When to Elicit Preferences in Multi-Objective Bayesian Optimization

Juan Ungredda
ungredda@esteco.com
ESTECO SpA
Trieste, Italy

Juergen Branke
juergen.branke@wbs.ac.uk
University of Warwick
Coventry, UK

ABSTRACT

We consider the case of interactive multi-objective Bayesian optimization where decision maker (DM) preferences can be elicited by asking the DM to select the more preferred among pairs of observations. Assuming that there is a cost to evaluating a solution as well as to eliciting preferences, and given a total budget, we propose an acquisition function that, in each iteration, decides whether to evaluate another solution or to query the DM. Thus, the approach automatically chooses how often and when to interact with the DM. It furthermore decides which pair of observations is likely to be most informative when shown to the DM. We show empirically that the proposed criterion is not only able to pick suitable pairs of observations, but also automatically results in a sensible balance between optimization and querying the DM.

ACM Reference Format:
Juan Ungredda and Juergen Branke. 2023. When to Elicit Preferences in Multi-Objective Bayesian Optimization. In *Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3583133.3596342

1 INTRODUCTION

Many real-world optimization problems have multiple, conflicting objectives. A popular way to tackle such problems is to search for a set of Pareto-optimal solutions with different trade-offs, and allow the decision maker (DM) to pick their most preferred solution from this set. This has the advantage that the DM doesn't have to specify their preferences explicitly before the optimization, which is generally considered very difficult.

In this paper, we consider the case of expensive multi-objective optimization problems, where the evaluation of the objective functions is costly, and thus the number of solutions that can be evaluated during optimization is small. This has two potential drawbacks:

(1) The number of evaluations may not be sufficient to find solutions close to the Pareto frontier.
(2) Given only a small number of evaluated solutions, it is unlikely that the Pareto-optimal solution most preferred by the DM would be among the small set of solutions found by the algorithm, even if these were truly Pareto-optimal [23]

Interactive approaches help in those cases, as they allow to focus the search on the most interesting area of the search space, speeding up convergence and returning more alternatives to choose from in the area of focus [3].

A popular approach to capture preferences is to query the decision maker (DM) a fixed number of times, e.g., asking the DM to select the more preferred from a given pair of observations. The feedback is then used to learn a model of the DM preferences and guide the optimization run, focusing on the most preferred region of the Pareto frontier. However, there is very little work on when and how often to query the DM, or which pair of solutions to present to the DM. We fill this gap by proposing a multi-objective Bayesian optimization algorithm that is able to decide, in each iteration, whether it would be more beneficial to evaluate another solution and learn more about the optimization problem, or to query the DM and learn more about their preferences. This is inspired by the BICO approach [24] which, for a single-objective problem with simulation input uncertainty, can decide whether it is better to evaluate a new solution or to collect more data on the input distribution. In the case of eliciting preference information, if our algorithm decides to query the DM, it will intelligently select the pair of solutions that maximizes the potential contribution to the optimization.

Assuming a fixed total budget, devoting too much effort to collect preference information may not leave sufficient resources for optimization. On the other hand, devoting too little effort to collect preference information means the identified solution(s) may be sub-optimal under the true utility parameters.

This paper is structured as follows. After a literature review in Section 2, we formally define the problem considered in Section 3. Section 4 explains the statistical models and derives the suggested sampling procedures. We perform numerical experiments in Section Section 5 and, finally, the paper concludes with a summary and some suggestions for future work in Section 6.

2 LITERATURE REVIEW

Bayesian optimization (BO) is a global optimization technique that builds a Gaussian process (GP) surrogate model of the fitness landscape, and then uses the estimated mean and variance at each location to decide which solution to evaluate next. It uses an acquisition function to explicitly make a trade-off between exploitation and exploration (e.g., [21]). A frequently used acquisition function is the expected improvement (EI) [12] which selects the point with the largest expected improvement over the current best known solution as the next solution to evaluate.

GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0120-7/23/07.
https://doi.org/10.1145/3583133.3596342

Recently, BO has been adapted to the multi-objective case, for a survey see [18]. One of the earliest approaches, ParEGO [15] simply uses the Tchebychev scalarization function to turn the multi-objective problem into a single objective problem, but it uses a different scalarization vector in every iteration where the next solution is decided according to EI. This scalarization has the advantage that all solutions in the Pareto front may be supported, in convex and concave regions [25]. EI-UU [1] is another method that translates the multi-objective problem into a single-objective problem using an achievement scalarization function. However, EI-UU uses linear scalarizations and integrates the expected improvement over all possible scalarizations, so it takes into account different scalarizations simultaneously rather than sequentially over generations. [8] trains a GP model for each objective, then chooses the next solution to be evaluated according to a hypervolume-based acquisition criterion. Other multi-objective BO approaches include [6, 14, 17].

Depending on the involvement of the DM in the optimization process, multi-objective optimization can be classified into a priori approaches, a posteriori approaches, and interactive approaches [3, 11, 27]. The field is very large, so we can only mention some of the most relevant papers here. A priori approaches ask the DM to specify their preferences ahead of optimization. This allows to turn the multi-objective optimization problem into a single objective optimization problem, but it is usually very difficult for a DM to specify their preferences before having seen the alternatives. One effective way to capture preferences is to ask the DM to specify a reference point [7].

Most multi-objective evolutionary algorithms (MOEA) are a posteriori approaches, attempting to identify a good approximation of the Pareto frontier, and the DM can then pick the most preferred solution from this set. This is much easier for a DM, but identifying the entire Pareto front may be computationally expensive. In particular in case of more than two objectives, the number of Pareto-optimal solutions increases drastically, making it harder to identify a good approximation, while it becomes more difficult for the DM to grasp the structure of a high-dimensional Pareto front in order to identify their most preferred solution [3].

Interactive approaches attempt to learn the DM's preferences during optimization and then focus the search on the most preferred region of the Pareto front. In particular, elicitation of pairwise comparisons from the training sample set is a widely used approach and requires a relatively small cognitive effort from the DM. Often, each pair is randomly selected and a pre-defined (static) interaction pattern determines *when* to elicit information from the DM. For example, the interactive MOEA in [4] queries, at regular intervals, randomly selected observations to the DM and examines the influence of the number of interactions to solve the MO problem. Similarly, [22] propose to elicit pairs at regular intervals, however, each pair is determined according to the potential contribution to a performance measure, e.g, selecting pairs that minimize the expected number of potentially optimal solutions. [13] improve over previous work by comparing several non-regular, but pre-defined, interaction patterns. Moreover, they propose a dynamic pattern that depends on the progress attained in the evolutionary search, imposing additional evolutionary pressure by narrowing down search when the evolutionary search is stagnating. The interaction with the DM relies on showing pairs of random observations.

Recently, a few interactive multi-objective BO approaches have also been proposed. [9] allows the DM to specify a reference point a priori, and use this to subsequently focus the BO search. [1] propose to learn preferences by selecting random pairs of previously evaluated solution, at each BO iteration. The preference information is then used to guide EI-UU by computing the improvement using only the scalarization functions compatible with the preference dataset. [23] suggest to interact with the DM once before the end of the run, rather than supplying the identified Pareto front at the end. The DM selects a single solution from an approximated continuous Pareto front which allows the algorithm a final effort to find the most preferred solution. Similar to the approach in our paper, [16] uses an acquisition function to select pairs of observations to show to the DM. It takes into account the benefit of eliciting information to the performance measure. However, the interaction strategy is pre-defined before the start of the optimization. [10] proposes that, at each interaction, the decision maker is shown a subset of non-dominated observations and provide preferences in the form of preferred ranges for each objective. Internally, a ParEGO algorithm is used where sample reference points within the hyperbox defined by the preferred ranges in the objective space guide the search. The process of preference elicitation and optimization is performed at regular intervals defined by the DM.

All aforementioned methods assume a fixed amount of preference data to guide the search of the algorithm and/or a predefined schedule to perform optimization and query the DM. We explicitly look at the trade-off between either running more optimization or instead collecting more preference data. Moreover, we propose a unified BO algorithm that, in each iteration, dynamically decides when to interact with the DM and what pairs of observations should be selected.

3 PROBLEM DEFINITION

We assume a D-dimensional real-valued space of possible *solutions*, i.e., $x \in \mathbb{X} \subset \mathbb{R}^D$. The objective function is an arbitrary black box $f : \mathbb{X} \to \mathbb{R}^K$ which takes as arguments a solution and returns a deterministic vector of *observations* $y \in \mathbb{R}^K$. The (unknown) DM preference over the observations can be characterized by a utility function $U : \mathbb{R}^K \to \mathbb{R}$. Thus, of all solutions in $\mathbb{X}$, the DM's most preferred solution is $x^* = \arg\max_{x \in \mathbb{X}} U(x)$.

We consider a budget of B units that can be spent either by choosing $x \in \mathbb{X}$ and calling $f(x)$, costing c_f, or by asking the DM to select the more preferred among a pair of observations to reduce the uncertainty over the DM preferences, costing c_s.

After consuming the budget, a *single* solution x_r is returned to the DM and its quality is determined by the difference in true user's utility between x_r and the best solution x^*, or Opportunity Cost (OC),

$$OC = U(f(x^*)) - U(f(x_r)),$$

which is to be minimized. This assumption of returning a single solution at the end reflects the idea that at some point, preference elicitation has to take place to identify the most preferred solution, and we assume that this has to happen within the given budget. Under finite resources, an algorithm must carefully determine how much effort to spend on preference elicitation vs. optimization.

4 METHODOLOGY

This section describes details of the proposed algorithm. Section 4.1 and Section 4.2 describe the statistical models used for the objectives and the utility, respectively. Sections 4.4 and 4.5 apply this to estimate the value of collecting black box evaluations and collecting DM information, respectively. At each iteration, the action is simply determined by what has the highest value. Together the modeling and automated value based data collection form the algorithm summarized as Algorithm 1 in Section 4.6.

4.1 Statistical Model of the Objectives

Let us denote the set of n evaluated points and their objective function values as $\mathscr{D}_f^n = \{(x,y)^1, \dots, (x,y)^n\}$. We also define the column vector of all n solutions sampled so far, X^n, as a $n \times D$ and the column vector of all sampled outputs for the j-th objective function as Y_j^n. Then, we propose to use an independent GP to model each objective function $y_j = f_j(x), \forall j = 1, \dots, K$, defined by a mean function $\mu_j^0(x) : \mathbb{X} \to \mathbb{R}$ and a covariance function $k_j^0(x,x') : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$. Given the dataset $\mathscr{D}_f^n$, predictions at new locations x for observation y_j are given by,

$$\begin{aligned}\mathbb{E}[f_j(x)] &= \mu_j^n(x)\\ &= \mu_j^0(x) - k_j^0(x, X^n) k_j^0(X^n, X^n)^{-1}(Y_j^n - \mu_j^0(X^n))\\ \mathrm{Cov}\,[f_j(x), f_j(x')] &= k_j^n(x,x')\\ &= k_j^0(x,x') - k_j^0(x,X^n)k_j^0(X^n,X^n)^{-1}k_j^0(X^n,x')\end{aligned}$$

The prior mean $\mu_j^0(x)$ is typically set to $\mu_j^0(x) = 0$ and the kernel $k_j^0(x,x')$ allows the user to encode known properties of the objective function $f_j(x)$ such as smoothness and periodicity. We use the popular squared exponential kernel that assumes $f_j(x)$ is a smooth function such that nearby x have similar outputs while widely separated points have unrelated outputs,

$$k_j^0(x,x') = \sigma_0^2 \exp\left(\frac{||x-x'||^2}{2l_X^2}\right)$$

where $\sigma_0 \geq 0$ and $l_X > 0$ are hyper-parameters estimated from the data $\mathscr{D}_f^n$ by maximum marginal likelihood.

4.2 Statistical Model over the Utility

We consider the approach of showing pairs of previously sampled observations $y, y' \in Y^n$ to the DM. Then, for each pair shown, the DM returns a feedback, $q \in \{1, -1\}$, that represents which of the two observations is preferred, e.g, a value $q = 1$ means the DM prefers observation y over y'. Similar to Sec. 4.1, let us then denote the dataset of m collected preference queries as $\mathscr{D}_S^m = \{([y,y'],q)^i\}_{i=1}^m$.

Let us assume that the DM's utility can be described by a parametric utility function $U(x,\theta)$ with parameters $\theta \in \Theta$. Although different parametric families of utility functions may be assumed, i.e., linear functions [26] to deep neural networks [5], we assume a Tchebychev achievement scalarization function over the observations.

$$U(x;\theta) = \max_{j=1,\dots,K}(\theta_j y_j(x)) + \rho \sum_{j=1,\dots,K} \theta_j y_j(x)$$

where ρ is usually a very small number and the parameter $\theta \in \Theta$ is defined as, $\theta = \{\theta \in [0,1]^K | \sum_j \theta_j = 1\}$.

Similar to [1] and [19], we adopt a Bayesian approach to obtain a distribution over parameters $\theta \in \Theta$. Commonly used likelihood functions include probit and logit [26]. However, for simplicity we assume fully accurate preference responses in this paper. Therefore, if each DM preference, q_m, is expressed according to the utility difference $\Delta = U(y;\theta) - U(y';\theta)$, then we may express the likelihood function as

$$\mathscr{L}(q_m;\Delta) = \mathbb{I}_{\{q_m = \mathrm{sign}(\Delta)\}}.$$

If we assume a flat Dirichlet prior $\mathbb{P}(\theta)$, the posterior distribution over θ given a dataset of collected preference information $\mathscr{D}_S^m$ is then given by,

$$\mathbb{P}[\theta|\mathscr{D}_S^m] = \prod_m \mathscr{L}(q_m;\Delta)\mathbb{P}(\theta).$$

4.3 Predictive Performance

After exhausting the budget B, the algorithm must return a *single* recommended solution, x_r, to the user. The *true* utility value of any given solution x is $U(f(x),\theta^*)$, where $\theta^* \in \Theta$ is the true underlying parameter. However, both f and θ^* are unknown, hence we need to make two approximations. Firstly, approximate $f(x)$ with the GP prediction for each objective, $\mu_j^n(x)$. Then, we replace θ^* with the posterior $\mathbb{P}[\theta|\mathscr{D}_S^m]$. Thus, the best estimate of the true utility of solution x, $U(f(x),\theta^*))$, given the data so far $\mathscr{D}_f^n$ and $\mathscr{D}_S^m$, is denoted as $G(x;\mathscr{D}_S^m,\mathscr{D}_f^n)$ and given by

$$G(x,\mathscr{D}_f^n,\mathscr{D}_S^m) = \mathbb{E}_\theta[U(x;\theta)] = \int U(x;\theta,\mathscr{D}_f^n)\mathbb{P}[\theta|\mathscr{D}_S^m]d\theta$$

where the utility $U(x;\theta,\mathscr{D}_f^n)$ is the Tchebychev achievement scalarization function and $\mathbb{P}[\theta|\mathscr{D}_S^m]$ is the probability of θ being the true parameters of the DM's utility function given the preference information collected so far $\mathscr{D}_S^m$.

Therefore, the recommended solution is the solution that maximizes the expected utility, $x_r = \arg\max_x G(x,\mathscr{D}_f^n,\mathscr{D}_S^m)$, with best expected utility value, $G^*(\mathscr{D}_S^m,\mathscr{D}_f^n) = G(x_r,\mathscr{D}_f^n,\mathscr{D}_S^m)$. To estimate $G(x,\mathscr{D}_f^n,\mathscr{D}_S^m)$, we first compute the Gaussian process mean prediction for each objective μ_j^n, and aggregate the predictions using the utility function U. Finally, the expectation over θ is computed using a Monte Carlo (MC) average approximation as,

$$G(x,\mathscr{D}_S^m,\mathscr{D}_f^n) \approx \hat{G}(x,\mathscr{D}_S^m,\mathscr{D}_f^n) = \frac{1}{n_\theta}\sum_{i=1}^{n_\theta} U(x;\theta_i,\mathscr{D}_f^n), \quad (1)$$

where we use n_θ sampled parameter values, $\theta_i \sim \mathbb{P}[\theta|\mathscr{D}_S^m]$.

4.4 Value of Information for Objective Function Data

To select the solutions at which the objective function is evaluated, we consider the one-step look-ahead incremental increase in predicted performance of taking the solution x^{n+1}, with c_f being the cost of evaluating the objective function,

$$\text{VoI}(x) = \mathbb{E}_{y^{n+1}}\left[\frac{G^*(\mathcal{D}_S^m, \mathcal{D}_f^{n+1}) - G^*(\mathcal{D}_S^m, \mathcal{D}_f^n)}{c_f}\middle| x^{n+1} = x\right] \quad (2)$$

Obtaining a closed-form expression for $\text{VoI}(x; \mathcal{D}_S^m, \mathcal{D}_f^n)$ is not possible but it can still be computed efficiently via Monte Carlo sampling. We first convert $\mu_j^{n+1}(x)$ to quantities that can be computed in the current step n through the reparameterization trick [20], $\mu_j^{n+1}(x) = \mu_j^n(x) + \tilde{\sigma}_j(x, x^{n+1})Z$ where $Z \sim N(0,1)$. The deterministic function $\tilde{\sigma}_j^n(x, x^{n+1})$ represents the standard deviation of $\mu_j^{n+1}(x)$ parameterized by x^{n+1} and given by $\tilde{\sigma}_j^n(x, x^{n+1}) = k_j^n(x, x^{n+1}) / \sqrt{k_j^n(x^{n+1}, x^{n+1})}$. Therefore, we may construct n_z models for each objective, $\{\mu_j^{n+1}(x; Z_j)\}_{j=1}^{n_z}$. If $\tilde{Z}_j = (Z_1, \ldots, Z_K)_j$ represents a sample of K normally distributed values for each objective, then, the expected utility of each model may be computed according to Eqn. 1 and optimized to obtain $\{G^*(\tilde{Z}_j; \mathcal{D}_S^m, \mathcal{D}_f^{n+1})\}_{j=1}^{n_z}$.

Solving Eqn. 2 involves solving n_z nested maximization problems, if all inner-optimization problems are solved sequentially. To reduce the computational time, we jointly optimize all inner optimization problems and the acquisition function optimizer, x^{n+1}, using a "one-shot" formulation [2]. This consists of pairing each sampled $\{\tilde{Z}_j\}_{j=1}^{n_z}$ with an x_j^* and define the set of design vectors that maximize the inner problems, $X_d^* = \{x_j^*\}_{j=1}^{n_z}$. Then, we estimate $\text{VoI}(x; \mathcal{D}_S^m, \mathcal{D}_f^n)$ as follows,

$$\text{VoI}(x; X_d^*, \{\tilde{Z}_j\}_1^{n_z}) = \frac{1}{n_z}\sum_{j=1}^{n_z} \hat{G}(x_j^*, \tilde{Z}_j, \mathcal{D}_S^m, \mathcal{D}_f^{n+1})$$

$$- \hat{G}(x_r, \tilde{Z}_j, \mathcal{D}_S^m, \mathcal{D}_f^{n+1}) \quad (3)$$

where $x_r = \arg\max_{x \in \mathbb{X}} \hat{G}(x, \mathcal{D}_S^m, \mathcal{D}_f^n)$. Note that the argument within the summation is strictly non-negative.

Eqn. 3 removes the inner maximization operation by a now extended optimization space defined by x and X_d^*. In each BO iteration, the random samples $\{\tilde{Z}_j\}_1^{n_z}$ are fixed hence $\text{VoI}(x; X_d^*, \{\tilde{Z}_j\}_{j=1}^{n_z}))$ is deterministic and, in the search for x^{n+1}, we *simultaneously* search over X_d and x^{n+1} hence the acquisition function estimate improves over the course of the search for the next candidate point x^{n+1^*},

$$x^{n+1^*} = \arg\max_{x^{n+1}} \max_{X_d} \text{VoI}(x^{n+1}; X_d, \{\tilde{Z}_j\}_1^{n_z})). \quad (4)$$

where, for simplicity, we denote

$$\text{VoI}_x^* = \max_{x, X_d} \text{VoI}(x; X_d, \{\tilde{Z}_j\}_1^{n_z})).$$

The optimal x^{n+1^*} is used as the next sample and the final optimized X_d^* is discarded as a "byproduct" of Eqn. 4. This method improves the acquisition function evaluation time but requires solving a problem with higher dimensions.

4.5 Value of Information for Decision Maker's Queries

Instead of evaluating the objective function , we may elicit data from the DM and augment the dataset $\mathcal{D}_S^{m+1} = \mathcal{D}_S^m \cup \{([y, y'], q)^{m+1}\}$. Therefore, we may quantify the benefit due to the additional query by taking the difference in expected utility obtained given the additional query $\{([y, y'], q)^{m+1}\}$.

$$\text{VoI}([y, y']) = \mathbb{E}_{q^{m+1}}\left[\frac{G^*(\mathcal{D}_S^{m+1}, \mathcal{D}_f^n) - G^*(\mathcal{D}_S^m, \mathcal{D}_f^n)}{c_s}\middle| [y, y']^{m+1} = [y, y']\right] \quad (5)$$

where c_s is the cost of eliciting preference information from the decision maker. As mentioned previously, this acquisition function is optimized using pairs of previously sampled observations. This may be further extended to optimize over a continuous output space, however, predicted observations shown to the DM are not guaranteed to be achievable.

The acquisition function is efficiently computed by considering that the total Value of Information is given by a weighted sum of the two possible outcomes of showing the pair to the decision maker. The weights are chosen to be the probability of the two outcomes given by π.

$$\begin{aligned}\text{VoI}([y, y']) = \pi(([y, y'], 1)^{m+1}; \mathcal{D}_S^m) \cdot G^*(\mathcal{D}_S^m \cup \{([y, y'], 1)^{m+1}\})\\ + \pi(([y, y'], -1)^{m+1}; \mathcal{D}_S^m) \cdot G^*(\mathcal{D}_S^m \cup \{([y, y'], -1)^{m+1}\})\end{aligned} \quad (6)$$

To estimate $\pi(([y, y'], 1); \mathcal{D}_S^m)$ we take the average over multiple likely responses as,

$$\pi(([y, y'], 1)^{m+1}; \mathcal{D}_S^m) \approx \frac{1}{n_\pi}\sum_{i=1}^{n_\pi} \mathcal{L}(1; U(y; \theta_i) - U(y'; \theta_i)).$$

where each parameter is generated according to the posterior distribution $\theta_i \sim \mathbb{P}[\theta | \mathcal{D}_S^m]$ and $\pi(([y, y'], -1)^{m+1}; \mathcal{D}_S^m)$ can be computed as $\pi(([y, y'], -1)^{m+1}; \mathcal{D}_S^m) = 1 - \pi(([y, y'], 1)^{m+1}; \mathcal{D}_S^m)$. Therefore, for each candidate pair, only two optimization problems must be solved to estimate the value of the acquisition function. To compare the Value of Information of both actions, black box evaluations and query to the DM, we use the same Monte Carlo samples to estimate G^*.

We determine the m-th candidate pair $[y, y']^*$ to show to the DM by evaluating Eqn. 6 using non-dominated pairs of previously sampled observations. The pair with the largest expected benefit is then selected,

$$[y, y']^{m+1^*} = \arg\max_{y, y' \in Y^n \times Y^n} \text{VoI}([y, y'])$$

For simplicity, we also denote

$$\text{VoI}^*_{[y, y']} = \max_{[y, y'] \in Y^n \times Y^n} \text{VoI}([y, y'])$$

Fig. 1 illustrates how the value of information for the DM's queries determines the m-th candidate. Fig. 1 (a) and (b) show a sampled set of observations (white dots) in a two-dimensional objective space. Fig. 1 (a) shows non-dominated combinations of observations that may be evaluated, when the dataset size for preference information is zero ($m = 0$). Each colored edge represents the estimated

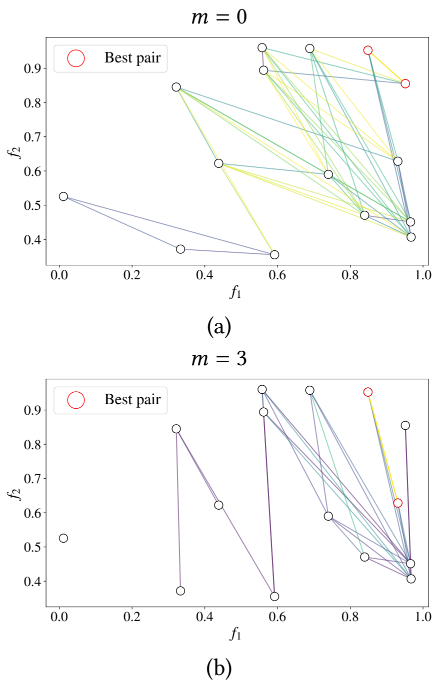

Figure 1: Both figures show a sampled set of observations (white dots) in a two-dimensional objective space during the maximization when the dataset for preference information is empty (a) and when 3 queries are collected (b). The best pair according to Eqn. 6 (red dots) is then selected.

Value of Information according to Eqn. 6, with lighter colors representing more preferable pairs with higher VoI. The best pair (red dots), according to its value of information, is selected and shown to the DM. Then, the dataset is updated and a set of compatible samples of parameters is generated. After performing 3 queries to the DM, Fig. 1 (b) shows a highly reduced set of potential pairs that may provide a benefit to the optimization process. In both figures, non-connected observations present a value of information of zero.

4.6 The Overall Algorithm

The proposed approach is summarized in Algorithm 1. On Line 0, the algorithm begins by fitting a Gaussian process model for each objective to a set of initial solutions $\mathscr{D}_f^n$ using a Latin hypercube (LHS) 'space-filling' experimental design. After initialization, the algorithm continues in an optimization loop until the budget B has been consumed. During each iteration, we compute the VoI of collecting a new solution (x^{n+1}, y^{n+1}) according to $\text{VoI}(x)$ (Line 2) and the VoI of eliciting preference information $\text{VoI}([y, y'])$ (Line 3). The action that gives the greatest value determines whether we collect a sample $(x, y)^{n+1}$ or $([y, y'], q)^{m+1}$ (Line 4). In the first case, the Gaussian process model is updated according to the new solution sample (Lines 5-8) and, for the second case, the posterior distribution $\mathbb{P}[\theta|\mathscr{D}_S^m]$ is updated according to the new $([y, y'], q)^{m+1}$ sample (Lines 10-13). At the end of B samples, the design x with the largest predicted performance $\hat{G}(x)$ is recommended to the user (Line 14).

Algorithm 1: Overall Algorithm.

Input: black box $f : X \to \mathbb{R}^K$ with cost c_f, querying cost c_s, and sampling budget B

0. Collect initial data from f, $\mathscr{D}_f^n$, and fit a Gaussian process for each objective, $\mu_j^n(x)$
1. **While** $b < \text{B}$ **do:**
2. Compute $x^{n+1^*} = \arg\max_x \max_{X_d} \text{VoI}(x; X_d, \{\tilde{Z}_j\}_1^{n_z})$.
3. Compute $[y, y']^{m+1^*} = \arg\max_{[y,y'] \in [\text{Y}^n, \text{Y}^n]} \text{VoI}([y, y'])$
4. **If** $\text{VoI}_x^* > \text{VoI}_{[y,y']}^*$:
5. Evaluate black box function, $y^{n+1} = f(x^{n+1})$
6. Update $\mathscr{D}_f^{n+1} \leftarrow \mathscr{D}_f^n \cup \{(x, y)^{n+1}\}$
7. Fit a Gaussian process to $\mathscr{D}_f^{n+1}$
8. Update budget consumed, $b \leftarrow b + c_f, n \leftarrow n + 1$
9. **Else:**
10. Query DM, q^{m+1}
11. Update $\mathscr{D}_S^{m+1} \leftarrow \mathscr{D}_S^m \cup \{([y, y'], q)^{m+1}\}$
12. Compute a posterior distribution $\mathbb{P}[\theta|\mathscr{D}_S^{m+1}]$
13. Update budget consumed, $b \leftarrow b + c_s, m \leftarrow m + 1$
14. **Return:** Recommend $x_r = \arg\max_x \hat{G}(x; \mathscr{D}_S^m, \mathscr{D}_f^n)$

5 RESULTS AND DISCUSSION

We test the proposed approach on a synthetic test function with two spherical objectives defined over $X = [0, 1]^2$. The center of each spherical function is placed as shown in Fig. 2 (**first row**) of each figure (red crosses). This produces the feasible regions shown in output space, Fig. 2 (**second row**). The aim of this comparison is to show that our proposed approach adapts to the characteristics of the Pareto frontier and selects a sensible number of interactions according to the problem. Notably, a larger distance between the center of each spherical function implies a wider Pareto frontier, and therefore, the required number of preference elicitations increases to achieve better results. In all test functions we set $n_z = n_\theta = 50$ and all results are averaged over 200 replications.

We compare our proposed method (Algorithm 1) against the following benchmarks,

- **Benchmark 1**: The preference elicitation is performed at regular intervals. Therefore, we collect objective function data for a fixed amount of iterations before collecting information from the DM. Each non-dominated pair is randomly selected from the set of sampled observations.
- **Benchmark 2**: A two stage algorithm following the a posteriori preference elicitation paradigm. Thus, the first stage is dedicated to only sample from the objective function, while in the second stage only preference information is elicited from the DM. Each pair is selected using randomly non-dominated observations.

- **Benchmark 3**: We use the same interaction pattern for Benchmark 1, however, each pair is optimized according to the proposed acquisition function for pair selection (Eqn. 6).

For each Benchmark algorithm, the preference information is used to inform the posterior distribution $\mathbb{P}[\theta|\mathscr{D}_S^m]$ to guide the search and pick a final recommended solution. The number of preference elicitations from the overall budget, B, for the Benchmark algorithms is chosen "a priori" and we show results for different budget allocations. For simplicity and without loss of generality, in all figures, we consider the same acquisition cost, $c_s = c_f = 1$. As presented in Algorithm 1, our proposed approach determines the budget allocation dynamically.

Results are shown in semi-log scale in each figure. The horizontal axis represents the number of pairs shown to the DM, whereas the vertical axis shows the confidence interval of the OC after the budget has been completely allocated. For the proposed approach, we also show a horizontal confidence interval representing the range of sample sizes chosen.

Fig. 2 (**third row**) indicates the performance of the different algorithms depending on the amount of preference queries executed. For the first column, the spherical functions are highly overlapping, and the Pareto front is therefore very narrow. Therefore, only a small number of pairs is required to achieve the best result (around 4). However, when the Pareto front widens, the number of preference elicitations required increases. This makes sense, as the solutions one would pick depending on DM preferences differ more significantly, making the result more depending on a good knowledge of the DM preferences. Clearly this trade-off depends on the problem and it would not be possible to know the ideal number of preference elicitations in advance. In all cases, our proposed approach (yellow results) balances the sampling allocation, finding equivalent or better results than even taking the optimal number of pairs for the other methods (for the third test problem with the widest Pareto front, Benchmark 3 might benefit from an even higher number of DM interactions). Furthermore, it is clear that selecting each pair according to $\text{VoI}_{[y,y']}$ (Algorithm 1 and Benchmark 3) yields superior performance compared to selecting each pair randomly (Benchmark 1 and 2), which confirms the usefulness of the criterion.

Intuitively, spreading the interaction through the optimization allows to take advantage of the preference information early in the optimization which we would expect to be beneficial. Comparing Benchmarks 1 and 2 which implement different interaction patterns, but both select pairs randomly, shows that asking the DM at regular intervals (Benchmark 1) provides only a minor improvement in performance compared to asking the DM towards the end of the optimization (Benchmark 2). Our approach presents similar results compared to Benchmark 3 even when the best number of pairs is selected. Note, however, that the best number of pairs in Benchmark 3 has to be pre-determined by the user and may be problem dependent, so comparing with the optimal setting gives the Benchmark 3 algorithm an unrealistic advantage.

6 CONCLUSIONS

We demonstrate how to dynamically determine which of the two actions (evaluating a solution or eliciting DM preferences) is more beneficial. In each iteration, our proposed approach estimates the value of each action and selects the action considered more valuable. A comparison with the standard EMO a posteriori paradigm which would first only optimize then elicit preference information from the DM to identify the most preferred solution, demonstrated that the allocation mechanism of the proposed approach is able to automatically identify a sensible balance between preference learning and objective function optimization, and outperforms even the best budget allocation from the comparison algorithms.

Future directions of research may include the evaluation on a wider range of test problems with different cost configurations and examining the scalability of the proposed approach in higher dimensions, where it should be most beneficial.

REFERENCES

[1] Raul Astudillo and Peter Frazier. 2020. Multi-attribute Bayesian optimization with interactive preference learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 4496–4507.

[2] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. 2020. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*.

[3] Juergen Branke. 2016. MCDA and Multiobjective Evolutionary Algorithms. In *Multiple Criteria Decision Analysis*. Springer, 977–1008.

[4] Juergen Branke, Salvatore Greco, Roman Słowiński, and Piotr Zielniewicz. 2015. Learning Value Functions in Interactive Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation* 19, 1 (2015), 88–102. https://doi.org/10.1109/TEVC.2014.2303783

[5] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc., 4299–4307. https://proceedings.neurips.cc/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf

[6] Sam Daulton, Maximilian Balandat, and Eytan Bakshy. 2021. Parallel Bayesian Optimization of Multiple Noisy Objectives with Expected Hypervolume Improvement. In *Neural Information Processing Systems*.

[7] Kalyanmoy Deb and Jayavelmurugan Sundar. 2006. Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. In *Genetic and Evolutionary Computation Conference*. Association for Computing Machinery, New York, NY, USA, 635–642.

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

[9] David Gaudrie, Rodolphe Le Riche, Victor Picheny, Benoit Enaux, and Vincent Herbert. 2020. Targeting solutions in Bayesian multi-objective optimization: sequential and batch versions. *Annals of Mathematics and Artificial Intelligence* 88, 1 (2020), 187–212.

[10] Jussi Hakanen and Joshua D Knowles. 2017. On using decision maker preferences with ParEGO. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 282–297.

[11] Andrzej Jaszkiewicz and Jürgen Branke. 2008. *Interactive Multiobjective Evolutionary Algorithms*. Springer Berlin Heidelberg, Berlin, Heidelberg, 179–193. https://doi.org/10.1007/978-3-540-88908-3_7

[12] Donald Jones, Matthias Schonlau, and William Welch. 1998. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization* 13 (Jan 1998), 455–492.

[13] Miłosz Kadziński, Michał K. Tomczyk, and Roman Słowiński. 2020. Preference-based cone contraction algorithms for interactive evolutionary multiple objective optimization. *Swarm and Evolutionary Computation* 52 (2020), 100602. https://doi.org/10.1016/j.swevo.2019.100602

[14] Andy J. Keane. 2006. Statistical improvement criteria for use in multiobjective design optimization. *AIAA journal* 44, 4 (2006), 879–891.

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

[16] Zhiyuan Jerry Lin, Raul Astudillo, Peter I. Frazier, and Eytan Bakshy. 2022. Preference Exploration for Efficient Bayesian Optimization with Multiple Outcomes. arXiv:2203.11382 [cs.LG]

[17] Victor Picheny. 2015. Multiobjective optimization using Gaussian process emulators via stepwise uncertainy reduction. *Statistics and Computing* 25, 6 (2015),

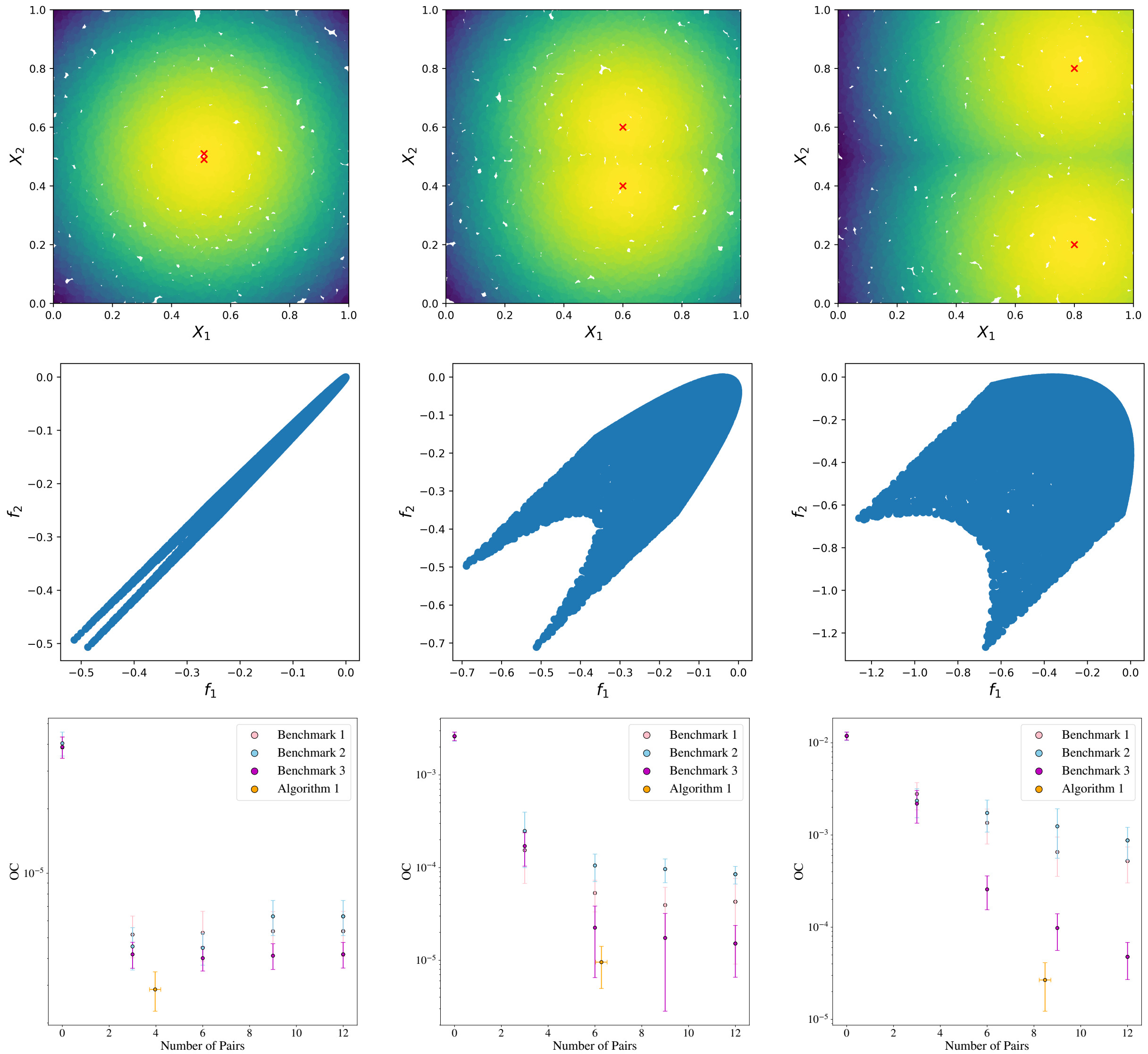

Figure 2: (first row) shows the Spherical test functions and the center of each spherical objective function in solution space such that only the distance between the each center varies. (second row) shows the resulting feasible region in objective space given by the test function above. (third row) Mean and 95% CI of OC. Each CI is generated using 200 replications and $B = 30$.

1265–1280.

[18] Sebastian Rojas-Gonzalez and Inneke Van Nieuwenhuyse. 2020. A survey on kriging-based infill algorithms for multiobjective simulation optimization. *Computers & Operations Research* 116 (2020), 104869.

[19] Marc Schoenauer, Riad Akrour, Michele Sebag, and Jean-Christophe Souplet. 2014. Programming by Feedback. In *Proceedings of the 31st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 32)*, Eric P. Xing and Tony Jebara (Eds.). PMLR, Bejing, China, 1503–1511.

[20] Warren Scott, Peter Frazier, and Warren Powell. 2011. The correlated Knowledge Gradient for simulation optimization of continuous parameters using Gaussian Process regression. *SIAM Journal on Optimization* 21, 3 (2011), 996–1026.

[21] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. 2016. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* 104, 1 (Jan 2016), 148–175.

[22] Michał K. Tomczyk and Miłosz Kadziński. 2019. EMOSOR: Evolutionary multiple objective optimization guided by interactive stochastic ordinal regression. *Computers I& Operations Research* 108 (2019), 134–154. https://doi.org/10.1016/j.cor.2019.04.008

[23] Juan Ungredda, Juergen Branke, Mariapia Marchi, and Teresa Montrone. 2022. Single Interaction Multi-Objective Bayesian Optimization. In *Parallel Problem Solving from Nature–PPSN XVII: 17th International Conference, PPSN 2022, Dortmund, Germany, September 10–14, 2022, Proceedings, Part I*. Springer, 132–145.

[24] Juan Ungredda, Michael Pearce, and Juergen Branke. 2022. Bayesian optimisation vs. input uncertainty reduction. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 32, 3 (2022), 1–26.

[25] Andrzej P. Wierzbick. 1977. Basic properties of scalarizing functionsl for multiobjective optimization. *Optimization* 8 (1977), 55–60.

[26] Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. 2017. A Survey of Preference-Based Reinforcement Learning Methods. *J. Mach. Learn. Res.* 18, 1 (Jan. 2017), 4945–4990.

[27] Bin Xin, Lu Chen, Jie Chen, Hisao Ishibuchi, Kaoru Hirota, and Bo Liu. 2018. Interactive Multiobjective Optimization: A Review of the State-of-the-Art. *IEEE Access* 6 (2018), 41256–41279.