

Bayesian preference elicitation for decision support in multi-objective optimization

Felix Huber^{1,2} | Sebastian Rojas Gonzalez³ | Raul Astudillo⁴

¹Institute of Applied Analysis and Numerical Simulation, University of Stuttgart, Germany

²Institute for Parallel and Distributed Systems, University of Stuttgart, Germany

³Surrogate Modeling Lab, Ghent University - imec, Belgium

⁴Department of Computing and Mathematical Sciences, California Institute of Technology, USA

Correspondence

Corresponding author: Sebastian Rojas Gonzalez

Email: sebastian.rojasgonzalez@ugent.be

Abstract

We present a novel approach to help decision-makers efficiently identify preferred solutions from the Pareto set of a multi-objective optimization problem. Our method uses a Bayesian model to estimate the decision-maker's utility function based on pairwise comparisons. Aided by this model, a principled elicitation strategy selects queries interactively to balance exploration and exploitation, guiding the discovery of high-utility solutions. The approach is flexible: It can be used interactively or a posteriori after estimating the Pareto front through standard multi-objective optimization techniques. Additionally, at the end of the elicitation phase, it generates a reduced menu of high-quality solutions, simplifying the decision-making process. Through experiments on test problems with up to nine objectives, our method demonstrates superior performance in finding high-utility solutions with a small number of queries. We also provide an open-source implementation of our method to support its adoption by the broader community.

KEY WORDS

Bayesian optimization, preference elicitation, multi-objective optimization, decision support

1 | INTRODUCTION

Many real-world problems require optimizing multiple *objectives* or *criteria* simultaneously. For example, in operations management, *decision-makers* (DMs) often aim to maximize expected return while minimizing risk (Winston, 2004). Similarly, in aerodynamic engineering, a common goal is to maximize the lift coefficient while minimizing the drag coefficient of an airfoil design (He et al., 2020). In such cases, a single solution that optimizes all objectives rarely exists; for instance, solutions with higher expected returns typically come with increased risk. As a result, a trade-off between objectives is inevitable, and DMs need to understand this trade-off to make informed choices.

A common approach in these scenarios is to estimate the *Pareto-optimal set* (also simply referred to as the *Pareto set*), which consists of solutions that cannot be improved in all objectives simultaneously; the image of the Pareto set is usually termed the *Pareto front*. Each solution in this set represents a different optimal trade-off between the competing objectives, providing DMs with a range of options. However, when the Pareto set is large, it becomes challenging to select a single solution for implementation (Cheikh et al., 2010). This issue is further exacerbated as the size of the Pareto set typically grows exponentially with the number of objectives, making it difficult to approximate and choose among the solutions effectively (Sanders, 2024).

In the literature, methods that approximate the entire Pareto set are known as *a posteriori* methods and are among the most widely studied. In contrast, *a priori* methods rely on DMs to specify their preferences before the optimization process begins, typically by combining the objectives into a single scalar-valued function. While this approach simplifies the selection of a final solution, it assumes that the DM can fully articulate their preferences beforehand, which is rarely the case in practice. Indeed, before the optimization process begins, DMs often have little knowledge about the trade-offs of the different criteria and the objective space in general (Misitano et al., 2021).

To address these limitations, a broad range of *interactive* methods have been proposed in the literature (Xin et al., 2018). These methods aim to progressively elicit the DM's preferences during the optimization process. By presenting a series of queries, they adaptively focus on the most relevant regions of the Pareto front. While this approach can be more flexible and

user-friendly, most interactive methods either make strong assumptions about the DM’s underlying preferences and responses or rely on data-inefficient heuristics to select the DM queries, often leading to suboptimal or biased solutions in practice.

In this paper, we propose a novel approach to support decision-making in multi-objective optimization, addressing several major drawbacks of existing methods. Specifically, we leverage non-parametric Bayesian preference learning to model the DM’s preferences, along with a principled elicitation strategy to select queries. Our approach offers three key advantages over existing methods:

1. It does not rely on strong assumptions about the DM’s utility function and can accommodate noisy responses.
2. It is query-efficient, providing a mechanism to balance exploration and exploitation while effectively eliciting the DM’s utility function.
3. It offers a natural scheme for generating a menu of user-specified size, consisting of diverse high-quality solutions, at the end of the elicitation phase.

Overall, our work provides principled and practical framework for decision support in general multi-objective problems via efficient and robust elicitation of the DM preferences. We also provide an implementation of our method to reproduce our experiments and support its use by the broader community[†].

2 | RELATED WORK

The literature on incorporating preference information into multi-objective optimization is extensive, with contributions spanning multiple fields such as multi-attribute utility theory, evolutionary algorithms, and Bayesian optimization (see, e.g., Wallenius et al. 2008, Xin et al. 2018, and Abdolshah et al. 2019). These research areas have traditionally evolved independently for the most part, leading to fragmented methodologies and inconsistent terminology. Despite recent attempts to bridge these fields (e.g., Branke 2016), significant challenges remain, particularly in terms of integrating preference information in a principled manner. Below, we discuss these various lines of research, focusing on connections to our work.

Originating from the operations research community, multi-attribute utility theory provides frameworks for eliciting and modeling the DM’s preferences (Keeney & Raiffa, 1993; Miettinen, 1999; Abbas, 2018). These methods typically involve constructing a utility function that captures the DM’s preferences across multiple attributes or criteria. However, as Miettinen (1999) highlights, most traditional approaches focus on problems with discrete or finite decision spaces, limiting their applicability to continuous, high-dimensional optimization problems. In cases involving continuous decision spaces, existing methods often yield a single estimate of the DM’s utility function, which can be problematic due to the substantial uncertainty that typically remains regarding the DM’s preferences.

Beyond multi-attribute utility theory, the operations research community has also explored preference-based optimization techniques that are more akin to classical gradient-based methods (Miettinen, 1999). Like our approach, these methods typically assume that an underlying utility function represents the DM’s preferences. However, they often make strong assumptions about its form, such as a specific parametric structure or concavity (see e.g., Zionts & Wallenius 1983; Mackin et al. 2011). Moreover, they also typically require the DM to respond to more challenging queries, such as those based on marginal substitution rates, and assume that the user responses are noise-free, which is typically not true in practice. Our work circumvents these limitations by employing a Bayesian approach to preference learning, which is both non-parametric and capable of efficiently handling noisy user feedback, thus making it more broadly applicable.

The incorporation of user preferences to guide the search process in optimizing complex, multi-modal landscapes has been widely studied by the evolutionary computation community (Branke, 2008; Deb et al., 2010; Branke et al., 2016; Tomczyk & Kadziński, 2019). These algorithms use preference information to steer the evolutionary search toward the desired regions of the Pareto front. However, most of these methods rely on heuristic mechanisms to select DM queries, which often require numerous interactions with the DM. In contrast, our approach utilizes Bayesian preference learning and decision theory to more efficiently elicit the DM’s preferences. For a comprehensive review of preference-based evolutionary methods, we refer to Xin et al. (2018). Recent advances such as K. P. Taylor et al. (2024) have proposed hybrid approaches to combine Bayesian learning and evolutionary optimization more effectively.

[†] <https://github.com/qres/BPE4MOO>

More recently, with the growing interest in artificial intelligence for decision-making, the machine learning and related communities have developed a wide array of preference learning techniques (Chajewska & Koller, 2000; Boutilier, 2002; Chu & Ghahramani, 2005; González et al., 2017; Benavoli & Azzimonti, 2024). However, most of this research has been conducted outside the scope of multi-objective optimization. An exception is a line of work in Bayesian optimization (Garnett, 2023), a well-studied framework for optimizing functions with expensive evaluations. Given its focus on sample-efficiency, the Bayesian optimization community has explored schemes to incorporate user preferences in multi-objective contexts to accelerate optimization (Abdolshah et al., 2019; Astudillo & Frazier, 2020; Gaudrie et al., 2020; Gibson et al., 2022; Lin et al., 2022; Heidari et al., 2023). Other efforts include applications in engineering settings, such as Lepird et al. (2015), and interactive surrogate-based design for energy systems (Aghaei Pour et al., 2022).

Among the various research directions discussed above, our work is closely aligned with a few key papers, which we describe in more detail below. Tomczyk & Kadziński (2019) also proposed a Bayesian approach to estimate the DM’s utility function, but in contrast to our work, they used a parametric prior distribution based on Chebyshev scalarizations, which were then used within an evolutionary algorithm. Queries were selected uniformly at random from populations generated by the evolutionary algorithm. Our work is also related to K. Taylor et al. (2021), which alternated between preference elicitation stages, where a Bayesian preference-based optimization approach was used to find DM-preferred solutions within populations generated by the evolutionary algorithm, and evolutionary stages, where the solutions from the previous stage were used as reference points to generate the next generation of solutions. In contrast, our method directly aims to identify informative queries to elicit the DM’s preferences without requiring an auxiliary evolutionary algorithm. Moreover, we model the DM’s utility function directly over the objective space, rather than solely in the decision space (as done in K. Taylor et al. 2021), which, as discussed in Section 4.4.2, is advantageous in most cases.

Finally, our work builds on and extends prior work by the authors. Astudillo & Frazier (2020) consider multi-objective Bayesian optimization with preference learning in settings where evaluating the objective function f is expensive, with a primary focus on optimizing f efficiently. In contrast, our work centers on efficiently eliciting the decision-maker’s utility function u through preference queries. The preference elicitation strategy we adopt, qEUBO, was originally proposed in Lin et al. (2022) as part of a framework that alternates between learning f and u in a multi-objective Bayesian optimization context. Here, we broaden the integration of multi-objective optimization and preference learning by allowing the objective functions to be known or cheaply computable. We examine both interactive and a posteriori variants of our method, and explore modeling u either over the decision space or the objective space. Our main contribution is a principled and practical framework for decision support in general multi-objective problems, particularly when preferences must be elicited efficiently through interaction. We also note that qEUBO was analyzed in Astudillo et al. (2023) in a pure preference-based Bayesian optimization setting, without the multi-objective context considered here.

3 | PROBLEM SETTING

Let $f: \mathbb{X} \rightarrow \mathbb{R}^m$ denote a continuous objective function over a compact decision space $\mathbb{X} \subset \mathbb{R}^d$. As is standard in multi-objective optimization, for any pair of decisions $x, x' \in \mathbb{X}$, we say that x Pareto-dominates x' , denoted by $x \succ_f x'$, if $f_j(x) \geq f_j(x')$ for all $j = 1, \dots, m$, with at least one strict inequality. The set $\mathbb{X}_f^* := \{x \in \mathbb{X} : \nexists x' \in \mathbb{X} \text{ such that } x' \succ_f x\}$ is termed the Pareto set of f .

We consider the problem of assisting a DM in identifying their most preferred solution within \mathbb{X}_f^* . Specifically, we assume that the DM’s preferences are governed by an underlying utility function $u: \mathbb{R}^m \rightarrow \mathbb{R}$, and our objective is to help the DM approximately solve the problem $\operatorname{argmax}_{x \in \mathbb{X}} u(f(x))$. We note that if u is monotone non-decreasing, at least one solution to this problem is guaranteed to lie in \mathbb{X}_f^* (Miettinen, 1999). Our approach, however, does not rely on this property and can also be applied when u is non-monotone.[‡]

A key challenge is that u cannot be observed directly. Instead, the only information available to our algorithm comes from the DM’s expressed preferences, which we assume are provided through (potentially noisy) *pairwise comparisons* between solutions; while we focus on pairwise comparisons in this work, our approach can easily be extended to best-out-of- q responses (see Astudillo et al. 2023 for details). Formally, we assume that we are allowed to interact with the DM up to N times. At each interaction, $n = 1, \dots, N$, our algorithm selects a pair of objective vectors $y_{n,1}, y_{n,2} \in \mathbb{R}^m$ and observes the DM’s response, encoded as $r_n \in \{1, 2\}$, where $r_n = 1$ if the DM prefers $y_{n,1}$ over $y_{n,2}$, and $r_n = 2$ in the opposite case. We introduce the notation $Y_n = (y_{n,1}, y_{n,2}) \in \mathbb{R}^m \times \mathbb{R}^m$ and refer to Y_n as the n -th query. In this work, we focus on *realizable* queries, i.e., queries of the

[‡] Nonetheless, in multi-objective optimization the true utility function is typically required to be monotone to ensure consistency with Pareto compliance.

Algorithm 1 Bayesian preference elicitation for decision support in multiobjective optimization

```

1: Sample initial dataset  $\mathcal{D}_0$ 
2: for  $n = 1, \dots, N$  do
3:   Compute posterior distribution on  $u$  given  $\mathcal{D}_{n-1}$ 
4:   Compute  $Y_n = f(X_n)$ , where  $X_n \in \operatorname{argmax}_{X \in \mathbb{X}^2} \operatorname{qEUBO}_{n-1}(X)$ 
5:   Observe  $r(Y_n)$ , the DM's response for query  $Y_n$ 
6:   Update dataset  $\mathcal{D}_n = \mathcal{D}_{n-1} \cup \{(Y_n, r(Y_n))\}$ 
7: end for
8: return  $\operatorname{argmax}_{x_1, \dots, x_k \in \mathbb{X}} \mathbf{E}_N[\max_{i=1, \dots, k} u(f(x_i))]$ 

```

form $Y_n = (f(x_{n,1}), f(x_{n,2}))$ for some $x_{n,1}, x_{n,2} \in \mathbb{X}$. Similarly, we denote the corresponding decision pairs as $X_n = (x_{n,1}, x_{n,2}) \in \mathbb{X}^2$ and, making slight abuse of notation, we write $Y_n = f(X_n)$.

Several related formulations have been proposed in the literature. Here, we briefly outline how our approach differs from existing ones. First, we make no specific assumptions about the form of u . Many prior works assume that u belongs to a specific parametric family (Deb et al., 2010; Tomczyk & Kadziński, 2019) or possesses certain properties, such as concavity (Zionts & Wallenius, 1983; Mackin et al., 2011). While our method is flexible enough to approximately incorporate such information if available, we do not impose restrictive assumptions on u .

Furthermore, unlike most existing algorithms that assume noise-free user responses (Zionts & Wallenius, 1983; Deb et al., 2010), we explicitly account for the possibility of noisy feedback. This allows our model to better handle real-world scenarios where the DM's preferences may not be consistently expressed.

Finally, while traditional approaches often focus on identifying a single optimal solution to present to the DM, we recognize that uncertainty in the DM's preferences typically remains. To address this, we propose generating a menu of high-quality solutions, taking into account the uncertainty over u that remains after interacting with the DM. By offering a diverse set of potential solutions, our approach aims to mitigate the risk associated with presenting only one option, providing the DM with a richer set of choices that better reflect their uncertain preferences.

4 | METHOD

Our approach consists of three main components: A probabilistic model of the DM's utility (Section 4.1), a preference elicitation strategy (Section 4.2), and a menu selection strategy (Section 4.3). The probabilistic model allows us to infer the solutions most preferred by the DM and is updated after each interaction. The preference elicitation strategy uses an acquisition function derived from the probabilistic model to quantify the potential benefit of asking a particular query. Finally, the menu selection strategy, applied at the end of the elicitation phase, selects a small set of high-quality decisions based on our current understanding of the DM's preferences as encoded by the model. We discuss each of these components in detail below. In addition, Section 4.4 presents alternative design choices that yield variants of our algorithm, which may be more suitable for specific scenarios. Our approach is summarized in Algorithm 1.

4.1 | Probabilistic model of the DM's utility

We model the DM's latent utility function using a Gaussian process (GP) for preference learning using pairwise comparisons (Chu & Ghahramani, 2005). This is done by specifying a GP prior distribution and a likelihood function that accounts for noise in the DM's responses. Recall that $Y_n = (y_{n,1}, y_{n,2}) \in \mathbb{R}^m \times \mathbb{R}^m$ represents the n -th query posed to the DM and $r_n \in \{1, 2\}$ denotes the corresponding response. To model the likelihood of the DM's responses, we adopt a logistic noise model defined as follows:

$$\mathbf{P}(r_n = i \mid Y_n) = \frac{e^{u(y_{n,i})/\lambda}}{e^{u(y_{n,1})/\lambda} + e^{u(y_{n,2})/\lambda}}, \quad i = 1, 2 \quad (1)$$

where $\lambda > 0$ is the noise level parameter. We note that this model can recover a noise-free likelihood by letting $\lambda \rightarrow 0$.

Let \mathcal{D}_0 denote a potentially empty initial dataset and $\mathcal{D}_{n-1} = \mathcal{D}_0 \cup \{(X_l, r(X_l))\}_{l=1}^{n-1}$ denote the data collected before the n -th interaction with the DM. The posterior distribution over u given \mathcal{D}_{n-1} can be updated using Bayes' rule. However, due to the combination of our choice of GP prior and logistic likelihood, the posterior is not analytically tractable. Instead, it can be approximated using methods such as the Laplace approximation (Cheikh et al., 2010) or variational inference (Nguyen et al., 2021). In our experiments, we employ the latter approach due to its computational efficiency and scalability; see Appendix A.1 for a description of the approach.

4.2 | Preference elicitation strategy

We select the queries to be presented to the DM by maximizing an acquisition function constructed from the current posterior distribution. In our experiments, we use the qEUBO acquisition function proposed by Lin et al. (2022) and Astudillo et al. (2023) due to its tractability, strong empirical performance, and theoretical guarantees. However, other acquisition functions could be employed as well. For completeness, we provide a brief description of the qEUBO acquisition function below.

The qEUBO acquisition function is defined as

$$\text{qEUBO}_{n-1}(X) = \mathbf{E}_{n-1} \left[\max_{i=1,2} u(f(x_i)) \right], \quad (2)$$

where the subscript $n-1$ indicates that the expectation is taken with respect to the posterior distribution of u after $n-1$ interactions with the DM. At each interaction, we select the query to be presented by maximizing this acquisition function. Specifically, during the n -th interaction, we set $Y_n = f(X_n)$, with

$$X_n \in \operatorname{argmax}_{X \in \mathbb{X}^2} \text{qEUBO}_{n-1}(X). \quad (3)$$

We note that maximizing the acquisition function requires potentially many evaluations of f . In many scenarios, this is computationally feasible. When f is expensive to evaluate, a potential alternative is to use a separate probabilistic model over f , as suggested by Lin et al. (2022). However, our focus is on problems where evaluating f enough times to maximize the acquisition function is feasible.

In Section 4.4.2 we discuss an alternative modeling approach, where the DM's responses are used to infer the mapping $x \mapsto u(f(x))$ directly, instead of u . This approach eliminates the need to evaluate f during the acquisition function maximization step but comes with a trade-off: It requires modeling a function over a potentially higher-dimensional decision space, since the number of decision variables d is typically larger than the number of objectives m .

4.3 | Menu selection strategy

An important advantage of our probabilistic approach over most existing interactive methods is that it provides a natural scheme for generating a menu of high-quality items to present to the DM after the elicitation phase is complete. This menu is designed to maximize the expected utility that the DM would receive when selecting the best item according to their underlying utility function, given the remaining uncertainty after the elicitation phase. Specifically, our approach selects the menu items by maximizing the expected utility of the best choice, defined as

$$\max_{x_1, \dots, x_k \in \mathbb{X}} \mathbf{E}_n \left[\max_{i=1, \dots, k} u(f(x_i)) \right], \quad (4)$$

where k is the desired menu size.

It is noteworthy that, for $k = 2$, the above expression coincides with the qEUBO acquisition function. This dual interpretation of qEUBO, both as an acquisition function during the elicitation phase and as a strategy for selecting menus afterward, has been previously observed in the literature (Viappiani & Boutilier, 2010). Although this is not critical to our setting, it intuitively implies that the queries presented by our algorithm are composed of high-quality points, similar to those we would select if we were to finalize the elicitation phase at that particular interaction.

4.4 | Alternative design choices

Having outlined the main version of our algorithm, we now explore several alternative design choices that lead to different variants. These alternatives may be advantageous in specific scenarios and can enhance the algorithm’s applicability.

4.4.1 | A priori and a posteriori versions of our algorithm

Our algorithm, as described above, can be naturally categorized as an interactive multi-objective optimization method. It is not an a priori method, as it does not require the user to specify a single utility function representing their preferences before the optimization process begins. Similarly, it is not an a posteriori method, as it does not rely on building an estimate of the Pareto front of f prior to the elicitation phase. Instead, the algorithm directly aims to find the point that maximizes the DM’s utility $u(f(x))$ by interactively querying the DM. Below, we discuss how our approach can be adapted to function as both an a priori and a posteriori method in specific scenarios.

A priori version with an informative prior

In certain cases, it is possible to construct an informative Bayesian prior distribution over the DM’s utility function using historical data or transfer learning. When such a prior is available, our approach can be adapted to operate as an a priori method. Specifically, we can generate a menu for the DM by following the procedure described in Section 4.3, computing the expectation with respect to the prior distribution without requiring any further interaction with the DM. This allows the DM to make an informed decision based on the prior knowledge alone, bypassing the need for an interactive elicitation phase.

A posteriori version with a precomputed estimate of the Pareto set

In general, accurately computing the entire Pareto set can be challenging. However, there may be scenarios where a reasonable approximation can be obtained using traditional multi-objective optimization techniques, such as evolutionary algorithms. In these situations, our approach can be used as an a posteriori method to help the DM identify their most preferred solution among points in the approximate Pareto set. Let $\hat{\mathbb{X}}_f^*$ denote this approximation. This can be achieved by simply replacing \mathbb{X} with $\hat{\mathbb{X}}_f^*$ in Algorithm 1 (i.e., maximizing the acquisition function and selecting the menu using only points in $\hat{\mathbb{X}}_f^*$).

4.4.2 | Modeling preferences over decision vectors vs. objective vectors

Since the DM typically expresses preferences over objective vectors $f(x)$, it is natural to model utility as a function over the objective space, i.e., $y \mapsto u(y)$ for $y \in \mathbb{R}^m$. Alternatively, one can model preferences directly over the decision space by considering the composite mapping $x \mapsto u(f(x))$ with $x \in \mathbb{X}$.

In general, modeling in objective space offers several advantages: (i) the number of decision variables often exceeds the number of objectives, making the lower-dimensional objective space more tractable; (ii) it allows for the incorporation of structural prior knowledge, such as monotonicity of the utility function (see Section 4.4.3); and (iii) it decouples preference modeling from the complexities of the objective function, avoiding the need for the utility model to capture nonlinearities or irregularities introduced by the mapping from decision to objective space.

Nonetheless, modeling in decision space can be beneficial in specific scenarios: For instance, when the number of decision variables is smaller than the number of objectives, or when evaluating f is computationally expensive and should be avoided during acquisition function optimization. This formulation also permits the use of gradient-based optimization, as a GP defined over decision space is directly differentiable. In contrast, modeling over the objective space typically requires zeroth-order optimization methods, as gradients of f are often unavailable or expensive to compute.

In our experiments in Section 5, we compare both modeling strategies to offer practical guidance for selecting the most suitable approach depending on the problem setting.

4.4.3 | Monotonicity of the utility function

In most settings, the decision-maker’s utility function $u(y)$ is known to be monotone increasing, reflecting a preference for improving all objectives. This structural assumption offers valuable prior knowledge that, if properly incorporated, can enhance

Algorithm 2 Generate points to induce monotonicity

Require: Dataset $\mathcal{Y}_n = \{y_{l,i} : l = 1, \dots, n, i = 1, 2\}$, number of objectives m , expansion parameter $\delta > 0$

- 1: Compute bounding box of current dataset \mathcal{D}_n :
- 2: $y_{\min,j} = \min_{y \in \mathcal{Y}_n} y_j$ for $j = 1, \dots, m$
- 3: $y_{\max,j} = \max_{y \in \mathcal{Y}_n} y_j$ for $j = 1, \dots, m$
- 4: Normalize objective space to $[0, 1]^m$ using y_{\min} and y_{\max}
- 5: Draw two points y_1, y_2 uniformly from an expanded box $[-\delta, 1 + \delta]^m$, such that y_1 dominates y_2
- 6: **return** $(y_{\min} + (y_{\max} - y_{\min})y_1, y_{\min} + (y_{\max} - y_{\min})y_2)$

both modeling accuracy and query efficiency. However, GP priors almost surely generate non-monotone sample paths, making it challenging to enforce monotonicity exactly within the model.

To address this, we adopt a simple yet effective data augmentation strategy to approximately impose monotonicity when modeling utility over the objective space. Specifically, we augment the training set with *virtual* queries of the form $(Y, r(Y))$, where $Y = (y_1, y_2)$ and y_1 dominates y_2 , with $r(Y) = 1$ (i.e., y_1 is preferred), as described in Algorithm 2. These synthetic comparisons encourage the model to behave monotonically in the vicinity of the augmented pairs, guiding query selection toward regions closer to the Pareto front.

Our experiments show that this simple strategy can yield meaningful performance gains. Nonetheless, other techniques for enforcing monotonicity, such as constrained GPs or basis-function approximations (Riihimäki & Vehtari, 2010; Da Veiga & Marrel, 2012; Maatouk & Bay, 2017; López-Lopera et al., 2018), remain a promising avenue for future work.

5 | EXPERIMENTS

In this section, we assess the ability of our approach to identify high-utility decisions based on a DM’s unknown utility function. To this end, we use well-known synthetic test functions from the multi-objective optimization literature, as well as a practical application example, each paired with a distinct utility function (described in Section 5.1). Our primary experiments, detailed in Section 5.4.1, evaluate the performance of our approach across various decision space dimensions and number of objectives (see Table 1). These experiments also explore two key design choices: Modeling the DM’s preferences over decision space versus objective space, and using our approach interactively versus a posteriori. By considering each possible combination of these two design choices, we obtain four versions of our algorithm. In our plots, we label these design choices as follows:

- *int*: Refers to the interactive version of our algorithm.
- *post*: Refers to the a posteriori version of our algorithm.
- *dec*: Refers to the version of our algorithm that models the utility function over the decision space.
- *obj*: Refers to the version of our algorithm that models the utility function over the objective space.

Thus, for example, the interactive version of our algorithm that models the utility function over objective space is labeled as *int-obj*. Additionally, we perform secondary experiments that examine the effects of approximately enforcing monotonicity in the probabilistic model (Section 5.4.2), noise in the DM’s responses (Section 5.4.3), and varying menu sizes (Section 5.4.4).

As discussed earlier, there is a plethora of methods for multi-objective optimization with preferences; comparing our approach against these methods is challenging for several reasons. For example, (1) there are many different ways of collecting and processing preference information (e.g., the frequency of interaction or type of information), (2) the performance evaluation of these methods is often linked to the modeling assumptions and thus method-dependent (Afsar et al., 2021), (3) the so-called *most preferred* solution can be selected in multiple ways, and (4) not all methods are scalable in both decision and objective spaces. As a comparison with an existing state-of-the-art method, we note that one of the secondary versions of our algorithm aligns closely with the method proposed by K. Taylor et al. (2021). This version of our algorithm models the DM’s utility over decision space and conducts the elicitation phase a posteriori over an approximation of the Pareto front (labeled as *post-dec*). This can be interpreted as the algorithm in K. Taylor et al. (2021), where instead of using the Expected Improvement, as proposed in

Objective function (d, m)	Utility function
DTLZ7 (5,3) disconnected, mixed Pareto front	$u(y) = \sum_{i=1}^m y_i$
DTLZ2 (9,6) continuous and concave Pareto front	$u(y) = -\left(\sum_{i=1}^m (z_i - y_i)^3\right)^{\frac{1}{3}}$ with $z_i = \begin{cases} 0.2 & \text{if } i \text{ is even,} \\ 0 & \text{otherwise} \end{cases}$
WFG3 (14,9) linear and degenerate Pareto front	$u(y) = -\theta^{-1} \log\left(\sum_{i=1}^m \exp(-\theta y_i)\right)$ for $\theta = 4$
Car Cab Design (7,9) non-convex Pareto front	$u(y) = \sum_{i=1}^m h_i(y_i)$ with piecewise linear functions h_i

TABLE 1 Objective functions and their corresponding utility functions used in our empirical evaluation.

their paper, all queries are collected using the qEUBO acquisition function, which has been shown to outperform Expected Improvement (Astudillo et al., 2023).

5.1 | Test functions

We evaluate the proposed framework on a suite of well-established multi-objective optimization benchmarks that present diverse characteristics and challenges (Huband et al., 2006). Specifically, we use DTLZ7 (5,3), DTLZ2 (9,6), WFG3 (14,9), and Car Cab Design (7,9), where the values in parentheses denote the dimensionality of the decision and objective spaces, respectively. This selection allows us to assess the framework’s performance across a range of problem scales and structural complexities. Formal definitions of the synthetic test functions and the Car Cab Design problem can be found in Huband et al. (2006) and Lin et al. (2022), respectively.

Each benchmark is paired with a distinct utility function to model the DM’s preferences (see Table 1). These utility functions are chosen to be monotonic over the objective space, ensuring that the global optimum lies on the Pareto set. Moreover, they represent commonly used classes of utility models across various domains. For example, DTLZ7 is coupled with a linear utility function, a widely adopted and interpretable model (Keeney & Raiffa, 1993; Miettinen, 1999; Abbas, 2018). DTLZ2 uses a 3-norm utility function relative to an ideal point, inspired by Tomczyk & Kadziński (2019). WFG3 is paired with an exponential utility function, another popular choice in the literature (Astudillo & Frazier, 2020; Abbas, 2018). For the Car Cab Design problem, we adopt a piecewise linear utility function following the setup in Lin et al. (2022).

5.2 | Performance metric

We evaluate performance using *regret*, defined as the difference between the true optimal utility and the utility of the best point in a menu of size k selected from a set \mathcal{X} :

$$\text{regret}(\mathcal{X}, k) = u(x^*) - \max_{i=1, \dots, k} u(\widehat{x}_i^*),$$

where $x^* = \operatorname{argmax}_{x \in \mathbb{X}} u(x)$ is the true optimal point, and $\widehat{x}_1^*, \dots, \widehat{x}_k^*$ are the points in \mathcal{X} that maximize the expected utility of the best item in the menu:

$$(\widehat{x}_1^*, \dots, \widehat{x}_k^*) \in \operatorname{arg} \max_{x_1, \dots, x_k \in \mathcal{X}} \mathbf{E}_n \left[\max_{i=1, \dots, k} u(f(x_i)) \right].$$

In our main experiments (Section 5.4.1), we set $\mathcal{X} = \{x_{l,i} : l = 1, \dots, n, i = 1, 2\}$, i.e., the set of all queried points up to step n . This ensures a fair comparison between the interactive and a posteriori versions of our algorithm, since the latter only queries

points from the approximated Pareto front $\widehat{\mathbb{X}}_f^*$ and the model is generally unreliable outside that region. In all other experiments, we use the full decision space and set $\mathcal{X} = \mathbb{X}$.

In experiments that do not focus on menu selection (i.e., all except those in Section 5.4.4), we set $k = 1$ to evaluate the regret of a single recommended point:

$$\text{regret}(\mathcal{X}, 1) = u(x^*) - u(\widehat{x}_1^*),$$

where $\widehat{x}_1^* = \arg \max_{x \in \mathcal{X}} \mathbf{E}_n[u(f(x))]$ is the point with the highest posterior mean utility.

5.3 | Experimental setup

All experiments begin with an initial set \mathcal{D}_0 of $2(d + 1)$ query pairs obtained from points drawn uniformly at random over the decision space. The DM’s preferences are then elicited for each pair, forming the initial dataset \mathcal{D}_0 (line 1 in Algorithm 1). Each algorithm is allocated a budget of 50 additional DM interactions beyond the initial set.

Our GP model of the utility function uses a Matérn covariance function with parameter $\nu = 5/2$ and a zero mean function. The covariance function’s length scales, the likelihood’s noise level, and the variational distribution’s parameters are estimated by maximizing a variational lower bound of the log-marginal likelihood, following the approach of Hensman et al. (2015).

In the a posteriori versions of our algorithm, we utilize standard evolutionary algorithms. Specifically, we use NSGA-II (Deb et al., 2002) for scenarios with three objectives and NSGA-III (Deb & Jain, 2013) for scenarios with more objectives. For the cases with three objectives, we use a population of 200 instances and optimize over 500 generations. For the cases with more than three objectives, we use a population of 1500 instances and optimize over 2500 generations.

5.4 | Results

In our main results in Section 5.4.1, the simulated DM responses are generated based on the true underlying utility function without any added noise, although this information is not known to our model. In addition, our main results do not use monotonicity. In Section 5.4.2 we discuss how enforcing monotonicity may enhance the efficiency and robustness of our method, and in Section 5.4.3 we show that our method remains robust even under moderate noise levels. Finally, our main results use a menu size of 1; in Section 5.4.4, we explore the impact of increasing the menu size.

5.4.1 | Main results

A comparison of the performance of the different algorithms is shown in Figure 1. Overall, the interactive algorithm that models the DM’s utility over the objective space (*int-obj*) achieves the best performance, particularly on problems with a larger number of objectives. We highlight below the main trends observed across the two key design choices.

First, interactive algorithms consistently outperform their a posteriori counterparts, with the exception of the DTLZ7 (5,3) problem. In this case, the small number of objectives allows a posteriori methods to begin with a high-quality approximation of the Pareto set, giving them an early advantage. However, as the number of objectives increases, accurately approximating the Pareto front becomes increasingly difficult. This leads to different performance dynamics depending on the number of DM interactions, as illustrated by the DTLZ2 (9,6) problem. A posteriori methods tend to converge quickly at first, benefiting from their initial approximation, but often plateau as this approximation lacks resolution. In contrast, interactive algorithms continue to improve by adaptively focusing on high-utility regions, enabling fine-grained refinement near the Pareto front. This trend is especially pronounced in the Car Cab Design (7,9) problem, where a posteriori methods show early gains but stagnate, while interactive methods steadily progress. The performance gap widens further in higher-dimensional settings such as WFG3 (14,9), suggesting that a posteriori approaches may be preferable only when a reliable Pareto front approximation is available.

Second, algorithms that model utility over the objective space generally outperform those that model it over the decision space. This is likely due to the lower dimensionality of the objective space, which simplifies utility modeling and improves data efficiency. An exception arises in the Car Cab Design (7,9) problem, where the objective space is higher-dimensional than the decision space; however, the performance gap remains modest. This pattern holds across both interactive and a posteriori settings, reinforcing the benefits of modeling in objective space.

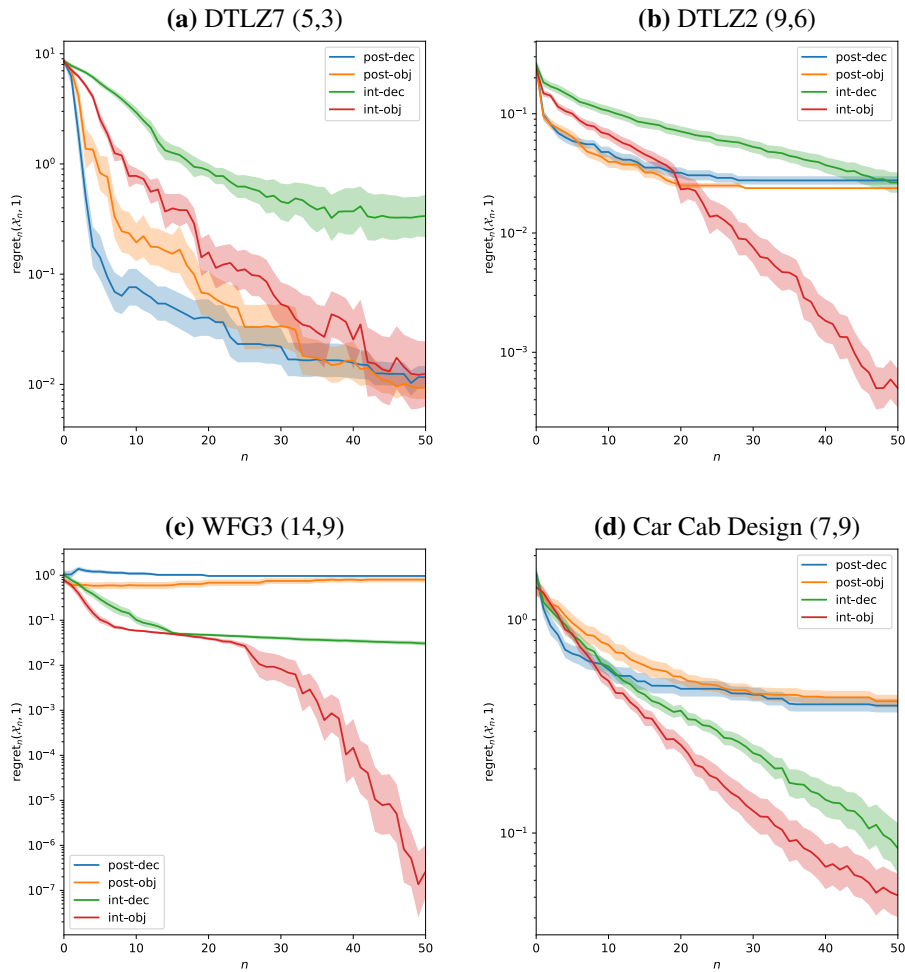


FIGURE 1 Mean regret and standard error for DTLZ7 with 5 decision variables and 3 objectives, DTLZ2 with 9 decision variables and 6 objectives, WFG3 with 14 decision variables and 9 objectives, and Car Cab Design with 7 decision variables and 9 objectives.

To illustrate the learning dynamics, Figure 2 shows a representative run of the algorithm on the bi-objective DTLZ7 (5,2) problem. The leftmost panel shows the true Pareto front (orange) together with the contour lines of the utility function. The subsequent panels illustrate how the algorithm incrementally explores the objective space and refines its utility model. Starting from a few initial queries (beige crosses), the algorithm adaptively selects additional queries (black crosses), enabling it to rapidly identify high-utility trade-offs and converge toward the Pareto front. The learned model captures key qualitative trends of the true utility, such as higher values in regions where both f_1 and f_2 are large. However, because the model is not perfectly monotonic, these trends are only approximated. Moreover, the absolute scale of the predicted utility does not necessarily coincide with that of the true utility, since the model relies solely on pairwise comparisons rather than direct observations.

5.4.2 | Monotonicity

To enforce monotonicity, we use 64 pairs of points generated according to Algorithm 2 with $\delta = 0.25$. The results in Figure 3 indicate that while the benefits of monotonicity are modest, they remain significant. This observation suggests that exploring more sophisticated approaches to induce monotonicity in the utility model could further improve performance, albeit with potentially higher computational costs. This improvement could be particularly valuable in high-dimensional objective spaces, where monotonicity may help avoid exploration of vast suboptimal regions far from the Pareto front.

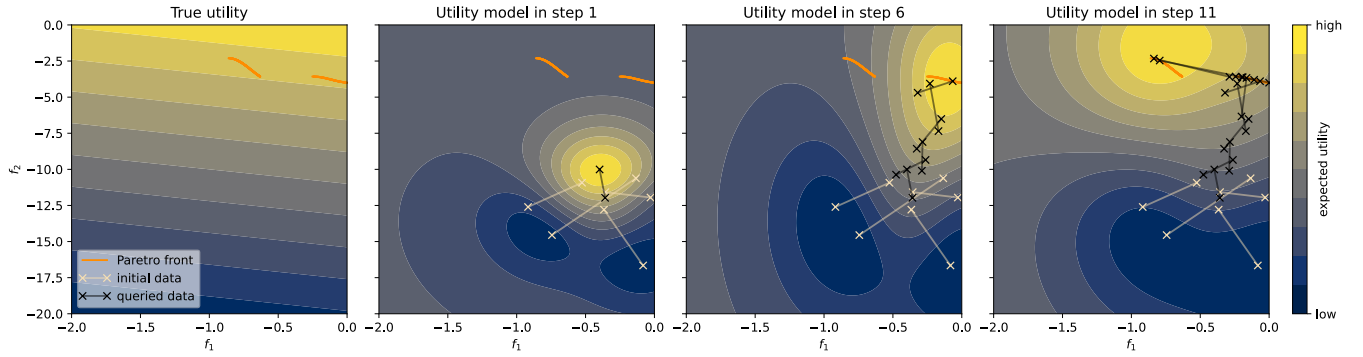


FIGURE 2 Convergence of our algorithm in objective space for DTLZ7 with 5 decision variables and 2 objectives. The leftmost panel shows the Pareto front (orange) together with contour lines of the true utility function. The subsequent panels depict the 4 initial queries (beige crosses), the algorithm’s selected queries (black crosses), and the model’s predicted expected utility after 1, 6, and 11 user interactions.

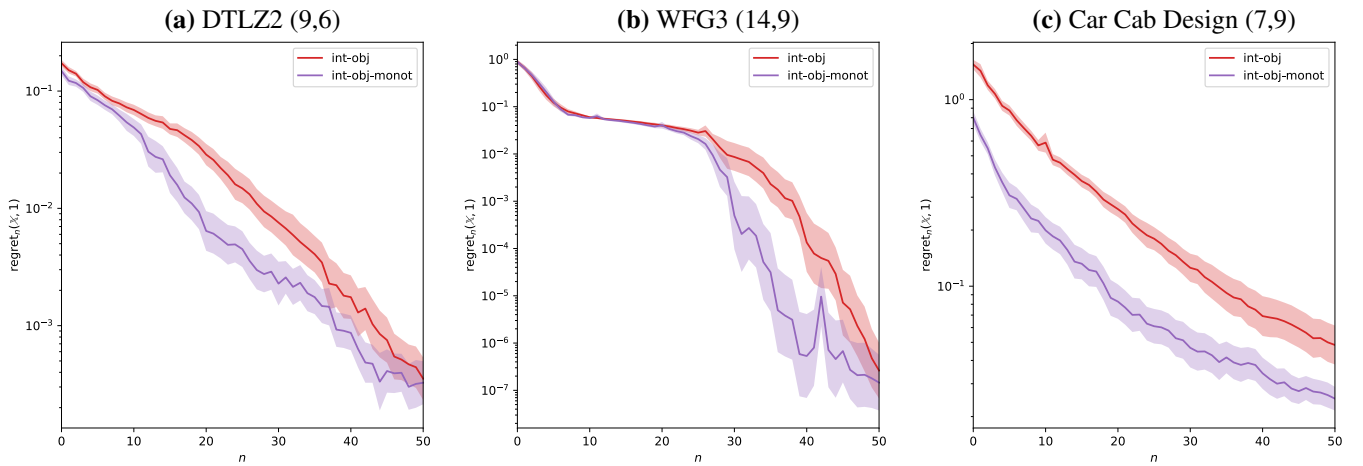


FIGURE 3 Effect of the monotonicity for DTLZ2 with 9 decision variables and 6 objectives, WFG3 with 14 decision variables and 9 objectives, and Car Cab Design with 7 decision variables and 9 objectives.

5.4.3 | Noisy responses

To assess the robustness of our method against noise in DM responses, we perform a reduced set of experiments where the simulated DM responses are corrupted by different levels of noise. The noisy responses are generated using a logistic likelihood (see Section 4.1), with a noise level parameter chosen such that the DM makes errors in approximately 15% or 30% of comparisons when comparing pairs of points from the top 1% of utility function values within the optimization domain \mathbb{X} . These levels are estimated using a large number of random points within \mathbb{X} .

As shown in Figure 4, while noise in the DM’s responses slows the convergence rate, the regret curves still exhibit a desirable trend, indicating our method’s resilience. This pattern suggests that our probabilistic model compensates effectively for noisy interactions, maintaining a stable trajectory toward high-utility solutions.

5.4.4 | Menu selection

In this section, we evaluate the impact of different menu sizes, specifically $k = 1, 4,$ and 16 . To improve computational efficiency, we adopt a greedy construction strategy: Instead of jointly optimizing over all k points, we build the menu incrementally by

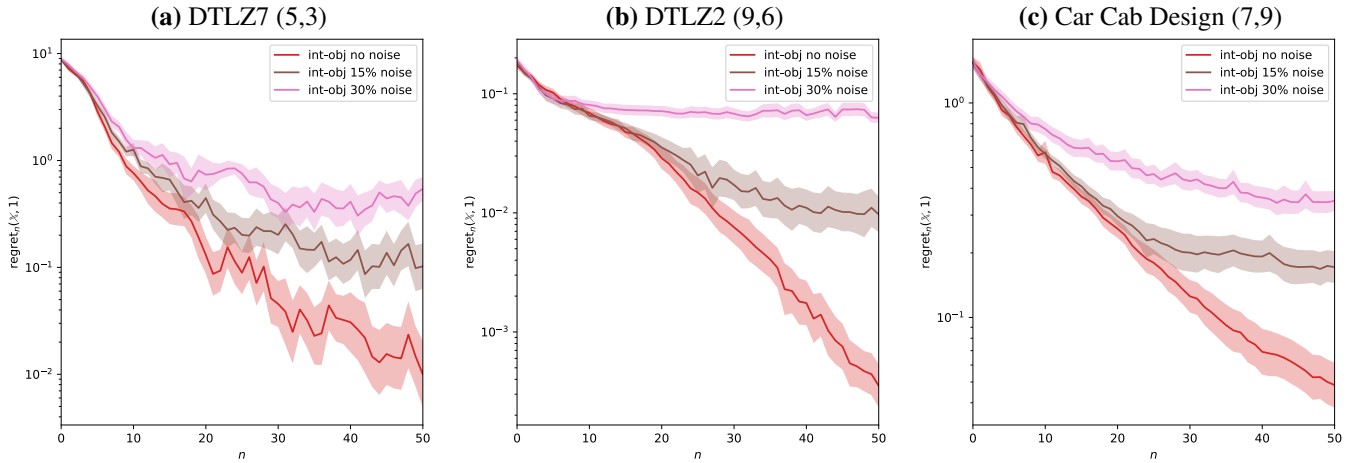


FIGURE 4 Effect of noise for DTLZ7 with 5 decision variables and 3 objectives and DTLZ2 with 9 decision variables and 6 objectives, and Car Cab Design with 7 decision variables and 9 objectives. The noise levels corresponding to 0%, 15% and 30% mistakes of the DM at the top 1% of the utility values in the domain are indicated by the ‘no noise’, ‘15% noise’, and ‘30% noise’ labels.

selecting one point at a time. At each step, we fix the previously selected points and optimize only the next one using Equation (4). This approach offers substantial computational savings while preserving strong theoretical guarantees; specifically, it achieves a $(1 - e^{-1})$ -approximation to the optimal menu obtained via joint optimization under a standard submodularity argument (Wilson et al., 2018).

As shown in Figure 5, increasing the menu size generally leads to improved performance, especially when the number of DM interactions is limited or the responses are noisy. However, as k grows, the same submodularity property implies diminishing returns, likely compounded by the increasing difficulty of the underlying optimization problem.

6 | CONCLUSION

In this work, we proposed an approach to assist a DM in finding their most preferred solution within the Pareto set of an objective function. Our method assumes that the DM’s preferences are represented by an underlying utility function, which we estimate from pairwise comparisons between solutions expressed by the DM. We employ a Bayesian framework using Gaussian processes to model the utility function and utilize an acquisition function to balance the exploration-exploitation trade-off when selecting queries for the DM. Furthermore, we demonstrated how to construct a menu of high-quality solutions after the elicitation phase, leveraging the uncertainty quantification from our probabilistic model. Through several experiments, we showed that our approach effectively identifies high-quality solutions according to the DM’s preferences using only a small number of queries.

There are many exciting directions for future research. On the application side, our approach offers a principled way to find high-quality solutions in problems involving a larger number of objectives than those typically handled by traditional methods, potentially enabling new and impactful applications. On the methodological side, exploring alternative probabilistic models beyond GPs for estimating the DM’s utility function would be of great interest. In particular, models that are both flexible and capable of incorporating prior structural information, such as monotonicity or concavity, could significantly enhance the efficiency and robustness of our method. Additionally, it would be valuable to investigate alternative frameworks for modeling DM preferences in scenarios where (i) assuming a utility function is not appropriate (e.g., when transitivity does not hold), and (ii) there are multiple stakeholders (e.g., when a single utility function may not suffice). Developing efficient elicitation strategies tailored to these alternative models could further expand the scope and applicability of preference-based multi-objective optimization.

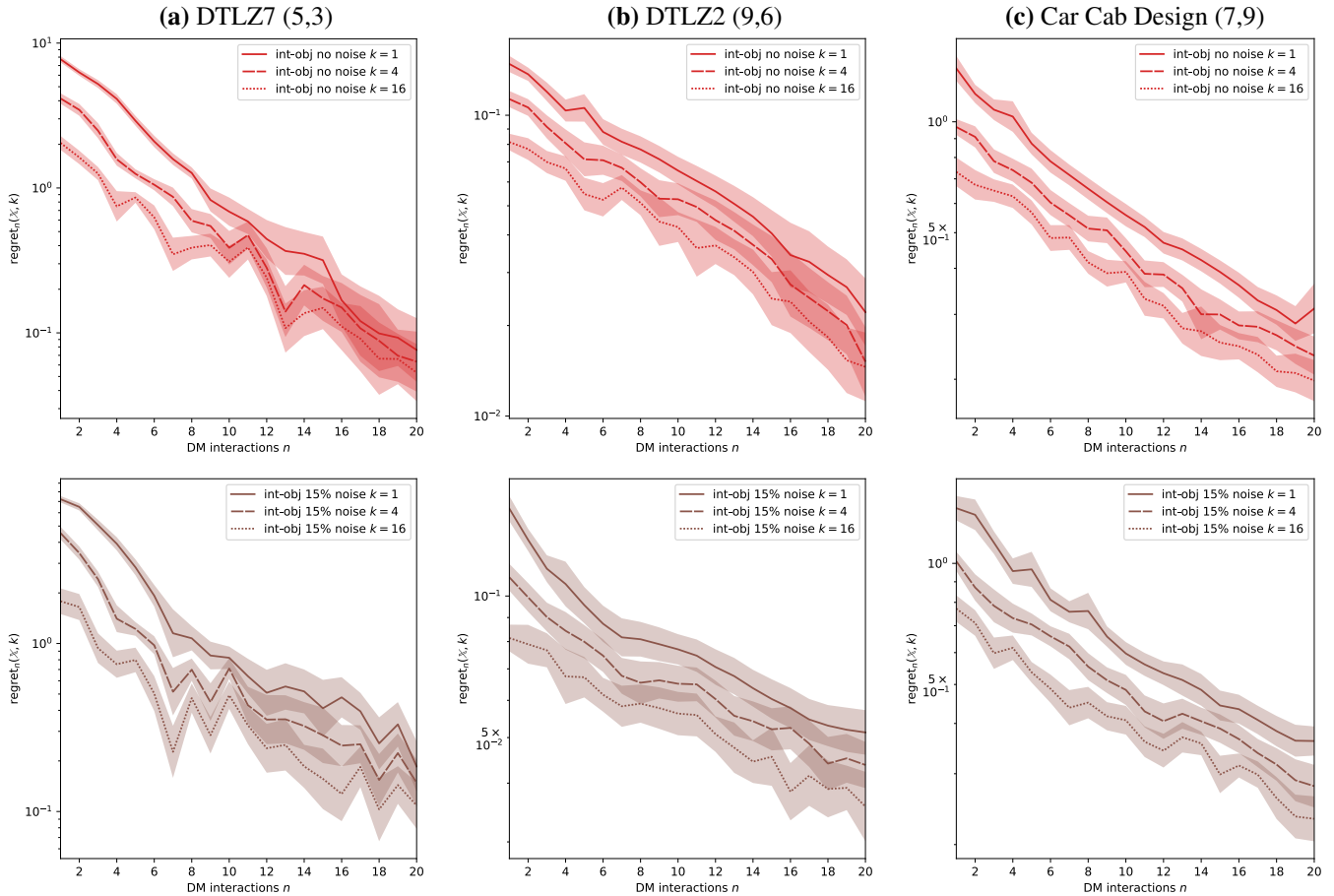


FIGURE 5 Regret for different menu sizes for DTLZ7 with 5 decision variables and 3 objectives (left), DTLZ2 with 9 decision variables and 6 objectives (center), and Car Cab Design with 7 decision variables and 9 objectives (right). The first row shows results for noise-free responses from the DM. The second row shows results for noisy responses with medium noise level (i.e., 15% mistakes of the DM at the top 1% of the utility values in the domain). Short, medium, and small dashes correspond to menus of size 1, 4, and 16, respectively.

ACKNOWLEDGMENTS

Felix Huber was funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy - EXC 2075 – 390740016 and acknowledges the support by the Stuttgart Center for Simulation Science (SimTech). Sebastian Rojas Gonzalez acknowledges support by FWO (Belgium) grant number 1216021N and the Belgian Flanders AI Research Program.

REFERENCES

- Abbas, A. E. (2018). *Foundations of multiattribute utility*. Cambridge: Cambridge University Press.
- Abdolshah, M., Shilton, A., Rana, S., Gupta, S., & Venkatesh, S. (2019). Multi-objective Bayesian optimisation with preferences over objectives. *Advances in neural information processing systems*, 32.
- Afsar, B., Miettinen, K., & Ruiz, F. (2021). Assessing the performance of interactive multiobjective optimization methods: A survey. *ACM Computing Surveys*, 54(4), 1–27.
- Aghaei Pour, P., Rodemann, T., Hakanen, J., & Miettinen, K. (2022). Surrogate assisted interactive multiobjective optimization in energy system design of buildings. *Optimization and Engineering*, 23(1), 303–327.
- Astudillo, R., & Frazier, P. (2020). Multi-attribute Bayesian optimization with interactive preference learning. In *Proceedings of the twenty third international conference on artificial intelligence and statistics* (Vol. 108, pp. 4496–4507). PMLR. Retrieved from <http://proceedings.mlr.press/v108/astudillo20a.html>

- Astudillo, R., Lin, Z. J., Bakshy, E., & Frazier, P. (2023). qEUBO: A decision-theoretic acquisition function for preferential Bayesian optimization. In *International conference on artificial intelligence and statistics* (pp. 1093–1114).
- Benavoli, A., & Azzimonti, D. (2024). A tutorial on learning from preferences and choices with gaussian processes. *arXiv preprint arXiv:2403.11782*.
- Boutillier, C. (2002). A POMDP formulation of preference elicitation problems. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 239–246). Austin: AAAI Press.
- Branke, J. (2008). Consideration of partial user preferences in evolutionary multiobjective optimization. *Multiobjective optimization: Interactive and evolutionary approaches*, 157–178.
- Branke, J. (2016). MCDA and multiobjective evolutionary algorithms. *Multiple criteria decision analysis: State of the art surveys*, 977–1008.
- Branke, J., Corrente, S., Greco, S., Słowiński, R., & Zielniewicz, P. (2016). Using Choquet integral as preference model in interactive evolutionary multiobjective optimization. *European Journal of Operational Research*, 250(3), 884–901.
- Chajewska, U., & Koller, D. (2000). Utilities as random variables: Density estimation and structure discovery. In *Proceedings of the sixteenth conference on uncertainty in artificial intelligence* (pp. 63–71).
- Cheikh, M., Jarboui, B., Loukil, T., & Siarry, P. (2010). A method for selecting Pareto optimal solutions in multiobjective optimization. *Journal of Informatics and mathematical sciences*, 2(1), 51–62.
- Chu, W., & Ghahramani, Z. (2005). Preference learning with Gaussian processes. In *Proceedings of the 22nd international conference on machine learning* (pp. 137–144).
- Da Veiga, S., & Marrel, A. (2012). Gaussian process modeling with inequality constraints. *Annales de la Faculte des sciences de Toulouse: Mathematiques*, 21(3), 529–555.
- Deb, K., & Jain, H. (2013). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. *IEEE transactions on evolutionary computation*, 18(4), 577–601.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2), 182–197.
- Deb, K., Sinha, A., Korhonen, P. J., & Wallenius, J. (2010). An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. *IEEE Transactions on Evolutionary Computation*, 14(5), 723–739.
- Garnett, R. (2023). *Bayesian optimization*. Cambridge University Press.
- Gaudrie, D., Le Riche, R., Picheny, V., Enaux, B., & Herbert, V. (2020). Targeting solutions in Bayesian multi-objective optimization: Sequential and batch versions. *Annals of Mathematics and Artificial Intelligence*, 88(1), 187–212. doi: 10.1007/s10472-019-09644-8
- Gibson, F. J., Everson, R. M., & Fieldsend, J. E. (2022). Guiding surrogate-assisted multi-objective optimisation with decision maker preferences. In *Proceedings of the genetic and evolutionary computation conference* (pp. 786–795).
- González, J., Dai, Z., Damianou, A., & Lawrence, N. D. (2017). Preferential Bayesian optimization. In *International conference on machine learning* (pp. 1282–1291).
- He, Y., Sun, J., Song, P., Wang, X., & Usmani, A. S. (2020). Preference-driven kriging-based multiobjective optimization method with a novel multipoint infill criterion and application to airfoil shape design. *Aerospace Science and Technology*, 96, 105555.
- Heidari, A., Rojas Gonzalez, S., Dhaene, T., & Couckuyt, I. (2023). Data-efficient interactive multi-objective optimization using ParEGO. In *Joint european conference on machine learning and knowledge discovery in databases* (pp. 519–526).
- Hensman, J., Matthews, A., & Ghahramani, Z. (2015). Scalable variational Gaussian process classification. In G. Lebanon & S. V. N. Vishwanathan (Eds.), *Proceedings of the eighteenth international conference on artificial intelligence and statistics* (Vol. 38, pp. 351–360). PMLR. Retrieved from <https://proceedings.mlr.press/v38/hensman15.pdf>
- Huband, S., Hingston, P., Barone, L., & While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5), 477–506.
- Keeney, R. L., & Raiffa, H. (1993). *Decisions with multiple objectives: Preferences and value tradeoffs*. Cambridge university press.
- Lepird, J. R., Owen, M. P., & Kochenderfer, M. J. (2015). Bayesian preference elicitation for multiobjective engineering design optimization. *Journal of Aerospace Information Systems*, 12(10), 634–645.
- Lin, Z. J., Astudillo, R., Frazier, P., & Bakshy, E. (2022). Preference exploration for efficient Bayesian optimization with multiple outcomes. In *International conference on artificial intelligence and statistics* (pp. 4235–4258).

- López-Lopera, A. F., Bachoc, F., Durrande, N., & Roustant, O. (2018). Finite-dimensional Gaussian approximation with linear inequality constraints. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3), 1224–1255. doi: <https://doi.org/10.1137/17M1153157>
- Maatouk, H., & Bay, X. (2017). Gaussian process emulators for computer experiments with inequality constraints. *Mathematical Geosciences*, 49(5), 557–582. doi: <https://doi.org/10.1007/s11004-017-9673-2>
- Mackin, P. D., Roy, A., & Wallenius, J. (2011). An interactive weight space reduction procedure for nonlinear multiple objective mathematical programming. *Mathematical Programming*, 127, 425–444.
- Miettinen, K. (1999). *Nonlinear multiobjective optimization* (Vol. 12). Springer.
- Misitano, G., Saini, B. S., Afsar, B., Shavazipour, B., & Miettinen, K. (2021). Desdeo: The modular and open source framework for interactive multiobjective optimization. *IEEE Access*, 9, 148277–148295.
- Nguyen, Q. P., Tay, S., Low, B. K. H., & Jaillet, P. (2021). Top-k ranking Bayesian optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10), 9135–9143. doi: <https://doi.org/10.1609/aaai.v35i10.17103>
- Riihimäki, J., & Vehtari, A. (2010). Gaussian processes with monotonicity information. In Y. W. Teh & M. Titterton (Eds.), *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (Vol. 9, pp. 645–652). PMLR. Retrieved from <https://proceedings.mlr.press/v9/riihimaki10a.html>
- Sanders, S. (2024). Does the paradox of choice exist in theory? A behavioral search model and Pareto-improving choice set reduction algorithm. *AI & SOCIETY*, 39(3), 913–923.
- Taylor, K., Ha, H., Li, M., Chan, J., & Li, X. (2021). Bayesian preference learning for interactive multi-objective optimisation. In *Proceedings of the genetic and evolutionary computation conference* (p. 466–475). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/3449639.3459299> doi: 10.1145/3449639.3459299
- Taylor, K. P., Ha, H., Li, M., Li, X., & Chan, J. (2024). Accelerated bayesian preference learning for efficient evolutionary multi-objective optimisation. In *Proceedings of the genetic and evolutionary computation conference companion* (pp. 371–374).
- Tomczyk, M. K., & Kadziński, M. (2019). Decomposition-based interactive evolutionary algorithm for multiple objective optimization. *IEEE Transactions on Evolutionary Computation*, 24(2), 320–334.
- Viappiani, P., & Boutilier, C. (2010). Optimal Bayesian recommendation sets and myopically optimal choice query sets. *Advances in neural information processing systems*, 23.
- Wallenius, J., Dyer, J. S., Fishburn, P. C., Steuer, R. E., Zionts, S., & Deb, K. (2008). Multiple criteria decision making, multiattribute utility theory: Recent accomplishments and what lies ahead. *Management Science*, 54(7), 1336–1349.
- Wilson, J., Hutter, F., & Deisenroth, M. (2018). Maximizing acquisition functions for bayesian optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 31). Curran Associates. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2018/hash/498f2c21688f6451d9f5fd09d53edda7-Abstract.html
- Winston, W. L. (2004). *Operations research: applications and algorithm*. Thomson Learning, Inc.
- Xin, B., Chen, L., Chen, J., Ishibuchi, H., Hirota, K., & Liu, B. (2018). Interactive multiobjective optimization: A review of the state-of-the-art. *IEEE Access*, 6, 41256–41279. doi: 10.1109/ACCESS.2018.2856832
- Zionts, S., & Wallenius, J. (1983). An interactive multiple objective linear programming method for a class of underlying nonlinear utility functions. *Management science*, 29(5), 519–529.

□

APPENDIX

A MODEL DETAILS

A.1 Variational approximation of the posterior

For data consisting of preferences for pairwise comparisons the posterior distribution cannot be computed in closed form. Therefore, we approximate the posterior distribution $p(u' \mid \mathcal{D}_n)$ using a variational approach with a Gaussian distribution $q(u') = \mathcal{N}(u' \mid \mu', \Sigma')$ at n_i inducing points with corresponding utility values $u' \in \mathbb{R}^{n_i}$ as introduced in Hensman et al. (2015).

For this, we fit the mean $\mu' \in \mathbb{R}^{n_i}$ and covariance $\Sigma' \in \mathbb{R}^{n_i \times n_i}$ of the inducing points similarly to Nguyen et al. (2021) by maximizing the evidence lower bound (ELBO)

$$\text{ELBO}(q) = \int_{\mathbb{R}^{n_i}} q(u') \log p(\mathcal{D}_n \mid u') - q(u') \log \frac{q(u')}{p(u')} \, du'$$

with the logistic likelihood (1). The predictive distribution of the model is then a Gaussian process depending on μ' and Σ' (Nguyen et al., 2021) and thus allows an efficient maximization of the acquisition function. Following Astudillo et al. (2023), we use all points from previous queries as inducing points and extend the set with quasi-random Sobol points to improve the model's numerical stability.