.90 (0.65)	**5.55 (0.86)**	0.415
7	12	3	**0.58 (0.19)**	0.64 (0.18)	0.760	**0.32 (0.08)**	0.54 (0.22)	0.025 *

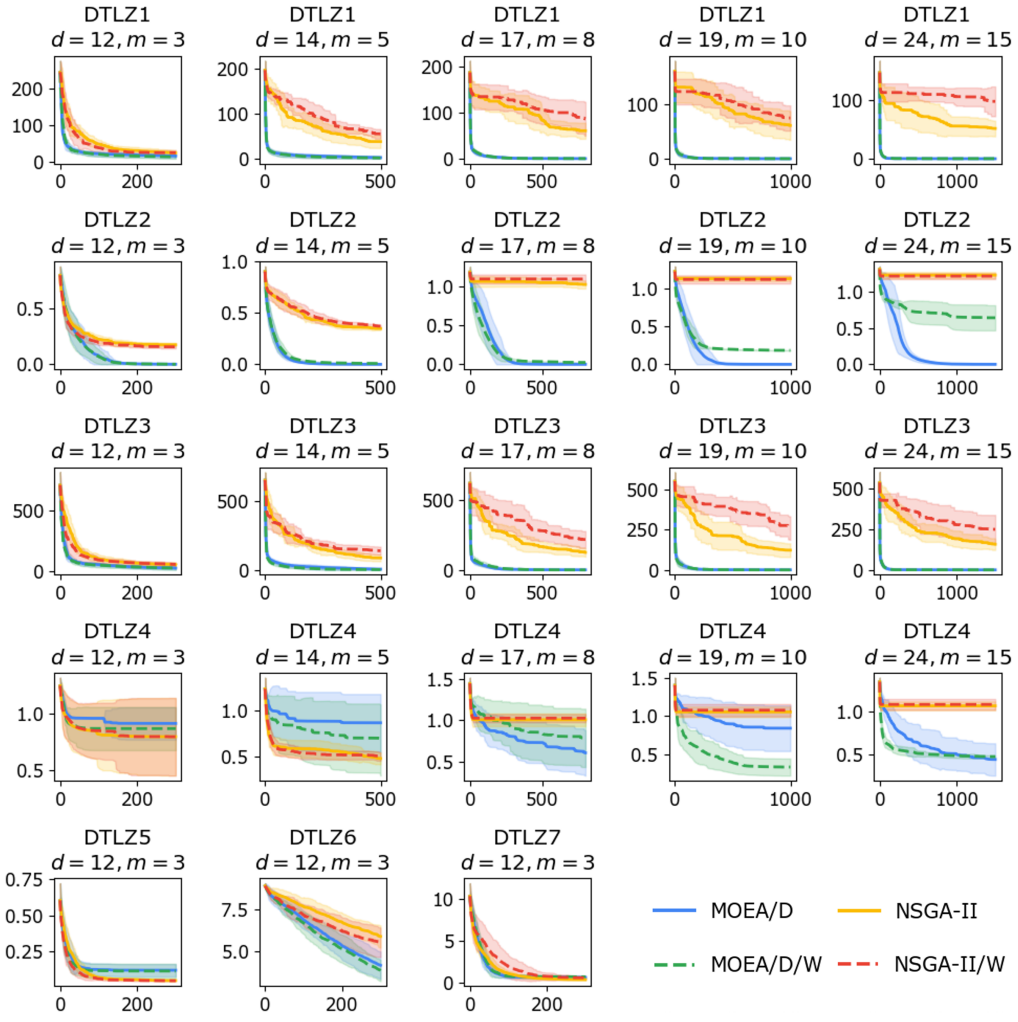

Fig. 3.14 IGD values across generations for all DTLZ test problems. Lines indicate the mean over 10 independent runs, while shaded regions represent the corresponding standard deviation.

Table 3.3 Average IGD values over 10 independent runs, with standard deviations shown in parentheses, for the WFG problems. For each problem, the best-performing algorithm within each category (NSGA-based and decomposition-based) is highlighted in bold, while the overall best result is underlined. p-values are reported for pairwise comparisons, with * and ** indicating statistical significance at the 5% and 1% levels, respectively.

WFG	d	m	MOEA/D	MOEA/D/W	p-value	NSGA-II	NSGA-II/W	p-value
1	24	3	**2.23 (0.08)**	2.26 (0.09)	0.221	**0.05 (0.11)**	2.19 (0.06)	0.011 *
1	28	5	**2.57 (0.06)**	2.62 (0.02)	0.008 **	**2.34 (0.06)**	2.42 (0.03)	0.019 *
1	34	8	**4.40 (0.02)**	4.43 (0.03)	0.083	**3.78 (0.04)**	3.87 (0.05)	0.008 **
1	38	10	**4.78 (0.03)**	4.81 (0.02)	0.006 **	**4.15 (0.06)**	4.30 (0.06)	0.008 **
2	24	3	2.25 (0.44)	**2.11 (0.53)**	0.106	**1.04 (0.37)**	1.18 (0.55)	0.359
2	28	5	2.55 (0.48)	**2.54 (0.44)**	0.919	**0.91 (0.29)**	1.11 (0.34)	0.083
2	34	8	5.82 (0.79)	**5.63 (0.84)**	0.097	**1.38 (0.04)**	1.58 (0.12)	0.006 **
2	38	10	6.72 (0.65)	**5.23 (0.62)**	0.006 **	**1.84 (0.12)**	1.95 (0.09)	0.154
3	24	3	**0.77 (0.14)**	0.82 (0.13)	0.221	0.45 (0.08)	**0.42 (0.10)**	0.683
3	28	5	1.20 (0.05)	**0.96 (0.13)**	0.006 **	0.45 (0.10)	**0.43 (0.08)**	0.919
3	34	8	**1.49 (0.27)**	1.51 (0.14)	0.855	**0.49 (0.12)**	0.51 (0.10)	0.610
3	38	10	**0.89 (0.52)**	1.87 (0.54)	0.789	**0.35 (0.12)**	0.41 (0.15)	0.415
4	24	3	0.91 (0.25)	**0.82 (0.09)**	0.476	**0.69 (0.02)**	0.76 (0.12)	0.359
4	28	5	2.80 (0.38)	**2.76 (0.79)**	0.838	**1.23 (0.06)**	1.34 (0.06)	0.011 *
4	34	8	5.88 (0.72)	**4.82 (0.28)**	0.008 **	**2.92 (0.15)**	3.28 (0.11)	0.006 **
4	38	10	8.54 (0.67)	**3.82 (0.48)**	0.006 **	**3.53 (0.26)**	4.24 (0.17)	0.006 **
5	24	3	**0.76 (0.05)**	1.03 (0.38)	0.067	**0.60 (0.07)**	0.64 (0.07)	0.067
5	28	5	**2.86 (0.22)**	2.90 (0.31)	0.834	**1.34 (0.03)**	1.36 (0.03)	0.154
5	34	8	5.27 (0.36)	**5.03 (0.47)**	0.100	3.38 (0.08)	**3.32 (0.10)**	0.126
5	38	10	5.34 (0.28)	**5.20 (0.29)**	0.059	4.24 (0.09)	**4.17 (0.17)**	0.262
6	24	3	**1.16 (0.54)**	1.19 (0.55)	0.726	**0.61 (0.04)**	0.64 (0.07)	0.053
6	28	5	3.66 (0.14)	**3.40 (0.37)**	0.036 *	**1.49 (0.04)**	1.51 (0.06)	0.262
6	34	8	6.83 (0.79)	**6.47 (0.22)**	0.126	**3.34 (0.15)**	3.57 (0.14)	0.014 *
6	38	10	9.08 (0.05)	**7.52 (0.19)**	0.006 **	**4.78 (0.09)**	5.12 (0.12)	0.006 **
7	24	3	0.90 (0.15)	**0.84 (0.06)**	0.262	**0.68 (0.09)**	0.79 (0.08)	0.032 *
7	28	5	3.19 (0.49)	**2.97 (0.23)**	0.185	**1.45 (0.03)**	1.48 (0.06)	0.221
7	34	8	7.67 (0.45)	**6.71 (0.85)**	0.014 *	**3.59 (0.14)**	3.87 (0.10)	0.006 **
7	38	10	9.47 (0.75)	**7.75 (0.82)**	0.008 **	**4.97 (0.11)**	5.23 (0.07)	0.006 **
8	24	3	**0.94 (0.12)**	0.96 (0.15)	0.541	**0.76 (0.08)**	0.81 (0.06)	0.221
8	28	5	3.10 (0.37)	**2.98 (0.53)**	0.407	**1.56 (0.03)**	1.64 (0.06)	0.011 *
8	34	8	6.78 (1.37)	**6.07 (1.59)**	0.185	**3.74 (0.06)**	4.04 (0.07)	0.006 **
8	38	10	8.24 (1.87)	**6.77 (1.81)**	0.053	**5.03 (0.11)**	5.38 (0.07)	0.006 **
9	24	3	**0.98 (0.17)**	1.08 (0.18)	0.359	**0.83 (0.16)**	0.86 (0.15)	0.610
9	28	5	**2.16 (0.30)**	2.30 (0.32)	0.308	2.05 (0.14)	**2.00 (0.11)**	0.541
9	34	8	6.66 (0.96)	**6.61 (0.54)**	0.838	**4.79 (0.17)**	4.89 (0.13)	0.221
9	38	10	8.73 (1.77)	**7.32 (0.98)**	0.103	**6.21 (0.25)**	6.52 (0.16)	0.008 **

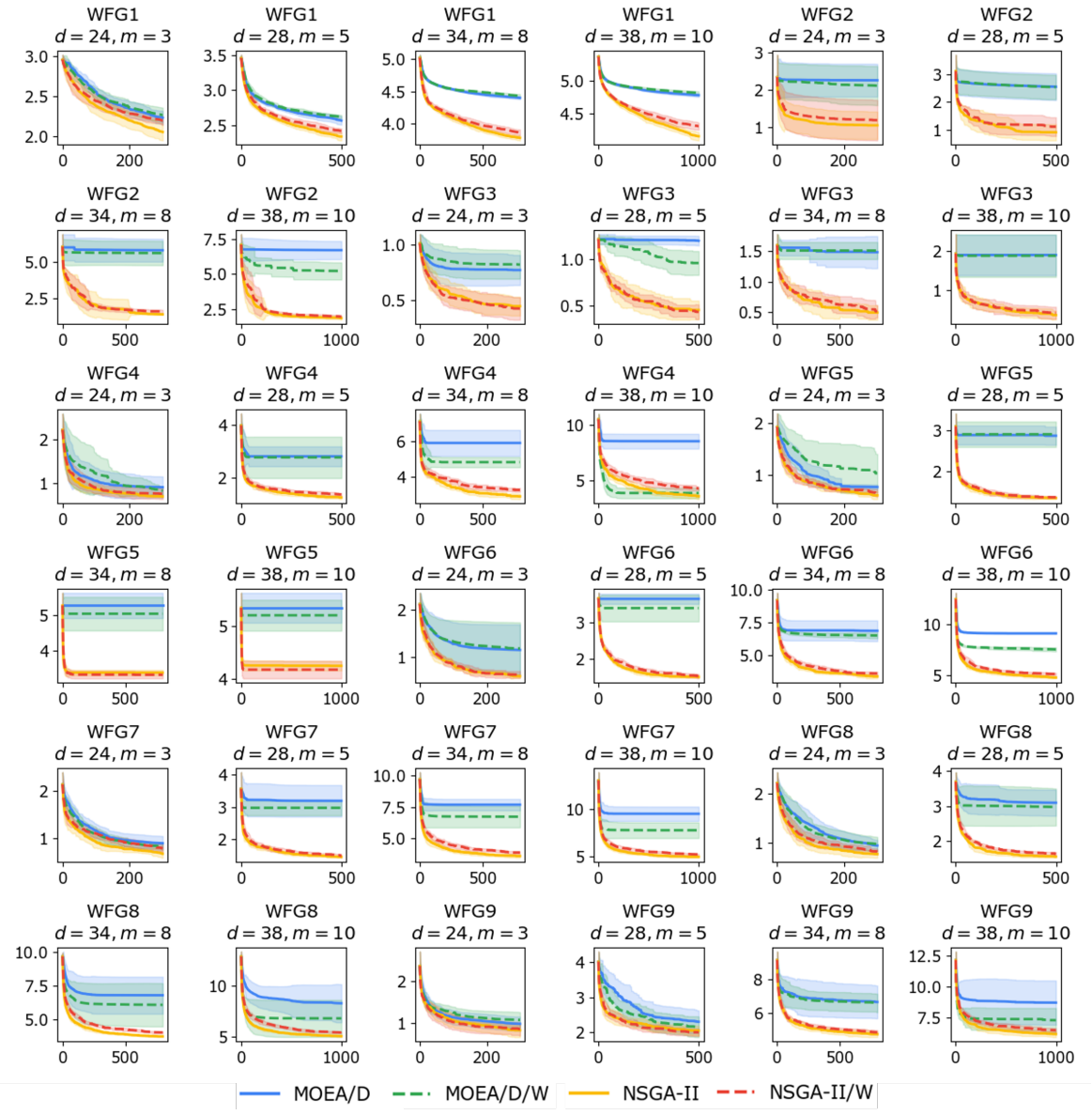

Fig. 3.15 IGD values across generations for all WFG test problems. Lines indicate the mean over 10 independent runs, while shaded regions represent the corresponding standard deviation.

Results on DASCMOP Test Problems Table 3.4 summarizes the IGD results on the DASCMOP problems. These instances, characterized by two or three objectives and several constraints, tend to favor Pareto-based approaches. In this setting, NSGA-II/W frequently outperformed NSGA-II, indicating that the Wasserstein selection can enhance the exploratory capability of NSGA-II in constrained, low-dimensional scenarios. Conversely, MOEA/D and MOEA/D/W achieved comparable IGD values on most instances, with only minor differences between the two.

Figure 3.16 highlights that the Wasserstein-enabled selection generally led to faster convergence for both algorithms, particularly in the bi-objective cases. This reinforces the idea that, while the improvements in overall solution quality may be modest, Wasserstein distance can still accelerate the search dynamics by guiding the exploration more effectively.

Overall Discussion In summary, the incorporation of the Wasserstein-enabled selection operator yielded mixed but insightful results. MOEA/D/W consistently demonstrated improved or at least comparable performance over MOEA/D, particularly on DTLZ and WFG problems with irregular or complex Pareto fronts. This confirms that decomposition-based approaches benefit the most from the Wasserstein distance, which helps mitigate their typical weaknesses when facing challenging front geometries. In contrast, NSGA-II/W showed only marginal improvements, primarily in constrained or low-dimensional settings such as DASCMOP. This can be explained by the fact that Pareto-based methods are inherently well-suited to handling complex or irregular Pareto fronts and therefore gain less from the additional Wasserstein-based selection mechanism.

Overall, these findings suggest that the Wasserstein distance is especially valuable when integrated with decomposition-based algorithms, where it provides a tangible advantage in terms of both solution quality and convergence speed. For Pareto-based approaches, its contribution is more limited, though still beneficial in specific contexts, such as constrained problems.

Table 3.4 Average IGD values over 10 independent runs, with standard deviations shown in parentheses, for the WFG problems. For each problem, the best-performing algorithm within each category (NSGA-based and decomposition-based) is highlighted in bold, while the overall best result is underlined. p-values are reported for pairwise comparisons, with * and ** indicating statistical significance at the 5% and 1% levels, respectively.

DASCMOP	Difficulty	m	MOEA/D	MOEA/D/W	p-value	NSGA-II	NSGA-II/W	p-value
1	1	2	0.37 (0.03)	**0.36 (0.03)**	0.683	**0.37 (0.02)**	0.37 (0.02)	0.308
1	2	2	**0.34 (0.01)**	0.35 (0.04)	0.760	0.32 (0.04)	**0.31 (0.02)**	0.359
1	3	2	1.13 (0.26)	**0.62 (0.40)**	0.019 *	0.38 (0.03)	**0.38 (0.03)**	0.262
1	4	2	0.61 (0.02)	**0.56 (0.09)**	0.032 *	0.47 (0.10)	**0.43 (0.07)**	0.053
1	13	2	**0.38 (0.01)**	0.38 (0.03)	0.838	0.38 (0.03)	**0.37 (0.02)**	0.053
2	1	2	0.34 (0.03)	**0.33 (0.03)**	0.308	**0.34 (0.03)**	0.34 (0.02)	0.919
2	2	2	0.30 (0.01)	**0.29 (0.03)**	0.610	**0.26 (0.02)**	0.28 (0.03)	0.221
2	3	2	**0.33 (0.01)**	0.36 (0.11)	0.476	0.35 (0.02)	**0.33 (0.01)**	0.019
2	4	2	0.46 (0.20)	**0.31 (0.03)**	0.103	0.29 (0.02)	**0.28 (0.02)**	0.541
2	13	2	**0.34 (0.01)**	0.35 (0.03)	0.838	0.34 (0.03)	**0.33 (0.02)**	0.760
3	1	2	**0.38 (0.06)**	0.39 (0.05)	0.919	**0.33 (0.05)**	0.34 (0.04)	0.415
3	2	2	**0.24 (0.02)**	0.27 (0.05)	0.006 **	0.24 (0.02)	**0.23 (0.01)**	0.083
3	3	2	0.32 (0.01)	**0.31 (0.03)**	0.053	0.28 (0.03)	**0.28 (0.01)**	0.610
3	4	2	**0.31 (0.04)**	0.38 (0.14)	0.006 **	**0.30 (0.01)**	0.30 (0.03)	0.610
3	13	2	0.41 (0.08)	**0.39 (0.07)**	0.541	0.42 (0.08)	**0.34 (0.07)**	0.011 *
4	1	2	**0.09 (0.04)**	0.15 (0.02)	0.006 **	0.13 (0.04)	**0.12 (0.05)**	0.683
4	2	2	**0.00 (0.00)**	0.09 (0.00)	0.006 **	**0.00 (0.00)**	0.00 (0.00)	0.032 *
4	3	2	0.35 (0.28)	**0.32 (0.07)**	0.415	**0.25 (0.08)**	0.31 (0.05)	0.053
4	4	2	**0.09 (0.08)**	0.10 (0.00)	0.476	**0.00 (0.00)**	0.01 (0.01)	0.067
4	13	2	**0.01 (0.008)**	0.09 (0.00)	0.006 **	0.00 (0.00)	**0.00 (0.00)**	0.103
5	1	2	**0.13 (0.07)**	0.15 (0.07)	0.185	0.13 (0.05)	0.11 (0.03)	0.415
5	2	2	**0.00 (0.00)**	0.05 (0.00)	0.006 **	0.00 (0.00)	**0.00 (0.00)**	0.041 *
5	3	2	**0.27 (0.14)**	0.42 (0.12)	0.032 *	**0.28 (0.11)**	0.39 (0.07)	0.053
5	4	2	0.15 (0.43)	**0.05 (0.00)**	0.103 *	**0.00 (0.00)**	0.24 (0.43)	0.025 **
5	13	2	**0.02 (0.03)**	0.05 (0.00)	0.083	0.00 (0.00)	**0.00 (0.00)**	0.008 **
6	1	2	**0.21 (0.09)**	0.22 (0.08)	0.760	0.19 (0.09)	0.18 (0.09)	0.919
6	2	2	0.14 (0.06)	**0.13 (0.08)**	0.683	0.10 (0.11)	0.08 (0.09)	0.683
6	3	2	0.55 (0.49)	**0.44 (0.12)**	0.838	0.81 (0.34)	0.63 (0.38)	0.359
6	4	2	**0.31 (0.31)**	0.35 (0.25)	0.610	**0.24 (0.21)**	0.35 (0.18)	0.185
6	13	2	**0.09 (0.08)**	0.18 (0.08)	0.041 *	**0.09 (0.09)**	0.12 (0.09)	0.760
7	1	3	**0.09 (0.02)**	0.15 (0.01)	0.006 **	**0.12 (0.04)**	0.12 (0.03)	1.000
7	2	3	**0.03 (0.00)**	0.12 (0.01)	0.006 **	0.04 (0.00)	**0.03 (0.00)**	0.041 *
7	3	3	**0.14 (0.05)**	0.15 (0.01)	0.683	**0.13 (0.04)**	0.14 (0.02)	0.308
7	4	3	0.01 (0.05)	0.12 (0.00)	0.006 **	0.03 (0.00)	**0.03 (0.00)**	0.476
7	13	3	0.03 (0.00)	**0.11 (0.00)**	0.006 **	**0.03 (0.00)**	0.04 (0.02)	1.000
8	1	3	**0.07 (0.07)**	0.24 (0.14)	0.008 **	**0.10 (0.02)**	0.11 (0.03)	0.359
8	2	3	**0.04 (0.00)**	0.34 (0.21)	0.006 **	0.05 (0.00)	**0.05 (0.00)**	0.541
8	3	3	**0.09 (0.08)**	0.24 (0.14)	0.011 *	**0.10 (0.02)**	0.11 (0.02)	0.221
8	4	3	**0.05 (0.01)**	0.23 (0.07)	0.006 **	0.04 (0.00)	**0.04 (0.00)**	0.221
8	13	3	**0.04 (0.00)**	0.22 (0.08)	0.006 **	**0.05 (0.00)**	0.05 (0.00)	0.760
9	1	3	**0.48 (0.29)**	0.64 (0.06)	0.221	**0.23 (0.15)**	0.31 (0.11)	0.262
9	2	3	**0.25 (0.18)**	0.62 (0.05)	0.006 **	**0.19 (0.0863)**	0.2143 (0.0994)	0.415
9	3	3	**0.61 (0.19)**	0.68 (0.06)	0.067	**0.21 (0.20)**	0.27 (0.12)	0.308
9	4	3	**0.38 (0.22)**	0.66 (0.04)	0.006 **	**0.23 (0.0784)**	0.29 (0.10)	0.126
9	13	3	**0.56 (0.08)**	0.61 (0.08)	0.262	0.52 (0.06)	**0.46 (0.03)**	0.011 *

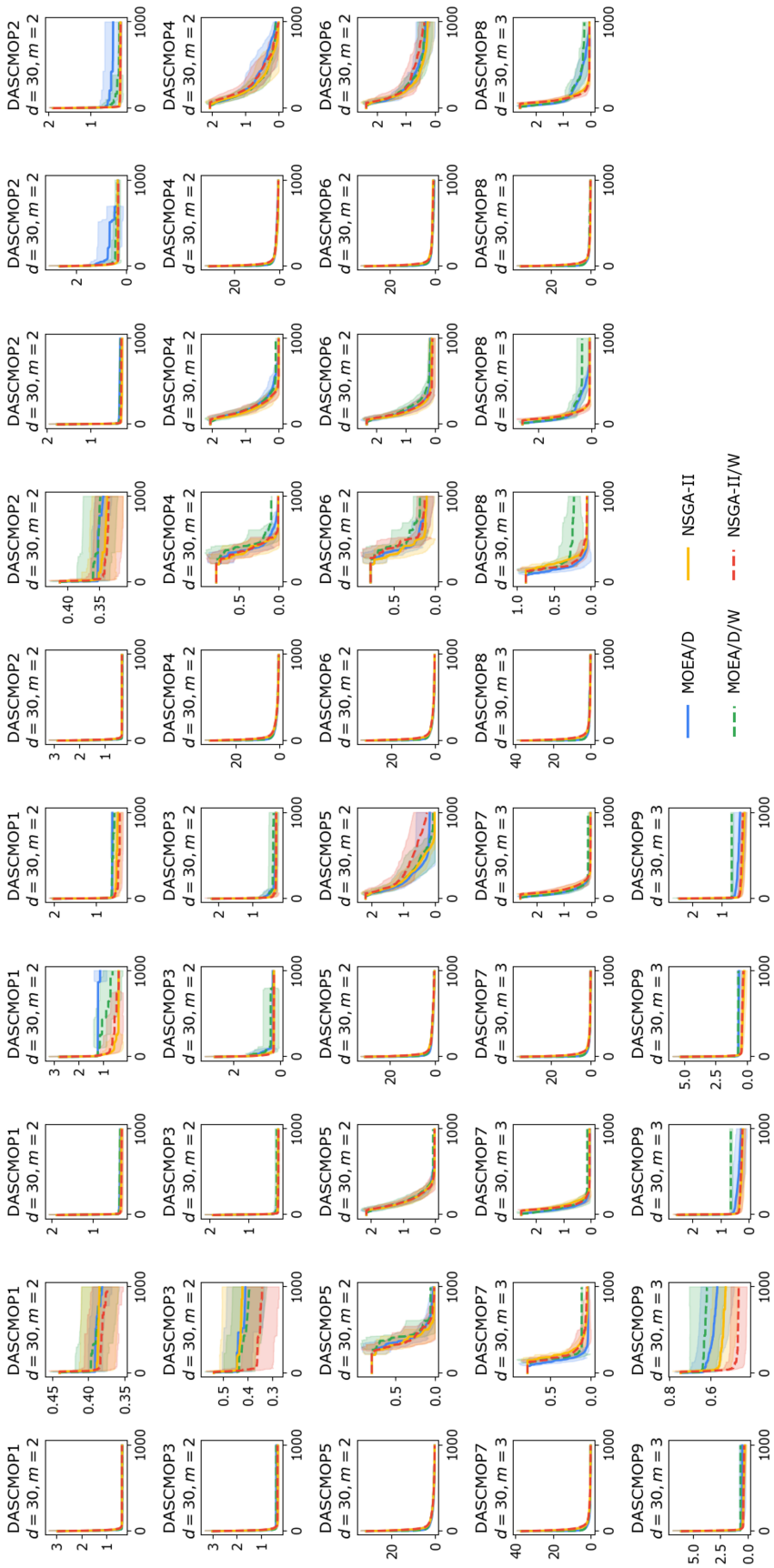

Fig. 3.16 IGD values across generations for all WFG test problems. Lines indicate the mean over 10 independent runs, while shaded regions represent the corresponding standard deviation.

3.3.2 Optimal Sensor Placement

The results presented in this section were originally published in [113, 115].

Problem Description

The sensor placement problem consists of determining the optimal locations of a limited number of sensors in a water distribution network (WDN) to effectively detect contamination events (Figure 3.17). The WDN is modeled as a graph $G = (V, E)$, where the nodes V represent junctions, tanks, reservoirs or consumption points, and the edges E represent pipes, pumps and valves. A sensor placement is encoded as a binary vector $s \in \{0,1\}^{|L|}$, where $L \subseteq V$ is the set of candidate locations, and $s_i = 1$ indicates that a sensor is installed at node i. The placement must satisfy a budget constraint that limits the number of sensors to at most p, i.e., $\|s\|_1 \leq p$.

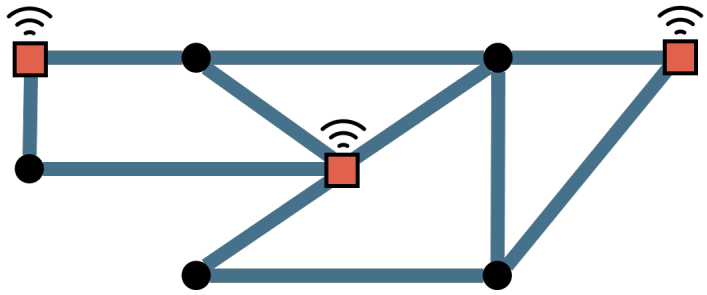

Fig. 3.17 An example of Water Distribution Network with sensors placed on the orange squares.

In this work, the objective is to find a feasible placement s that minimizes the impact of contamination events, assumed to occur uniformly at random across the network nodes. For each contamination event $a \in A$ and each sensor i, the impact d_{ai} is defined as the time until detection or the volume of contaminated water consumed prior to detection if a sensor is placed at node i. The impact of a placement s for an an event a is given by the minimal impaonn the installed sensors:

$$\hat{d}_a(s) = \min_{i:s_i=1} d_{ai}. \tag{3.14}$$

Assuming a uniform probability distribution over contamination events, the two objectives considered for a placement s are:

$$f_1(s) = \text{Average impact} = \frac{1}{|A|} \sum_{a \in A} \hat{d}_a(s), \tag{3.15}$$

$$f_2(s) = \text{Standard deviation of impact} = \sqrt{\frac{1}{|A|} \sum_{a \in A} \left(\hat{d}_a(s) - f_1(s)\right)^2}. \tag{3.16}$$

These two objectives are computed both for the detection time and for the volume of contaminated water, resulting in a total of four objectives: minimizing the average and standard deviation of detection time, and minimizing the average and standard deviation of contaminated water volume.

For a given placement s, the detection time of an event is defined as the earliest time at which any of the sensors detects a contaminant concentration above a specified threshold. Similarly, the contaminated water volume is the total amount of water consumed before detection occurs. These impact measures are computed by simulating contamination events at each node and evaluating the response of the network.

The resulting optimization problem is a multi-objective combinatorial problem with conflicting objectives and a strict cardinality constraint, making it computationally challenging. The aim is to identify sensor configurations that achieve a good trade-off among the objectives while remaining feasible.

Distributional Representation. In the context of Optimal Sensor Placement, the distributional representation of objective values adopted in NSGA-II/W can be directly adapted. Instead of encoding each candidate solution as a vector of m objective values, the distribution of impact measures across contamination scenarios can be used. Specifically, a sensor placement can be represented by a discrete distribution of detection times and contaminated water volumes over all contamination scenarios (Figure 3.18). This formulation enables the use of the Wasserstein distance to compare solutions, capturing differences in the entire distribution of impacts rather than only in aggregated statistics, and thus providing a more informative evaluation of candidate placements under uncertainty.

Data. In the experimental analysis, two benchmark networks and two real-world water distribution networks (WDNs) were considered (Figure 3.19). The *Hanoi* network is a commonly adopted benchmark in the literature, consisting of 32 nodes (1 reservoir and 31 junctions) and 34 pipes. The *Anytown* network is another synthetic benchmark, composed

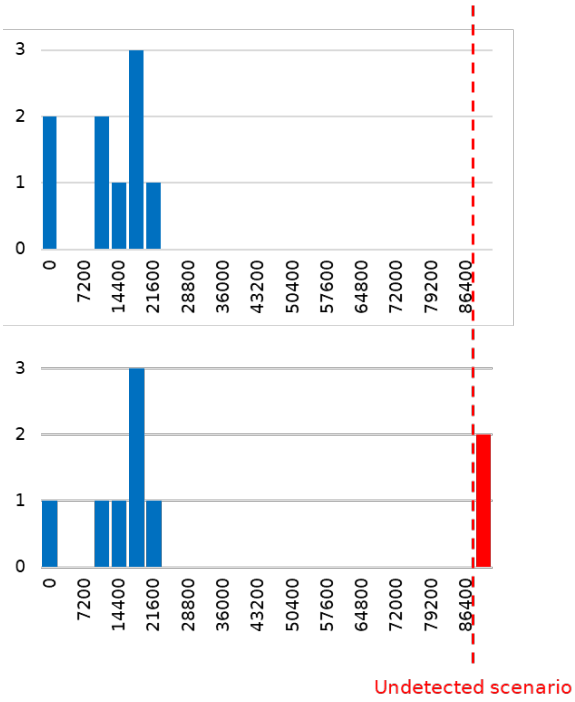

Fig. 3.18 Two example of detection times distributions over different contamination scenarios. The x-axis represents the detection time (in seconds) while the y-axis the number of contamination scenarios detected.

of 25 nodes (1 reservoir, 2 tanks, and 22 junctions) connected by 46 edges (3 pumps and 43 pipes). As real-world cases, the *Neptun* network represents the WDN of the Romanian city of Timișoara, comprising 333 nodes (1 reservoir and 332 junctions) and 339 edges (27 valves and 312 pipes). Finally, the *Apulian5* network models the WDN of an Italian town in Apulia, with 1364 nodes (1362 junctions and 2 reservoirs) and 1518 edges (1475 pipes and 43 valves).

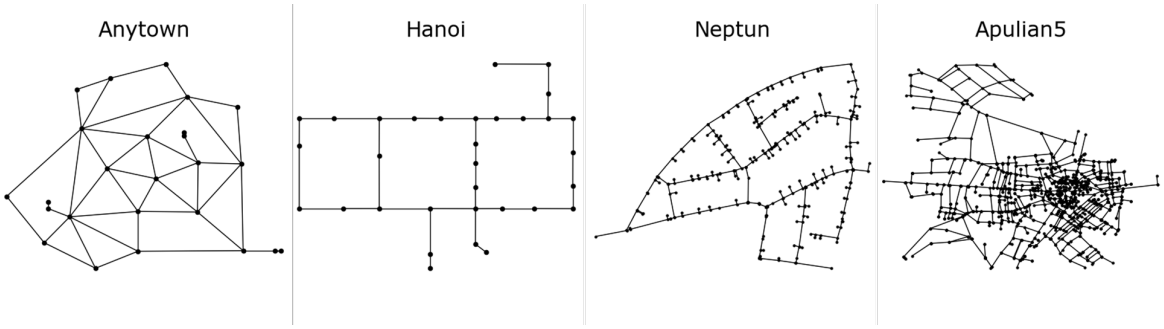

Fig. 3.19 The four water distribution networks used in the experiments.

Implementation Details

To evaluate the performance of sensor placements in detecting contamination events, extensive hydraulic and water quality simulations of the water distribution networks were performed. The simulations compute, for each potential contamination event and sensor placement, the time to detection and the volume of contaminated water consumed before detection. These quantities are derived by simulating contaminant diffusion within the network, assuming that a contaminant is injected at one node at a time and monitoring its spread throughout the system.

The simulations were implemented in Python using the Water Network Tool for Resilience (WNTR) [84], a Python library built on top of the EPANET 2.0 simulator. WNTR provides functionality to compute the hydraulic state of a network and simulate the water quality over time, tracking the concentration of a contaminant injected at a specific location.

To streamline the simulation workflow and facilitate reproducibility, a dedicated Python library called **WaCo**[3] (WAter COntamination) has been developed. WaCo provides a simplified interface to set up and run the hydraulic simulations, extract relevant impact measures, and analyze the results. It is publicly available and can be installed via `pip`. WaCo consists of two main modules:

- **sim**: a wrapper around WNTR that performs hydraulic and water quality simulations. In particular, it generates the so-called *trace matrix*, which records contaminant concentration at each node over time for each possible contamination event.
- **analyzer**: utilities for computing detection times and volumes of contaminated water from the trace matrix.

The typical workflow begins by reading a water network model in EPANET format and simulating contaminant propagation for all possible injection points. The `sim` module produces the trace matrix as a DataFrame, where rows correspond to simulation times and nodes, and columns correspond to injection nodes. From the trace matrix, the detection time at each sensor location is computed as the first time when the contaminant concentration exceeds a specified detection threshold (default 10%). This information is returned in a detection time matrix that lists, for each node and injection point, the corresponding detection time.

In addition to detection times, WaCo can compute the total volume of contaminated water consumed before detection. This requires, in addition to the trace matrix, the time-dependent water demand at each node, which is also simulated using the `sim` module. The `analyzer`

[3] https://andreaponti5.github.io/waco/

module then integrates the demands up to the detection time to estimate the contaminated water volume.

WaCo has been used to generate the impact matrices (detection time and contaminated water volume) employed in the sensor placement optimization experiments. The library simplifies the otherwise complex process of configuring, running, and analyzing simulations, and ensures consistency between simulation parameters (e.g., time step, duration) and analysis procedures. All simulations were performed with a time step of 1 hour and duration of 24 hour, chosen to balance computational cost and accuracy, and contamination events were assumed to occur uniformly at random across all candidate injection nodes.

Experimental Settings

In the hydraulic simulations, the impact of contamination is recorded at each node, while only junctions are considered as potential injection points. Consequently, the number of decision variables in the optimization problem corresponds to the number of nodes in the network. The maximum number of sensors (budget) is set to 4 for Anytown and Hanoi, 8 for Apulian5, and 25 for Neptun.

For each network, two experimental setups are considered: (i) a bi-objective formulation, where the objectives are the average and the standard deviation of the detection time, and (ii) a four-objective formulation, which additionally includes the average and the standard deviation of the volume of contaminated water. The bi-objective problems are solved with a population size of 25, whereas the four-objective problems use a population size of 35. The number of generations is set to 500 for Anytown and Hanoi, 750 for Apulian5, and 1000 for Neptun.

Its important to note that all the algorithms used the proposed combinatorial binary crossover operator.

Experimental Results

The results in Table 3.5 reveal different behaviors of the algorithms depending on the size of the network and the formulation of the problem.

For the smaller benchmark networks (Anytown and Hanoi), all algorithms reached equally good solutions. This confirms that,, for relatively simple instances, the choice of optimization strategy is less critical, as the Pareto front can be identified with limited computational effort.

When moving to larger and more realistic cases, such as Apulian5 and Neptun, performance differences become more evident. In the Apulian5 network, decomposition-based approaches (e.g., MOEA/D and MOEA/D/W) generally achieved the best performance.

Table 3.5 Average Hypervolume values over 10 independent runs, with standard deviations shown in parentheses. For each settings, the best-performing algorithm within each category (NSGA-based and decomposition-based) is highlighted in bold, while the overall best result is underlined.

Network	m	b	MOEA/D	MOEA/D/W	NSGA-II	NSGA-II/W
Anytown	2	4	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)
Anytown	4	4	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)
Hanoi	2	4	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)
Hanoi	4	4	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)
Apulian5	2	8	**0.9288 (0.0393)**	0.8891 (0.0276)	0.9455 (0.0179)	**0.9508 (0.0206)**
Apulian5	4	8	**0.9348 (0.0255)**	0.9228 (0.0372)	**0.9779 (0.0189)**	0.9591 (0.0101)
Neptun	2	25	**0.8615 (0.1013)**	0.8148 (0.0660)	**0.4775 (0.0462)**	0.4321 (0.0261)
Neptun	4	25	**0.8893 (0.0656)**	0.8105 (0.0428)	**0.2681 (0.0207)**	0.2568 (0.0149)

This indicates that when dealing with large-scale and highly complex scenarios, decomposition provides a more robust mechanism for guiding the search, as dominance relations become less discriminative in very high-dimensional spaces. In contrast, for the Neptun network, dominance-based methods (e.g., NSGA-II and NSGA-II/W) consistently outperformed dominance-based approaches, especially as the number of objectives increased. This suggests that for problems of this scale, where the Pareto front remains rich but still manageable, dominance-based strategies are effective in preserving diversity and capturing trade-offs.

It is worth noting that all algorithms employed the crossover operator introduced in Section 3.2.3. Compared to a standard n-point crossover, this operator substantially accelerates convergence for larger networks, such as Apulian5 and Neptun. For example, when using a 5-point crossover, none of the algorithms were able to identify a feasible solution within the termination criteria (750 generations for Apulian5 and 1000 generations for Neptun).

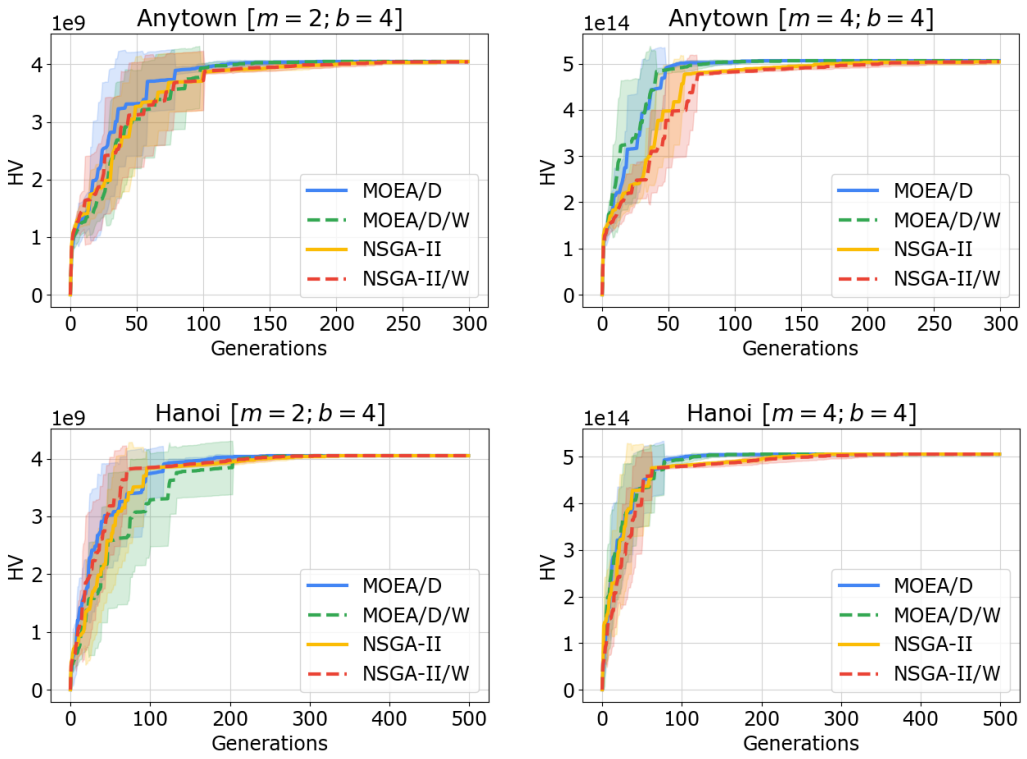

Fig. 3.20 Hypervolume values across generations for the Anytown and Hanoi networks. Lines indicate the mean over 10 independent runs, while shaded regions represent the corresponding standard deviation.

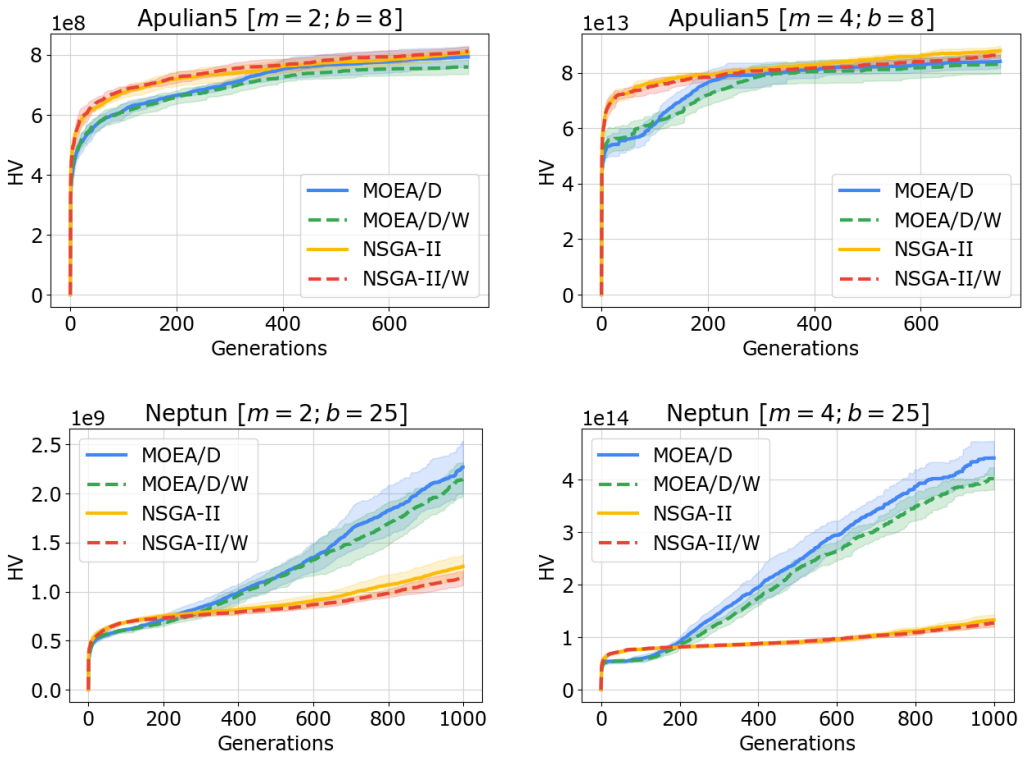

Fig. 3.21 Hypervolume values across generations for the Apulian5 and Neptun networks. Lines indicate the mean over 10 independent runs, while shaded regions represent the corresponding standard deviation.

3.3.3 Recommendation Systems

The results presented in this section were originally published in [36].

Problem Description

Recommendation systems are widely used to help users discover items of interest within large catalogs, by suggesting personalized lists of items based on past interactions, preferences, and behavior. Formally, consider a set of users $U = \{u_1, \ldots, u_M\}$ and a set of items $O = \{o_1, \ldots, o_N\}$. The task of the recommender engine is to assign to each user u_i a list $S_L(u_i) \subseteq O$ of L items to recommend, known as the *top-L recommendation list*.

From an optimization perspective, the problem of generating effective recommendations can be formulated as the search for top-L lists that achieve the best trade-off among several desirable properties of the recommendations. Among the most common objectives considered in the literature are:

- **Accuracy:** how well the recommended items align with the true preferences of the user, often measured using the ratings $r(u_i, o_j)$ assigned by users to items. The average accuracy over all users is defined as:

$$accuracy = \frac{1}{M \cdot L} \sum_{u_i \in U} \sum_{o_j \in S_L(u_i)} r(u_i, o_j) \tag{3.17}$$

- **Coverage:** the proportion of distinct items recommended across all users, which encourages diversity and broader exposure of the catalog:

$$coverage = \frac{1}{N} \left| \bigcup_{u_i \in U} S_L(u_i) \right| \tag{3.18}$$

- **Novelty:** the tendency of the recommender to suggest less popular or unexpected items, which can be quantified through the self-information of the recommended items:

$$novelty = \frac{1}{M} \sum_{u_i \in U} \sum_{o_j \in S_L(u_i)} \frac{N_j}{L}, \quad N_j = \log_2 \frac{M}{d_j} \tag{3.19}$$

  where d_j is the number of users who have rated item o_j.

These objectives are typically conflicting: for instance, maximizing accuracy often favors popular items, which may hurt novelty and reduce coverage. Conversely, increasing novelty and coverage may come at the cost of accuracy. Therefore, the task is naturally posed

as a multi-objective optimization problem, where the goal is to discover a set of top-L recommendation lists that represent good trade-offs among these competing objectives. In this work, the distributions of these objectives over the users are also explicitly modeled and leveraged to guide the optimization.

Distributional Representation. In this setting, the distributional formulation introduced in NSGA-II/W can be naturally extended. Rather than summarizing each objective by a single aggregated value (e.g., the average across all users), each recommendation list is associated with a distribution of objective values over the users. Concretely, accuracy, coverage, and novelty are represented as three one-dimensional discrete distributions, each describing how the objective is realized across the user set (Figure 3.22). This representation allows solutions to be compared using the Wasserstein distance, which accounts for the full shape of the distributions rather than relying solely on averages. By doing so, it captures variability and disparities among users, leading to a more nuanced and informative evaluation of candidate recommendation lists.

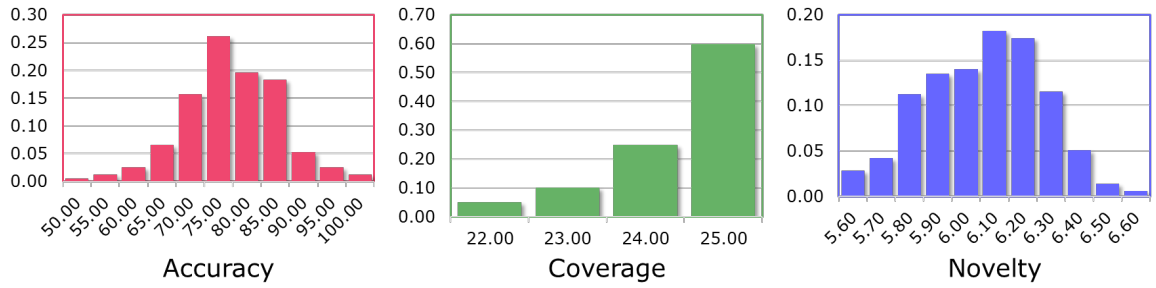

Fig. 3.22 An example of the distributional representation of the three objective. The y-axis represents the frequency of the values over different users.

Data. The experiments have been conducted on the well-known *MovieLens 100K* dataset, collected by the GroupLens Research Project at the University of Minnesota. This dataset contains 100,000 ratings (on a scale from 1 to 5) provided by 943 users on 1,682 movies. Each user has rated at least 20 movies, ensuring a minimum level of engagement. In addition to ratings, the dataset includes simple demographic information about users, such as age, gender, occupation, and zip code. The ratings were collected through the MovieLens website (movielens.umn.edu) over a seven-month period from September 19, 1997 to April 22, 1998. The data has been preprocessed to remove users with fewer than 20 ratings or missing demographic information. This cleaned dataset has become a standard benchmark in the recommender systems literature due to its size, diversity, and accessibility.

Implementation Details

To set up the optimization problem for finding optimal top-L recommendation lists, it was first necessary to predict the missing ratings in the user–item matrix. This was accomplished using the `Surprise` Python library [72], which is a scikit designed specifically for building and analyzing recommender systems based on explicit rating data.

Surprise offers a variety of ready-to-use prediction algorithms, including baseline models, neighborhood-based methods, and matrix factorization techniques such as SVD, SVD++, PMF, and NMF. It also provides flexible tools for handling datasets, evaluating models, and performing cross-validation experiments with minimal effort.

In this work, the SVD (Singular Value Decomposition) algorithm has been adopted to estimate the missing ratings. This algorithm, popularized by Simon Funk during the Netflix Prize competition, is a matrix factorization technique that approximates the user–item rating matrix as the product of lower-dimensional latent factor matrices. Specifically, it seeks to minimize a regularized squared error between the observed ratings and the predicted ratings by adjusting user and item latent factors, along with optional user/item biases. The minimization is performed using stochastic gradient descent, where the learning rate and regularization terms are hyperparameters of the model.

The SVD algorithm estimates the unknown rating $\hat{r}_{ui}$ for user u and item i as:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^\top p_u$$

where:

- μ is the global average rating.
- b_u and b_i are the bias terms for user u and item i, respectively.
- p_u and q_i are the latent factor vectors for user u and item i, respectively.

The model parameters are learned by minimizing the regularized loss:

$$\min_{b_u,b_i,p_u,q_i} \sum_{(u,i)\in\mathscr{K}} \left(r_{ui} - \hat{r}_{ui}\right)^2 + \lambda\left(||p_u||^2 + ||q_i||^2 + b_u^2 + b_i^2\right)$$

where $\mathscr{K}$ is the set of observed ratings and λ is the regularization coefficient.

Once trained, the SVD model was used to predict all missing ratings in the dataset. The resulting fully populated rating matrix — containing both observed and predicted ratings for every user–item pair — served as the basis for computing the accuracy, coverage, and novelty metrics used in the optimization problem.

Experimental Settings

The optimization task is to generate, for each user, a top-L recommendation list that simultaneously maximizes the three objectives introduced before. This naturally leads to a very high-dimensional decision space: each user must be assigned L recommended items, resulting in a total of $M \times L$ decision variables. In the considered dataset, there are $M = 943$ users, and in the experiments the length of the recommendation lists was fixed to $L = 25$. Consequently, the dimensionality of the problem amounts to 23,575 variables. Each variable corresponds to the identifier of an item included in the recommendation list, and is modeled as an integer-valued decision variable.

All algorithms were implemented within the same evolutionary computation framework to ensure comparability. Candidate solutions are represented as integer vectors of length $M \times L$, where each entry encodes an item identifier. Genetic operators were tailored to this representation:

- **Mutation:** an inverse mutation operator was employed, which selects a subsequence of the recommendation list and inverts its order. This operator is particularly suited to permutation-like representations, as it preserves the feasibility of recommendation lists while introducing diversity.
- **Crossover:** a 4-point crossover operator was applied, exchanging multiple contiguous segments between pairs of parent solutions to generate offspring. This promotes effective recombination of building blocks corresponding to user-specific recommendation sublists.

The population size was set to 66 individuals, and the algorithms were evolved for a total of 1,000 generations. These parameters were chosen to balance computational tractability with sufficient evolutionary pressure to explore the high-dimensional search space. To account for stochastic variability, all results were averaged over 10 independent runs, and standard deviations were reported to assess robustness.

Experimental Results

The results of the experiments on the MovieLens1k dataset are reported in Figure 3.23, where the evolution of the hypervolume indicator over 1000 generations is shown for MOEA/D, NSGA-II, and their Wasserstein-based variants. The shaded regions represent the standard deviation across independent runs, providing insights into the robustness of each algorithm.

Overall, the results highlight clear differences between decomposition-based and dominance-based approaches in this high-dimensional recommendation setting. MOEA/D and its variant

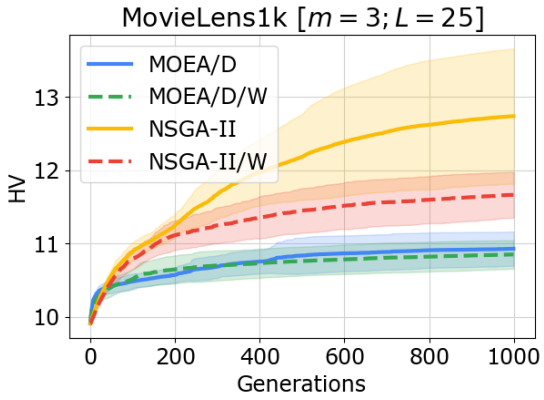

Fig. 3.23 MovieLens1k

converge relatively quickly to stable solutions but achieve comparatively lower hypervolume values. This behavior is consistent with the previous findings: while MOEA/D is efficient at maintaining convergence along predefined directions in the objective space, it often struggles to achieve sufficient diversity with high-dimensional decision spaces and complex Pareto fronts. In contrast, NSGA-II and NSGA-II/W exhibit superior performance in terms of final HV values, suggesting that dominance-based selection is more effective at exploring the trade-offs among accuracy, coverage, and novelty. The advantage becomes particularly evident after the early generations, where NSGA-II continues to improve steadily, whereas MOEA/D tends to plateau.

A second important observation concerns the robustness of the results. The shaded areas indicate that both Wasserstein-enhanced variants (MOEA/D/W and NSGA-II/W) display markedly lower variability across independent runs compared to their baseline counterparts. This suggests that the incorporation of the Wasserstein distance contributes to more stable search dynamics and less sensitivity to stochastic effects, which is a desirable property in large-scale and noisy recommendation problems. The impact of the Wasserstein distance itself on the absolute HV values depends on the underlying optimization paradigm. In MOEA/D, the inclusion of the Wasserstein distance slightly improves stability but does not significantly alter the final performance plateau. In NSGA-II, instead, the Wasserstein variant sacrifices some of the asymptotic HV gains observed in the baseline NSGA-II, but it achieves much narrower confidence intervals, making it more reliable across runs. This points to a trade-off between maximizing the average performance and ensuring robustness, with the Wasserstein-based methods offering a more conservative but stable exploration of the search space.

In summary, the experimental results suggest that dominance-based methods (NSGA-II) are more effective than decomposition-based methods (MOEA/D) in tackling the three-objective, high-dimensional recommendation problem. Moreover, the integration of the Wasserstein distance consistently enhances robustness, reducing variability across runs, and providing more predictable performance, albeit sometimes at the expense of the highest attainable HV values.

Chapter 4

Multi-Information Source Optimization

This chapter provides a theoretical and practical overview of Bayesian optimization, emphasizing the distinction between multi-fidelity and multi-information source optimization. It begins with foundational concepts, including Gaussian Process modeling, acquisition function optimization, and extensions to handle multiple sources of varying cost and accuracy. The chapter then presents the thesis contribution: an Augmented Gaussian Process framework designed for combinatorial structures, along with novel acquisition strategies and their implementation within `BoTorch`. Finally, the effectiveness of the proposed methods is demonstrated through experiments on benchmark functions and real-world problems, including binary quadratic programming and risk-averse optimal sensor placement.

4.1 Background

This section provides the theoretical background for multi-information source optimization. Section 4.1.1 introduces the fundamentals of Bayesian optimization, while Section 4.1.2 focuses on its formulation with Gaussian processes. Section 4.1.3 then discusses the optimization of acquisition functions, a crucial component in Bayesian optimization. Finally, Section 4.1.4 presents the concepts of multi-fidelity and multi-information source optimization, which form the basis for the methods developed in the following sections.

4.1.1 Bayesian Optimization

Bayesian Optimization (BO) is a powerful framework for solving global optimization problems where the objective function is expensive to evaluate, black-box (i.e., lacking a closed-form expression or gradient), and potentially non-convex and multi-modal [60, 2, 63]. It has found wide applicability in domains such as engineering design, machine learning hyperpa-

rameter tuning, robotics, and scientific experiments: anywhere evaluations are costly and sample efficiency is paramount.

Formally, the goal is to find a solution

$$x^* \in \underset{x \in \mathscr{X} \subset \mathbb{R}^d}{\arg\min} f(x), \tag{4.1}$$

where the search space $\mathscr{X}$ is typically defined as a box-bounded domain, i.e., $\mathscr{X} = \Pi_{j=1}^{d}[l_j, u_j]$, with l_j and u_j denoting the lower and upper bounds of the j-th variable.

BO builds and iteratively updates a surrogate model of the true function $f(x)$, using a limited set of evaluations. At a generic iteration n, the information collected so far is encapsulated in the set of observed inputs $\mathbf{X} = \{x^{(i)}\}_{i=1}^{n}$ and their corresponding outputs $\mathbf{y} = \{y^{(i)}\}_{i=1}^{n}$, where each $y^{(i)} = f(x^{(i)})$ under the assumption of a noise-free setting. The next query point, $x^{(n+1)}$, is selected by maximizing an acquisition function $\mathscr{A}(x; \mu(x), \sigma(x))$, which depends on the predictive mean $\mu(x)$ and uncertainty $\sigma(x)$ returned by the surrogate model:

$$x^{(n+1)} \in \underset{x \in \mathscr{X}}{\arg\max;} \mathscr{A}(x; \mu(x), \sigma(x)). \tag{4.2}$$

This reformulation turns the outer, expensive optimization problem in (4.1) into a sequence of inner, inexpensive problems involving only the surrogate model. As a result, BO balances the trade-off between exploitation (i.e., choosing points expected to yield good objective values) and exploration (i.e., sampling uncertain regions to improve the model).

The choice of acquisition function is critical. Various acquisition strategies have been proposed in the literature and are typically categorized into two main families [140]:

- Improvement-based functions, such as Expected Improvement (EI), Probability of Improvement (PI), and Upper/Lower Confidence Bound (UCB), which focus on rapidly finding the minimum function value $f(x^*)$.
- Information-based functions, such as Predictive Entropy Search (PES) and Max-value Entropy Search (MES), which focus on reducing the uncertainty about the optimizer x^* itself.

Although both approaches are valid, information-based acquisition functions are generally more sample-efficient, at the cost of higher computational overhead. They are particularly suited for problems where function evaluations are extremely costly, and only a small number of queries are feasible.

A recent and unifying view is proposed in [103], showing that most acquisition functions, regardless of their family, can be interpreted as instances of a generalized decision-theoretic

framework based on entropy reduction. This insight not only connects existing strategies but also guides the design of new acquisition mechanisms.

Regardless of the acquisition function, BO fundamentally relies on the predictive capabilities of the surrogate model. The standard choice in the BO literature is Gaussian Process (GP) regression, which provides a closed-form posterior for both $\mu(x)$ and $\sigma(x)$, and offers principled uncertainty quantification.

GP models are particularly effective for low- to moderate-dimensional continuous domains. Nevertheless, recent research has extended GP-based BO to handle discrete, categorical, and high-dimensional search spaces [126, 46, 57, 105, 106], broadening its applicability.

4.1.2 Bayesian Optimization with Gaussian Processes

Central to BO is the use of a surrogate probabilistic model that approximates the unknown objective function based on past observations and quantifies the uncertainty of the predictions. A common and effective choice for this surrogate model is the Gaussian Process (GP).

A Gaussian Process is a non-parametric, probabilistic model defined as a collection of random variables, any finite number of which have a joint Gaussian distribution. In the context of regression, a GP defines a distribution over functions, fully specified by a mean function $\mu(x)$ and a covariance function $k(x,x')$. This makes GP regression particularly suited for BO, as it provides both a prediction for each input x and an associated uncertainty.

From a machine learning perspective, GP regression is a kernel method [141, 70], where the covariance function is given by a valid kernel $k(x,x')$. Common kernel choices include the Squared Exponential (Gaussian), Matérn, Exponential (Laplacian), and Rational Quadratic kernels [158, 67]. The choice of the kernel determines the structural assumptions about the unknown function $f(x)$, such as smoothness or periodicity.

Given a set of observed input-output pairs $(\mathbf{X},\mathbf{y})$, training a GP consists of estimating the hyperparameters of the kernel function, typically via Maximum Likelihood Estimation (MLE) or Maximum A Posteriori (MAP) inference.

Once the hyperparameters are learned, the GP defines a posterior predictive distribution for any new input x. The predictive mean and variance are given by:

$$\mu(x) = m(x) + \mathbf{k}(x,\mathbf{X})\left[\mathbf{K} + \lambda^2\mathbf{I}\right]^{-1}(\mathbf{y} - \mathbf{m}(\mathbf{X})) \tag{4.3}$$

$$\sigma^2(x) = k(x,x) - \mathbf{k}(x,\mathbf{X})\left[\mathbf{K} + \lambda^2\mathbf{I}\right]^{-1}\mathbf{k}(x,\mathbf{X})^\top \tag{4.4}$$

where:

- $m(x)$ is the prior mean function (typically assumed to be zero),

- $\mathbf{m}(\mathbf{X})$ is the vector of prior means at the training inputs,
- $\mathbf{k}(x, \mathbf{X})$ is the vector of kernel evaluations between x and the training inputs $\mathbf{X}$,
- $\mathbf{K}$ is the $n \times n$ kernel matrix with entries $K_{ij} = k(x^{(i)}, x^{(j)})$,
- λ^2 is the variance of a Gaussian noise model, capturing observation noise $\varepsilon^{(i)} \sim \mathcal{N}(0, \lambda^2)$ in the data.

In noise-free scenarios, the function is typically assumed to be deterministic $y^{(i)} = f(x^{(i)})$, and one may set $\lambda = 0$. However, in practice, even in the absence of observation noise, a small artificial noise term is often introduced (i.e., $\lambda > 0$) to improve numerical stability. This regularization is particularly important when two training points are very close to each other, which could lead to an ill-conditioned or non-invertible kernel matrix $\mathbf{K}$.

The predictive distribution provided by the GP enables BO to balance exploration (sampling in uncertain regions) and exploitation (sampling near known optima). A detailed summary of the standard BO procedure is presented in Algorithm 1 and an example is shown in Figure 4.1.

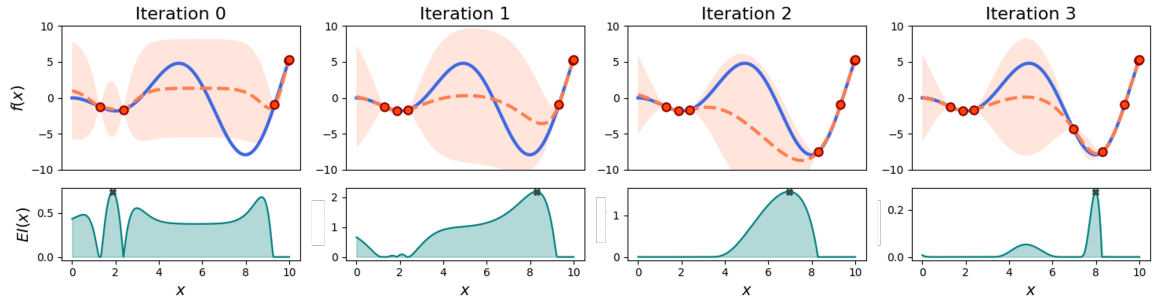

Fig. 4.1 Three iterations of BO for the Forrester test function. The GP on top and the acquisition function values on the bottom.

4.1.3 Optimizing the Acquisition Function

In many cases, acquisition functions are non-convex and, as dimensionality increases, they can become prohibitive to optimize efficiently. The combination of non-convexity and high dimensionality poses a significant challenge, particularly for non-myopic acquisition functions. In the following, several approaches to address this problem are analyzed.

Algorithm 1: vanilla BO algorithm

N, the overall number of queries

n_0, the number of initial random queries

$\mathbf{X} = \{x^{(i)}\}_{i=1:n_0}$ with $x^{(i)} \sim \mathscr{X}$ (e.g., via Latin Hypercube Sampling, Sobol sequence, etc.)

$\mathbf{y} = \{y^{(i)}\}_{i=1:n_0}$ with $y^{(i)} = f(x^{(i)})$ (if noise-free, $y^{(i)} = f(x^{(i)}) + \varepsilon^{(i)}$, otherwise)

$n \leftarrow n_0$

while $n \leq N$ **do**

  $\mu(x), \sigma(x)$ learned from $(\mathbf{X}, \mathbf{y})$ (i.e., $\mu(x), \sigma(x) \leftarrow \text{fitGP}(x, \mathbf{y})$ in GP-based BO)

  $x^{(n+1)} \in \arg\max_{x \in \mathscr{X}} \mathscr{A}(x; \mu(x), \sigma(x))$

  $y^{(n+1)} = f(x^{(n+1)})$ (or $y^{(n+1)} = f(x^{(n+1)}) + \varepsilon^{(n+1)}$, in the noisy setting)

  $\mathbf{X} \leftarrow \mathbf{X} \cup \{x^{(n+1)}\}$

  $\mathbf{y} \leftarrow \mathbf{y} \cup \{y^{(n+1)}\}$

  $n \leftarrow n + 1$

end

Result: $(x^+, y^+) \in (\mathbf{X}, \mathbf{y}) : y^+ = \min_{i=1:N} \{y^{(i)}\}$

Gradient-Based Optimization of Monte Carlo Acquisitions

A common strategy is to estimate the acquisition function using Monte Carlo (MC) methods. Assume that the acquisition $\mathscr{A}$ is defined as an expectation over a multivariate normal distribution $p(y|X, \mathscr{D}) = \mathscr{N}(y; \mu, \Sigma)$ specified by a Gaussian Process surrogate. Let build an unbiased MC estimator of the acquisition $\mathscr{A}(X) \approx \mathscr{A}_m(X) := \frac{1}{m} \sum_{i=1}^{m} \ell(y^{(i)})$. Given this estimator it is necessary to verify whether:

$$\nabla \mathscr{A}(X) \approx \nabla \mathscr{A}_m(X) := \frac{1}{m} \sum_{i=1}^{m} \nabla \ell(y^{(i)}) \tag{4.5}$$

where $\nabla \ell$ denotes the gradient of the utility function with respect to X. The process of differentiating through an MC estimator to its generative distribution's parameter is called reparameterization trick [123, 160] and consists of two components.

The first step consists in reparameterizing samples from the generative distribution p as draws from a simpler distribution $\hat{p}$. If the generative distribution p is a multivariate normal with parameters $\theta = (\mu, \Sigma)$, then the corresponding mapping is $\phi(z; \theta) = \mu + Lz$, where

$z \sim \mathcal{N}(0, I)$ and L is the Cholesky factor of Σ such that $\Sigma = LL^T$. Rewriting Equation 4.9 as a Gaussian integral and reparameterizing, it becomes

$$\mathscr{A}(X) = \int_a^b \ell(y)\mathcal{N}(y;\mu,\Sigma)dy = \int_{a'}^{b'} \ell(\mu + Lz)\mathcal{N}(z;0,I)dz \tag{4.6}$$

Then, for a given draw $y^{(i)} \sim \mathcal{N}(\mu,\Sigma)$, the sample path derivative of ℓ w.r.t. X is

$$\nabla\ell(y^{(i)}) = \frac{d\ell(y^{(i)})}{dy^{(i)}} \frac{dy^{(i)}}{d\mathscr{M}(X)} \frac{d\mathscr{M}(X)}{dX} \tag{4.7}$$

where $y^{(i)} = \phi(z^{(i)}; \mathscr{M}(X))$ and $\mathscr{M}$ is the surrogate model. Therefore, reinterpreting y as a function of z poses the problems of individual MC sample's differentiability.
The second step consists in interchanging differentiation and integration by taking the expectation over sample path derivatives. It is necessary to find out if the average sample gradient $\nabla\mathscr{A}_m$ is composed by differentiable terms, as the MC estimator $\mathscr{A}_m$

$$\nabla\mathscr{A}_m(X) = \nabla\mathbb{E}_y[\ell(y)] \stackrel{?}{=} \mathbb{E}_y[\nabla\ell(y)] \approx \nabla\mathscr{A}_m(X) \tag{4.8}$$

The necessary and sufficient conditions that make this interchange true are that integrand ℓ must be continuous and its first derivative ℓ' must exist and be integrable. In [154] has been shown that these conditions are met using a Gaussian Process with a twice differentiable kernel when there are no duplicates in the set X. The authors in [159], starting from this result, demonstrated the differentiability of a broad class of MC acquisition functions. In particular, they analyzed all the acquisitions that can be written as expectations.

Optimization of Myopic Maximal Acquisitions

For myopic maximal (MM) acquisition functions, greedy optimization strategies are especially effective. These acquisitions are defined as the expected maximum of a point-wise utility $\hat{\ell}$, i.e.,

$$\mathscr{A}(X) = \mathbb{E}_y[\max \hat{\ell}(y)]. \tag{4.9}$$

Examples include Expected Improvement (EI) and Probability of Improvement (PI). Greedy strategies iteratively select the point with the largest immediate expected utility, making them efficient in practice [38, 52].

These strategies are particularly powerful for submodular functions, which exhibit a diminishing returns property: adding a new element yields less gain when the existing set is larger. In [159], the authors proved that the class of MM acquisition functions is inherently

submodular. Consequently, greedy optimization yields near-optimal results with theoretical guarantees [86].

A related approach reformulates the acquisition as a marginal acquisition function [40, 140], such as:

$$\mathbb{E}_{y_{<j}|\mathscr{D}}\left[\bar{\mathscr{A}}(x_j;\mathscr{D}_j)\right], \tag{4.10}$$

where $\mathscr{D}_j = \mathscr{D} \cup \{(x_i, y_i)\}_{i<j}$ is a "fantasy state". Monte Carlo integration is typically used to estimate these expectations. Moreover, any joint acquisition function $\mathscr{A}$ (as in Equation 4.9) can be expressed as a marginal one $\bar{\mathscr{A}}$ using the expected discrete derivative of the utility. This reformulation often results in differentiable, closed-form expressions that are cheaper to evaluate and have lower sample variance.

Bayesian Optimization via Sample Average Approximation

An alternative to standard gradient-based optimization using the reparameterization trick is the *Sample Average Approximation* (SAA) approach. This is employed in the `BoTorch` framework [7]. Instead of drawing new samples from the reparameterized distribution at each iteration, a fixed set of base samples is drawn once and reused throughout the entire optimization process. Conditioned on this sample set, the acquisition function estimate becomes deterministic:

$$\hat{X}_m^* \in \underset{X \in \mathscr{X}^q}{\arg\max}\ \hat{\mathscr{A}}_m(X;\mathscr{D},\psi). \tag{4.11}$$

The gradient $\nabla\hat{\mathscr{A}}_m(X;\mathscr{D},\psi)$ can be computed as an average over sample-level gradients via auto-differentiation.

In [7], convergence results for the SAA approach were provided in the context of Bayesian Optimization with randomized quasi-Monte Carlo (RQMC) methods. Under relatively weak assumptions, they show that the optimizer $\hat{X}_m^*$ converges almost surely to the optimizer of the true acquisition $\mathscr{A}$, with convergence in probability occurring at an exponential rate.

The main advantage of SAA is that it enables the use of deterministic optimization algorithms, which tend to converge faster and are typically less sensitive to hyperparameters than stochastic first-order methods.

4.1.4 Multi-fidelity and Multi-Information Source

Although Bayesian Optimization (BO) is known for its sample efficiency, in many real-world scenarios the cost to evaluate the objective function $f(x)$ can be prohibitively high. This limits the number of allowable queries to the point that even BO may fail to identify a good solution within a feasible budget.

In such contexts, the notion of cost is problem-dependent. It may refer to time, monetary expenses, energy consumption, or physical resources. For example, the evaluation of $f(x)$ may involve performing wind tunnel tests in aerodynamics, running high-fidelity simulations in computational engineering, fabricating new materials, or training deep neural networks on large datasets. In all these cases, the query cost can be substantial and must be minimized.

Fortunately, many of these applications allow for the use of cheap-to-evaluate approximations of the expensive objective function $f(x)$, commonly referred to as the *ground-truth*. These approximations, denoted by $\{f_s(x)\}_{s\in\mathscr{S}}$, can arise from coarser simulations, reduced datasets, or surrogate models. Examples include using simulation software prior to conducting physical experiments [66, 68], or relying on digital twins [95, 104] to emulate physical systems.

The goal is to optimize the ground-truth $f(x)$ while leveraging cheaper approximations to reduce the overall cost. Formally, this leads to the following budget-constrained optimization problem:

$$\begin{aligned} x^* \in \underset{x\in\mathscr{X}}{\arg\min} \quad & f(x) \\ \text{s.t.} \quad & \sum_{i=1}^{N} c_{s^{(i)}} \leq C \end{aligned} \tag{4.12}$$

where $c_{s^{(i)}}$ is the cost to query the information source $s^{(i)} \in \mathscr{S}$ at iteration i, and C is the total budget.

At each iteration, the algorithm must select a source-location pair (x,s) to query. The response $y_s = f_s(x) + \varepsilon_s$ is collected, incurring a cost c_s, and used to update the surrogate models. Importantly, the final solution must be evaluated on the ground-truth $f(x)$ to ensure its validity.

The nature of the source space $\mathscr{S}$ dictates the specific optimization setting. When $\mathscr{S} \subset \mathbb{R}$ and admits a known ranking of fidelities, the problem is typically framed as Multi-Fidelity Bayesian Optimization (MFBO). Conversely, when $\mathscr{S}$ is a finite set with no inherent ordering or when fidelity varies across $\mathscr{X}$, it is called Multiple Information Source Optimization (MISO).

MFBO assumes that each information source has a known, fixed fidelity level, enabling a hierarchical organization. In contrast, MISO relaxes these assumptions and allows for sources with unknown or input-dependent fidelity, better reflecting many real-world scenarios.

The following sections explore these two paradigms in detail.

Multi-Fidelity Bayesian Optimization

Multi-Fidelity Bayesian Optimization (MFBO), introduced in [82], assumes that all sources are ordered by their fidelity and cost. A key example is provided in [145], where a multi-task GP model is used to jointly model function evaluations at different fidelity levels, enabling efficient hyperparameter tuning for machine learning algorithms.

MFBO methods generally rely on GP-based surrogates that incorporate fidelity as an additional input dimension. Notable contributions include [59, 108, 162, 76, 143, 146, 102]. These methods report substantial efficiency gains, but are also subject to failure when their assumptions do not hold.

For example, [143] showed that MFBO can underperform standard BO when low-fidelity sources are poor approximations of the ground-truth. Similarly, [100] and [97] highlighted the risks of assuming a strict fidelity hierarchy.

Continuous Fidelity When $\mathscr{S} \subset \mathbb{R}$, the acquisition function $\mathscr{A}$ is defined over the joint space $\mathscr{X} \times \mathscr{S}$:

$$(x^{(n+1)}, s^{(n+1)}) \in \underset{(x,s)\in\mathscr{X}\times\mathscr{S}}{\arg\max}\ \mathscr{A}\left(x, s; \mu(x,s), \sigma(x,s)\right) \tag{4.13}$$

Here, μ and σ are the predictive mean and standard deviation of a GP defined over $\mathbb{R}^{d+1}$, allowing seamless integration into standard BO algorithms.

Discrete Fidelity When $\mathscr{S}$ is a finite set, e.g., $\mathscr{S} = \{s_1, \ldots, s_S\}$, each source is treated as a discrete fidelity level. While `BoTorch` suggests using the same GP structure as in the continuous case, it may be preferable to use a multi-task GP [13] if no natural ordering exists. The acquisition function is evaluated separately for each $s \in \mathscr{S}$:

$$\tilde{x}_s = \underset{x\in\mathscr{X}}{\arg\max}\,\mathscr{A}\left(x, s; \mu(x,s), \sigma(x,s)\right) \tag{4.14}$$

The final selection is:

$$(x^{(n+1)}, s^{(n+1)}) = (\tilde{x}_{\bar{s}}, \bar{s}) \quad \text{where} \quad \bar{s} = \underset{s\in\mathscr{S}}{\arg\max}\,\mathscr{A}(\tilde{x}_s, s) \tag{4.15}$$

One of the most known multi-fidelity acquisition functions is the Knowledge Gradient (`KG`) [62, 161–163]. The core idea of the Knowledge Gradient acquisition function is to evaluate the value of sampling a point by considering how much it improves the posterior distribution of the function over the entire domain. This principle can be extended to also

consider fidelity, choosing points and fidelity levels that provide the highest value per unit cost. In a multi-fidelity setting, the KG acquisition function measures the trade-off between the information gained from sampling at a specific point and fidelity and the cost associated with that evaluation.

Another famous approach is based on a multi-fidelity extension of the Max-value Entropy Search acquisition function [157]. The key idea in behind the Multi-Fidelity Max-value Entropy Search (`MF-MES`) [146] acquisition function is to focus on the entropy of the maximum value of the highest fidelity function f_1^* rather than the input points x^*. `MF-MES` computes the information gain from querying different fidelity levels to maximize the high-fidelity function with lower costs. The entropy computation is reduced by considering the optimal function value rather than its location, which simplifies calculations to involve mainly one-dimensional integrals. The mutual information measures how much a query at any fidelity level reduces the uncertainty in the maximum value of the highest fidelity function, and it is selected in a cost-efficient manner by maximizing the information gain per unit cost. Additionally, MF-MES can be parallelized to handle asynchronous queries, enabling efficient use of computational resources across multiple fidelities.

The General-purpose Information-Based Bayesian Optimization (`GIBBON`) acquisition function [102], builds upon the `MF-MES` framework but with the goal of generalization across more complex optimization problems. It is designed to work across various complex optimization problems such as noisy, multi-fidelity, and batch BO over continuous or discrete search spaces. The main idea of `GIBBON` is to approximate the information gain that each evaluation provides about the unknown maximum value of a function. By efficiently calculating this information gain, `GIBBON` directs evaluations to regions of the search space that are expected to improve optimization. Unlike other MES-based methods, `GIBBON` simplifies the process by avoiding the need for complex numerical integration, resulting in significantly reduced computational overhead.

Multiple Information Source Optimization

Multiple Information Source Optimization (MISO) relaxes the assumptions of MFBO and is suitable when the sources cannot be ranked or exhibit location-dependent fidelity. This setting is common in real-life applications where simulations, experts, or experimental data may vary in quality across $\mathscr{X}$. The first work in this direction was [88], addressing input-dependent fidelities. Follow-up methods include [111, 65, 18, 29], all relying on GP regression but using separate GP models for each source.

In MISO, each source $s \in \mathscr{S}$ is modeled independently with its own GP using the observed data $(\mathbf{X}_s, \mathbf{y}_s)$. Let $\mu_s(x)$ and $\sigma_s(x)$ denote the predictive mean and standard deviation from

the GP associated with source s. These are fused into a single model:

$$(x^{(n+1)}, s^{(n+1)}) = \underset{(x,s)\in\mathscr{X}\times\mathscr{S}}{\arg\max}\ \mathscr{A}(x, s; \mu(x), \sigma(x)) \tag{4.16}$$

The fused predictions $\mu(x)$ and $\sigma(x)$ can be computed via reification or other ensemble strategies.

Reification combines multiple GPs into a single predictive model. For reference locations $\hat{\mathbf{X}}$, the fused mean and variance are computed as:

$$\mu_{\text{fused}}(x) = \mathbf{k}(x, \hat{\mathbf{X}}) \left[\hat{\mathbf{K}} + \Sigma(\hat{\mathbf{X}})\right]^{-1} \mu_{\text{wink}}(\hat{\mathbf{X}}) \tag{4.17}$$

$$\sigma^2_{\text{fused}}(x) = k(x,x) - \mathbf{k}(x, \hat{\mathbf{X}}) \left[\hat{\mathbf{K}} + \Sigma(\hat{\mathbf{X}})\right]^{-1} \mathbf{k}(\hat{\mathbf{X}}, x) \tag{4.18}$$

where $\mu_{\text{wink}}(x)$ and $\sigma^2_{\text{wink}}(x)$ are defined as:

$$\mu_{\text{wink}}(x) = \frac{\mathbf{e}^\top \tilde{\Sigma}^{-1}(x)\mu(x)}{\mathbf{e}^\top \tilde{\Sigma}^{-1}(x)\mathbf{e}}, \quad \sigma^2_{\text{wink}}(x) = \frac{1}{\mathbf{e}^\top \tilde{\Sigma}^{-1}(x)\mathbf{e}} \tag{4.19}$$

The covariance matrix $\tilde{\Sigma}(x)$ includes pairwise source correlations based on:

$$\tilde{\rho}_{ab}(x) = \frac{\sigma_a(x)}{\sqrt{(\mu_a(x) - \mu_b(x))^2 + \sigma_a^2(x)}} \tag{4.20}$$

In addition to reification, other fusion methods exist, such as co-kriging [59, 54], originally used in engineering design. Though effective in some applications, these methods inherit the limitations of MFBO when the assumptions on source correlation or hierarchy fail. Recent work, such as [153], explores the applicability of co-kriging in settings like inertial confinement fusion. Lastly, strategies from transfer learning and multi-task optimization may also be adapted to MISO, especially when auxiliary tasks or objectives can help improve the learning process.

4.2 Augmented Gaussian Process on Combinatorial Structures

This section presents the main contributions of the thesis to the field of Multi-Information Source Optimization. Sections 4.2.1 and 4.2.2 provide the foundations, introducing the Augmented Gaussian Process and its corresponding acquisition function, originally proposed in [18]. Building on this framework, the `MISO-AGP` algorithm is extended to combinatorial

optimization problems, leveraging genetic algorithms to handle discrete and constrained acquisition functions (Section 4.2.3) [129]. Furthermore, a risk-averse formulation of the optimal sensor placement problem is developed as a practical application of the proposed methodology. Finally, the `MISO-AGP` algorithm has been integrated into the open-source Python library `BoTorch` (Section 4.2.4), ensuring its accessibility and facilitating further research.

4.2.1 Augmented Gaussian Process

The Augmented Gaussian Process (AGP) originally proposed in [18] and further applied in [19, 29], relies on Gaussian Process sparsification to select a subset of informative queries. This subset, drawn from all the queries collected across multiple information sources, is used as inducing locations to construct an AGP that approximates the ground-truth function $f(x)$.

GP sparsification techniques aim to restrict the full Gaussian Process model to a smaller representative set of inducing locations. This set should provide sufficient coverage of the input space to avoid variance starvation [156], while also remaining small enough to ensure the scalability of the GP model on large datasets. A range of sparsification strategies has been proposed [152, 44, 142, 45, 138, 137, 134]. Among them, the approach in [142] offers high accuracy but suffers from high computational demands. Its improved variant [81] reduces computational time at the cost of increased memory requirements.

Earlier contributions to this field focused on developing criteria for insertion [45] and deletion [44] of observations from the set of inducing locations (also referred to as the *basis vector set*), enabling exact model updates.

Subsequently, [138] proposed a method for selecting inducing locations, referred to as *support patterns*, using information-theoretic measures such as Information Gain and the Kullback–Leibler divergence, thereby providing a stable approximation to the marginal log-likelihood of the training data.

A unifying framework for sparse GP approximations was proposed in [121], which demonstrates equivalence with Bayesian models and encompasses various sparsification techniques. One of these is the Deterministic Training Conditional (DTC), closely related to the work of [44, 45].

Further contributions by [137] argue for the importance of Bayesian modelling beyond mere scalability, particularly in high-level decision-making or experimental design tasks where data collection is sequential and actively managed. The paper highlights that Gaussian Processes do not inherently model sparsity and proposes the use of Laplace priors to enhance performance, despite the increased complexity of inference. Inducing location selection is guided by expected information gain, with a computationally efficient approximation

suggested for its estimation. Once the inducing points are identified, a Gaussian posterior is used to restore the GP structure, closely aligning with BO, which is itself an active learning strategy.

The GP sparsification strategy introduced in [18] adopts an insertion-based approach. The set of inducing locations is initialized using queries from the ground-truth information source, denoted as $f(x) = f_1(x)$. The set is then augmented by including additional queries from alternative sources, based on a criterion that accounts for both the discrepancy between their respective GPs and the predictive uncertainty of the ground-truth GP.

Let $\eta(\mathscr{G}, \mathscr{G}', x)$ denote the discrepancy between two Gaussian Processes $\mathscr{G}$ and $\mathscr{G}'$ at a point $x \in \mathscr{X}$. This discrepancy is defined as the absolute difference between their predictive means:

$$\eta(\mathscr{G}, \mathscr{G}', x) = |\mu(x) - \mu'(x)| \tag{4.21}$$

This formulation avoids the need for learning an additional discrepancy model, such as a separate GP as proposed in [111].

The complete set of inducing locations, denoted $\widehat{D}$, is obtained as the union of the ground-truth queries D_1 and a selected subset $\bar{D}$:

$$\bar{D} = \bigcup_{s=2:S} \left\{ (x,y) \in D_s : \eta(\mathscr{G}_1, \mathscr{G}_s, x) < \alpha \sigma_1(x) \right\} \tag{4.22}$$

Here, α is a user-defined threshold parameter, with a suggested default value of $\alpha = 1$ [18]. This selection mechanism includes only those queries from cheaper information sources that fall within the confidence bounds of the ground-truth GP, thereby enhancing the informative content of the training set without significantly compromising accuracy. The resulting inducing set $\widehat{D}$ effectively augments the available ground-truth data. This strategy incurs a computational cost of $O(1)$, which matches the efficiency of the most scalable methods in [138, 134].

Finally, the AGP $\widehat{\mathscr{G}}$ is trained on the selected set $\widehat{D}$, yielding predictive mean and variance functions $\widehat{\mu}(x)$ and $\widehat{\sigma}^2(x)$, respectively. This model differs from previous approaches that either combine all queries into a unified discrepancy model or directly merge the outputs of the individual GPs. The AGP, by contrast, relies on a strategically selected subset of informative queries, yielding a sparse yet accurate surrogate of the ground-truth function.

4.2.2 Augmented Confidence Bound

In the `MISO-AGP` framework, the proposed Augmented Confidence Bound acquisition function selects the next source-location pair by evaluating, for every source $s \in \mathscr{S}$ and location

$x \in \mathscr{X}$, an optimistic improvement over the current augmented best seen value $\widehat{y}^+$, and penalizing this improvement according to both the cost c_s of querying the s-th information source and the discrepancy between its associated GP $\mathscr{G}_s$, and the augmented GP $\widehat{\mathscr{G}}$. The acquisition function is formally defined as:

$$(x^{(n+1)}, s^{(n+1)}) \in \underset{x \in \mathscr{X},\, s \in \mathscr{S}}{\arg\max} \frac{\widehat{y}^+ - \left[\widehat{\mu}(x) - \xi\,\widehat{\sigma}(x)\right]}{c_s\big(1 + \eta(\widehat{\mathscr{G}}, \mathscr{G}_s, x)\big)} \tag{4.23}$$

The numerator captures the potential improvement, where $\widehat{y}^+$ is the lowest value observed so far within the set of AGP inducing locations $\widehat{D}$, and $\widehat{\mu}(x) - \xi\,\widehat{\sigma}(x)$ corresponds to the Lower Confidence Bound (LCB) of the AGP at location x, with ξ regulating the exploration-exploitation trade-off.

This formulation encourages exploration by prioritizing source-location pairs that are both promising according to the AGP and associated with low cost and discrepancy. The addition of 1 in the denominator ensures numerical stability, avoiding division by zero. Since $\widehat{y}^+$ is constant for each iteration, maximizing the acquisition function in (4.23) is equivalent to minimizing the AGP's LCB normalized by the penalization term $c_s(1 + \eta(\widehat{\mathscr{G}}, \mathscr{G}_s, x))$.

The discrepancy $\eta(\widehat{\mathscr{G}}, \mathscr{G}_s, x)$ is calculated as the absolute difference between the predictive means of the two GPs at location x, as defined in Equation (4.21). This simple formulation avoids the need to model the discrepancy as a separate GP, as is done in other multi-source optimization strategies [111].

To determine the next query point, the acquisition function in (4.23) is maximized independently for each source $s = 1, \ldots, S$, yielding S candidate solutions. Among these, the one attaining the highest value determines the next query pair $(x^{(n+1)}, s^{(n+1)})$. This procedure is analogous to that adopted in multi-fidelity Bayesian optimization with discrete fidelity levels, although in the MISO-AGP framework the model is defined over the original d-dimensional domain $\mathscr{X}$, rather than the extended $(d+1)$-dimensional space $\mathscr{X} \times \mathscr{S}$ (Figure 4.2).

Upon selection, the function value $y^{(n+1)} = f_{s^{(n+1)}}(x^{(n+1)})$ is observed at a cost $c_{s^{(n+1)}}$, and the corresponding dataset is updated:

$$D_{s^{(n+1)}} \leftarrow D_{s^{(n+1)}} \cup \left\{ \left(x^{(n+1)}, y^{(n+1)} \right) \right\}$$

The process is repeated until the total query cost exceeds a predefined budget.

It is important to note that the best augmented observed $\widehat{y}^+$ evolves differently from the best seen values that are typically used in Bayesian Optimization or other MISO approaches that rely on fused GPs. Since $\widehat{y}^+$ is computed based on the set of inducing locations $\widehat{D}$, which

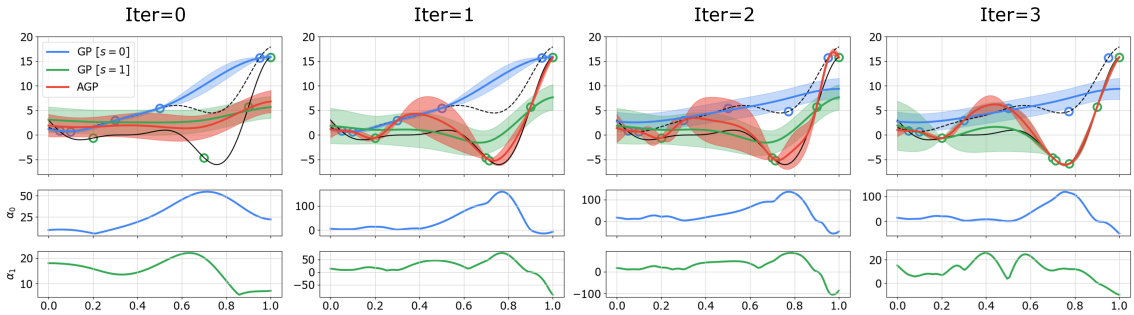

Fig. 4.2 Three iterations of the MISO-AGP algorithm considering the Forrester test function on two different sources. The GPs and AGP on top and the acquisition function values of the two sources on the bottom.

is re-evaluated at each iteration according to Equations (4.21) and (4.22), the sequence of $\widehat{y}^+$ values may be non-monotonic. That is, a previously best-performing point may be excluded from $\widehat{D}$ in future iterations, causing an increase in $\widehat{y}^+$.

For completeness, the definitions of various best-seen values are reported below. In a vanilla BO settings (ground-truth only), the best seen is given by:

$$y^+ = \min\left\{y^{(i)} : (x^{(i)}, y^{(i)}) \in D_1,\ i = 1,\dots,n\right\}. \tag{4.24}$$

In the context of MISO using a fused-GP the best seen is:

$$y^+ = \min\left\{y^{(i)} : (x^{(i)}, y^{(i)}) \in D_s,\ i = 1,\dots,n_s,\ s = 1,\dots,S\right\}. \tag{4.25}$$

Finally, using the `MISO-AGP` algorithm, the best seen is given by:

$$\widehat{y}^+ = \min\left\{y^{(i)} : (x^{(i)}, y^{(i)}) \in \widehat{D},\ i = 1,\dots,p\right\}. \tag{4.26}$$

In Equations (4.24, 4.25, 4.26), n is the total number of queries on the ground-truth, n_s the number of queries for source s, and p the number of inducing locations in $\widehat{D}$. Unlike the sequences generated by Equations (4.24) and (4.25), which are monotonic in minimization (or maximization) problems, the sequence defined by Equation (4.26) may be non-monotonic due to the dynamic update of the inducing locations. The complete MISO-AGP algorithm is summarized in Algorithm 2.

Algorithm 2: MISO-AGP algorithm

set MISO-AGP's parameter α
$C \leftarrow$ maximum cost
$N \leftarrow$ maximum number of function evaluations
$n \leftarrow 0$
$c \leftarrow 0$
while $c < C$ ***AND*** $n < N$ **do**
  # updating GPs on all the sources
  for $s = 1, ..., S$ **do**
    $\mu_s(x), \sigma_s(x)$ learned from D_s
  end
  # generating the augmented GP
  $\widehat{D} \leftarrow D_1 \cup \bar{D}$, with $\bar{D}$ as defined in 4.22
  $\widehat{\mu}(x), \widehat{\sigma}(x)$ learned from $\widehat{D}$)
  # computing the augmented best seen
  $\widehat{y}^+ \leftarrow \min_i \widehat{y}_i$, with $\widehat{y}_i : (\widehat{\mathbf{x}}_i, \widehat{y}_i) \in \widehat{D}$
  # selecting the next source-point to query
  $(x^{(n+1)}, s^{(n+1)}) \in \underset{x \in \mathscr{X}, s \in \mathscr{S}}{\arg\max} \frac{\widehat{y}^+ - \left[\widehat{\mu}(x) - \xi \widehat{\sigma}(x)\right]}{c_s\left(1+\eta(\widehat{\mathscr{G}}, \mathscr{G}_s, x)\right)}$
  # query at $\left(s^{(n+1}, x^{(n+1)}\right)$ *and observe* $y^{(n+1)}$
  $y^{(n+1)} \leftarrow f_{s^{(n+1)}}\left(x^{(n+1)}\right)$
  # updating the dataset associated to $s^{(n+1)}$
  $D_{s^{(n+1)}} \leftarrow D_{s^{(n+1)}} \cup \left\{\left(x^{(n+1)}, y^{(n+1)}\right)\right\}$
  # updating cost and function evaluations
  $n \leftarrow n+1$
  $c \leftarrow c + c_{s^{(n+1)}}$
end
 # providing the best solution observed on the ground-truth
Result: $(x^+, y^+) \in D_1 : y^+ = \min_{i=1:|D_1|} \{y_i\}$

4.2.3 Optimizing Combinatorial Acquisition Functions

In contrast to standard MISO problems, which typically involve box-bounded continuous search spaces, many real-world applications require solving combinatorial optimization problems. In such cases, solutions lie in discrete domains.

As discussed in [64], several strategies allow the handling of discrete or integer variables without altering the underlying Bayesian Optimization (BO) framework. A common approach treats the discrete components as continuous during modeling, while restricting the acquisition function optimization to feasible discrete configurations only. This enables the use of standard surrogate models (e.g., AGPs) and acquisition functions without modification. The only adaptation required is in the optimization strategy for the acquisition function, since traditional gradient-based methods are not applicable in discrete domains.

When additional constraints are present (e.g., cardinality limits or domain-specific restrictions), the problem becomes a constrained combinatorial optimization task. In such contexts, derivative-free algorithms such as Genetic Algorithms (GAs) have been widely adopted. GAs allow the integration of problem-specific heuristics, making them suitable for optimizing acquisition functions over discrete, constrained search spaces.

In the proposed framework, the Pymoo implementation of GA [12] is employed to perform acquisition function optimization. This design maintains feasibility across generations and facilitates efficient exploration of the discrete solution space.

4.2.4 MISO-AGP in BoTorch

The MISO-AGP approach was initially developed in `R`, as introduced in [18]. A subsequent Python implementation has been released in the well-known Bayesian optimization Python library `BoTorch` [7].

It is important to note that the term *augmented* had already been used in `BoTorch` in a distinct context. Specifically, it referred to test functions whose search space dimensionality was *augmented* by including an additional fidelity dimension. This usage is unrelated to the AGP model considered here.

The Model

The `SingleTaskAugmentedGP` model extends the standard `SingleTaskGP` from `BoTorch` by incorporating data from multiple sources within the AGP framework (see Section 4.1). The model is constructed using the modular components of `BoTorch` and GPyTorch, enabling efficient development of custom Gaussian Process (GP) models.

The core functionality of `SingleTaskAugmentedGP` consists of enhancing the predictive performance of a GP model trained on high-fidelity data by augmenting it with selected observations from lower-fidelity sources. The selection is based on a discrepancy criterion that compares predictive means and uncertainties across fidelity-specific GPs. Only the most reliable low-fidelity observations are used for augmentation. The training process involves the following steps:

1. A `SingleTaskGP` is independently fitted to each information source.
2. A discrepancy function is evaluated between each source and the high-fidelity GP to identify reliable samples.
3. Selected low-fidelity points are merged with high-fidelity observations to construct the augmented training set.
4. A final GP is trained on the augmented dataset.

The implementation supports automatic hyperparameter tuning via marginal log-likelihood maximization using the `fit_gpytorch_mll` function. Thanks to GPyTorch's scalable variational inference and `BoTorch`'s differentiable programming interface, the model remains efficient and scalable. Moreover, compatibility with `BoTorch`'s infrastructure allows easy customization of mean and kernel modules.

The Acquisition Function

Although the discussion in this work refers to minimization problems, `BoTorch` follows a maximization convention. This does not pose conceptual issues, since minimization of a function $f(x)$ over $\mathscr{X}$ is equivalent to the maximization of its negation, i.e.,

$$\min_{x \in \mathscr{X}} f(x) \equiv \max_{x \in \mathscr{X}} \left(-f(x)\right).$$

As a consequence, in the `BoTorch` implementation, the acquisition function is expressed in maximization form, and observed function values are stored as $y^{(i)} = -f(x^{(i)})$, with no loss of generality.

The `AugmentedUpperConfidenceBound` (AUCB) is an acquisition function specifically designed for use with the AGP model. It extends `BoTorch`'s standard `UpperConfidenceBound` by accounting for multi-source settings. In particular, AUCB penalizes the acquisition score based on both the cost of querying a source and the discrepancy between the AGP model and the source-specific GP model.

At each iteration, the next evaluation point and source are selected as:

$$(x^{(n+1)}, s^{(n+1)}) \in \underset{x \in \mathscr{X},\, s \in \mathscr{S}}{\arg\max} \frac{[\widehat{\mu}(x) + \xi\, \widehat{\sigma}(x)] - \widehat{y}^{+}}{c_s \left(1 + \eta(\widehat{\mathscr{G}}, \mathscr{G}_s, x)\right)}, \tag{4.27}$$

where:

- $\widehat{\mu}(x)$ and $\widehat{\sigma}(x)$ are the posterior mean and standard deviation estimated by the AGP model.
- $\widehat{y}^{+}$ is the current best value (in the maximization setting).
- c_s is the cost associated with querying source s.
- $\eta(\widehat{\mathscr{G}}, \mathscr{G}_s, x)$ is the discrepancy between the AGP and the source-specific GP $\mathscr{G}_s$.
- $\xi > 0$ is the exploration-exploitation trade-off parameter.

To align with the minimization framework adopted in the previous sections, note that:

$$\widehat{\mu}(x) \simeq -f(x), \quad \widehat{y}^{+} \simeq -\min f(x),$$

and thus the numerator in (4.27) becomes:

$$[-\widehat{\mu}(x) + \xi\, \widehat{\sigma}(x)] + \widehat{y}^{+} = \widehat{y}^{+} - (\widehat{\mu}(x) - \xi\, \widehat{\sigma}(x)),$$

which corresponds to the standard form of the Lower Confidence Bound (LCB) acquisition function used for minimization. The denominator remains unaffected by the problem's direction.

Consequently, the `BoTorch` implementation is fully consistent with the theoretical framework of MISO-AGP as presented in the earlier sections.

4.3 Experiments

The proposed MISO-AGP algorithm has been evaluated on a set of widely used benchmark functions (Section 4.3.1) as well as on a real-world applications: binary quadratic programming (Section 4.3.2) and optimal sensor placement (Section 4.3.3). This section reports the experimental setup and the corresponding results. All code and data used in the experiments are publicly available on GitHub[1,2].

[1] https://github.com/andreaponti5/miso-bocs
[2] https://github.com/andreaponti5/miso-agp

4.3.1 Benchmark Functions

Problem Description

To evaluate the performance of the proposed multi-information source optimization algorithm, a set of benchmark problems was considered. These include both functions natively available in the `BoTorch` library and additional test functions commonly used in the literature. The chosen problems exhibit diverse dimensionalities, landscapes, and fidelity structures, thus providing a comprehensive testbed for assessing algorithmic performance.

The first problem is the **Branin** function [163], a two-dimensional benchmark widely used in optimization. It is characterized by three global minima located at

$$x^* = (-\pi, 12.275), (\pi, 2.275), (9.42478, 2.475),$$

each attaining a function value of $f(x^*) = 0.397887$. The multi-fidelity extension of this function is obtained by introducing a fidelity parameter s that modifies the quadratic term, as shown in Equation 4.28. Figure 4.3 illustrates the behavior of the function across the three information sources.

$$\begin{aligned} f(x,s) = \left(x_2 - \left(\frac{5.1}{4\pi^2} - 0.1\,(1-s)\right)x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 \\ +10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10 \end{aligned} \tag{4.28}$$

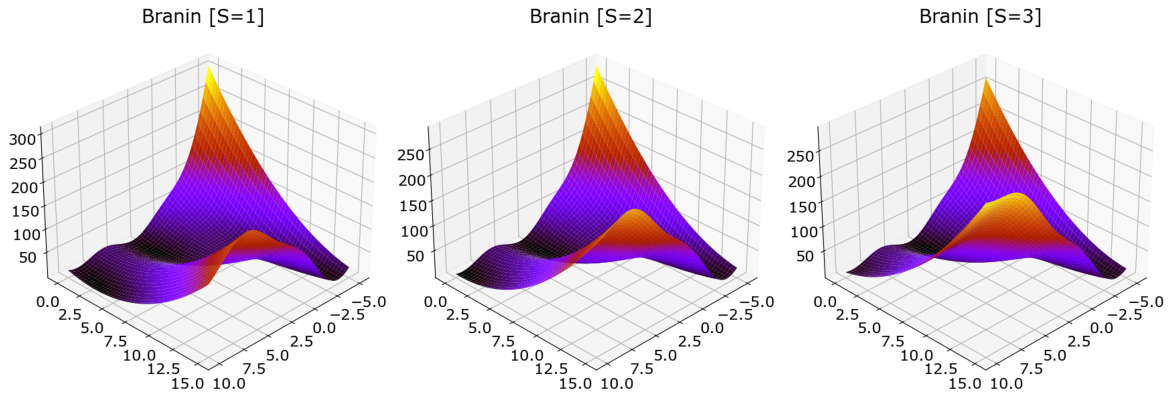

Fig. 4.3 The three considered fidelities of the Branin test function.

The second `BoTorch`-native benchmark is the six-dimensional **Hartmann** function [163]. This problem has a unique global minimum located at

$$x^* = (0.20169, 0.150011, 0.476874, 0.275332, 0.311652, 0.6573)$$

with an associated function value of $f(x^*) = -3.32237$. Similar to the Branin function, its multi-fidelity structure is introduced by perturbing one of the coefficients in the exponential terms, as defined in Equation 4.29.

$$\begin{aligned} f(x,s) = &-(\alpha_1 - 0.1\,(1-s)) \exp\left(-\sum_{j=1}^{d} A_{1j}\left(x_j - P_{1j}\right)^2\right) \\ &+ \sum_{i=2}^{4} \alpha_i \exp\left(-\sum_{j=1}^{d} A_{ij}\left(x_j - P_{ij}\right)^2\right). \end{aligned} \tag{4.29}$$

Beyond the `BoTorch` benchmarks, additional test functions were selected to further challenge the algorithms. The **Forrester** function, a one-dimensional benchmark defined on $[0,1]$, is a common choice in multi-fidelity optimization. Its global minimum is located at $x^* = 0.75724876$, where the function value is $f(x^*) = 6.020740$. In this case, three sources are available: the ground-truth function, together with two lower-fidelity variants that provide biased approximations (Equation 4.30). Figure 4.4 depicts the three information sources.

$$\begin{aligned} f(x,1) &= (6x-2)^2 \sin(12x-4), \\ f(x,2) &= 0.5 f(x,1) + 10(x-0.5) + 5, \\ f(x,3) &= 0.5 f(x,1) + 10(x-0.5) - 5. \end{aligned} \tag{4.30}$$

Another widely studied test function is **Rosenbrock**'s function [96], which presents a global minimum inside a long, narrow, parabolic-shaped valley. The function is unimodal, with the minimum located at $x^* = (1, \ldots, 1)$ and $f(x^*) = 0$. While the valley is relatively easy to locate, convergence to the exact minimum is notoriously difficult. Multi-fidelity variants of Rosenbrock have been proposed in [16, 151], where medium- and low-fidelity approximations are constructed to model reduced accuracy evaluations. These lower-fidelity sources are described in Equation (4.31) and illustrated in Figure 4.5.

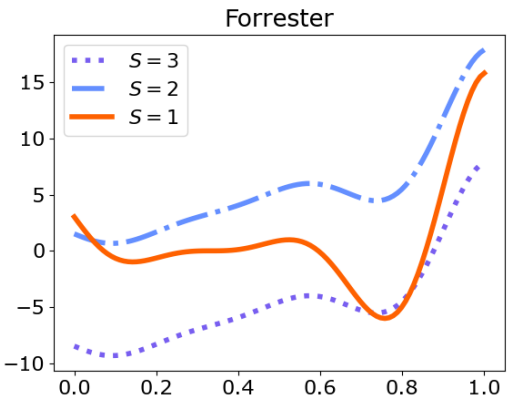

Fig. 4.4 The three considered sources of the Forrester test function.

$$
\begin{aligned}
f(x,1) &= \sum_{i=1}^{d-1} 100\left(x_{i+1}-x_i^2\right)^2 + (1-x_i)^2, \\
f(x,2) &= \sum_{i=1}^{d-1} 50\left(x_{i+1}-x_i^2\right)^2 + (-2-x_i)^2 - \sum_{i=1}^{d} 0.5x_i, \\
f(x,3) &= \frac{f_1(x) - 4 - \sum_{i=1}^{d} 0.5x_i}{10 + \sum_{i=1}^{d} 0.25x_i}.
\end{aligned}
\tag{4.31}
$$

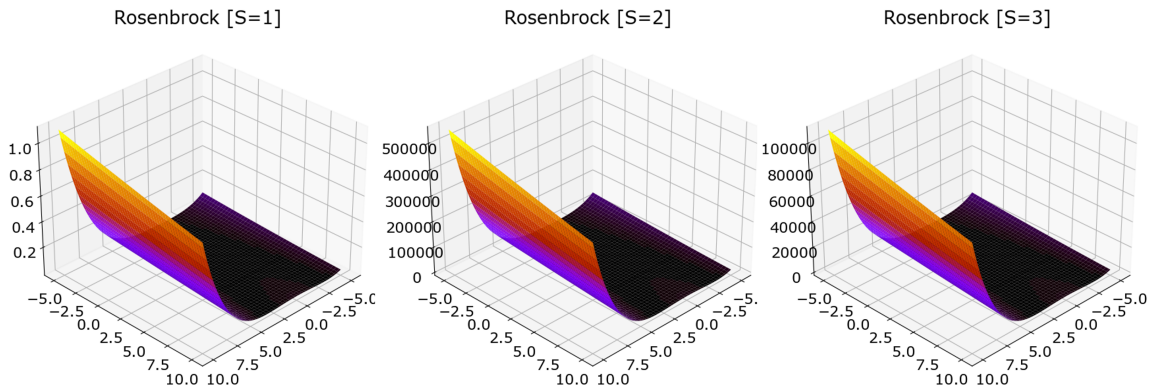

Fig. 4.5 The three sources considered for the Rosenbrock test function.

Taken together, these benchmark problems provide a diverse set of test cases, ranging from low-dimensional functions with multiple global minima to high-dimensional landscapes with narrow valleys and complex fidelity structures. This variety ensures that the evaluation

of multi-fidelity optimization algorithms captures both the opportunities and challenges inherent in real-world applications.

Experimental Settings

Each experiment began with an initial design consisting of $d+1$ observations, generated using Latin Hypercube Sampling to ensure a well-spread coverage of the search space. After initialization, $20(d+1)$ sequential queries were performed to iteratively refine the solutions. To obtain statistically reliable results, 30 independent runs were conducted for MISO-AGP, MF-MES, and MF-GIBBON, while the MF-KG method was executed only once due to its significantly higher computational cost. For fairness, all algorithms shared the same initial random design in each run. It is worth noting that, with the exception of MISO-AGP, fidelity was modeled both as a continuous and as a discrete variable in separate experiments.

Each function was evaluated under different dimensional settings and fidelity configurations, as summarized in Table 4.1. The Branin function was tested in two dimensions with two- and three-fidelity variants. The Hartmann function was evaluated in six dimensions, again under two- and three-fidelity settings. Finally, the Rosenbrock function was examined in both two and ten dimensions, with three fidelities in each case.

Table 4.1 Benchmark functions and experimental configurations.

Function	**Search space**	**Dimensions (d)**	**Fidelities**
Forrester	$[(0,1)]$	1	{0.5, 0.75, 1.0}
Branin	$[(-5,10),(0,15)]$	2	{0.5, 1.0}, {0.5, 0.75, 1.0}
Hartmann	$[(0,1)]^d$	6	{0.5, 1.0}, {0.5, 0.75, 1.0}
Rosenbrock	$[(-2,2)]^d$	2, 10	{0.5, 0.75, 1.0}

An important distinction must be made between MISO-AGP and the `BoTorch`-based methods (MF-MES, MF-GIBBON, MF-KG). MISO-AGP operates directly on the original d-dimensional search space of the objective function, whereas the `BoTorch` implementations augment the search space by including fidelity as an additional dimension, resulting in a $(d+1)$-dimensional formulation. This design choice reflects different modeling assumptions regarding fidelity and may influence the comparative performance of the algorithms.

Since the fidelity parameter in these experiments was discrete, two alternative strategies were employed for optimizing the acquisition function. In the first approach, fidelity was treated as a continuous variable during the optimization and subsequently rounded to the nearest discrete value. In the second approach, fidelity was explicitly handled as a discrete attribute: the acquisition function was optimized separately for each fidelity level, and

the candidate point corresponding to the highest acquisition value across all fidelities was selected.

Experimental Results

Figure 4.6 reports the evolution of the best observed value (i.e., best seen) as a function of the accumulated query cost, when fidelity is modeled as a continuous variable. Under this setting, the MF-KG consistently achieves the strongest performance across all benchmark problems. MISO-AGP emerges as the closest competitor, maintaining results that are comparable in quality while requiring significantly less computational effort. This difference in wall-clock time highlights a crucial trade-off: although MF-KG is theoretically appealing and empirically powerful, its computational overhead severely limits its practical applicability, especially when moving from synthetic benchmarks to expensive real-world scenarios.

Figure 4.7 presents the results obtained when fidelity is treated as a discrete parameter. In this case, both MF-MES and MF-GIBBON show notable improvements over their continuous-fidelity counterparts, narrowing the performance gap with MISO-AGP. Nevertheless, MISO-AGP retains its position as the second-best performer after MF-KG, and becomes the most attractive option once computational efficiency is taken into account. This finding suggests that the way fidelity is modeled can substantially impact the effectiveness of certain acquisition functions, particularly for entropy-based methods such as MF-MES and MF-GIBBON.

Overall, the experimental evidence highlights two main insights. First, MF-KG is largely insensitive to how fidelity is represented: it consistently delivers the most cost-effective solutions in terms of query cost. However, this advantage comes at the price of excessive wall-clock time, which makes the method impractical for many real-life applications involving costly evaluations. Second, MISO-AGP achieves a favorable balance between performance and efficiency. While slightly less effective than MF-KG in terms of query cost, its computational requirements remain manageable, making it the most practical alternative among the tested approaches. This result is particularly important in view of applying Multi-Fidelity Bayesian optimization to real-world scientific and engineering problems, where both query cost and wall-clock time are critical factors.

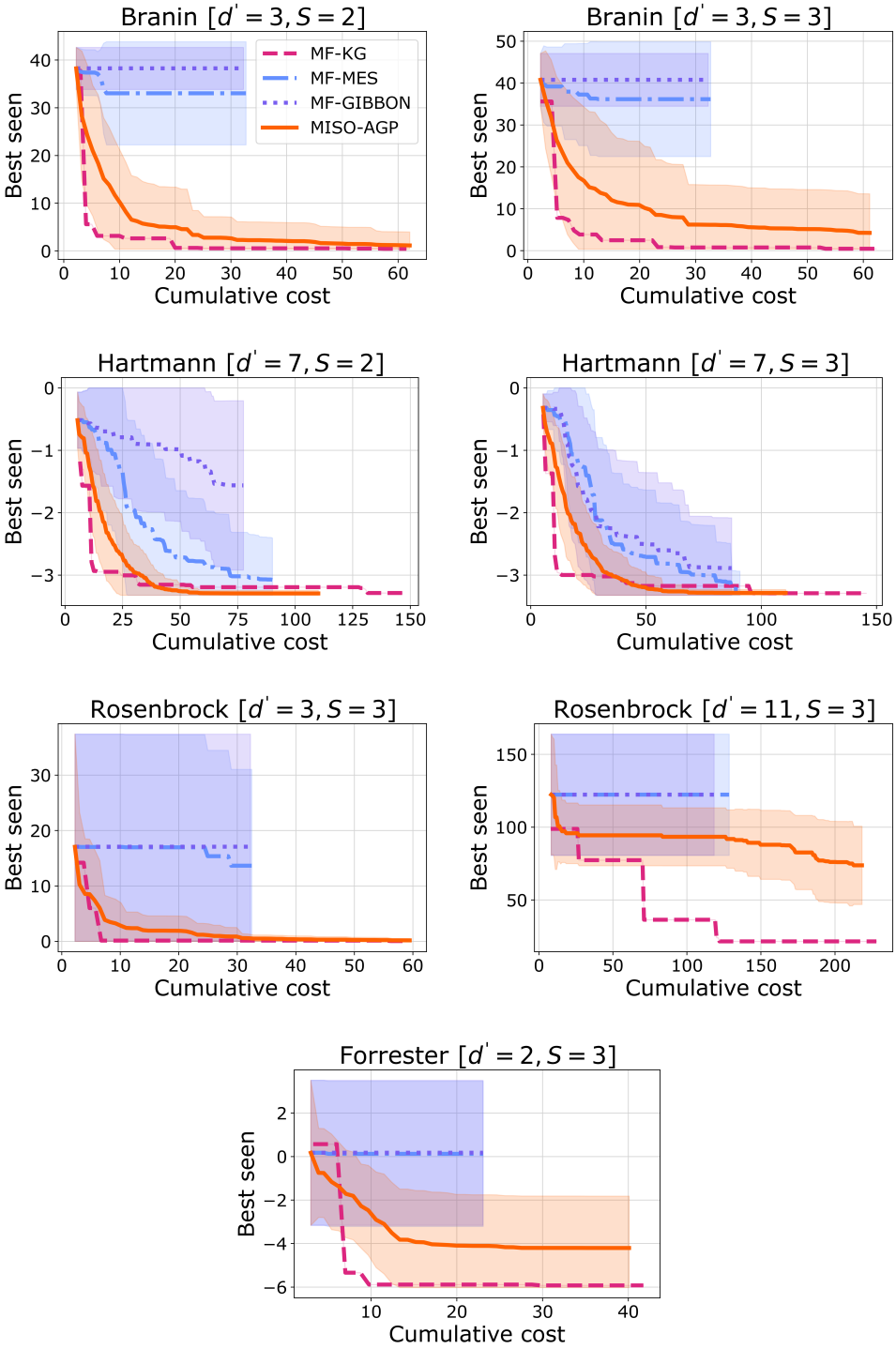

Fig. 4.6 MISO-AGP against multi-fidelity approaches (which treat **fidelity as a continuous variable**). Contrary to MISO-AGP, the three multi-fidelity approaches work on a space having $d' = d + 1$ dimensions, with d the dimensionality of the original search space.

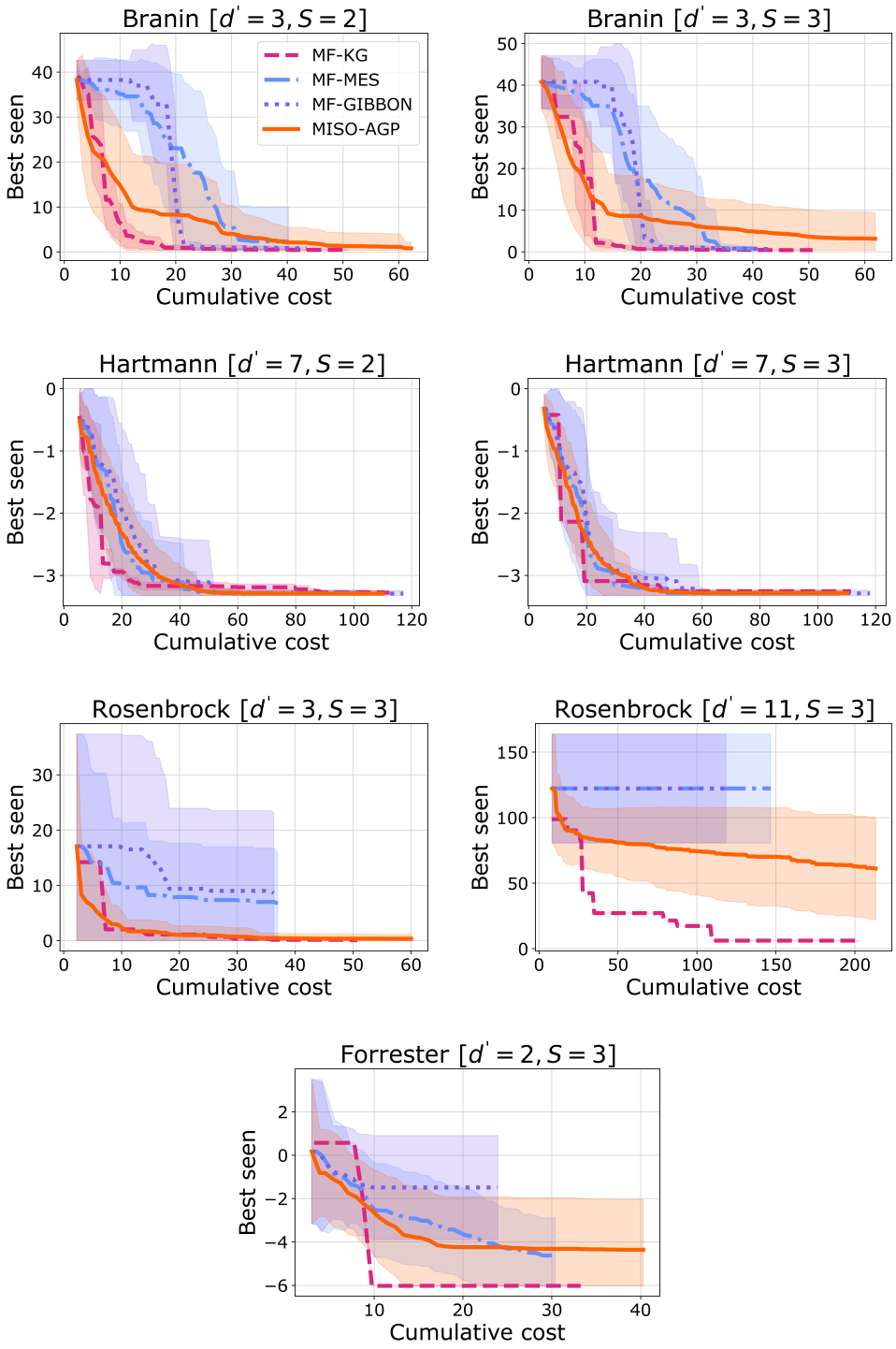

Fig. 4.7 MISO-AGP against multi-fidelity approaches (which treat **fidelity as a discrete variable**). Contrary to MISO-AGP, the three multi-fidelity approaches work on a space having $d' = d + 1$ dimensions, with d the dimensionality of the original search space.

4.3.2 Binary Quadratic Programming

The results presented in this section were originally published in [129].

Problem Description

The Binary Quadratic Programming (BQP) [8] problem consists of maximizing a quadratic function with ℓ_1-regularization, defined as

$$f(x) - \lambda P(x) = x^\top Q x - \lambda \|x\|_1, \quad x \in \{0,1\}^d.$$

Here, $Q \in \mathbb{R}^{d \times d}$ is a random matrix with independent standard Gaussian entries, which is multiplied element-wise by a correlation matrix $K \in \mathbb{R}^{d \times d}$, where each entry is defined as $K_{ij} = \exp\big(-(i-j)^2/L_c^2\big)$. The correlation length L_c controls the decay of correlations: for small values of L_c, Q is nearly diagonal, whereas larger values of L_c produce denser matrices, thereby increasing the complexity of the optimization task. The BQP problem was extended to the multi-information source optimization (MISO) setting by associating different sources with different numbers of realizations of Q. Two scenarios were considered:

1. A ground-truth source based on 50 realizations and a cheaper source based on 25 realizations (50% of the computational cost).

2. A ground-truth source based on 50 realizations and a cheaper source based on 5 realizations (10% of the computational cost).

In both scenarios, each algorithm was run 10 times on each instance for every realization of Q. Experiments were conducted for two parameter settings: $(L_c = 10, \lambda = 0)$ and $(L_c = 100, \lambda = 1)$ with fixed dimensionaly $d = 10$ as suggested in [8]. Figure 4.8 shows the distributions of BQP values for the different sources.

Experimental Results

The performance of MISO-AGP, MF-MES, and MF-GIBBON was evaluated across the four configurations of the Binary Quadratic Programming problem, obtained by varying the regularization parameter λ, the correlation length L_c, and the relative cost (and fidelity) of the cheap source.

Case 1: $\lambda = 0$, $L_c = 10$, cheap source at 50% of the ground truth. As shown in Figure 4.9, MISO-AGP achieves, on average, both a lower best-seen value (left) and a smaller accumulated query cost (right) compared to MF-MES and MF-GIBBON. Although the final

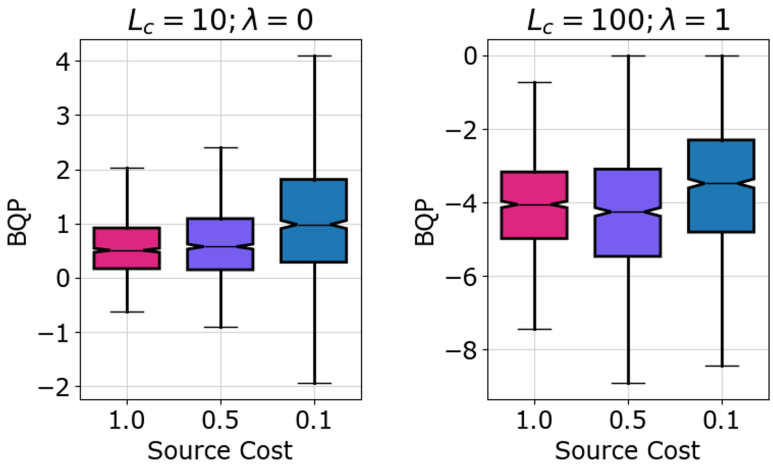

Fig. 4.8 Distribution of BQP values across the three information sources, for each considered setting.

best-seen of MISO-AGP was lower, the difference was not statistically significant according to the Wilcoxon test (p-value > 0.05). Nevertheless, MISO-AGP proved significantly more efficient in terms of cumulative cost. In this setting, MISO-AGP queried the ground truth in 79% of the total evaluations, while MF-MES and MF-GIBBON relied on it only 25% and 20% of the time, respectively. This behavior reflects the AGP's ability to detect discrepancies between sources and to discard cheap observations when the correlation with the ground truth is weak.

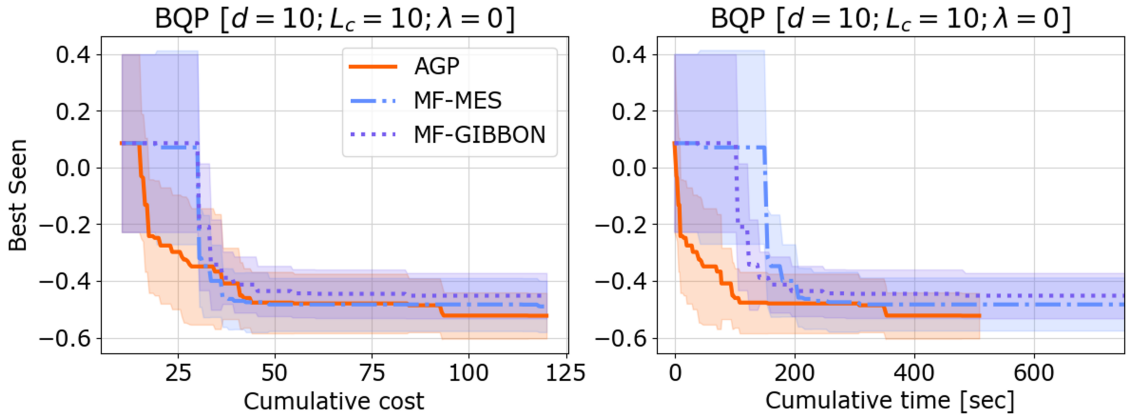

Fig. 4.9 The best seen of the BQP problems given by the tested algorithms over the cumulated query cost (left) and the wall-clock time (right). The figure refers to the case of the cheap source at 50% of the ground truth

Case 2: $\lambda = 0$, $L_c = 10$**, cheap source at 10% of the ground truth.** In this case (Figure 4.10), MISO-AGP again outperforms the competitors in terms of both best-seen and accumulated query cost. The algorithm relied on the ground truth in 91% of the total queries, compared to 29% for MF-MES and 20% for MF-GIBBON. Interestingly, both MISO-AGP and MF-MES increased their reliance on the ground truth even though the relative cost of the cheap source decreased from 50% to 10%. This behavior indicates that, although the algorithms initially exploit the cheap source due to its low cost, they rapidly identify its poor correlation with the ground truth and subsequently favor the expensive but more reliable source.

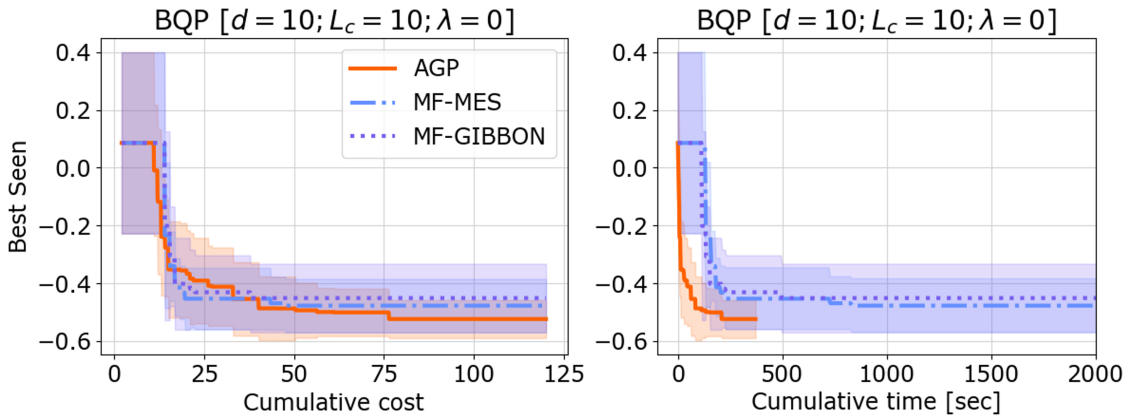

Fig. 4.10 The best seen of the BQP problems given by the tested algorithms over the cumulated query cost (left) and the wall-clock time (right). The figure refers to the case of the cheap source at 10% of the ground truth

Case 3: $\lambda = 1$, $L_c = 100$**, cheap source at 50% of the ground truth.** Under this more challenging configuration (Figure 4.11), MISO-AGP shows worse results than MF-MES and MF-GIBBON in terms of final best-seen, with statistically significant differences (Wilcoxon test: against MF-MES, $p = 0.0143$; against MF-GIBBON, $p = 0.0141$). However, MISO-AGP still maintained a lower cumulative runtime, confirming its computational efficiency. Despite the performance gap, MISO-AGP queried the ground truth in 83% of the total evaluations, compared to 31% for MF-MES and 20% for MF-GIBBON. This again highlights the AGP's tendency to rely on the expensive source when the cheap source provides poor local approximations.

Case 4: $\lambda = 1$, $L_c = 100$**, cheap source at 10% of the ground truth.** Finally, in this scenario (Figure 4.12), MISO-AGP regains its advantage, achieving on average both a lower

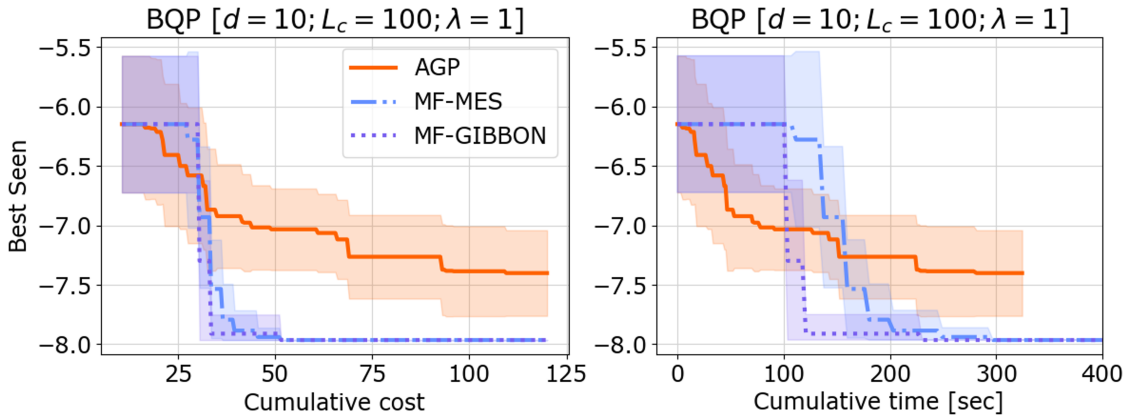

Fig. 4.11 The best seen of the BQP problems given by the tested algorithms over the cumulated query cost (left) and the wall-clock time (right). The figure refers to the case of the cheap source at 50% of the ground truth

best-seen value and a smaller accumulated query cost. The final best-seen of MISO-AGP was significantly better than those obtained by MF-MES and MF-GIBBON (Wilcoxon test). Here, MISO-AGP relied on the ground truth in 87% of the total queries, slightly more than in the previous case, while MF-MES and MF-GIBBON used it 33% and 20% of the time, respectively. The explanation is consistent with earlier cases: even when the cheap source is inexpensive, the AGP identifies its weak correlation with the ground truth and prioritizes the expensive source to guide the search more reliably.

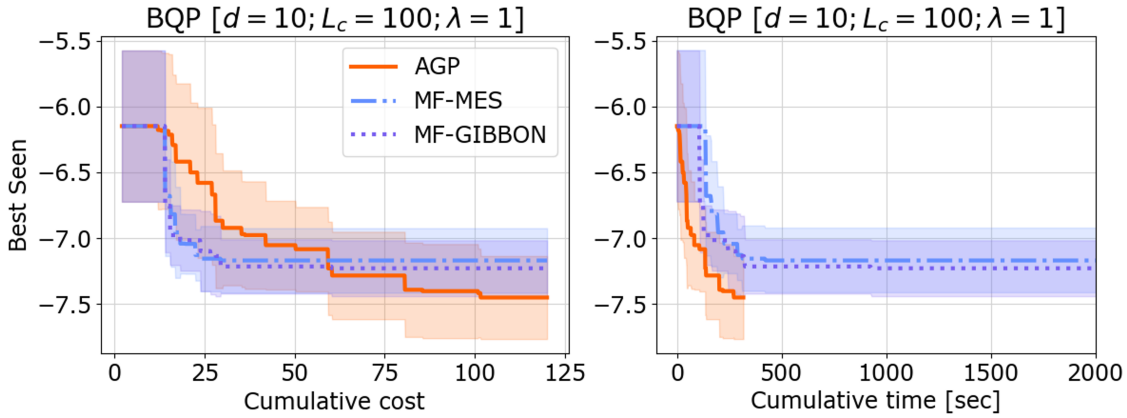

Fig. 4.12 The best seen of the BQP problems given by the tested algorithms over the cumulated query cost (left) and the wall-clock time (right). The figure refers to the case of the cheap source at 10% of the ground truth

Overall, the experiments highlight the robustness of MISO-AGP. While its performance can degrade under certain conditions (large L_c and $\lambda = 1$), it consistently demonstrates superior efficiency and a principled mechanism for balancing multiple sources of information, discarding unreliable ones when necessary.

4.3.3 Risk-Averse Optimal Sensor Placement

The results presented in this section were originally published in [129].

Problem Description

The optimal sensor placement (OSP) problem can be adapted to a risk-averse formulation, where the objective is to minimize the impact of contamination events under worst-case or tail-risk considerations. Unlike the multi-objective setting introduced in Chapter 3, where aggregated statistics such as the average and standard deviation of detection time or contaminated water volume were considered, the focus here is on the Conditional Value-at-Risk (CVaR) of detection times. CVaR is a widely used risk measure that captures the expected value of the worst $\alpha\%$ outcomes of a random variable, thus penalizing solutions that perform poorly in extreme scenarios.

As before, a water distribution network (WDN) is represented as a graph $G = (V, E)$, where nodes correspond to junctions, reservoirs, or consumption points, and edges represent pipes and other hydraulic components. A sensor placement is encoded as a binary vector $x \in \{0,1\}^{|L|}$, with $L \subseteq V$ the set of candidate locations. The budget constraint enforces that at most b sensors can be installed, i.e., $\sum_{i=1}^{|L|} x_i \leq b$. Given a placement x and a set of contamination events $A \subseteq V$, the detection time for each event $a \in A$ is defined as the earliest time at which any installed sensor detects the contaminant. The distribution of detection times over all scenarios in A is then used to compute CVaR, which serves as the impact measure to minimize.

In the multi-information source setting, the stochasticity of the problem is exploited through the use of different sets of contamination scenarios with varying costs. Let $A_1 = V$ denote the complete set of contamination events, representing the high-fidelity (ground-truth) source, and $A_2 \subset A_1$ a reduced subset of scenarios, representing a cheaper but less accurate source. In the following, $|A_2| = |A_1|/2$, so that evaluating a candidate placement on A_2 incurs approximately half the computational cost of evaluating it on A_1. Accordingly, the

optimization problem can be formulated as:

$$\begin{aligned} x^* = & \underset{x \in \{0,1\}^{|L|}}{\arg\min}\ \mathrm{CVaR}(x \mid A_1) \\ & \text{s.t.} \sum_{i=1}^{|L|} x_i \leq b, \end{aligned} \tag{4.32}$$

where $\mathrm{CVaR}(x \mid A_1)$ denotes the CVaR of detection times computed from the full scenario set A_1.

Since direct optimization with A_1 is computationally demanding, the MISO framework enables a sequential strategy in which solutions are evaluated alternately using the expensive source (A_1) and the cheap source (A_2). Let $\{(s^{(1)},x^{(1)}),\ldots,(s^{(n)},x^{(n)})\}$ denote the sequence of queries, where $x^{(j)}$ is the j-th candidate solution and $s^{(j)} \in \{1,2\}$ indicates the information source. Querying $s^{(j)} = 1$ corresponds to computing $\mathrm{CVaR}(x^{(j)} \mid A_1)$ at unit cost, while querying $s^{(j)} = 2$ corresponds to computing $\mathrm{CVaR}(x^{(j)} \mid A_2)$ at half cost. The objective is to converge to x^* while minimizing the total accumulated query cost.

This formulation highlights how the OSP problem can naturally be embedded into the MISO setting, where trade-offs between evaluation accuracy and cost must be explicitly balanced.

Experimental Settings

To account for the stochasticity, each of the three algorithms was executed over five independent runs. Within each run, the algorithms shared the same initial random solutions to ensure a fair comparison.

The maximum number of deployable sensors b was set according to the size of the network, with smaller budgets for the benchmark cases and larger budgets for real-world networks: $b = 4$ for Anytown and Hanoi, $b = 10$ for Neptun, and $b = 15$ for Apulian5.

The experiments were conducted on the four water distribution networks already introduced in Chapter 3, namely the two synthetic benchmarks (Anytown and Hanoi) and the two real-world systems (Neptun and Apulian5). Contamination events were simulated using WNTR [84], a Python wrapper of EPANET [125]. Each simulation spanned a 24-hour period, with contaminant concentrations recorded hourly at each node. Sensors were allowed only at junctions, excluding tanks and reservoirs. In the two small benchmark networks, all nodes were eligible sensor locations, whereas in the larger Neptun and Apulian5 systems the candidate locations L were restricted to subsets of nodes chosen by uniform spatial sampling,

ensuring adequate coverage of the entire network while keeping the dimensionality of the optimization problem manageable.

Figure 4.13 illustrates the four water distribution networks, with the admissible sensor locations highlighted in red. Table 4.2 summarizes the main characteristics of the networks, including the number of nodes, the size of the candidate location set, and the contamination scenarios associated with the two information sources: the high-fidelity source (A_1) corresponding to all possible contamination events, and the cheap source (A_2) obtained by halving the number of scenarios.

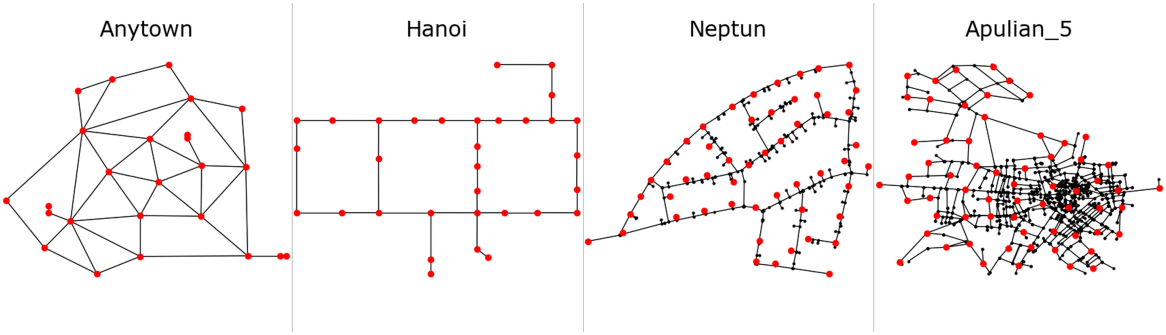

Fig. 4.13 The four water networks considered in the experiments. Red points indicate the candidate sensor locations.

Table 4.2 Main characteristics of the four water distribution networks used in the experiments. The last column reports the number of contamination scenarios for the two information sources (high fidelity $|A_1|$ and cheap fidelity $|A_2|$).

Network	**Nodes**	**Candidate locations**	**Contamination events** $|A_1|/|A_2|$
Anytown	25	25	22 / 11
Hanoi	32	32	31 / 16
Neptun	333	51	332 / 166
Apulian5	1364	63	1364 / 682

An important aspect in the multi-information source setting is the discrepancy between the two sources. The more accurate the approximation provided by the cheap source, the more effective it can be in guiding optimization at a reduced cost. Figure 4.14 reports the distribution of the CVaR of detection times across the two sources, showing close agreement for most networks, with the exception of Anytown, where a noticeable gap is observed.

Differently from standard MISO problems, which are usually defined over box-bounded continuous search spaces, the risk-averse OSP formulation is inherently combinatorial. The search space is given by $\{0,1\}^{|L|} \times \{1,2\}$, where the first $|L|$ dimensions represent the binary sensor placement vector x, and the last dimension indicates the information source used for

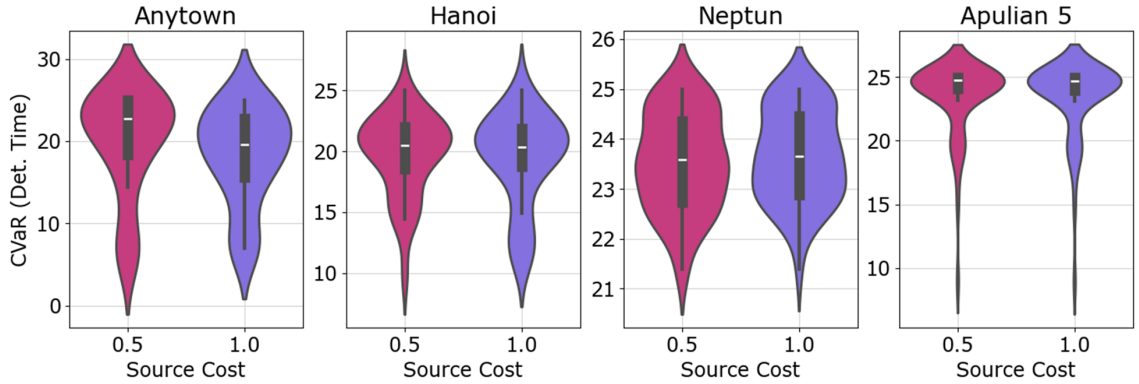

Fig. 4.14 Distribution of CVaR values of detection times (hours) across the two information sources, shown separately for each of the four networks.

evaluating the objective function. An illustrative example of a candidate solution at a generic iteration of the optimization process is reported in Figure 4.15.

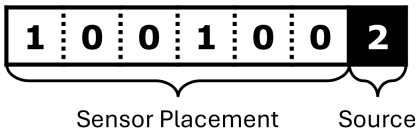

Fig. 4.15 Example of a solution in the MISO framework for the OSP problem. Two sensors are deployed at locations $i = 1$ and $i = 4$, with the CVaR computed using the cheap information source $s = 2$, corresponding to the reduced scenario set A_2.

As discussed in Section 4.2.3, several strategies exist for handling combinatorial or integer variables without altering the general BO framework. In this work, the acquisition function has been optimized through the Genetic Algorithm (GA) available in the `pymoo` library [12]. For mutation, a standard bit-flip operator has been employed with probability $1/|L|$. For crossover, the problem-specific operator previously introduced in [115] and described in Section 3.2.3 has been adopted. This operator ensures that offspring remain feasible whenever the parents are feasible, i.e., the number of deployed sensors is preserved across generations.

Experimental Results

The comparison among the three approaches has been carried out in terms of the best-seen objective values with respect to both query cost and wall-clock time (Figure 4.16). Overall, MISO-AGP and MF-GIBBON exhibit comparable performance trends, while MF-MES

consistently achieves lower performance. An important observation is that MISO-AGP shows a markedly lower variability across independent runs, indicating greater robustness compared to the other methods. This property becomes increasingly relevant as the dimensionality of the problem grows, since the performance gap between MISO-AGP and the other approaches appears to widen in larger networks. This suggests that the proposed method is particularly well suited to handle high-dimensional instances of the OSP problem.

A limitation of MISO-AGP emerges in terms of computational efficiency. In two of the four test cases, namely Hanoi and Neptun, the wall-clock time required by MISO-AGP exceeds that of the competing methods. This additional cost is mainly due to the need to fit two single-fidelity GPs and the AGP at every iteration. However, when considering the lowest cumulative time across the approaches, the best-seen values achieved by MISO-AGP are consistently lower or at least very close to those obtained by the other methods. This indicates that, despite the higher computational burden per iteration, MISO-AGP is able to make more effective use of evaluations.

A possible explanation lies in the different use of the information sources. Both MF-GIBBON and MF-MES tend to overexploit the cheap source, which may lead to suboptimal trade-offs in terms of query efficiency. In contrast, MISO-AGP appears to maintain a more balanced exploration between the cheap and expensive sources, enabling more reliable convergence toward high-quality solutions.

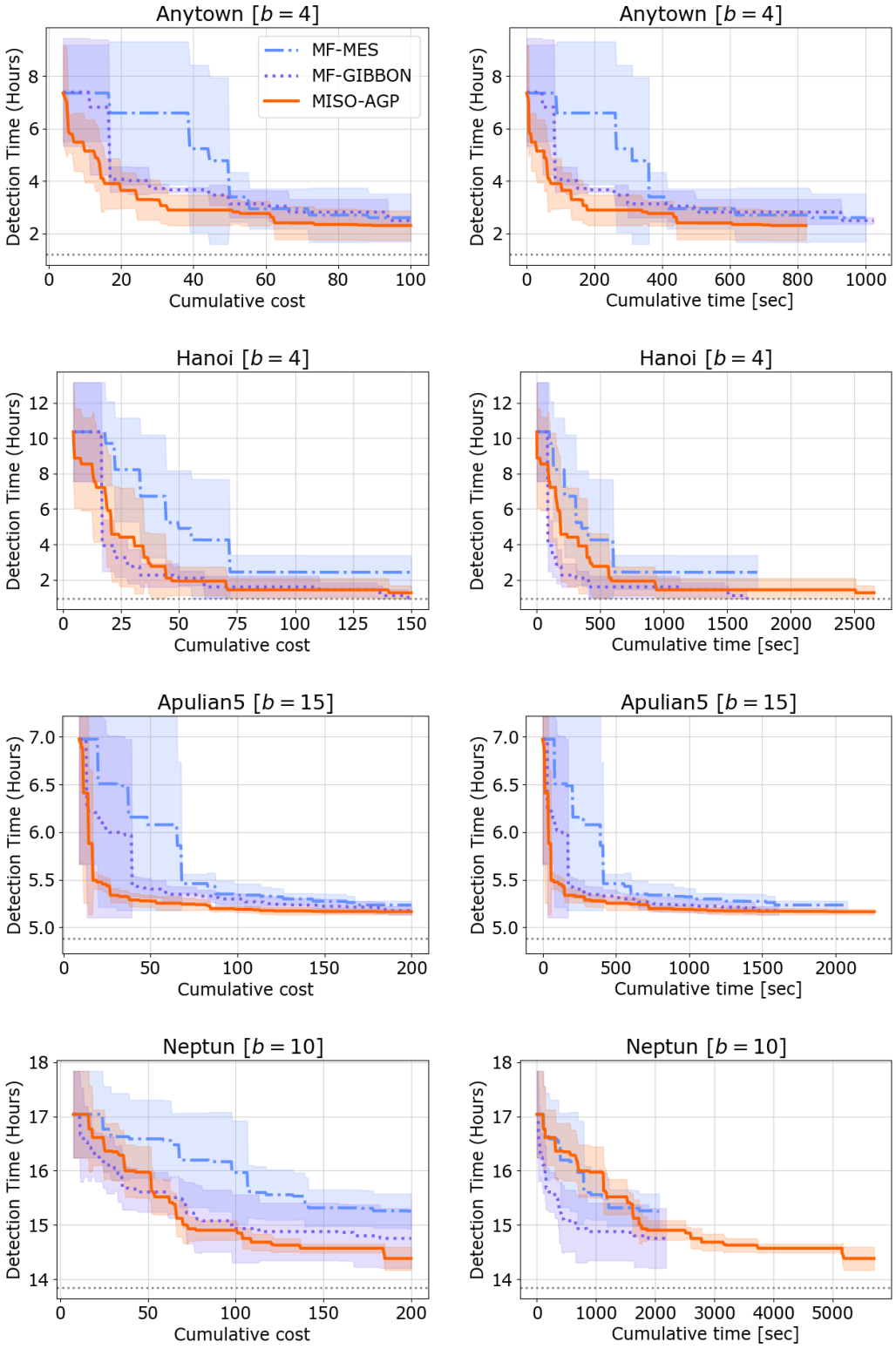

Fig. 4.16 Comparison between the best seen (i.e., the lowest observed CVaR of the detection times) over the cumulative cost (left) and the wall-clock time (right). The shadow represents the standard deviation while the line the mean over 5 independent runs.

Chapter 5

Multi-Objective Multi-Fidelity Optimization

This chapter addresses the integration of multi-objective and multi-fidelity optimization within the context of Bayesian optimization. It begins by introducing the foundations of multi-objective Bayesian optimization and its extension to multi-fidelity settings. The chapter then discusses the concepts of fairness and energy efficiency in machine learning, presenting strategies for fair and green model development. Building on these principles, the thesis contribution is introduced: an extension of the Augmented Gaussian Process framework to handle multi-objective problems. Finally, the chapter illustrates the performance of the proposed approach through experiments on various machine learning algorithms and datasets, evaluating both predictive accuracy and fairness metrics.

5.1 Background

5.1.1 Multi-Objective Bayesian Optimization

Multi-objective optimization addresses problems involving the simultaneous optimization of multiple, typically conflicting, objective functions. Formally, the general MO problem can be expressed as:

$$\min_{x \in \mathcal{X}} f(x) \tag{5.1}$$

where $\mathcal{X} \subset \mathbb{R}^d$ is the search space, typically assumed to be box-bounded (e.g., $\mathcal{X} = [0,1]^d$), and $f : \mathcal{X} \to \mathbb{R}^M$ is a vector-valued objective function, with M denoting the number of objectives.

Due to inherent trade-offs among the objectives, a single optimal solution rarely exists. Instead, the aim is to approximate the Pareto front, i.e., the set of non-dominated objective vectors, along with the corresponding Pareto set in the input space. A solution $x \in \mathscr{X}$ is said to *dominate* another solution $x' \in \mathscr{X}$ if the following two conditions are satisfied:

$$f_m(x) \leq f_m(x') \quad \forall m \in \{1,\dots,M\} \tag{5.2}$$

$$\exists j \in \{1,\dots,M\} : \; f_j(x) < f_j(x') \tag{5.3}$$

This relation is usually written as $f(x) \prec f(x')$.

When the objective functions f are expensive-to-evaluate black-box functions, computing the full Pareto front becomes intractable. In such contexts, Multi-Objective Bayesian Optimization (MOBO) provides a sample-efficient approach by modeling the objectives using probabilistic surrogate models (typically Gaussian Processes), and by employing acquisition functions to guide the search toward informative and diverse solutions.

The design of acquisition functions in MOBO must account for multiple objectives. Several strategies have been proposed in literature, which can be broadly classified into the following categories:

- Scalarization-based methods: These approaches transform the vector-valued objective function $f(x)$ into a scalar function via a parameterized aggregation function. Each scalarized instance can be solved using standard single-objective BO. By varying the scalarization parameters, different regions of the Pareto front can be explored [107, 167]. However, scalarization may fail to capture the geometry of the Pareto front, especially in non-convex cases.

- Hypervolume-based method: These methods directly aim at maximizing the dominated hypervolume, i.e., the volume in objective space covered by the Pareto front approximation with respect to a reference point. The corresponding acquisition function, called Expected Hypervolume Improvement (EHVI), quantifies the expected increase in hypervolume resulting from evaluating a new point. EHVI has become a standard choice in MOBO due to its ability to balance convergence and diversity [56, 164, 41].

- Information-theoretic methods: These approaches aim to reduce the uncertainty over the Pareto set or front. Acquisition functions are derived from mutual information or entropy reduction criteria, and tend to favor regions of the input space that are expected to yield significant information gain about the optimal front [9, 10, 144].

Among the most established MOBO methods, the following two are widely adopted and representative of the scalarization and hypervolume-based paradigms:

Pareto Efficient Global Optimization (ParEGO): ParEGO [85] is a scalarization-based approach that models a randomly weighted scalarization of the objectives. Specifically, at each iteration, a weight vector $\lambda \in \mathbb{R}^M$ is sampled from the unit simplex, and the objective function is scalarized using a function such as the augmented Tchebycheff norm:

$$f_\lambda(x) = \max_{m=1,\dots,M} \lambda_m |f_m(x) - z_m^*| + \rho \sum_{m=1}^{M} \lambda_m |f_m(x) - z_m^*| \tag{5.4}$$

where z^* is a reference (typically ideal) point and ρ is a small positive scalar that ensures sufficient exploration. This scalarized objective is modeled via a standard GP, and optimized using any single-objective acquisition function (e.g., Expected Improvement). By changing the weights over iterations, ParEGO approximates the full Pareto front.

Expected Hypervolume Improvement (EHVI): EHVI extends the Expected Improvement acquisition function to the multi-objective case by measuring the expected increase in dominated hypervolume:

$$\text{EHVI}(x) = \mathbb{E}\left[\Delta\mathscr{H}\big(f(x)\big)\right] \tag{5.5}$$

where $\Delta\mathscr{H}$ denotes the improvement in hypervolume when the predicted outcome $f(x)$ is added to the current Pareto front approximation. The expectation is taken with respect to the predictive posterior distribution provided by the GP models. EHVI can be computed analytically in low-dimensional cases (e.g., $M \leq 3$), while various approximation schemes exist for higher-dimensional problems [164].

Both ParEGO and EHVI are fully compatible with the Bayesian optimization framework and have demonstrated strong performance across a variety of multi-objective benchmarks. The choice between scalarization and hypervolume-based methods typically depends on the characteristics of the Pareto front and computational considerations.

5.1.2 Multi-Objective Multi-Fidelity Bayesian Optimization

In many real-world decision-making problems, multiple conflicting objectives must be optimized simultaneously. When each objective evaluation is costly - as is often the case in scientific experiments, engineering simulations, or real-world deployments - Bayesian Optimization offers a principled approach to sample-efficient search. However, in such settings, it is also common for multiple fidelity levels of the objectives to be available. These fidelities may represent, for instance, coarser simulations, subsampled data, or simplified models that provide cheaper but less accurate evaluations.

Multi-Fidelity Bayesian Optimization leverages these heterogeneous sources to accelerate optimization by selectively querying cheaper approximations when appropriate. When combined with the challenges of multi-objective optimization, where the goal is to approximate the entire Pareto front of trade-off solutions, the result is the *Multi-Objective Multi-Fidelity Bayesian Optimization* (MOMFBO) problem. In this setting, the optimizer must simultaneously manage:

- Trade-offs between competing objectives;
- Trade-offs between cost and accuracy across fidelity levels;
- Efficient approximation of the Pareto front within a limited budget.

A typical MOMFBO algorithm consists of three main components: *(i)* a surrogate model that jointly captures the behavior of all objectives across all fidelities (e.g., using multi-output or hierarchical Gaussian Processes), *(ii)* an acquisition function that selects the next query point and fidelity level, and *(iii)* a fidelity-aware optimization strategy that balances cost and informativeness.

Several algorithms have been recently proposed to address this complex setting. Below, a few representative strategies are discussed.

Trust Region-Based Multi-Fidelity Multi-Objective BO Trust Region-Based Multi-Fidelity Multi-Objective BO (Trust-MOMF) [74] introduces a trust-region framework into the MOMFBO setting. The method decouples exploration and exploitation spatially by restricting candidate generation to a local region around previously evaluated high-quality solutions. Within this trust region, a multi-objective acquisition function, such as Expected Hypervolume Improvement (EHVI), is optimized. Low-fidelity evaluations are used to inform and expand the trust region, while expensive high-fidelity queries are limited to regions where predictions are deemed sufficiently reliable. This approach offers several advantages:

- It focuses high-fidelity evaluations in promising subregions, enhancing sample efficiency;
- It allows broader exploration via low-fidelity queries without compromising the quality of Pareto front estimates;
- It maintains compatibility with existing multi-objective acquisition functions, requiring minimal structural modifications.

MOMF with Decoupled Evaluations A class of sequential algorithms [73] tackles MOMFBO by decoupling the selection of the fidelity level from the selection of the input location. At each iteration, the algorithm first determines the most informative fidelity to use, based on a cost-benefit trade-off, and subsequently selects the input configuration that maximizes the acquisition function under the chosen fidelity. This stage-wise decomposition offers increased modularity and transparency:

- The fidelity selection can be guided by information-theoretic criteria, expected cost, or model uncertainty;
- The location selection can leverage standard acquisition strategies, such as EHVI or Pareto-aware variants of Expected Improvement;
- The decoupled structure facilitates extensions to asynchronous or resource-constrained settings.

Hypervolume-Based Knowledge Gradient The Hypervolume-Based Knowledge Gradient (HV-KG) algorithm [48] extends the Knowledge Gradient (KG) framework to multi-objective optimization using the hypervolume indicator as a utility metric. In the multi-fidelity variant, the acquisition function quantifies the expected gain in hypervolume that would result from querying a point at a given fidelity level. This results in a fidelity-aware, information-theoretic acquisition function that directly targets improvements to the Pareto front. Key properties of HV-KG include:

- Explicit integration of fidelity cost into the acquisition function, allowing budget-aware optimization;
- Focus on improving the Pareto frontier via hypervolume gains;
- Applicability in the presence of noisy or biased lower-fidelity data.

While several promising methods have been developed, the field of MOMFBO remains nascent, and multiple open challenges remain. These include:

- Designing acquisition functions that scale to high-dimensional Pareto fronts and large fidelity hierarchies;
- Capturing complex correlations and biases across objectives and fidelity levels in surrogate models;
- Handling heterogeneous fidelities with varying noise levels, missing outputs, or unknown cost functions.

Addressing these challenges is essential to broaden the applicability of Bayesian Optimization to real-world problems involving multi-criteria decision-making and hierarchical simulations. Applications include engineering design, robotics, environmental monitoring, and hyperparameter tuning in machine learning, all of which often involve multiple competing goals and evaluation modalities.

5.1.3 Fair and Green Machine Learning

Recent years have witnessed a growing concern regarding the sustainability and ethical implications of Machine Learning systems. In response, two key paradigms have emerged: Fair Machine Learning, which focuses on mitigating algorithmic bias, and Green Machine Learning, which emphasizes the reduction of computational and environmental costs associated with model training and inference.

Fair machine learning aims to prevent discriminatory behaviors in algorithms, particularly in sensitive applications such as hiring, lending, and criminal justice. Achieving fairness typically involves incorporating fairness metrics, such as demographic parity, equal opportunity, or equalized odds, into the model selection or optimization process. These fairness criteria often stand in trade-off with traditional performance measures like accuracy or precision, requiring algorithms that can explicitly model and explore such trade-offs.

Green machine learning, on the other hand, seeks to reduce the energy consumption and carbon footprint of ML pipelines. This can involve minimizing training time, limiting the number of model evaluations, or selecting architectures that are less resource-intensive. Energy and resource metrics, such as GPU hours or estimated CO_2 emissions, are thus increasingly treated as optimization objectives alongside task performance.

Both fairness and sustainability naturally lead to the need for Multi-Objective Optimization frameworks. When evaluations are expensive, either due to the use of large models, real-world data pipelines, or resource-consuming simulations, Bayesian Optimization becomes a suitable strategy. In such contexts, MOMFBO provides a principled approach to efficiently navigate trade-offs while leveraging cheaper but less accurate approximations of the objectives.

MOMFBO extends traditional multi-objective Bayesian optimization by incorporating information from multiple sources or fidelities, such as smaller datasets, fewer training epochs, or simplified model architectures. These low-fidelity sources provide inexpensive but noisy estimates of target objectives like accuracy, fairness, or energy consumption. By modeling the correlations between fidelities using Gaussian Processes or surrogate models, MOMFBO is able to allocate resources adaptively across fidelities and objectives.

In the context of Fair and Green Machine Learning, lower-fidelity evaluations can be defined using reduced datasets to minimize the carbon footprint, while fairness may be treated as a competing objective to traditional performance metrics such as accuracy.

Using MOMFBO, it becomes possible to:

- Efficiently approximate the Pareto front between competing objectives such as fairness and accuracy.
- Explore the trade-off between performance and resource consumption.
- Prioritize high-fidelity evaluations only when necessary, thereby reducing overall cost.

5.1.4 AutoGluon

AutoGluon is an open-source AutoML framework developed by AWS that provides a unified environment for automated model development, emphasizing empirical performance, reproducibility, and scalability. Unlike traditional machine learning pipelines that require extensive manual intervention, AutoGluon systematically explores model architectures, hyperparameters, and ensembling strategies using algorithmic search techniques. At its core, the framework employs multi-layer stacking ensembles, combining diverse model families such as gradient-boosted trees, neural networks, and k-nearest neighbors, which enhances robustness and predictive performance.

AutoGluon leverages adaptive hyperparameter optimization methods, including Bayesian optimization and multi-fidelity evaluation, to efficiently allocate computational resources and accelerate convergence. It further integrates mechanisms such as early stopping, meta-learning for warm-starting configurations, and automatic feature preprocessing, which reduce human bias and improve generalization. Its modular design supports tabular, text, image, and multimodal learning, making it a versatile tool for benchmarking AutoML algorithms and conducting large-scale experimental studies under a consistent and reproducible framework.

Several approaches for fairness-aware Bayesian optimization have been implemented within AutoGluon. One example is Fair Constrained Bayesian Optimization (FairCBO) [109], which extends classical Bayesian optimization to incorporate fairness constraints. This enables model-agnostic optimization of opaque functions while ensuring compliance with user-defined fairness criteria. FairCBO models both the target objective $f(x)$ and the fairness metric $c(x)$ as Gaussian processes, allowing probabilistic predictions of performance and constraint satisfaction across the hyperparameter space. Optimization is guided by a constrained acquisition function, the constrained Expected Improvement (cEI), defined as

$$cEI(x) = EI(x)\,P(c(x) \leq \varepsilon),$$

where $EI(x)$ is the standard Expected Improvement and $P(c(x) \leq \varepsilon)$ represents the probability of satisfying the fairness constraint. Initially, the algorithm emphasizes exploration of feasible regions by greedily maximizing the probability of satisfying the constraint until the first fair configuration is identified. Afterward, the full *cEI* criterion directs the search. FairCBO naturally extends to multiple simultaneous fairness constraints by assuming independence and combining their satisfaction probabilities. By focusing the search on hyperparameters that satisfy fairness requirements, FairCBO achieves data-efficient optimization while avoiding the computational overhead of multi-objective approaches, offering a practical and theoretically grounded solution for fairness-constrained hyperparameter tuning.

Another approach, proposed in [133] and building on advances in multi-objective hyperparameter optimization (MO-HPO) [132], adapts bandit-based resource allocation strategies to the multi-objective setting. Classical Hyperband [93] efficiently allocates a small initial resource r_0 to randomly sampled configurations, terminates unpromising candidates early, and reallocates larger resources to promising configurations in successive rounds. To extend Hyperband to multi-objective optimization, some methods employ random linear scalarization, reducing each configuration's performance vector to a single scalar to enable ranking. While scalarization simplifies candidate selection, it has limitations: it may fail to recover the full Pareto front, restricts exploration to certain directions, and is sensitive to rescaling of objectives. These approaches also replace Hyperband's synchronous scheduler with the asynchronous successive halving (ASHA) scheduler, which immediately reallocates workers to new configurations as soon as current evaluations finish.

In this work, this second approach has been leveraged as a baseline for comparison with the proposed MISO-AGP framework.

5.1.5 Fair-by-Design Machine Learning Algorithms

In addition to fairness-aware optimization approaches, another class of methods explicitly incorporates fairness constraints directly into the model design, often referred to as *fair-by-design* algorithms. These methods enforce fairness during model training rather than treating it as an external constraint applied post-hoc. Two representative examples implemented within the `fairml` R package are `zlrm` and `fgrrm`.

`zlrm` implements the Fair Logistic Regression with covariance constraints proposed in [165]. This method introduces linear constraints on the covariance between the sensitive attributes and the model's decision boundary, ensuring that predictions are statistically independent of protected characteristics. By embedding fairness directly into the objective function, `zlrm` provides a model that balances predictive performance with fairness guaran-

tees in a principled manner, making it suitable for binary classification tasks where fairness is a critical requirement.

`fgrrm` extends this concept to generalized linear models with ridge regularization, as proposed in [136]. The Fair Generalized Ridge Regression Model supports a wide range of outcome families, including Gaussian, binomial, Poisson, multinomial, and Cox (proportional hazards) models. Fairness constraints are integrated directly into the regularized optimization problem, allowing the model to achieve fair predictions across diverse statistical settings while maintaining flexibility in handling various response types. This approach is particularly useful in applications where outcomes are not limited to binary labels and may involve counts, categorical variables, or survival data.

Overall, fair-by-design methods offer a complementary strategy to fairness-aware hyperparameter optimization: rather than adapting the search process to satisfy fairness, they build fairness guarantees into the model itself. In practice, these approaches provide interpretable and theoretically grounded mechanisms for producing fair predictions across a variety of machine learning tasks.

5.2 Multi-Objective AGP

5.2.1 Extending MISO-AGP to Multi-Objective Settings

The MISO-AGP framework can be naturally extended to handle multi-objective optimization problems, resulting in a method referred to as MOMISO-AGP (Multi-Objective Multi-Information Source Optimization via Augmented Gaussian Processes). This extension enables the optimization of vector-valued objective functions under query cost constraints, leveraging multiple information sources with varying fidelities and associated costs.
Formally, the problem can be expressed as follows:

$$\begin{aligned} x^* \in \arg\min_{x \in \mathcal{X}} \quad & f_1(x) \\ \text{s.t.} \quad & \sum_{i=1}^{N} c_{s^{(i)}} \leq C \end{aligned} \tag{5.6}$$

where $f_1(x)$ is the vector-valued ground-truth objective, and $f_s(x)$ for $s \in 2,\ldots,S$ are the cheaper approximations from auxiliary information sources. The query cost for each source $s^{(i)}$ is $c_{s^{(i)}}$, N is the total number of queries, and C is the maximum total cost budget. Each objective $m \in 1,\ldots,M$ and each information source $s \in 1,\ldots,S$ is modeled independently

with a Gaussian Process (GP), resulting in $S \times M$ individual models:

$$\left\{\mu_{sm}(x), \sigma_{sm}(x)\right\}_{s=1:S, m=1:M}$$

where each pair $\mu_{sm}(x)$, $\sigma_{sm}(x)$ denotes the GP predictive mean and uncertainty for objective m from source s, conditioned on the respective observation set $\mathbf{X}_s, \mathbf{Y}_s$. For each objective m, an AGP model is built by augmenting the ground-truth observations with reliable observations from the cheaper sources. Reliability is defined per objective-source pair using a discrepancy-based criterion:

$$\begin{aligned} \mathscr{I}_{sm} = \Big\{ i \,:\, \Big|\mu_{1m}(x^{(i)}) - \mu_{sm}(x^{(i)})\Big| \leq \alpha \sigma_{1m}(x^{(i)}),\, x^{(i)} \in \mathbf{X}_s \Big\}, \\ \forall s \in \{2,\ldots,S\},\, \forall m \in \{1,\ldots,M\} \end{aligned} \tag{5.7}$$

Using these indices, the AGP for objective m is trained on the augmented dataset:

$$\widehat{\mathbf{X}}_m \leftarrow \mathbf{X}_1 \cup \left\{ x^{(i)} \in \mathbf{X}_s : i \in \mathscr{I}_{sm}, \forall s \neq 1 \right\} \tag{5.8}$$

where $[m]$ denotes the m-th column of the output matrix. Each AGP yields a predictive mean $\widehat{\mu}_m(x)$ and uncertainty $\widehat{\sigma}_m(x)$ for objective m. To determine the next query, a two-step procedure is adopted:

1. **Selecting the input location** x': the next input point is selected by maximizing the Expected Hypervolume Improvement (EHVI), a widely used acquisition function in multi-objective Bayesian optimization:

$$x' = \underset{x \in \mathscr{X}}{\arg\max}\, \mathrm{EHVI}(x, \mathscr{P}, \mathbf{r}) \tag{5.9}$$

   where $\mathscr{P}$ is the current approximated Pareto front and $\mathbf{r}$ is the Nadir point in the objective space.

2. **Selecting the information source** s': the source to query at x' is selected by minimizing the product of its cost and cumulative discrepancy from the ground-truth across all objectives:

$$s' = \underset{s \in \{1,\ldots,S\}}{\arg\min}\; c_s \sum_{m=1}^{M} \left|\mu_{1m}(x') - \mu_{sm}(x')\right| \tag{5.10}$$

To ensure that AGP models are not overly biased by cheap sources, a safeguard mechanism is included: if, for any objective, the number of augmenting observations exceeds the number of ground-truth observations, then the ground-truth ($s' = 1$) is queried directly.

Once the (s', x') pair is selected, $f_{s'}(x')$ is queried, potentially with noise, and the relevant GP models for source s' are updated. This iterative process continues until the budget C is exhausted.

The MOMISO-AGP algorithm provides a principled and cost-aware strategy for optimizing multiple competing objectives using multiple information sources, without requiring correlation assumptions among objectives or sources. The framework is generic and can be applied to a variety of application domains, including those where objectives reflect fairness, robustness, environmental impact, or other competing performance criteria.

5.3 Experiments

5.3.1 Experimental Setup

This section describes the experimental framework employed to evaluate multi-objective hyperparameter optimization (HPO) approaches, incorporating predictive performance, fairness, and computational efficiency. The experimental design ensures reproducibility and comparability across all considered methods.

Problem Description

The main problem under investigation is multi-objective hyperparameter optimization of machine learning algorithms, with three objectives considered:

1. Predictive performance, quantified via the misclassification error over a validation set:

$$\mathrm{MSC}(x) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(\hat{y}_i(x) \neq y_i), \tag{5.11}$$

   where x represents a hyperparameter configuration, N is the number of validation instances, y_i is the true label, and $\hat{y}_i(x)$ is the predicted label.

2. Fairness, measured as the Difference in Statistical Parity (DSP) between groups defined by sensitive features:

$$\mathrm{DSP}(x) = \left| P(\hat{Y} = 1 \mid S = 0, x) - P(\hat{Y} = 1 \mid S = 1, x) \right|, \tag{5.12}$$

   where S denotes the sensitive attribute (e.g., gender or race), and $\hat{Y}$ is the predicted label.

3. Carbon footprint, approximated via a computational cost proxy associated with each evaluation. Multi-fidelity evaluations are performed on smaller subsets of the dataset to reduce this cost, thereby providing energy-efficient estimates of the objectives.

The multi-fidelity setup allows reduced-cost evaluations to inform the optimization while minimizing the environmental impact associated with full dataset computations.

Datasets

Four benchmark datasets widely employed in fairness-related research were considered: ADULT, COMPAS, GERMAN CREDIT, and LAW SCHOOL ADMISSIONS [29, 91]. Each dataset poses a non-trivial trade-off between predictive accuracy and fairness.

- ADULT: Predicts whether personal income exceeds $50K/year from the 1994 U.S. Census data (30,162 instances, 14 features; sensitive features: gender, race).
- COMPAS: Predicts recidivism within two years for offenders in Florida (2013-2014; 5,855 instances, 16 features; sensitive: gender, race).
- GERMAN CREDIT: Predicts consumer loan defaults in Germany (1,000 instances, 21 features; sensitive: gender).
- LAW SCHOOL ADMISSIONS: Predicts passing the bar exam on the first try from a 1991 U.S. survey (20,800 instances, 11 features; sensitive: gender, race).

Prior to hyperparameter optimization, all categorical features were one-hot encoded, increasing the final number of features.

Machine Learning Algorithms

Four supervised machine learning algorithms were considered for hyperparameter optimization:

- Multi-Layer Perceptron (MLP), 10 hyperparameters.
- Random Forest (RF), 2 hyperparameters.
- eXtreme Gradient Boosting (XGB), 7 hyperparameters.
- Support Vector Machine (SVM) with RBF kernel, 2 hyperparameters.

The hyperparameter search spaces for each algorithm are reported in Table 5.1. For MLP and XGB, the search spaces follow [132], while RF and SVM were defined specifically for this study.

Table 5.1 Search space for the algorithms' hyperparameters. The range of the hyperparameter `max_features` of RF depends on the dataset: $|F|$ denotes the number of features in the dataset, excluded the target one.

Algorithm	Hyerparameter	Type	Domain	Scaling
MLP	n_layers	integer	{1,2,3,4}	linear
	layer_1	integer	{2,...,32}	linear
	layer_2	integer	{2,...,32}	linear
	layer_3	integer	{2,...,32}	linear
	layer_4	integer	{2,...,32}	linear
	alpha	real	$[10^{-6},10^{-1}]$	$\log_{10}$
	learning_rate_init	real	$[10^{-6},10^{-1}]$	$\log_{10}$
	beta_1	real	[0.001,0.99]	$\log_{10}$
	beta_2	real	[0.001,0.99]	$\log_{10}$
	tol	real	$[10^{-5},10^{-2}]$	$\log_{10}$
RF	n_estimators	integer	{100,...,1000}	linear
	max_features	integer	$\{2,\ldots,\lvert F\rvert\}$	linear
XGBoost	n_estimators	integer	{1,...,256}	linear
	learning_rate	real	[0.01,1.0]	$\log_{10}$
	gamma	real	[0.0,0.1]	linear
	reg_alpha	real	$[10^{-3},10^{3}]$	$\log_{10}$
	reg_alpha	real	$[10^{-3},10^{3}]$	$\log_{10}$
	subsample	real	[0.01,1.0]	linear
	max_depth	integer	{1,2,...,16}	linear
SVM	C	real	[0.0001,10000]	$\log_{10}$
	gamma	real	[0.0001,10000]	$\log_{10}$

Experimental Settings

The experimental evaluation included three multi-objective, energy- and fairness-aware HPO approaches:

- `FanG-HPO` [29]: The multi-information source Bayesian optimization approach using Augmented Gaussian Processes (AGPs) for each objective and EHVI-based acquisition functions as described in Section 5.2.
- `BoTorch-MOMF` [74]: A multi-objective, multi-fidelity Bayesian optimization method employing multi-output Gaussian Processes jointly modeling misclassification error, DSP, and evaluation cost as described in Section 5.1.2.

- `AutoGluon-FairBO` [133]: A fairness-aware implementation of HyperBand using successive halving and random scalarization to reduce the multi-objective problem to a single scalar objective as described in Section 5.1.4.

Additionally, two fairness-by-design algorithms were considered for comparison:

- `zlrm` [165]: Fair Logistic Regression with covariance constraints.
- `fgrrm` [135]: Fair Generalized Ridge Regression Model supporting Gaussian, binomial, Poisson, multinomial, and Cox outcome families.

To ensure an unbiased comparison, all HPO methods were initialized using the same set of hyperparameter configurations and were constrained to a common evaluation budget.

Multi-Fidelity Setup Two information sources were considered for BoTorch-MOMF and FanG-HPO:

- High-fidelity (full dataset), nominal query cost $c_1 = 1$.
- Low-fidelity (50% stratified sample), nominal query cost $c_2 = 0.5$.

AutoGluon-FairBO relies on successive halving over 10-fold cross-validation (10FCV). Queries early-stopped before completing all folds are assigned a cost $c_2 = n_f/10$, where n_f is the number of folds completed. Completed 10FCV runs correspond to $c_1 = 1$. Fairness-by-design algorithms do not involve hyperparameter optimization and have no associated query costs.

Termination Criterion AutoGluon-FairBO was executed with a maximum of 200 hyperparameter evaluations per run. The cumulative query cost from each run defined the budget for BoTorch-MOMF and FanG-HPO, ensuring a consistent comparison. The first $2(d+1)$ configurations evaluated by AutoGluon-FairBO were used to initialize the two other methods, with d being the number of hyperparameters for the respective algorithm.

Evaluation Protocol For each ML algorithm – dataset pair:

1. Run AutoGluon-FairBO with a limit of 200 queries and compute cumulative query cost.
2. Initialize BoTorch-MOMF and FanG-HPO using the first two sets of configurations evaluated by AutoGluon-FairBO.

3. Terminate BoTorch-MOMF and FanG-HPO when the cumulative query cost reaches the same budget observed for AutoGluon-FairBO.

4. Perform ten independent runs per ML algorithm – dataset pair. For `zlrm` and `fgrrm`, five independent runs were performed, using an unfairness constraint of 0.1 for each sensitive feature.

Energy-Efficiency Mechanisms The approaches differ in their strategy for energy-efficient optimization [147]:

- `AutoGluon-FairBO`: Early discarding of unpromising configurations via HyperBand successive halving.
- `BoTorch-MOMF`: Multi-fidelity performance measurements explicitly model the query cost and fidelity as part of the acquisition function.
- `FanG-HPO`: Multi-information source optimization using AGPs with a two-step EHVI acquisition function balancing objective performance and evaluation cost.

5.3.2 Experimental Results

This section reports and discusses the most relevant results of the experimental analysis. To ensure clarity, each result is explicitly stated and followed by a short commentary. The presentation is organized into four parts: *(i)* a comparison between Fairness-aware ML and Fairness-aware AutoML algorithms, *(ii)* an evaluation of the cost-effectiveness of the considered BO-based approaches, *(iii)* an analysis of their ecological performance profiles, and *(iv)* additional observations regarding the usage of information sources.

Fairness-related Results

A first set of experiments aimed at comparing Fairness-aware ML algorithms with Fairness-aware AutoML methods. The comparison was carried out in terms of Pareto optimality. For each dataset–ML algorithm pair, all dominant hyperparameter configurations obtained by the three BO-based approaches over 10 independent runs were collected. Only configurations evaluated on the full dataset were considered, and the resulting set of non-dominated solutions was referred to as the *super Pareto front*. Figure 5.1 illustrates the super Pareto fronts together with the MCE–DSP trade-offs of Fairness-aware ML algorithms. The following main findings were derived.

- Fairness-aware AutoML dominates Fairness-aware ML algorithms. Across all four datasets, at least one super Pareto front obtained via AutoML dominates the trade-offs achieved by Fairness-aware ML baselines. This observation is consistent with recent findings in the literature, which reported that post-processing of Fairness-aware ML algorithms is often insufficient to achieve competitive Pareto trade-offs [43].
- Bi-objective HPO of RF produces super Pareto fronts that are systematically smaller than those obtained for other ML algorithms. Both the hypervolume and the number of Pareto-optimal configurations are limited. In particular, the DSP values of RF solutions remain consistently high, indicating that RF, although accurate, is less competitive in terms of fairness.
- The bi-objective HPO of XGBoost (XGB) yields the best overall outcomes. For all datasets, the super Pareto front associated with XGB is larger than those of the other algorithms, both in terms of HV and in the number of Pareto-optimal solutions.

Cost-effectiveness of Fairness-aware HPO Methods

When comparing AutoML systems, it is not sufficient to evaluate only the final performance after the budget is exhausted. Instead, the evolution of the best observed hypervolume with respect to the cumulative query cost provides a more informative assessment of cost-effectiveness. In this analysis, the HV was computed from the approximated Pareto front consisting of dominant configurations evaluated on the full dataset, at increasing values of the cumulative query cost. The costs were defined according to the two information sources introduced in Section 5.3.1. Representative results are reported in Figures 5.2, 5.3, 5.4 and 5.5.

- For MLP, successive halving (AutoGluon-FairBO) is less cost-effective than multi-fidelity and multiple-information source BO methods (BoTorch-MOMF and FanG-HPO). Although AutoGluon-FairBO initially achieves higher hypervolume values due to its initialization strategy, both BoTorch-MOMF and FanG-HPO surpass it at relatively low cumulative costs.
- The pathological behaviour of RF, noted earlier, persists across all three BO-based approaches. Despite gradual improvements, neither BoTorch-MOMF nor FanG-HPO consistently close the gap with AutoGluon-FairBO, except for BoTorch-MOMF on the GERMANCREDIT dataset. This effect is explained by the degenerate Pareto fronts produced for RF.

- For XGB, multi-fidelity and multiple-information-source approaches (BoTorch-MOMF and FanG-HPO) again prove more cost-effective than AutoGluon-FairBO. The trend mirrors the one observed for MLP.
- For SVM, FanG-HPO consistently outperforms the other approaches in terms of cost-effectiveness, emerging as the most reliable choice across all datasets.

Ecological Performance Profiles

Nominal query cost provides only an indirect proxy of resource efficiency. To better approximate environmental impact, cost-effectiveness curves were redrawn in terms of cumulative runtime, following the ecological performance profile framework proposed by [147]. Runtime was measured exclusively for hyperparameter evaluations (i.e., queries), as the overhead of the optimization algorithms was negligible in comparison. Results are shown in Figures 5.6, 5.7, 5.8 and 5.9.

- Overall, successive halving (AutoGluon-FairBO) is less ecological than multi-fidelity and multiple-information-source methods. BoTorch-MOMF and FanG-HPO achieve significantly lower cumulative runtimes, except for RF, where BoTorch-MOMF performs worse due to the pathological nature of RF Pareto fronts.
- XGB is the most promising algorithm in terms of fairness-aware HPO. Its Pareto fronts are consistently rich, and when optimized with FanG-HPO, it achieves both the best ecological profiles and the largest sets of Pareto-optimal models. From the perspective of developers and practitioners, this represents one of the most valuable outcomes of the study.

Additional Observations

Finally, the behaviour of the BO-based approaches with respect to their usage of the expensive (high-fidelity) source was examined. Table 5.2 reports the average frequency of expensive queries across datasets and algorithms.

FanG-HPO is the approach that most frequently queries the expensive source. This behaviour is expected given its design, but it is noteworthy that despite more frequent use of the ground-truth source, its cumulative runtime remains comparable to (and often smaller than) that of BoTorch-MOMF. This suggests an efficient and adaptive exploitation of available sources.

Across datasets and algorithms, FanG-HPO consistently provides the largest sets of Pareto-optimal models, further confirming its effectiveness as a fairness-aware AutoML approach.

Table 5.2 Percentage of hyperparameter configurations evaluated on the entire dataset: mean and standard deviation on 10 independent runs.

ML algorithm	Dataset	autogluon-FairBO	BoTorch-MOMF	FanG-HPO
MLP	ADULT	3.64% (0.90%)	40.47% (3.86%)	60.24% (6.68%)
	COMPAS	3.97% (0.36%)	40.37% (4.05%)	55.33% (8.72%)
	GERMAN CREDIT	3.74% (0.37%)	41.20% (4.04%)	61.36% (7.28%)
	LAW SCHOOL ADMISSIONS	4.36% (0.48%)	41.52% (3.82%)	58.12% (5.34%)
RF	ADULT	4.91% (0.97%)	41.30% (4.28%)	91.25% (1.88%)
	COMPAS	4.55% (0.66%)	45.67% (6.58%)	92.69% (0.81%)
	GERMAN CREDIT	4.09% (0.33%)	40.43% (4.45%)	84.18% (4.50%)
	LAW SCHOOL ADMISSIONS	4.41% (0.94%)	43.12% (4.85%)	90.17% (1.76%)
XGB	ADULT	3.73% (0.82%)	44.88% (6.30%)	84.09% (3.71%)
	COMPAS	4.22% (0.66%)	41.55% (5.56%)	73.28% (10.62%)
	GERMAN CREDIT	3.84% (0.24%)	38.28% (6.61%)	53.73% (9.78%)
	LAW SCHOOL ADMISSIONS	4.85% (0.74%)	41.60% (5.25%)	66.38% (14.78%)
SVM	ADULT	5.67% (0.13%)	8.05% (6.77%)	68.06% (13.65%)
	COMPAS	4.78% (0.35%)	16.23% (7.41%)	53.50% (6.38%)
	GERMAN CREDIT	4.67% (1.30%)	8.33% (3.52%)	47.87% (1.69%)
	LAW SCHOOL ADMISSIONS	5.28% (1.03%)	11.12% (7.45%)	52.14% (7.54%)

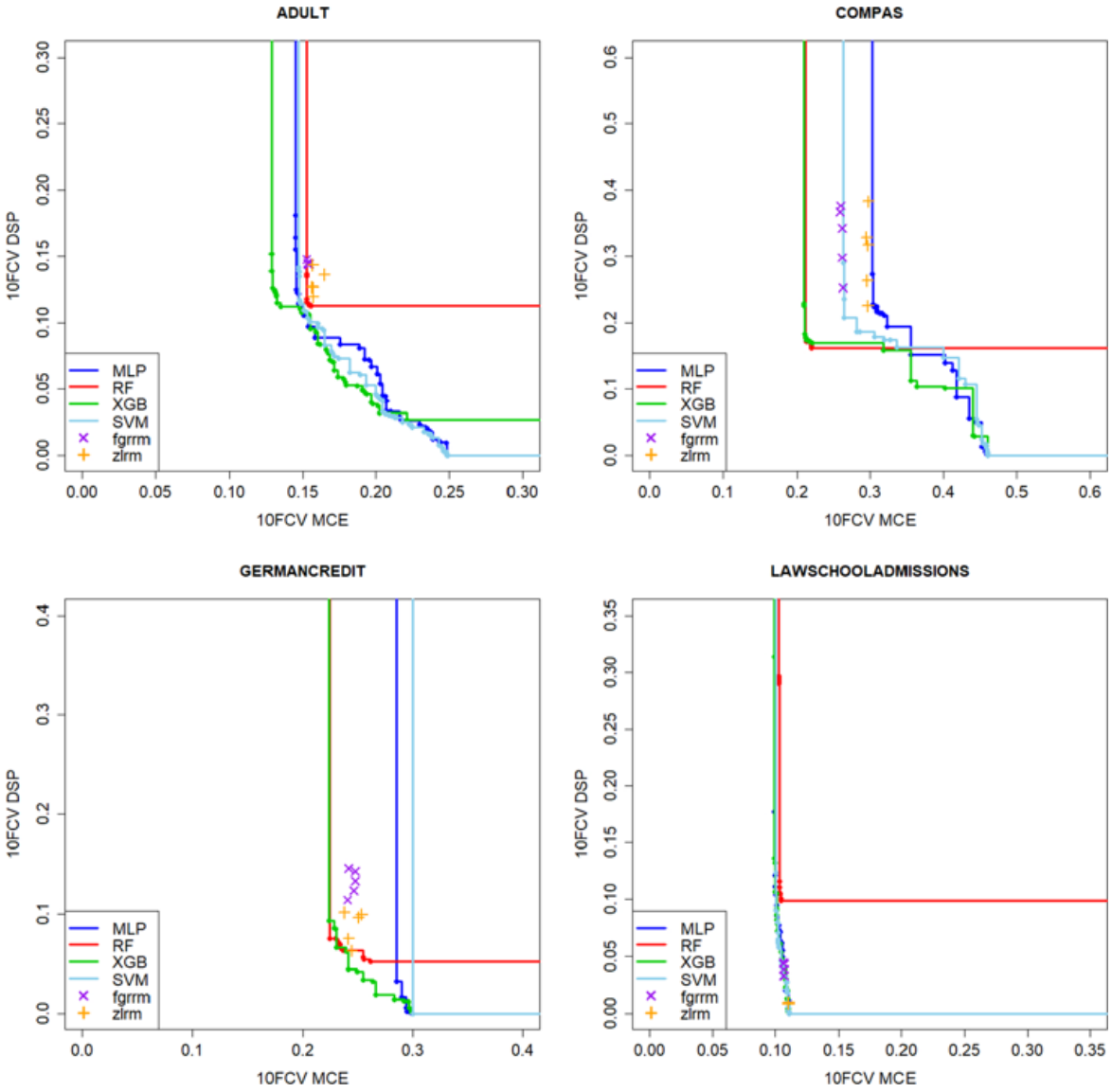

Fig. 5.1 Comparison between MCE–DSP trade-offs of Fairness-aware ML algorithms and the super Pareto fronts obtained through HPO of four ML algorithms. The super Pareto fronts are constructed by pooling together all non-dominated configurations identified by the three BO-based approaches (AutoGluon-FairBO, BoTorch-MOMF, and FanG-HPO) over 10 independent runs.

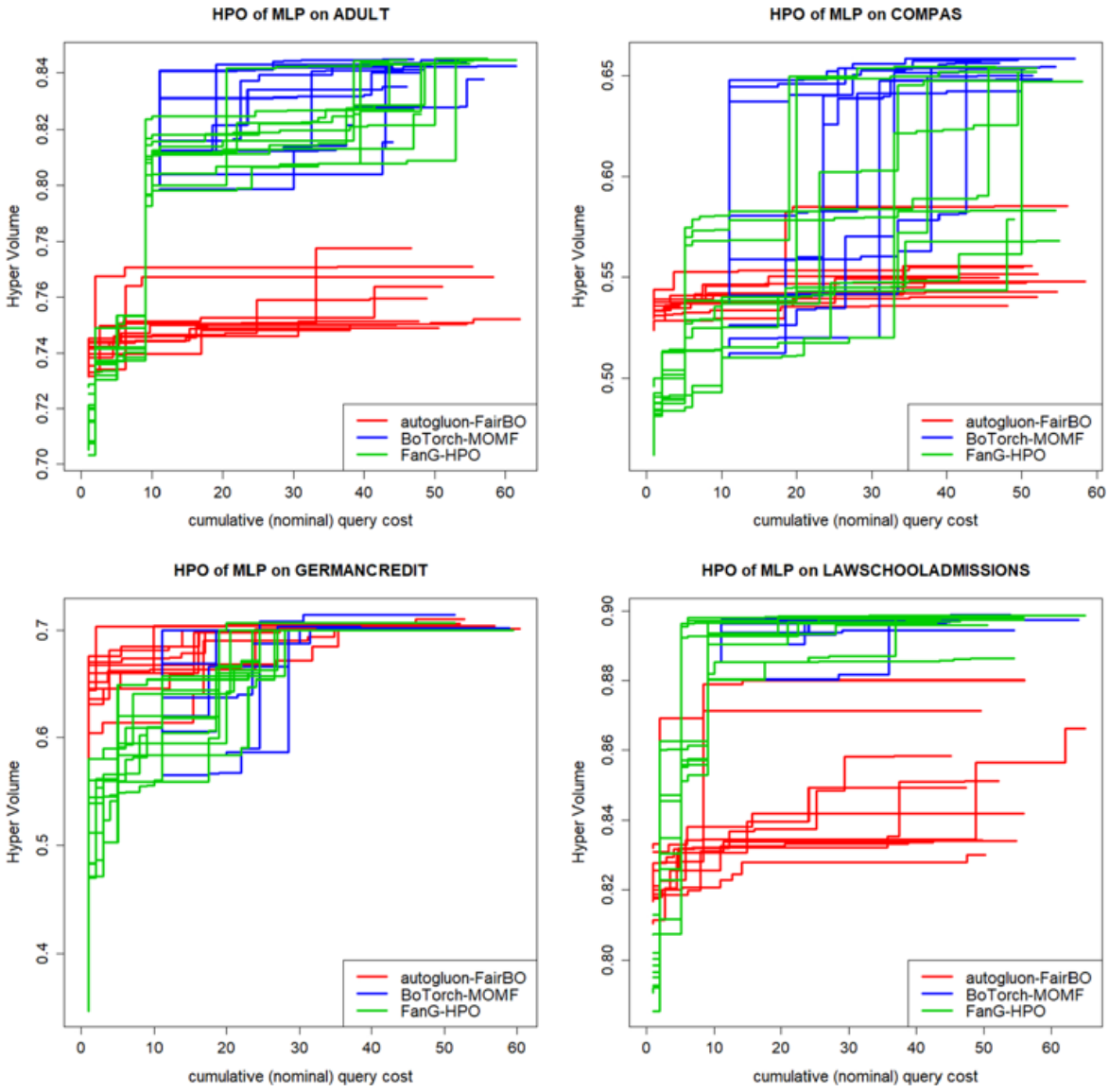

Fig. 5.2 Cost-effectiveness of the three BO-based approaches for bi-objective HPO of an MLP classifier, aggregated over 10 independent runs on the four datasets.

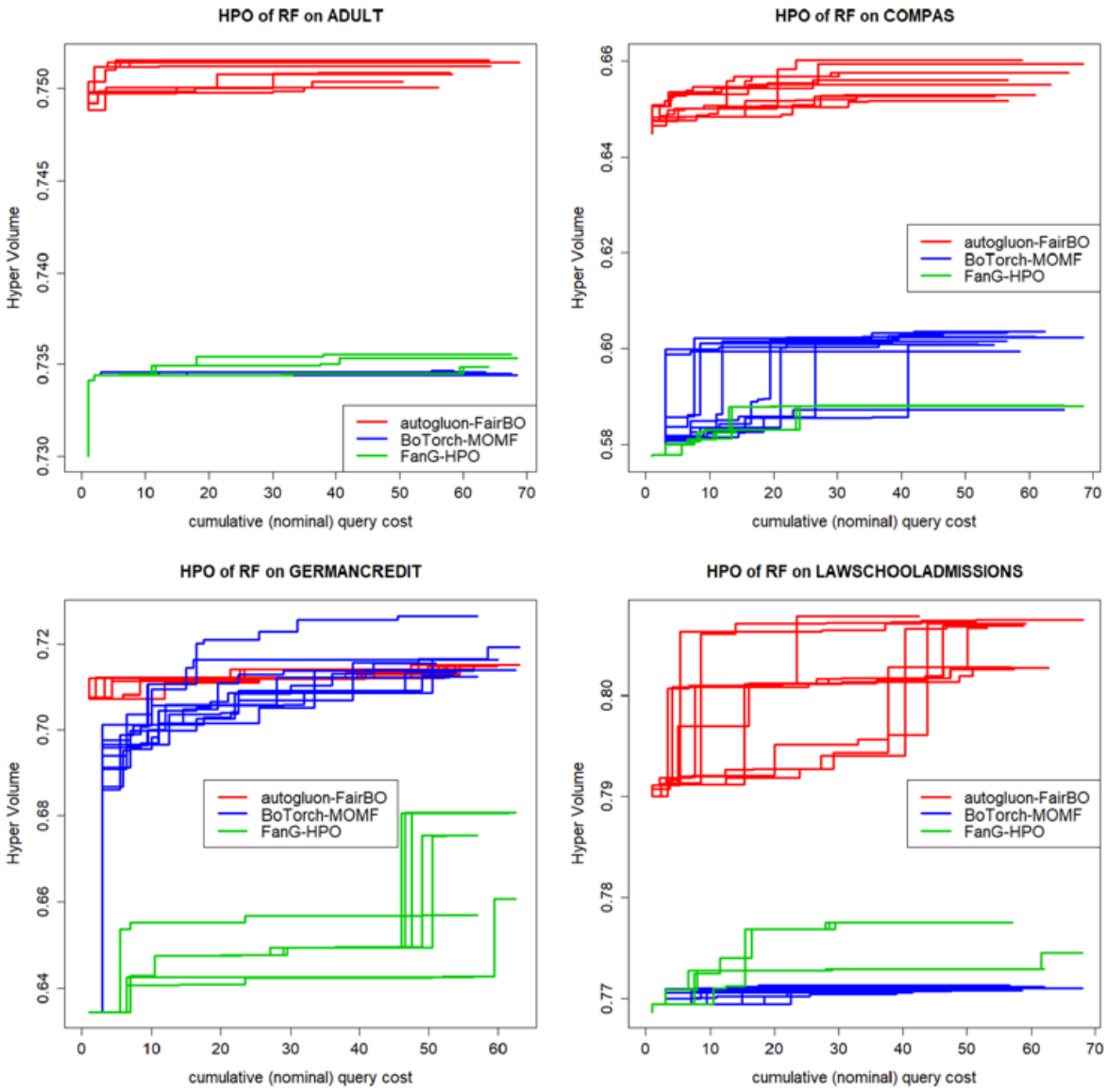

Fig. 5.3 Cost-effectiveness of the three BO-based approaches for bi-objective HPO of an RF classifier, aggregated over 10 independent runs on the four datasets.

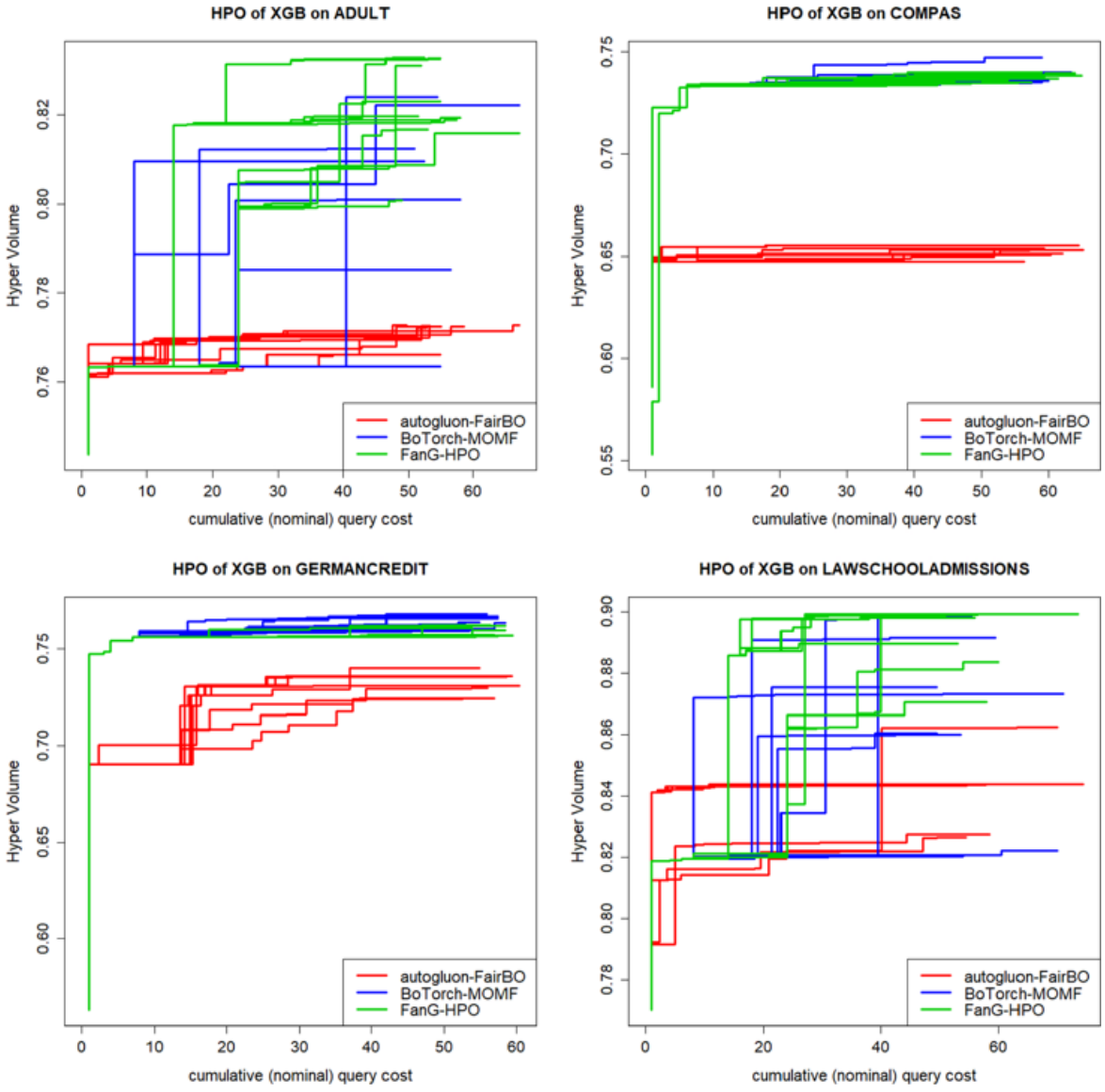

Fig. 5.4 Cost-effectiveness of the three BO-based approaches for bi-objective HPO of an XGB classifier, aggregated over 10 independent runs on the four datasets.

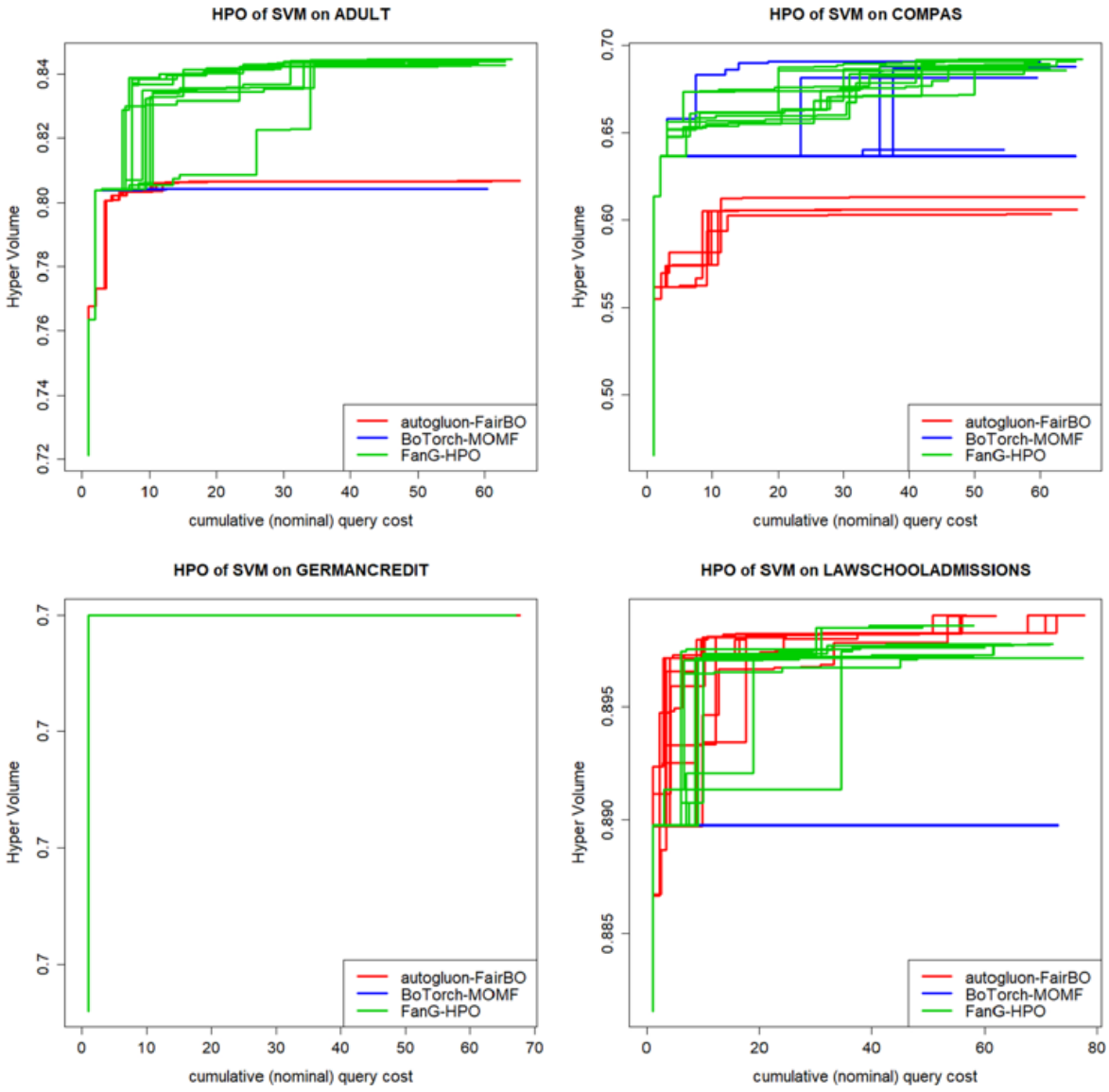

Fig. 5.5 Cost-effectiveness of the three BO-based approaches for bi-objective HPO of an SVM classifier, aggregated over 10 independent runs on the four datasets.

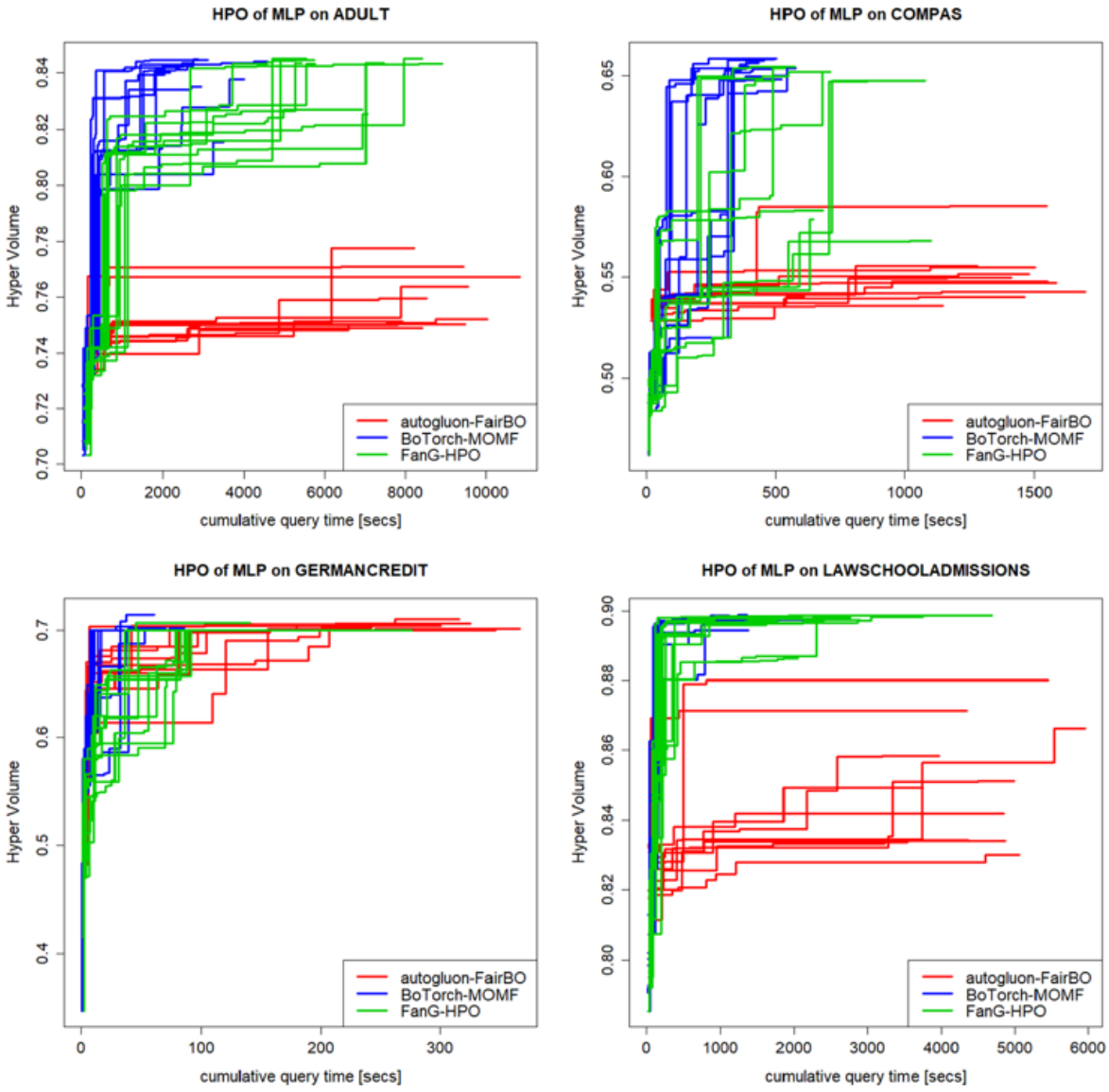

Fig. 5.6 Ecological performance profiles (runtime-based) of the three BO-based approaches for MLP.

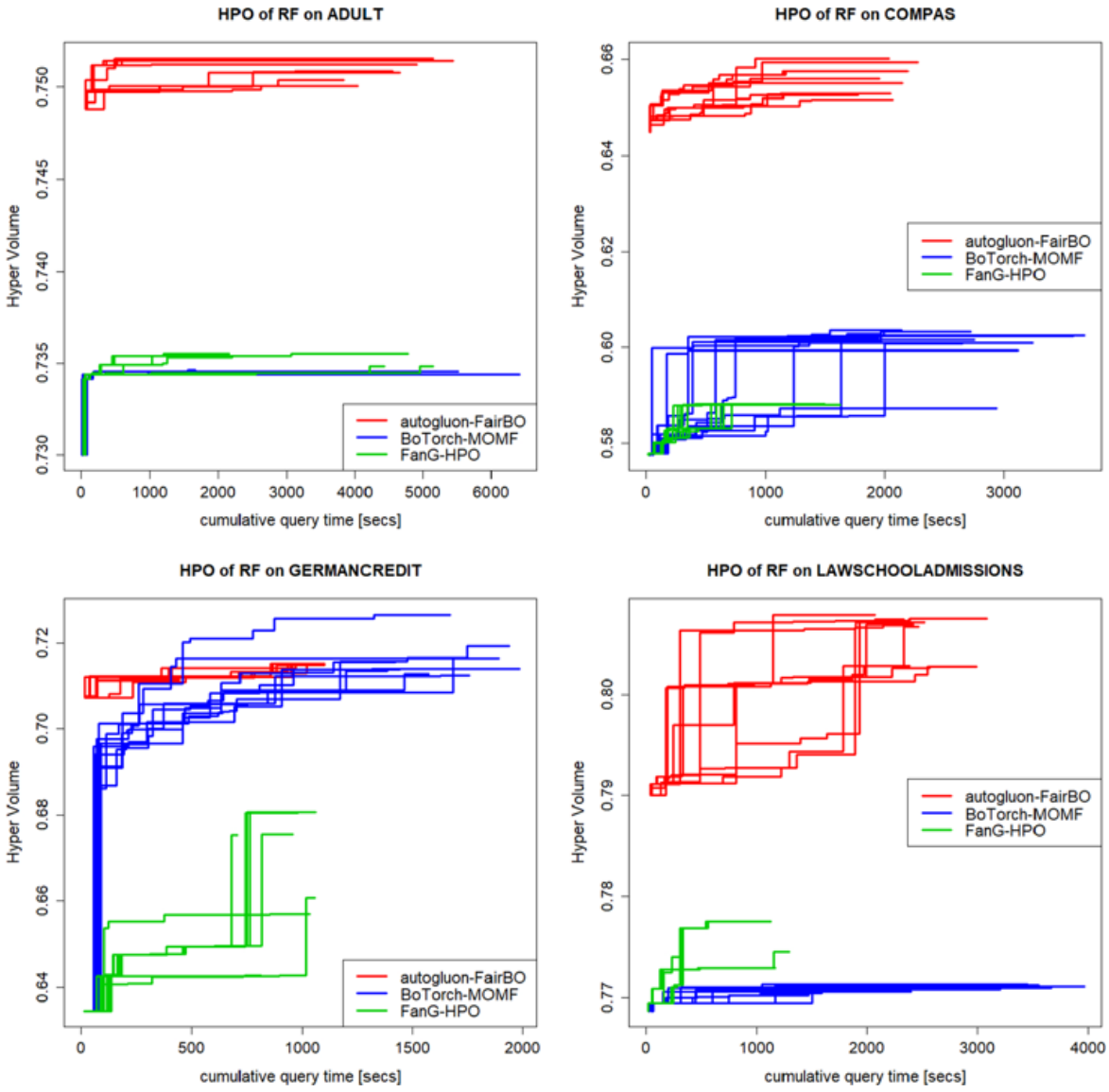

Fig. 5.7 Ecological performance profiles (runtime-based) of the three BO-based approaches for RF.

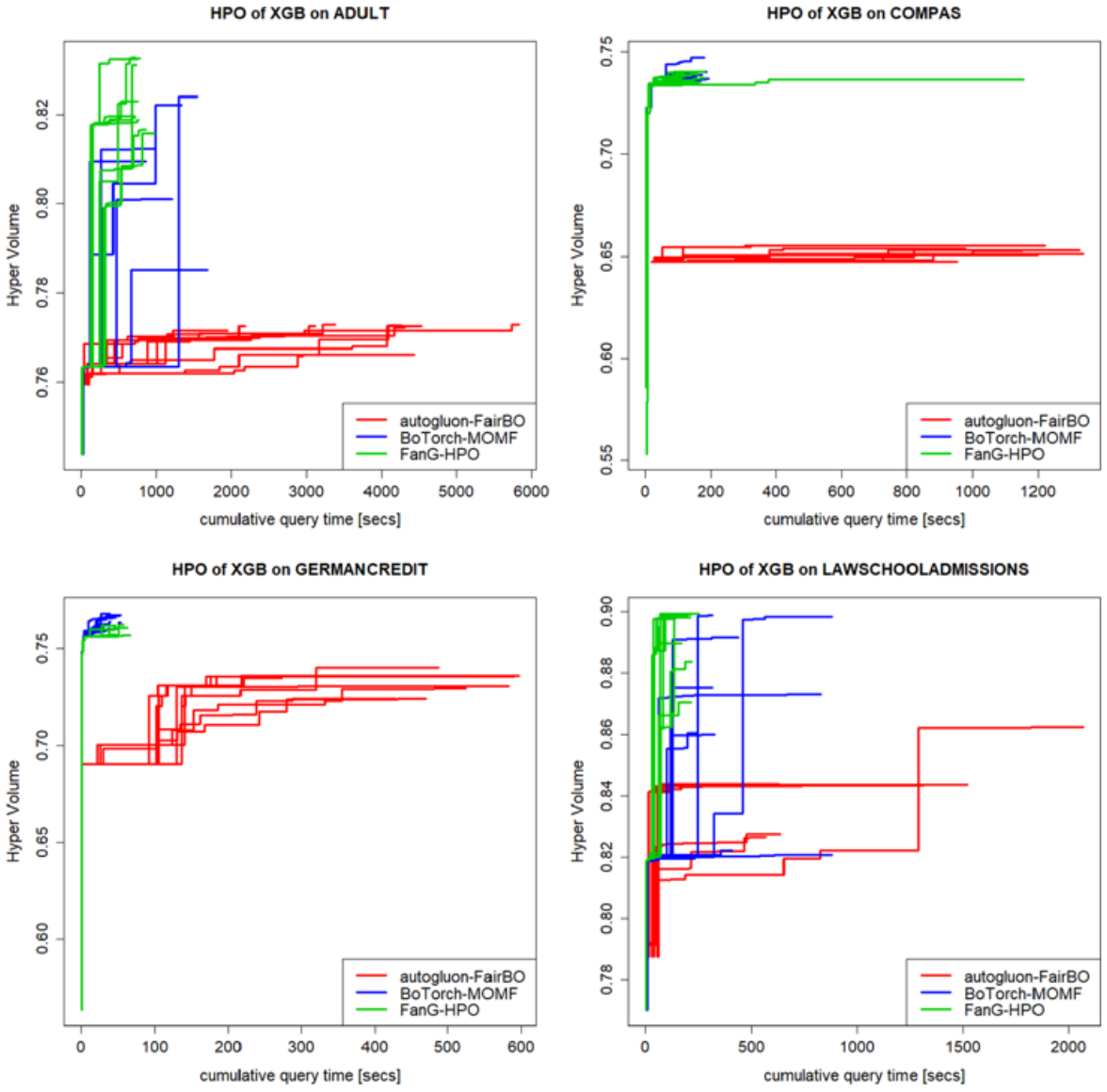

Fig. 5.8 Ecological performance profiles (runtime-based) of the three BO-based approaches for XGB.

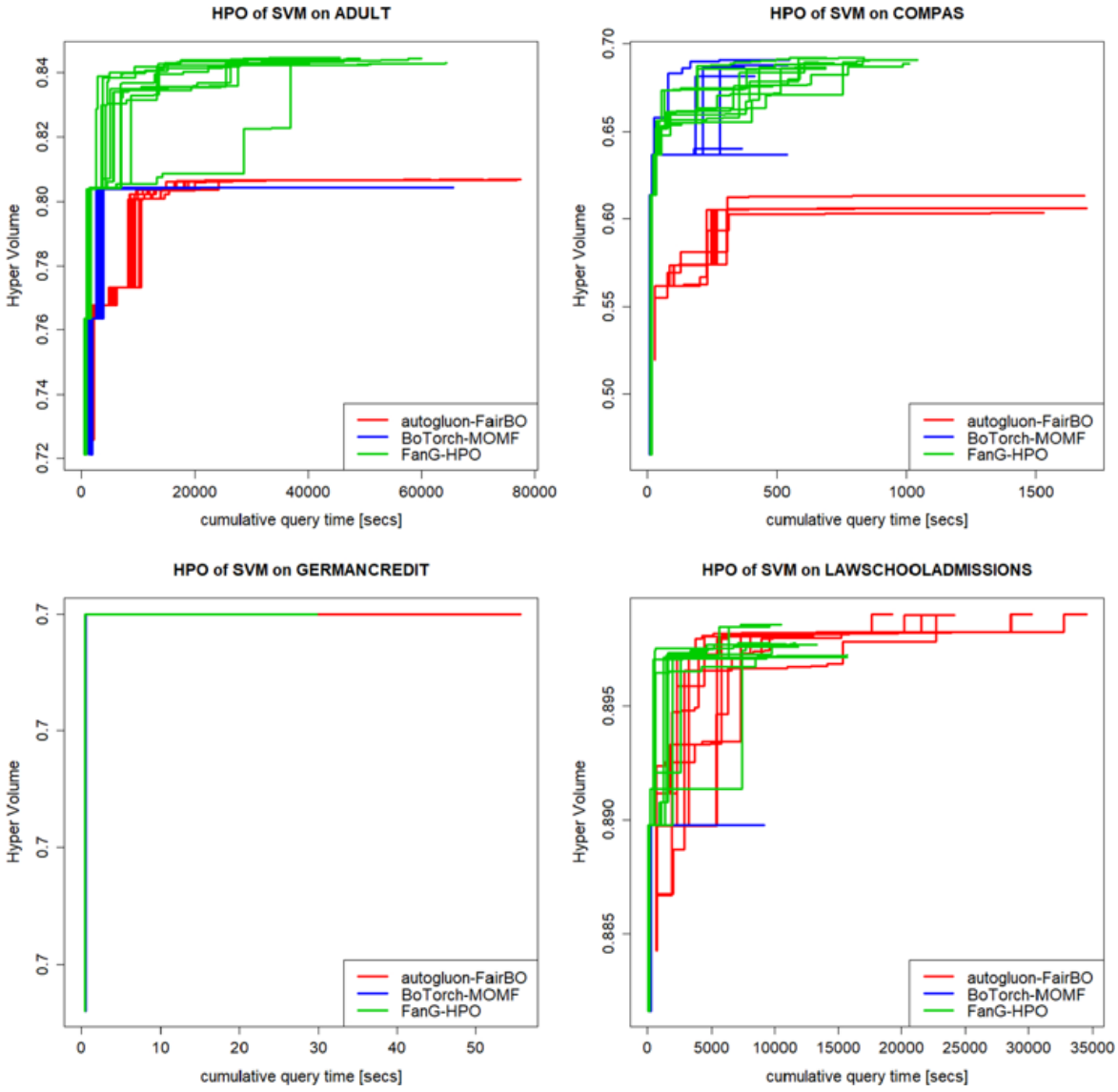

Fig. 5.9 Ecological performance profiles (runtime-based) of the three BO-based approaches for SVM.

Chapter 6

Conclusions

This thesis has addressed the general problem of optimizing expensive, black-box functions under realistic conditions where multiple objectives, multiple information sources, and complex decision spaces coexist. Such scenarios arise naturally in modern scientific and machine learning applications, including model selection, resource-aware optimization, sensor placement, and recommender systems.

The common thread connecting all contributions is the development of methods that are both efficient (i.e., able to learn from few costly evaluations) and expressive (i.e., capable of capturing multi-objective trade-offs, multi-fidelity structures, and discrete or combinatorial design spaces). To this end, the work integrates concepts from Bayesian optimization, Gaussian processes, and evolutionary multi-objective algorithms, while introducing novel theoretical and algorithmic components.

The research developed in this dissertation can be grouped into four main areas of contribution.

Wasserstein-enabled Multi-Objective Evolutionary Algorithms

A first line of work focused on enhancing evolutionary algorithms through the use of the Wasserstein distance. Specifically, an extension of the NSGA-II algorithm was proposed, where the selection mechanism relies on Wasserstein distances among individuals, promoting a more geometrically informed diversity preservation in the Pareto front. Similarly, a Wasserstein-based variant of MOEA/D was introduced, in which the decomposition weights are ordered using the Wasserstein distance, leading to a smoother coverage of the objective space. Finally, a novel binary crossover operator was designed, exploiting the Wasserstein metric to blend parent solutions in a way that preserves diversity while maintaining convergence pressure. All these extensions were implemented within the `pymoo` framework

and experimentally validated on standard benchmarks and real-world problems, showing consistent improvements in terms of convergence speed and Pareto front diversity.

Augmented Gaussian Processes for Combinatorial and Multi-Objective Optimization

A second major contribution lies in the extension of the Augmented Gaussian Process (AGP) model to broader and more challenging settings. First, the AGP framework was extended to handle combinatorial and discrete domains by integrating genetic algorithms for the optimization of acquisition functions. This approach allows the Bayesian optimization framework to tackle discrete search spaces while retaining the uncertainty-aware nature of Gaussian processes. Second, AGP was extended to the multi-objective case, leading to the formulation of the MISO-AGP (Multi-Information Source Optimization with Augmented Gaussian Processes). This extension enables the joint modeling of multiple fidelities and objectives in a coherent probabilistic framework. Both formulations were implemented and evaluated within the `BoTorch` library, and the combinatorial version was contributed to its open-source codebase, thus making the method available to the research community.

Multi-Objective and Multi-Fidelity Formulations of Classical Problems

A further contribution concerns the reformulation of several classical optimization problems under the proposed frameworks. In particular:

- the Optimal Sensor Placement (OSP) problem and the Recommender System problem were reformulated as multi-objective optimization tasks, allowing to explicitly model and explore the trade-offs among competing goals;
- the Risk-Averse OSP and Binary Quadratic Programming problems were extended to multi-fidelity settings, where different sources of information or simulation accuracies can be adaptively combined;
- the Hyperparameter Optimization of machine learning algorithms was formulated as a multi-objective, multi-fidelity problem, providing a unified view of fairness, accuracy, and computational efficiency.

These formulations not only expand the modeling flexibility of traditional problems, but also demonstrate the practical applicability of the proposed methods in diverse and realistic scenarios.

Experimental Validation and Software Contributions

The methodological advances presented in this thesis were supported by extensive experimental campaigns. For evolutionary algorithms, benchmark results confirmed the effectiveness of Wasserstein-based mechanisms in improving convergence and diversity. For Bayesian optimization, the proposed AGP extensions demonstrated robustness and scalability on both synthetic and real-world problems, including those with combinatorial structures. Moreover, the thesis has contributed two significant open-source implementations: (i) the Wasserstein-enabled evolutionary algorithms integrated in `pymoo`, and (ii) the AGP module for `BoTorch`, which extends the library's capabilities to multi-fidelity and combinatorial settings. These software releases ensure reproducibility and foster future developments by the research community.

Perspectives and Future Work

While this thesis provides several advances in the field of Bayesian and evolutionary optimization, many promising directions remain open for future investigation.

First, the integration of Wasserstein-based metrics within Bayesian optimization itself, e.g., for defining acquisition functions that measure distances in objective space, could unify evolutionary and probabilistic perspectives. Second, scaling the proposed methods to very high-dimensional or large-scale problems remains a challenge; sparse Gaussian processes, neural surrogate models, and distributed optimization schemes offer potential solutions. Finally, extending multi-objective, multi-fidelity frameworks to streaming or dynamic environments, where objectives and sources evolve over time, represents an exciting frontier with strong connections to continual learning and adaptive control.

Closing Remarks

Overall, this dissertation contributes to bridging key gaps in modern optimization by combining principles of efficiency, adaptability, and interpretability. It introduces new algorithms, reformulates classical problems, and provides open-source implementations that aim to make multi-objective and multi-fidelity optimization more practical and generalizable. Beyond specific technical results, the broader message of this work is that intelligent optimization, rooted in probabilistic modeling and geometry-aware diversity, can play a crucial role in making data-driven decision processes more fair, efficient, and sustainable.

References

[1] Akimoto, Y., Shimizu, T., and Yamaguchi, T. (2019). Adaptive objective selection for multi-fidelity optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 880–888.

[2] Archetti, F. and Candelieri, A. (2019). *Bayesian optimization and data science*, volume 849. Springer.

[3] Archetti, F., Ponti, A., Candelieri, A., and Sabbatella, A. (2025). Bayesian optimization, machine learning, and probabilistic numerics. In *AIP Conference Proceedings*, volume 3315, page 400049. AIP Publishing LLC.

[4] Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.

[5] Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2009). Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point. In *Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*, pages 87–102.

[6] Backurs, A., Dong, Y., Indyk, P., Razenshteyn, I., and Wagner, T. (2020). Scalable nearest neighbor search for optimal transport. In *International Conference on machine learning*, pages 497–506. PMLR.

[7] Balandat, M., Karrer, B., Jiang, D., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. (2020). Botorch: a framework for efficient monte-carlo bayesian optimization. *Advances in neural information processing systems*, 33:21524–21538.

[8] Baptista, R. and Poloczek, M. (2018). Bayesian optimization of combinatorial structures. In *International conference on machine learning*, pages 462–471. PMLR.

[9] Belakaria, S. and Deshwal, A. (2019). Max-value entropy search for multi-objective bayesian optimization. In *International Conference on Neural Information Processing Systems (NeurIPS)*.

[10] Belakaria, S., Deshwal, A., Jayakodi, N. K., and Doppa, J. R. (2020). Uncertainty-aware search framework for multi-objective bayesian optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10044–10052.

[11] Beume, N., Fonseca, C. M., Lopez-Ibanez, M., Paquete, L., and Vahrenhold, J. (2009). On the complexity of computing the hypervolume indicator. *IEEE Transactions on Evolutionary Computation*, 13(5):1075–1082.

[12] Blank, J. and Deb, K. (2020). Pymoo: Multi-objective optimization in python. *Ieee access*, 8:89497–89509.

[13] Bonilla, E. V., Chai, K., and Williams, C. (2007). Multi-task gaussian process prediction. *Advances in neural information processing systems*, 20.

[14] Bonneel, N., Peyré, G., and Cuturi, M. (2016). Wasserstein barycentric coordinates: histogram regression using optimal transport. *ACM Trans. Graph.*, 35(4):71–1.

[15] Bonneel, N., Rabin, J., Peyré, G., and Pfister, H. (2015). Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45.

[16] Bryson, D. E. and Rumpfkeil, M. P. (2016). Variable-fidelity surrogate modeling of lambda wing transonic aerodynamic performance. In *54th AIAA Aerospace Sciences Meeting*, page 0294.

[17] Candelieri, A. (2023). Resource allocation via bayesian optimization: an efficient alternative to semi-bandit feedback. In *International Conference on Numerical Computations: Theory and Algorithms*, pages 34–48. Springer.

[18] Candelieri, A. and Archetti, F. (2021). Sparsifying to optimize over multiple information sources: an augmented gaussian process based algorithm. *Structural and Multidisciplinary Optimization*, 64:239–255.

[19] Candelieri, A., Perego, R., and Archetti, F. (2021a). Green machine learning via augmented gaussian processes and multi-information source optimization. *Soft Computing*, 25(19):12591–12603.

[20] Candelieri, A., Ponti, A., and Archetti, F. (2021b). Risk aware optimization of water sensor placement. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 295–296.

[21] Candelieri, A., Ponti, A., and Archetti, F. (2022a). Bayesian optimization in wasserstein spaces. In *International Conference on Learning and Intelligent Optimization*, pages 248–262. Springer.

[22] Candelieri, A., Ponti, A., and Archetti, F. (2022b). Explaining exploration–exploitation in humans. *Big Data and Cognitive Computing*, 6(4):155.

[23] Candelieri, A., Ponti, A., and Archetti, F. (2022c). Safe-exploration of control policies from safe-experience via gaussian processes. In *International Conference on Learning and Intelligent Optimization*, pages 232–247. Springer.

[24] Candelieri, A., Ponti, A., and Archetti, F. (2023a). Generative models via optimal transport and gaussian processes. In *International Conference on Learning and Intelligent Optimization*, pages 135–149. Springer.

[25] Candelieri, A., Ponti, A., and Archetti, F. (2023b). Multi-objective and multiple information source optimization for fair & green machine learning. In *International Conference on Numerical Computations: Theory and Algorithms*, pages 49–63. Springer.

[26] Candelieri, A., Ponti, A., and Archetti, F. (2023c). Uncertainty quantification and exploration–exploitation trade-off in humans. *Journal of Ambient Intelligence and Humanized Computing*, 14(6):6843–6876.

[27] Candelieri, A., Ponti, A., and Archetti, F. (2023d). Wasserstein enabled bayesian optimization of composite functions. *Journal of Ambient Intelligence and Humanized Computing*, 14(8):11263–11271.

[28] Candelieri, A., Ponti, A., and Archetti, F. (2024a). A constrained-jko scheme for effective and efficient wasserstein gradient flows. In *International Conference on Learning and Intelligent Optimization*, pages 66–80. Springer.

[29] Candelieri, A., Ponti, A., and Archetti, F. (2024b). Fair and green hyperparameter optimization via multi-objective and multiple information source bayesian optimization. *Machine Learning*, 113(5):2701–2731.

[30] Candelieri, A., Ponti, A., and Archetti, F. (2025a). Bayesian optimization over the probability simplex. *Annals of Mathematics and Artificial Intelligence*, 93(1):77–91.

[31] Candelieri, A., Ponti, A., and Archetti, F. (2025b). Gaussian process regression over discrete probability measures: on the non-stationarity relation between euclidean and wasserstein squared exponential kernels. *Journal of Global Optimization*, pages 1–26.

[32] Candelieri, A., Ponti, A., and Archetti, F. (2025c). Multiple information source bayesian optimization.

[33] Candelieri, A., Ponti, A., Fersini, E., Messina, E., and Archetti, F. (2023e). Safe optimal control of dynamic systems: Learning from experts and safely exploring new policies. *Mathematics*, 11(20):4347.

[34] Candelieri, A., Ponti, A., Giordani, I., and Archetti, F. (2022d). Lost in optimization of water distribution systems: better call bayes. *Water*, 14(5):800.

[35] Candelieri, A., Ponti, A., Giordani, I., and Archetti, F. (2023f). On the use of wasserstein distance in the distributional analysis of human decision making under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 91(2):217–238.

[36] Candelieri, A., Ponti, A., Giordani, I., Bosio, A., and Archetti, F. (2023g). Distributional learning in multi-objective optimization of recommender systems. *Journal of Ambient Intelligence and Humanized Computing*, 14(8):10849–10865.

[37] Caruana, R. (1997). Multitask learning. *Machine learning*, 28(1):41–75.

[38] Chen, Y. and Krause, A. (2013). Near-optimal batch mode active learning and adaptive submodular optimization. In *International Conference on Machine Learning*, pages 160–168. PMLR.

[39] Coello Coello, C. A. and Reyes Sierra, M. (2004). A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *Mexican international conference on artificial intelligence*, pages 688–697. Springer.

[40] Contal, E., Buffoni, D., Robicquet, A., and Vayatis, N. (2013). Parallel gaussian process optimization with upper confidence bound and pure exploration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 225–240. Springer.

[41] Couckuyt, I., Deschrijver, D., and Dhaene, T. (2014). Fast calculation of multiobjective probability of improvement and expected improvement criteria for pareto optimization. *Journal of Global Optimization*, 60(3):575–594.

[42] Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2016). Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865.

[43] Cruz, A. F. and Hardt, M. (2023). Unprocessing seven years of algorithmic fairness. arXiv preprint arXiv:2306.07261.

[44] Csató, L. and Opper, M. (2001). Sparse representation for gaussian process models. In *Advances in neural information processing systems*, pages 444–450.

[45] Csató, L. and Opper, M. (2002). Sparse on-line gaussian processes. *Neural computation*, 14(3):641–668.

[46] Cuesta Ramirez, J., Le Riche, R., Roustant, O., Perrin, G., Durantin, C., and Gliere, A. (2022). A comparison of mixed-variables bayesian optimization approaches. *Advanced Modeling and Simulation in Engineering Sciences*, 9(1):6.

[47] Daulton, S., Balandat, M., and Bakshy, E. (2020). Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. *Advances in neural information processing systems*, 33:9851–9864.

[48] Daulton, S., Balandat, M., and Bakshy, E. (2023). Hypervolume knowledge gradient: a lookahead approach for multi-objective bayesian optimization with partial information. In *International Conference on Machine Learning*, pages 7167–7204. PMLR.

[49] Daulton, S., Eriksson, D., Balandat, M., and Bakshy, E. (2022). Multi-objective bayesian optimization over high-dimensional search spaces. In *Uncertainty in Artificial Intelligence*, pages 507–517. PMLR.

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

[51] Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization: theoretical advances and applications*, pages 105–145. Springer.

[52] Desautels, T., Krause, A., and Burdick, J. W. (2014). Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *Journal of Machine Learning Research*, 15:3873–3923.

[53] Désidéri, J.-A. (2012). Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318.

[54] Dowd, P. A. and Pardo-Igúzquiza, E. (2024). The many forms of co-kriging: A diversity of multivariate spatial estimators. *Mathematical Geosciences*, 56(2):387–413.

[55] Ehrgott, M. (2000). Weighted sum scalarization. In *Multicriteria Optimization*, pages 55–75. Springer.

[56] Emmerich, M., Beume, N., and Naujoks, B. (2005). An emo algorithm using the hypervolume measure as selection criterion. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 62–76. Springer.

[57] Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., and Poloczek, M. (2019). Scalable global optimization via local bayesian optimization. *Advances in neural information processing systems*, 32.

[58] Fan, Z., Li, W., Cai, X., Li, H., Wei, C., Zhang, Q., Deb, K., and Goodman, E. (2020). Difficulty adjustable and scalable constrained multiobjective test problem toolkit. *Evolutionary computation*, 28(3):339–378.

[59] Forrester, A. I., Sóbester, A., and Keane, A. J. (2007). Multi-fidelity optimization via surrogate modelling. *Proceedings of the royal society a: mathematical, physical and engineering sciences*, 463(2088):3251–3269.

[60] Frazier, P. I. (2018a). Bayesian optimization. In *Recent advances in optimization and modeling of contemporary problems*, pages 255–278. Informs.

[61] Frazier, P. I. (2018b). A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*.

[62] Frazier, P. I., Powell, W. B., and Dayanik, S. (2008). A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439.

[63] Garnett, R. (2023). *Bayesian optimization*. Cambridge University Press.

[64] Garrido-Merchán, E. C. and Hernández-Lobato, D. (2020). Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes. *Neurocomputing*, 380:20–35.

[65] Ghoreishi, S. F. and Allaire, D. (2019). Multi-information source constrained bayesian optimization. *Structural and Multidisciplinary Optimization*, 59:977–991.

[66] Ghoreishi, S. F., Molkeri, A., Arróyave, R., Allaire, D., and Srivastava, A. (2019). Efficient use of multiple information sources in material design. *Acta Materialia*, 180:260–271.

[67] Gramacy, R. B. (2020). *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences*. Chapman and Hall/CRC.

[68] Herbol, H. C., Poloczek, M., and Clancy, P. (2020). Cost-effective materials discovery: Bayesian optimization across multiple information sources. *Materials Horizons*, 7(8):2113–2123.

[69] Ho, N., Nguyen, X., Yurochkin, M., Bui, H. H., Huynh, V., and Phung, D. (2017). Multilevel clustering via wasserstein means. In *International conference on machine learning*, pages 1501–1509. PMLR.

[70] Hofmann, T., Schölkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, 36(3).

[71] Huband, S., Barone, L., While, L., and Hingston, P. (2005). A scalable multi-objective test problem toolkit. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 280–295. Springer.

[72] Hug, N. (2020). Surprise: A python library for recommender systems. *Journal of Open Source Software*, 5(52):2174.

[73] Irshad, F., Karsch, S., and Döpp, A. (2021). Expected hypervolume improvement for simultaneous multi-objective and multi-fidelity optimization. *arXiv preprint arXiv:2112.13901*, 10.

[74] Irshad, F., Karsch, S., and Döpp, A. (2024). Leveraging trust for joint multi-objective and multi-fidelity optimization. *Machine Learning: Science and Technology*, 5(1):015056.

[75] Ishibuchi, H., Masuda, H., Tanigaki, Y., and Nojima, Y. (2015). Modified distance calculation in generational distance and inverted generational distance. In *International conference on evolutionary multi-criterion optimization*, pages 110–125. Springer.

[76] Kandasamy, K., Dasarathy, G., Oliva, J., Schneider, J., and Poczos, B. (2019). Multi-fidelity gaussian process bandit optimisation. *Journal of Artificial Intelligence Research*, 66:151–196.

[77] Kandasamy, K., Dasarathy, G., Oliva, J. B., Schneider, J., and Póczos, B. (2016). Gaussian process bandit optimisation with multi-fidelity evaluations. *Advances in neural information processing systems*, 29.

[78] Kandasamy, K., Dasarathy, G., Schneider, J., and Póczos, B. (2017). Multi-fidelity bayesian optimisation with continuous approximations. In *International conference on machine learning*, pages 1799–1808. PMLR.

[79] Kandasamy, K., Neiswanger, W., Schneider, J., Poczos, B., and Xing, E. P. (2018). Neural architecture search with bayesian optimisation and optimal transport. *Advances in neural information processing systems*, 31.

[80] Kantorovich, L. (1942). On the transfer of masses (in russian). In *Doklady Akademii Nauk*, volume 37, page 227.

[81] Keerthi, S. and Chu, W. (2005). A matching pursuit approach to sparse gaussian process regression. *Advances in neural information processing systems*, 18.

[82] Kennedy, M. C. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13.

[83] Khan, N., Goldberg, D. E., and Pelikan, M. (2002). Multi-objective bayesian optimization algorithm. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, pages 684–684.

[84] Klise, K. A., Hart, D., Moriarty, D. M., Bynum, M. L., Murray, R., Burkhardt, J., and Haxton, T. (2017). Water network tool for resilience (wntr) user manual. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

[85] Knowles, J. (2006). Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE transactions on evolutionary computation*, 10(1):50–66.

[86] Krause, A. and Golovin, D. (2014). Submodular function maximization. *Tractability*, 3:71–104.

[87] Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR.

[88] Lam, R., Allaire, D. L., and Willcox, K. E. (2015). Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0143.

[89] Laumanns, M. and Ocenasek, J. (2002). Bayesian optimization algorithms for multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 298–307. Springer.

[90] Laumanns, M., Thiele, L., and Zitzler, E. (2006). An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169(3):932–942.

[91] Le Quy, T., Roy, A., Iosifidis, V., Zhang, W., and Ntoutsi, E. (2022). A survey on datasets for fairness-aware machine learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(3):e1452.

[92] Li, K., Deb, K., Zhang, Q., and Kwong, S. (2014). An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE transactions on evolutionary computation*, 19(5):694–716.

[93] Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816.

[94] Li, S., Xing, W., Kirby, R., and Zhe, S. (2020). Multi-fidelity bayesian optimization via deep neural networks. *Advances in Neural Information Processing Systems*, 33:8521–8531.

[95] Luo, G., Han, D., Zhang, Y., and Ruan, H. (2024). A digital twin for advancing battery fast charging based on a bayesian optimization-based method. *Journal of Energy Storage*, 93:112365.

[96] Mainini, L., Serani, A., Rumpfkeil, M. P., Minisci, E., Quagliarella, D., Pehlivan, H., Yildiz, S., Ficini, S., Pellegrini, R., Di Fiore, F., et al. (2022). Analytical benchmark problems for multifidelity optimization methods. *arXiv preprint arXiv:2204.07867*.

[97] March, A. and Willcox, K. (2012). Provably convergent multifidelity optimization algorithm not requiring high-fidelity derivatives. *AIAA journal*, 50(5):1079–1089.

[98] Meng, Y., Yan, X., Liu, W., Wu, H., and Cheng, J. (2020). Wasserstein collaborative filtering for item cold-start recommendation. In *Proceedings of the 28th ACM Conference on user modeling, adaptation and personalization*, pages 318–322.

[99] Miettinen, K. (1999). *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media.

[100] Mikkola, P., Martinelli, J., Filstroff, L., and Kaski, S. (2023). Multi-fidelity bayesian optimization with unreliable information sources. In *International Conference on Artificial Intelligence and Statistics*, pages 7425–7454. PMLR.

[101] Monge, G. (1781). Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pages 666–704.

[102] Moss, H. B., Leslie, D. S., Gonzalez, J., and Rayson, P. (2021). Gibbon: General-purpose information-based bayesian optimisation. *Journal of Machine Learning Research*, 22(235):1–49.

[103] Neiswanger, W., Yu, L., Zhao, S., Meng, C., and Ermon, S. (2022). Generalizing bayesian optimization with decision-theoretic entropies. *Advances in Neural Information Processing Systems*, 35:21016–21029.

[104] Nobar, M., Keller, J., Rupenyan, A., Khosravi, M., and Lygeros, J. (2024). Guided bayesian optimization: Data-efficient controller tuning with digital twin. *IEEE Transactions on Automation Science and Engineering*.

[105] Papenmeier, L., Nardi, L., and Poloczek, M. (2022). Increasing the scope as you learn: Adaptive bayesian optimization in nested subspaces. *Advances in Neural Information Processing Systems*, 35:11586–11601.

[106] Papenmeier, L., Nardi, L., and Poloczek, M. (2023). Bounce: Reliable high-dimensional bayesian optimization for combinatorial and mixed spaces. *Advances in Neural Information Processing Systems*, 36:1764–1793.

[107] Paria, B., Kandasamy, K., and Póczos, B. (2020). A flexible framework for multi-objective bayesian optimization using random scalarizations. In *Uncertainty in Artificial Intelligence*, pages 766–776. PMLR.

[108] Peherstorfer, B., Kramer, B., and Willcox, K. (2017). Combining multiple surrogate models to accelerate failure probability estimation with expensive high-fidelity models. *Journal of Computational Physics*, 341:61–75.

[109] Perrone, V., Donini, M., Zafar, M. B., Schmucker, R., Kenthapadi, K., and Archambeau, C. (2021). Fair bayesian optimization. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 854–863.

[110] Peyré, G., Cuturi, M., et al. (2019). Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

[111] Poloczek, M., Wang, J., and Frazier, P. (2017). Multi-information source optimization. *Advances in neural information processing systems*, 30.

[112] Ponti, A. and Archetti, F. (2023). The unreasonable effectiveness of optimal transport distance in the design of multi-objective evolutionary optimization algorithms. In *International Conference on Numerical Computations: Theory and Algorithms*, pages 151–164. Springer.

[113] Ponti, A., Candelieri, A., and Archetti, F. (2021a). A new evolutionary approach to optimal sensor placement in water distribution networks. *Water*, 13(12):1625.

[114] Ponti, A., Candelieri, A., and Archetti, F. (2021b). Optimal sensor placement by distribution based multiobjective evolutionary optimization. In *International Conference on Learning and Intelligent Optimization*, pages 315–332. Springer.

[115] Ponti, A., Candelieri, A., and Archetti, F. (2021c). A wasserstein distance based multiobjective evolutionary algorithm for the risk aware optimization of sensor placement. *Intelligent Systems with Applications*, 10:200047.

[116] Ponti, A., Candelieri, A., Giordani, I., and Archetti, F. (2023a). Intrusion detection in networks by wasserstein enabled many-objective evolutionary algorithms. *Mathematics*, 11(10):2342.

[117] Ponti, A., Giordani, I., Candelieri, A., and Archetti, F. (2023b). A leak localization algorithm in water distribution networks using probabilistic leak representation and optimal transport distance. In *International Conference on Learning and Intelligent Optimization*, pages 31–45. Springer.

[118] Ponti, A., Giordani, I., Candelieri, A., and Archetti, F. (2024). Wasserstein-enabled leaks localization in water distribution networks. *Water*, 16(3):412.

[119] Ponti, A., Giordani, I., Mistri, M., Candelieri, A., and Archetti, F. (2022a). The "unreasonable" effectiveness of the wasserstein distance in analyzing key performance indicators of a network of stores. *Big Data and Cognitive Computing*, 6(4):138.

[120] Ponti, A., Irpino, A., Candelieri, A., Bosio, A., Giordani, I., and Archetti, F. (2022b). Network vulnerability analysis in wasserstein spaces. In *International Conference on Learning and Intelligent Optimization*, pages 263–277. Springer.

[121] Quinonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959.

[122] Raissi, M. and Karniadakis, G. (2016). Deep multi-fidelity gaussian processes. *arXiv preprint arXiv:1604.07484*.

[123] Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and variational inference in deep latent gaussian models. In *International conference on machine learning*, volume 2, page 2. Citeseer.

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

[125] Rossman, L. A. et al. (1994). Epanet users manual.

[126] Ru, B., Alvi, A., Nguyen, V., Osborne, M. A., and Roberts, S. (2020). Bayesian optimisation over multiple continuous and categorical inputs. In *International Conference on Machine Learning*, pages 8276–8285. PMLR.

[127] Rubner, Y., Tomasi, C., and Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121.

[128] Sabbatella, A., Archetti, F., Ponti, A., Giordani, I., and Candelieri, A. (2024a). Bayesian optimization for instruction generation. *Applied Sciences*, 14(24):11865.

[129] Sabbatella, A., Ponti, A., Candelieri, A., and Archetti, F. (2024b). Bayesian optimization using simulation-based multiple information sources over combinatorial structures. *Machine Learning and Knowledge Extraction*, 6(4):2232–2247.

[130] Sabbatella, A., Ponti, A., Giordani, I., and Archetti, F. (2024c). A bayesian approach for prompt optimization in llms. In *International Conference on Learning and Intelligent Optimization*, pages 348–360. Springer.

[131] Sabbatella, A., Ponti, A., Giordani, I., Candelieri, A., and Archetti, F. (2024d). Prompt optimization in large language models. *Mathematics*, 12(6):929.

[132] Schmucker, R., Donini, M., Perrone, V., Zafar, M. B., and Archambeau, C. (2020). Multi-objective multi-fidelity hyperparameter optimization with application to fairness. In *NeurIPS Workshop on Meta-Learning*, volume 2.

[133] Schmucker, R., Donini, M., Zafar, M. B., Salinas, D., and Archambeau, C. (2021). Multi-objective asynchronous successive halving. *arXiv preprint arXiv:2106.12639*.

[134] Schreiter, J., Nguyen-Tuong, D., and Toussaint, M. (2016). Efficient sparsification for gaussian process regression. *Neurocomputing*, 192:29–37.

[135] Scutari, M., Panero, F., and Proissl, M. (2021). Achieving fairness with a simple ridge penalty. *arXiv preprint arXiv:2105.13817*.

[136] Scutari, M., Panero, F., and Proissl, M. (2022). Achieving fairness with a simple ridge penalty. *Statistics and Computing*, 32(5):77.

[137] Seeger, M., Steinke, F., and Tsuda, K. (2007). Bayesian inference and optimal design in the sparse linear model. In *Artificial Intelligence and Statistics*, pages 444–451. PMLR.

[138] Seeger, M. W., Williams, C. K., and Lawrence, N. D. (2003). Fast forward selection to speed up sparse gaussian process regression. In *International Workshop on Artificial Intelligence and Statistics*, pages 254–261. PMLR.

[139] Sener, O. and Koltun, V. (2018). Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31.

[140] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2015). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

[141] Shawe-Taylor, J. (2004). Kernel methods for pattern analysis. *Cambridge University Press google schola*, 2:181–201.

[142] Smola, A. J. and Bartlett, P. L. (2001). Sparse greedy gaussian process regression. In *Advances in neural information processing systems*, pages 619–625.

[143] Song, J., Chen, Y., and Yue, Y. (2019). A general framework for multi-fidelity bayesian optimization with gaussian processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3158–3167. PMLR.

[144] Suzuki, S., Takeno, S., Tamura, T., Shitara, K., and Karasuyama, M. (2020). Multi-objective bayesian optimization using pareto-frontier entropy. In *International Conference on Machine Learning*, pages 9279–9288. PMLR.

[145] Swersky, K., Snoek, J., and Adams, R. P. (2013). Multi-task bayesian optimization. *Advances in neural information processing systems*, 26.

[146] Takeno, S., Fukuoka, H., Tsukada, Y., Koyama, T., Shiga, M., Takeuchi, I., and Karasuyama, M. (2020). Multi-fidelity bayesian optimization with max-value entropy search and its parallelization. In *International Conference on Machine Learning*, pages 9334–9345. PMLR.

[147] Tornede, T., Tornede, A., Hanselle, J., Mohr, F., Wever, M., and Hüllermeier, E. (2023). Towards green automated machine learning: Status quo and future directions. *Journal of Artificial Intelligence Research*, 77:427–457.

[148] Van Veldhuizen, D. A. (1999). *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. Air Force Institute of Technology.

[149] Villani, C. (2009). The wasserstein distances. In *Optimal transport: old and new*, pages 93–111. Springer.

[150] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272.

[151] Wackers, J., Visonneau, M., Ficini, S., Pellegrini, R., Serani, A., and Diez, M. (2020). Adaptive n-fidelity metamodels for noisy cfd data. In *AIAA AVIATION 2020 FORUM*, page 3161.

[152] Wahba, G. (1990). *Spline models for observational data*, volume 59. Siam.

[153] Wang, J., Chiang, N., Gillette, A., and Peterson, J. L. (2024). A multifidelity bayesian optimization method for inertial confinement fusion design. *Physics of Plasmas*, 31(3).

[154] Wang, J., Clark, S. C., Liu, E., and Frazier, P. I. (2020). Parallel bayesian global optimization of expensive functions. *Operations Research*, 68(6):1850–1865.

[155] Wang, X., Jin, Y., Schmitt, S., and Olhofer, M. (2023). Recent advances in bayesian optimization. *ACM Computing Surveys*, 55(13s):1–36.

[156] Wang, Z., Gehring, C., Kohli, P., and Jegelka, S. (2018). Batched large-scale bayesian optimization in high-dimensional spaces. *arXiv preprint arXiv:1706.01445*.

[157] Wang, Z. and Jegelka, S. (2017). Max-value entropy search for efficient bayesian optimization. In *International Conference on Machine Learning*, pages 3627–3635. PMLR.

[158] Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

[159] Wilson, J., Hutter, F., and Deisenroth, M. (2018). Maximizing acquisition functions for bayesian optimization. *Advances in neural information processing systems*, 31.

[160] Wilson, J. T., Moriconi, R., Hutter, F., and Deisenroth, M. P. (2017). The reparameterization trick for acquisition functions. *arXiv preprint arXiv:1712.00424*.

[161] Wu, J. and Frazier, P. (2016). The parallel knowledge gradient method for batch bayesian optimization. *Advances in neural information processing systems*, 29.

[162] Wu, J. and Frazier, P. I. (2018). Continuous-fidelity bayesian optimization with knowledge gradient.

[163] Wu, J., Toscano-Palmerin, S., Frazier, P. I., and Wilson, A. G. (2020). Practical multi-fidelity bayesian optimization for hyperparameter tuning. In *Uncertainty in Artificial Intelligence*, pages 788–798. PMLR.

[164] Yang, K., Emmerich, M., Deutz, A., and Bäck, T. (2019). Efficient computation of expected hypervolume improvement using box decomposition algorithms. *Journal of Global Optimization*, 75(1):3–34.

[165] Zafar, M. B., Valera, I., Gomez-Rodriguez, M., and Gummadi, K. P. (2019). Fairness constraints: A flexible approach for fair classification. *Journal of Machine Learning Research*, 20(75):1–42.

[166] Zhang, Q. and Li, H. (2007). Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731.

[167] Zhang, R. and Golovin, D. (2020). Random hypervolume scalarizations for provable multi-objective black box optimization. In *International Conference on Machine Learning*, pages 11096–11105. PMLR.

[168] Zhang, Y., Hoang, T. N., Low, B. K. H., and Kankanhalli, M. (2017). Information-based multi-fidelity bayesian optimization. In *NIPS workshop on Bayesian optimization*, volume 49. Journal of Machine Learning Research JMLR. org Cambridge, MA.

[169] Zhang, Y. and Yang, Q. (2018). An overview of multi-task learning. *National Science Review*, 5(1):30–43.

[170] Zhang, Y. and Yang, Q. (2021). A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 34(12):5586–5609.

[171] Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., and Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and evolutionary computation*, 1(1):32–49.

[172] Zhuang, Y., Chen, X., and Yang, Y. (2022). Wasserstein k-means for clustering probability distributions. *Advances in Neural Information Processing Systems*, 35:11382–11395.

[173] Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195.

[174] Zitzler, E., Laumanns, M., and Thiele, L. (2001). Spea2: Improving the strength pareto evolutionary algorithm. *TIK report*, 103.