Bayesian Preference Learning for Interactive Multi-objective Optimisation

Kendall Taylor, Huong Ha, Minyi Li, Jeffrey Chan, and Xiaodong Li
School of Computing Technologies, RMIT University
Melbourne, Australia
{kendall.taylor,huong.ha,minyi.li,jeffrey.chan,xiaodong.li}@rmit.edu.au

ABSTRACT

This work proposes a Bayesian optimisation with Gaussian Process approach to learn decision maker (DM) preferences in the attribute search space of a multi-objective optimisation problem (MOP). The DM is consulted periodically during optimisation of the problem and asked to provide their preference over a series of pairwise comparisons of candidate solutions. After each consultation, the most preferred solution is used as the reference point in an appropriate multiobjective optimisation evolutionary algorithm (MOEA). The rationale for using Bayesian optimisation is to identify the most preferred location in the decision search space with the least number of DM queries, thereby minimising DM cognitive burden and fatigue. This enables non-expert DMs to be involved in the optimisation process and make more informed decisions. We further reduce the number of preference queries required, by progressively redefining the Bayesian search space to reflect the MOEA's decision bounds as it converges toward the Pareto Front. We demonstrate how this approach can locate a reference point close to an unknown preferred location on the Pareto Front, of both benchmark and real-world problems with relatively few pairwise comparisons.

CCS CONCEPTS

• **Applied computing** → **Multi-criterion optimization and decision-making**; • **Theory of computation** → Interactive computation; *Active learning*;

KEYWORDS

Multi-objective optimisation, preferences, evolutionary algorithms, interactive algorithms, Bayesian optimisation, preference learning, pairwise comparisons, active learning.

ACM Reference Format:
Kendall Taylor, Huong Ha, Minyi Li, Jeffrey Chan, and Xiaodong Li . 2021. Bayesian Preference Learning for Interactive Multi-objective Optimisation. In *2021 Genetic and Evolutionary Computation Conference (GECCO '21), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3449639.3459299

GECCO '21, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8350-9/21/07…$15.00
https://doi.org/10.1145/3449639.3459299

1 INTRODUCTION

Preference-based multi-objective optimisation incorporates a decision maker's (DM's), preference for a certain type of solution, or objective values, into the optimisation process [9, 17]. In an a-priori approach, preferences are elicited before optimisation and used to drive and focus the search for suitable solutions. At the other end of the scale, a-posteriori methods use preferences acquired after optimisation to reduce the solution set to contain fewer yet preferred solutions. While both techniques have their merits, the progressive (or interactive) elicitation of preferences during the optimisation process is considered to have additional benefits [4, 42].

By interleaving elicitation with the optimisation process, the interactive approach allows the DM to learn about their problem and potential solutions. This helps to refine their preferences and helps focus the optimisation algorithm's search toward a section (or sections) of the Pareto Front. The latter leads to a lower computational cost as the entire Pareto optimal set need not be found. At the completion of optimisation, the DM is more informed about their problem and only has to choose from a preferred subset of possible solutions rather than a complete optimal set [8, 9, 24]. Further, in a real-world setting, the DM's involvement throughout the process can facilitate greater ownership of the result and increase the chance of successful implementation.

Attempts have been made to combine operations research's elicitation strategies with those of evolutionary optimisation [3, 9, 41]. For the expert DM, this has resulted in approaches where they can use their domain knowledge to express their preferences in the form of objective trade-offs, desirability thresholds, weights and outranking measures (amongst others) [3, 9]. Unfortunately for the non-expert DM with little or no domain knowledge, such techniques can be daunting, require considerable cognitive burden and are often considered non-intuitive. An alternate intuitive approach is the reference point (or direction) method in which the DM specifies their aspiration levels for a problem's objectives [41, 59]. While successful in finding regions of interest along the Pareto Front, it is often assumed preferences are given, and the elicitation process is rarely discussed.

To facilitate the non-expert to express their preferences requires a different approach. Such a task requires low burden methods that help the DM explore possible solutions and easily discover and articulate their preferences. Pairwise comparisons (PWC), where the user indicates their preference for one option over another, is considered one of the lowest burden elicitation methods [19, 30]. However, a single PWC yields little information and many comparisons may be needed to obtain a total order over n options. At best this will be $\Omega(n \log n)$ but with randomly selected pairs can be as high as $O(n^2)$.

However, efficient and effective preference learning with PWCs can be accomplished in a practical, low burden setting using the machine learning technique of *active learning*. Active learning is a technique whereby training data can be minimised by labelling data that is selected on the basis of maximising information gain or minimising uncertainty [48]. Active learning is a sequential process where the answers to previous queries inform the selection of subsequent queries. The aim is to learn the maximum amount of information with a minimal number of queries. This approach blends well with interactive preference elicitation and learning techniques where preferences are obtained progressively from a DM during a multi-objective optimisation session. One type of active learning is Bayesian optimisation (BO) combined with Gaussian Processes (GPs), this is the focus of preference learning in this work.

Bayesian optimisation (BO) is a global optimisation method which uses Bayes Theorem and sequential sampling of a surrogate function (often a GP), to learn the maximum (or minimum), of an unknown, expensive to evaluate blackbox function [12, 29, 49].

BO has been shown to be effective in learning preferences with a minimum number of queries [11, 16]. In the case presented here, the unknown blackbox function is a DM's preferences, considered for the sake of simplicity as a value, or utility function. Such a function is expensive because it takes time to elicit responses from queries. The number of queries that can be presented requires cognitive effort limited by fatigue (and possibly patience).

It needs to be emphasised that BO has an entirely different role in this work than which it has traditionally served in multi-objective optimisation. Rather than being used as the principal means of problem optimisation with DM preferences incorporated into the BO method (such as [2, 28, 31]), our approach uses BO to learn a DM's preferences *outside* of a non-BO reference-point based MOEA.

Using BO, the number of queries required to find a blackbox function optimum can still be more than reasonably expected of a human DM to answer. In the work of [13] for example, the performance of BO on a four-dimensional search space required more than 40 queries to reach an acceptable level of accuracy. For a six-dimensional problem, the authors concluded that the search space was too large to find the optimum within their 50 query budget. Given that multi-objective optimisation problems often contain a high number of variables in the decision search space, the task of using a BO preference model to reduce DM queries is made more difficult. Simply combining the two techniques will not achieve sufficient reduction in DM cognitive burden. The approach taken in this work progressively reduces the BO search space in tandem with the convergence of an MOEA's generated solutions toward the Pareto Front. This technique allows the BO process to find the search space's optimal region with significantly fewer queries.

Therefore, this work aims to demonstrate how a non-expert DM's preferences can be learned using BO and interactively incorporated into an existing MOEA optimisation process. For this work, we incorporate the Bayesian learning of preferences with the reference point based algorithm R-NSGA-III [58]. The result is a method requiring low cognitive burden from a DM and with minimal or no domain knowledge. A framework for low burden preference elicitation from non-expert DMs also has the potential to expand the use of Multi-objective optimisation (MOO), into areas where non-experts proliferate, (such as travel itinerary creation; new dwelling specification, and; consumer product recommendation), by increasing the 'accessibility' of preference information.

Specifically, this work seeks to provide answers to the following questions:

(1) How can we elicit preferences and learn their structure for a multi-objective optimisation problem while minimising the decision maker's cognitive burden?
(2) How can we design and incorporate an active learning BO approach into an interactive MOO process and exploit any synergies that exist?
(3) Does the adopted approach (Bayesian method) help produce quality solutions with relative light cognitive burden?

1.1 Related Work

Within the field of machine learning, Gaussian Processes have been successfully applied to preference learning in [16], where a new likelihood function was proposed to learn preference relations in a Bayesian framework. The approach in [16] was extended further in [15] to incorporate instance ranking using PWCs. Both prior and likelihood models used in [15, 16] are single-user task-based specifications that correspond to the work in this paper. Using discrete choice data, [11] proposed an active learning process for Bayesian learning of individual user preferences. Similar to our work, they assume a cold-start scenario with PWCs. We adopt a similar approach to [11] and implement a probit regression model with Laplace approximation to relate binary preferences to a continuous utility function. This process is central to learning preferences via PWCs and BO.

The use of BO to solve multiobjective problems with preference data has been successfully implemented in [1, 2, 6, 28, 31, 32]. Each of these approaches have used BO as the optimiser of a multi-objective problem and incorporate DM preferences either interactively or a-priori. The use of BO as the main optimiser differentiates these works from that presented in this paper. Here, we use BO solely to discover a DM's preferences and use the acquired information to update a reference point in an interactive MOEA.

Methods developed by [24, 38] use a polynomial value function and radius basis function network respectively, to learn preferences in the context of multiobjective optimisation. However, both methods place a considerable burden on the decision-maker with the number of queries required. In the case of [24], 70 PWC decisions were required in the best case and 150 in the worst. For [38], 157 judgements were required of the decision-maker with each query requiring an absolute numeric score to be given for each option presented. This work however, seeks to minimise the number of queries asked of the user to reduce cognitive burden by modelling preferences using BO and active learning.

A recent paper [56] investigates indirect preference elicitation techniques for MOO and specifically examines the selection of comparison solutions for presentation to a DM. Results from this work reveal the benefits of selecting comparisons based on maximising information gain. While the active learning strategies examined were incorporated into an interactive decomposition MOEA, Bayesian methods including BO were not considered.

2 BACKGROUND

2.1 Multi-objective Optimisation

Multiobjective optimisation involves finding solutions to optimisation problems defined by at least two conflicting objectives. The general form of a multiobjective optimisation problem (MOP) can be defined as [18, 23, 61]:

$$\text{Minimise/Maximise} \quad [f_1(x), ..., f_m(x)]; \tag{1}$$

$$\text{subject to} \quad x \in \Omega, \tag{2}$$

where *decision space* is represented by Ω, the set of all possible solutions and the objective functions f_i map values from the decision space to real values $f_i : \Omega \to R$ where $i = 1, 2, ..., m$, concatenation of the values $[f_1(x), ..., f_m(x)] \in R^m$ form a m-dimensional *objective space* and m is the number of objective functions.

2.2 Bayesian Optimisation

BO is a global optimisation technique that uses Bayes Theorem to build a probabilistic model of the object function [12, 33, 49]. That is, given a model, and some evidence in the form of data or observations, the *posterior* probability of the model is proportional to the *likelihood* of the evidence, multiplied by the model's *prior* probability [12]. Specifically, this is expressed as:

$$P(M|E) \propto P(E|M)P(M), \tag{3}$$

where M is the model and E the evidence.

BO uses a small set of initial function evaluations to initialise the model. An acquisition function is then used to find the next most-informative location to query. After evaluating this suggested location, the model is updated, and a new suggestion is generated. The process continues until an evaluation budget is exhausted or a solution acceptable to the DM is found. The acquisition function constructs a utility function from the model's posterior distribution.

BO is most often employed where objective functions are complex, noisy, expensive to evaluate or a closed-form expression does not exist [12, 49]. For this work, BO is used to learn a DM's preferences over candidate solutions generated by an MOEA. This is achieved by assuming the existence of a latent value function representing the user's preferences and optimising this function to find the global maximum representing the DM's most preferred options. The objective function in this case is the unknown function determining the user's preferences.

2.2.1 *Gaussian Processes.*

As is common with BO, Gaussian Processes (GPs) are often used as a surrogate for the objective function. A GP is a non-parametric stochastic process that defines a Gaussian distribution over functions with a continuous domain. It is defined as *a collection of random variables, any finite number of which have a joint Gaussian distribution* [46]. A GP defines a prior over possible functions that transforms into a posterior over functions after incorporating evaluated data [27, 44]. The method can find functions that approximate a user's preferences quickly and accurately [16].

A multivariate Gaussian is defined by some mean $\mu(x)$ and a covariance matrix $\Sigma(x)$. A positive definite kernel function κ determines the covariance matrix such that $\Sigma_{i,j} = \kappa(x_i, x_j)$. The kernel takes as input two points (usually vectors in Euclidean space), and returns a scalar representing the similarity between the inputs [12, 26, 44]:

$$k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}, \Sigma = Cov(X, X') = k(x, x'). \tag{4}$$

Consequently, a GP is completely specified by its mean and covariance, such that:

$$f(\mathrm{x}) \sim GP(m(\mathrm{x}), \kappa(\mathrm{x} \cdot \mathrm{x}')), \tag{5}$$

where,

$$\begin{aligned} m(\mathrm{x}) &= \mathbb{E}[f(\mathrm{x})], \\ \kappa(\mathrm{x} \cdot \mathrm{x}') &= \mathbb{E}[(f(\mathrm{x}) - m(\mathrm{x}))(f(\mathrm{x}') - m(\mathrm{x}'))^T]. \end{aligned} \tag{6}$$

2.2.2 *Kernels (covariance functions).*

The kernel κ, embeds any existing problem knowledge into the process by defining the shape of the distribution and characteristics of the function we need to predict. The kernel and its hyperparameters are an important part of the BO process as they define the similarity between points and thus control the magnitude and smoothness of sampled GPs. A poorly chosen kernel that incorrectly models smoothness for a given problem will result in poor performance of a BO model that degrades further with increased dimensionality [47].

Many types of kernel are used in BO, the most popular are stationary kernels which are invariant to input space transformations (see [46] for an account of non-stationary and dot-product kernels). Some of the more common of this type include the *Squared Exponential*, *Linear*, *Rational Quadratic*, and *Matérn*. For learning a preference function this work uses the Matérn kernel as its generality and flexibility provides a good starting point when learning functions with an unknown form. Further, the Matérn kernel is $\lceil v \rceil - 1$ times differentiable enabling its length-scale hyperparameter to be learnt by Maximum Likelihood Estimation during the BO process. The Matérn kernel can be expressed in the form:

$$\kappa(x, x') = \frac{1}{\Gamma(v)2^{v-1}} \Big(\frac{\sqrt{2v}}{l} d(x, x') \Big)^v K_v \Big(\frac{\sqrt{2v}}{l} d(x, x') \Big), \tag{7}$$

where K_v is a Bessel function, v and l are positive parameters and Γ is a gamma function [46].

2.2.3 *Acquisition function.*

When choosing where to suggest the next evaluation, the acquisition function balances the exploitation of the discovered optimal location with the need to explore areas with high uncertainty. This is an important feature to ensure that the algorithm finds a global optima and does not over-exploit local optima. Without loss of generality, when maximising an objective the acquisition function should suggest locations where the predicted mean is high as well as areas where the variance is high, or both. Commonly used acquisition functions include Probability of Improvement (PI) [37], Expected Improvement (EI) [33, 43], Upper Confidence Bound (UCB) [21, 22, 52], and Thompson Sampling (TS) [14, 54].

With the exception of TS, the other three functions include a parameter for adjusting the balance between exploitation and exploration. Nonetheless, PI and EI are both structurally exploitative, and TS tends toward exploitation as suggested locations are identified from the maximum of the sampled GPs. UCB however, is optimistic in uncertain conditions and therefore tends toward exploration, a trait useful for multi-modal and large search spaces (like those encountered during preference learning). UCB also has strong theoretical results, proving it will converge to the global optimum (within the context of multi-armed bandit problems) [52]. For these two reasons UCB has been chosen as the acquisition function for this work. UCB is defined as:

$$UCB(x) = \mu(x) + \beta^{1/2}\sigma(x), \tag{8}$$

where β is a positive parameter that trades off the regions of high mean $\mu(x)$ (exploitation), and high variance $\sigma(x)$ (exploration).

BO works very well with multiple dimensions however the $O(n^3)$ complexity of the technique (due to the need to invert the covariance matrix for inference), becomes more of an issue as dimensionality increases [12, 35].

2.3 Preference elicitation and learning

The task of learning a DM's preferences first involves the acquisition, or elicitation of preferences via some form of interaction with, and information extraction from, the DM. A wide variety of methods exist and largely focus on allocating importance to attributes of potential solutions [57]. This is often achieved either directly with the DM using numerical values to score attributes based on preference or by setting trade-off limits or aspiration levels, or indirectly via ranking objects in order of preference, either as a list or in pairs (PWCs).

A PWC presents the DM with two options who then express their preference for one over the other. This method overcomes the difficulty of specifying preferences as numeric values, which studies have shown to be more difficult for human DMs than relative comparison, leading to increased inaccuracies. Unfortunately, the complete elicitation of preferences requires a considerable number of PWCs and consequent cognitive burden with each PWC supplying only a small amount of preference information.

Our preference model is based on multi-attribute utility theory [36] and the representation of preferences as a value (or utility) function. This model relies on the assumed existence of a latent utility function determining the DM's preferences that can only be queries indirectly. Formally, a value function is a mapping $f : \mathcal{X} \to \mathbb{R}$ that assigns a degree of value $f(x)$ to each item (object) x and induces a complete order on $\mathcal{X}$ and obtain a ranking for x. In the context of preferences, x is an option or choice defined by d attributes and the ordered set $\mathcal{X}$ defines the transitive set of preference relations $\{x_1 \succ x_2, ..., x_{n-1} \succ x_n\}$.

It is important to note that it is not always the case that such a function is completely formed a-priori, rather a preference function may be developed or refined after a DM learns more about a problem and its possible solutions. Preference learning using BO does not depend on a fully formed preference function existing prior to querying a DM. As the BO model is rebuilt, and the likelihoods of solution preferences recalculated after each query, the model is flexible enough to accommodate preference changes during the optimisation of a multi-objective problem.

We minimise the number of PWCs used via the partial modelling of a DM's value function and the use of active learning to ensure comparisons are chosen to elicit the greatest amount of information. We focus on identifying the DM's most preferred regions of a MOP's objective space and construct comparisons combining the currently

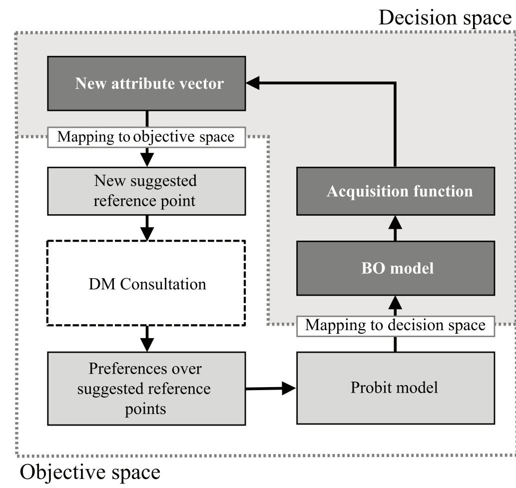

Figure 1: The process of learning preferences using BO and a probit comparison model. The sections illustrate which parts work on attribute vectors (from the decision space) and those that work on reference points (from the objective space).

most preferred option with an alternative selected for its probability of being a superior choice.

The process of consulting the DM involves generating PWCs of potential reference points and presenting them to the DM for evaluation. This could take the form of either (or both), graphical or numerical representations of objective value differences incorporated into a user interface. In our work, PWCs are created using the current most preferred option s_{pref} (a solution from the objective space), and a new unseen option s' mapped from an attribute vector a' (generated in the decision space). For the initial consultation, a small sample of attribute vectors ($n > 2$) is generated randomly. In the case of a MOP, each attribute is a decision variable and its value is selected from within the variable's bounds. For the initial consultation s_{pref} does not yet exist and the first PWC presented to the DM will contain two unseen options.

The acquisition of preferences occurs using solution vectors from the objective space, while the BO model learns using attribute vectors from the decision space. The learning process is illustrated in Figure 1. The comparison data is then processed using a probit model to produce solution likelihoods for inclusion into the BO model. This model is derived from the Thurstone-Mosteller law of comparative judgment [55] and relates the binary data from PWCs to a continuous latent function.

Following the work of [12], the preference value functions of two options r and c is: $v(r_i) = f(r_i) + \epsilon$ and $v(c_i) = f(c_i) + \epsilon$ where ϵ represents noise from a Gaussian distribution. The probability of an option r being preferred to option c can be expressed as:

$$P(r_i \succ c_i | f(r_i), f(c_i)) = \Phi\Big(\frac{f(r_i) - f(c_i)}{\sqrt{2}\sigma_{noise}}\Big), \tag{9}$$

where Φ is the Cumulative distribution function of a standard Gaussian distribution.

To model the DM's value function, an estimate of the posterior distribution of the function needs to be made in order to suggest new sampling locations. This is achieved using Laplace approximation with details provided in [10] and [12].

After all consultations have been completed the attribute vector a_{pref} is considered the focus of the DM's preferred region in the decision space, and s_{pref} the focus in the objective space.

The modelling of the user's entire preference function is not required as we are only interested in solutions of greatest preference. The Bayesian model suggests potential comparisons which are presented to the user periodically. The model is then updated with the results and identifies a set of objective values to be used as a reference point in the multiobjective optimisation algorithm solving the main problem.

3 INTEGRATION WITH MULTI-OBJECTIVE OPTIMISATION

Within MOO, preferences elicited with PWCs are often used to help construct value functions using mathematical programming techniques, ordinal regression or neural networks [5, 20, 38, 50]. Such approaches use the inferred function to set objective weightings or compliment the dominance principles used in population generation. This work differs in not using the value function from the Bayesian preference model in the MOO process. Instead, we use the DM's most preferred solution as a reference point in a reference point based MOEA. At the conclusion of each DM consultation, the most preferred objective vector is used as the reference point for the MOEA's iterations until the next consultation.

While BO can find global optima with a minimal number of queries, a high dimensional search space may temper its efficiency, requiring more queries than a DM would prefer to answer. This problem can be mediated by either increasing the preference information acquired, or by minimising (or truncating) the search space. Fortunately, the integration with multi-objective optimisation presents some synergies that can be exploited to overcome a large Bayesian search space defined by decision variable bounds. After each MOEA generation, candidate solutions are generated closer to the problem's Pareto Front. Consequently, the corresponding decision variables' bounds are likely to be reduced (it should be noted however that such a reduction is problem dependant). This is further compounded when solutions are increasingly concentrated within a region of interest (RoI), as is the goal of preference-based MOEAs (refer to Figure 2 for an illustration of this process). We harness this reduction in decision variable bounds to truncate the Bayesian model's search space, thereby increasing the rate of convergence and minimising DM queries.

Learning preferences using BO is combined with a reference-point based MOEA by interleaving the learning process within the MOEA optimisation process. It is important to note that the BO process is *independent* of the MOEA and only requires the ability to pause operation of the MOEA at the beginning of a generation, read the current results, and update the reference point and its RoI size. Figure 3 provides an activity diagram outlining the steps involved in such an integration.

Before beginning the process a decision on the desired DM interaction pattern needs to be made. Such a pattern includes setting which generation to begin interaction, the interval of subsequent consultations, the total number of consultations, and the number of PWCs to present to the DM at each consultation. Each of these tasks is non-trivial and depends on the problem type and the desires

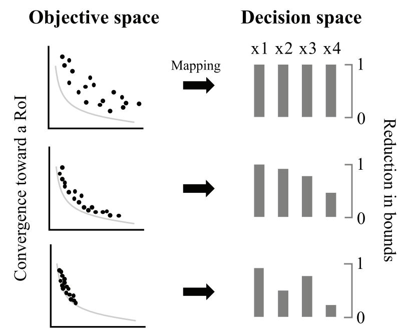

Figure 2: Illustration of decision variable (x1, x2, x3 and x4) domain reduction in conjunction with objective space solution convergence to a RoI.

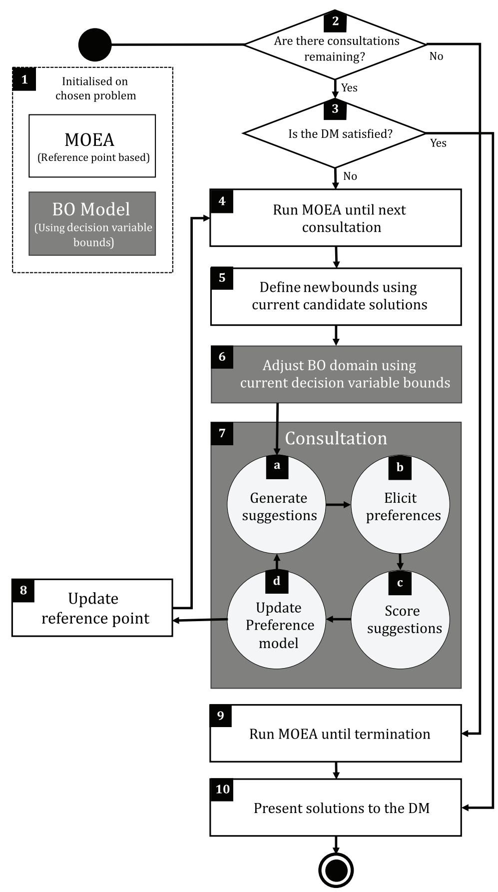

Figure 3: Activity diagram illustrating a simplified version of the process integrating Bayesian preference learning with a reference point based MOEA

of the DM. [34] provides an outline of several common patterns, while [40] and [51] (for example), use specific methods to determine when to interact during optimisation. We adopt a simple approach, and use consultations evenly spaced throughout the optimisation run (determined by a fixed number of generations), and the number of PWCs used for each consultation is limited to a maximum of four (see Table 1 for experimental details).

Once an interaction pattern has been set, the process proceeds as follows (step numbers correspond with those in Figure 3):

- Step 1 : Initialise the MOEA on the desired MOP with a starting reference point located in the middle of the objective bounds with a large RoI (this encourages diverse solutions before the actual consultation begins). Initialise the Bayesian preference model using the decision variable bounds of the MOP;
- Step 2 : If there are no further consultations proceed to (9);
- Step 3 : Present a selection of candidate solutions to the DM. If they are satisfied, proceed to step (10), otherwise continue;
- Step 4 : Run the MOEA and pause its progress when the next consultation is scheduled;
- Step 5 : Using the decision variable bounds of the current set of candidate solutions define a new set of preference model bounds;
- Step 6 : Update the bounds of Bayesian preference model with the bounds defined in (5) (with a high likelihood of reducing the search space of the Bayesian model);
- Step 7 : Consult with the DM (repeat for the desired number of PWCs per consultation);
  - (a) If this is the first consultation, a set of $n > 2$ reference points are randomly generated, and paired combinations are created. If this is not the first consultation then the most preferred reference point thus far is paired with one suggested by the Bayesian model;
  - (b) Present the PWCs (created in step 7a), to the DM and gather preferences;
  - (c) "Score" suggested reference points using the probit model described above;
  - (d) Update the Bayesian preference model by fitting it with the newly observed preferences and corresponding suggested reference points.
- Step 8 : Update the MOEA's reference point to the most preferred suggested reference point from step (7). If step (7) was the first consultation, then set the RoI size of the MOEA to the desired size for subsequent generations. Proceed to step (2);
- Step 9 : Run the MOEA until termination
- Step 10 : Present the subset of Pareto optimal solutions to the DM for final selection

The result of this process is an interleaved combination of probabilistic preference learning, and preference focused multiobjective problem optimisation. The cognitive burden on the user is minimised. Simultaneously, the search for solutions to the overriding optimisation problem is concentrated on the part of the search space yielding the most preferred solutions.

4 EXPERIMENTAL SETUP

To assess the effectiveness of learning preferences with BO, and integrating into a reference point based MOEA optimisation process, we compare our method's suggested comparison points with those chosen using a Non-BO approach.

Experiments are performed using the ZDT [62] problem set for bi-objective optimisation and the DTLZ [25] set for three objective problems. A selection of two and three objective real-world problems from the RE [53] problem set are also used (see Table 1 for details). The chosen RE problems all have continuous variables with known or approximated Pareto Fronts and include: RE2-4-1 (Four bar truss design); RE2-2-4 (Hatch cover design); RE3-5-4 (Vehicle crashworthiness design), and; RE3-4-7 (Rocket injector design).

The MOEA used is R-NSGA-III from the Pymoo multiobjective optimisation for Python package [7]. R-NSGA-III is a decomposition reference point based MOEA which can find solutions in multiple RoIs given multiple reference points. In this work only one reference point has been used to simplify evaluation. Initialisation used the default values from Pymoo with the number of reference directions set to 12 for two dimensional problems and 18 for three dimensional.

The BO model is initialised with dimensions reflecting the MOP's decision space variables. The acquisition function used is Upper Confidence Bound (UCB), and the exploitation/exploration ratio parameter is set to 2.5 to encourage a balance between exploration and exploitation within the search space. A Matérn 5/2 kernel is implemented with a length-scale set to 1.0 for the prior, which (along with the scale-factor), will be fitted using the Maximum Likelihood Estimation method.

4.1 Non-BO baseline method

The Non-BO process operates in a manner very similar to the preference elicitation procedure outlined above. However, rather than learning a preference function and using BO to suggest new reference points for the DM to evaluate, it generates random attribute vectors within the bounds of the decision search space, which are then paired with the current most preferred reference point for evaluation. The progressive reduction of this search space during the MOEA's operation is performed the same as with the BO approach, as is the mapping of attribute vectors to the objective space.

4.2 Performance metric

The artificial decision maker (DM) uses a 'golden' point (G-Pnt), located on the Pareto Front of a problem to represent their preferred solution (or RoI). This point fulfils a role similar to the 'golden' value used in [38, 39], that is, it represents a pre-optimisation specified location on a problem's Pareto Front that is the focus of our artificial DM's preferences. G-Pnts are solely used to elicit preferences from an *artificial DM*, whereby solutions closer to a G-Pnt are preferred. To evaluate our approach, G-Pnts were randomly selected from diverse regions of each problem's Pareto Front, and their locations keep static throughout the optimisation process. With each PWC presented to the DM, a preferred solution is identified. This preference is determined by the solution in the pair closest to the G-Pnt in the objective search space (using a normalised Euclidean distance). Neither the MOEA nor the Bayesian preference model are aware of the G-Pnt locations or even their existence.

For 50 runs of each method, the distance from the most preferred reference point is compared to a series of G-Pnts located on the Pareto Front of benchmark problems from the ZDT, DTLZ and RE problem sets. G-Pnts represent the DM's RoI and are unknown to the preference models and the MOEA. When PWCs are presented to our artificial DM, preference is expressed for the point closest to the G-Pnt. In the situation with a human DM, such a G-Pnt

Table 1: Problem variables, consultations, and pairwise comparisons (PWCs). Periodic consultations are distributed using uniform intervals (specified by a generation number: First|Last|Increment), and the number of consultations and PWCs per interaction were varied depending on problem difficulty.

Problem		Obj.	Vars	Cons.	Interval	PWCs/ Cons.	PWCs	Generations
ZDT*	1	2	7	4	20|80|20	3	12	100
	2	2	7	4	20|80|20	3	12	100
	3	2	7	4	20|80|20	3	12	100
	4	2	7	4	50|200|50	4	16	250
DTLZ**	1	3	7	9	220|380|20	3	27	400
	2	3	7	9	10|90|10	3	27	100
	3	3	7	9	10|90|10	3	27	100
	5	3	7	9	10|90|10	3	27	100
	6	3	7	9	10|90|100	3	27	150
	7	3	7	9	120|280|20	3	27	300
RE	2-4-1	2	4	4	25|100|25	3	12	150
	2-2-4	2	2	5	10|50|10	4	20	100
	3-5-4	3	5	9	50|450|50	3	27	500
	3-4-7	3	4	9	10|90|10	3	27	100

* ZDT5 was not evaluated as it uses discrete variables, and ZDT6 is not presented as all but the first decision variable converge to zero when solutions approach the Pareto Front, thereby invalidating the BO approach.
** DTLZ4 was not included as the non-linear mapping between parametric and decision variables results in neither the BOS nor baseline NBOS method being able to locate any preference area other than in the dense set of solutions along the $f_M - f_1$ plane [25]

need not exist as preferences will be determined by the unknown decision-making process of the human DM. For the sake of simplicity, we assume our artificial DM has consistent, non-contradictory preferences; however, this may not be the case with a real-world DM. It is important to note that the DM (either real or synthetic), does not explicitly choose a reference point nor specify preferences in any other way except by expressing their preference toward one option over another using PWCs.

For each combination of preference learning method (BO and Non-BO), multi-objective problem and G-pnt, 50 independent runs were conducted. At their conclusion, the closest suggested reference point to the G-pnt was identified for each set of 50 runs and comparisons made between both the Bayesian and Non-BO preference learning approaches. Suggestions located closer to the G-pnt are considered better than those further away.

5 RESULTS AND DISCUSSION

Overwhelmingly, the Bayesian approach performed better than the Non-BO method in locating a reference point close to the artificial DM's G-Pnts. Of the 66 problem instances evaluated (8 problems with 3 'golden' points and 6 problems with 7 'golden' points), approximately 95% reported statistically significant differences between mean values at the 0.05 significance level using the Wilcoxon signed-rank test, (see Table 2).

While the BO preference model performed extremely well, there were (predictably), variations in performance between problems, and differences evaluating the same problem but with various G-Pnt locations. The three objective DTLZ6 problem had the most number of BO model wins for each target G-Pnt while the bi-objective ZDT3 and three objective DTLZ1 problems had the lowest number. Neither ZDT3 nor DTLZ1 recorded 40 or more wins out of the 50 runs for any G-Pnt location.

DTLZ1 was the only problem from the DTLZ problem set that recorded a result (G-Pnt location 4), that was not statistically significant at $\alpha = 0.05$. Examination of the box plot for this problem (Figure 4a), illustrates the lower interquartile range for each G-Pnt and the influence that outliers had on the Non-BO method's mean values. The means for all target locations for the Bayesian method were all within the interquartile range as opposed to the Non-BO approach, which were not. There is little doubt that for the Non-BO method, the incidence of outliers was much greater than for the Bayesian approach, indicating the latter's superior convergence toward the target location.

Given that suggested reference points are generated using attribute vectors (decision variables), it is unsurprising that the performance is not consistent across all G-Pnt locations in the objective space. Some locations appear to be easier to find than others even if in close proximity in the objective space. For example, the 'golden' points two and five for DTLZ1 are relatively close together on the Pareto Front, yet the interquartile ranges of the closest references points for both Bayesian and Non-BO methods for 'golden' point two are noticeably smaller than those for G-Pnt five.

The overall lower performance of BO on the DTLZ1 problem may be due to a known weakness of BO: as mentioned earlier (see 2.2.2), performance is dependent on an appropriately chosen kernel; further, the method relies on the kernel's hyperparameters being learnt correctly after an initial number of data observations. If the kernel chosen for this work (i.e. a 5/2 Matérn kernel) was not a good fit for the problem, or the technique used to tune its

Table 2: Wins/Loses of the BO versus Non-BO approaches over 50 runs of each problem instance. A win is classified as a smaller distance between the closest suggested reference point to the decision maker's G-Pnt. All results were statistically significant using the Wilcoxin Signed Rank test with $\alpha = 0.05$ except for those underlined. Problems with a two dimensional Pareto Front were evaluated with three 'golden' points (G-Pnts), while those with three dimensions were evaluated with seven G-Pnts.

Problem		'Golden' point (G-Pnt) ID 1	2	3	4	5	6	7
ZDT	**1**	45/5	44/6	43/7	–	–	–	–
	2	42/8	39/11	44/6	–	–	–	–
	3	36/14	32/18	32/18	–	–	–	–
	4	36/14	37/13	41/9	–	–	–	–
DTLZ	**1**	38/12	38/12	35/15	29/21	34/16	36/14	38/12
	2	45/5	44/6	49/1	47/3	48/2	48/2	46/4
	3	43/7	43/7	47/3	42/8	41/9	43/7	44/6
	5	32/18	42/8	40/10	–	–	–	–
	6	50/0	48/2	50/0	–	–	–	–
	7	48/2	41/9	49/1	46/4	35/15	47/3	49/1
RE	**2-4-1**	41/9	37/13	30/20	–	–	–	–
	2-2-4	41/9	36/14	34/16	–	–	–	–
	3-5-4	36/14	40/10	42/8	37/13	33/17	32/18	29/21
	3-4-7	37/13	48/2	39/11	46/4	41/9	44/6	49/1

'–' denotes non-applicable 'golden' point (G-Pnt) IDs.

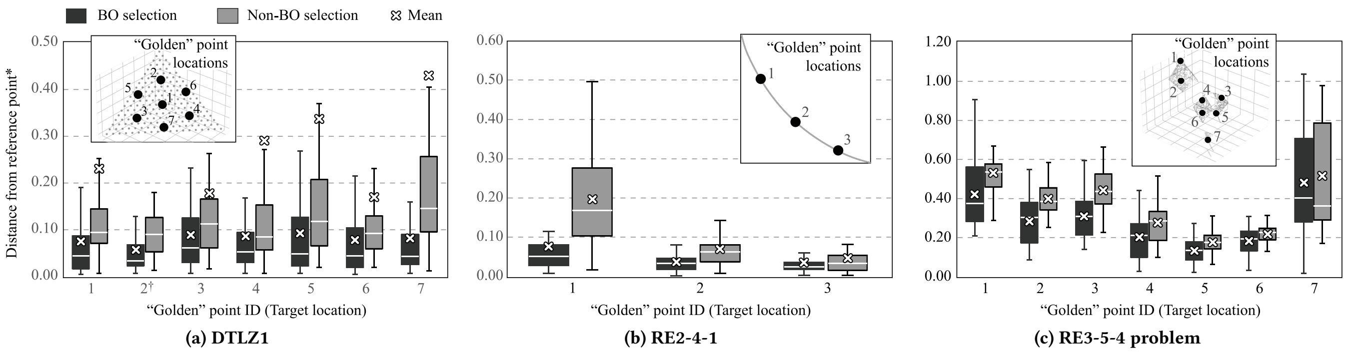

Figure 4: Box plots of the proximity of the final suggested reference point for three selected problems and their target 'golden' points (G-Pnts) over 50 runs

* Normalised euclidean distance

† The mean of the Non-BO method for DTLZ1, G-Pnt number two is 1.128 but is not displayed as it falls outside the chart area.

hyperparameters (i.e. Maximum Likelihood Estimation), failed to find optimal settings, then performance will be degraded.

An evaluation of the four real-world problems from the RE set of benchmark problems there were two instances where results between BO and Non-BO selection were not statistically significant. For the problem RE2-4-1, based on the *four bar truss* problem, the BO approach's ability to find 'golden' point three was very similar to that of Non-BO selection (see Figure 4b). The mean for BO was 0.035 with an interquartile range of 0.01, while Non-BO selection had 0.047 and 0.018 respectively. The MOEA (R-NSGA-III), used for this problem found solutions close to the Pareto Front in only a few generations. This, combined with a tendency to find solutions more readily to the right of this problem's Pareto Front results in the Non-BO method performing very well.

RE3-5-4 is based on the real-world problem *vehicle crashworthiness design* and has a disjoint Pareto Front with 4 distinct sections of optimal solutions. While the Bayesian model performed significantly better than Non-BO for all but the 7th G-Pnt, the average distance of the best reference point and its interquartile ranges varied considerably compared to many of the other problems tested. This is obviously not an easy problem to solve, and the location of the G-Pnt in the objective space determined how well both methods performed. It is important to note that if the MOEA used to generate solutions to the problem has difficulty in certain areas of the search space then finding a target location will also be difficult as the preference search space will not be sufficiently reduced in the correct region. As can be seen in Figure 4c, the 7th G-Pnt for this problem is located in a small narrow region of the Pareto Front. While some of the runs using the BO approach have been able to locate close to this target location, the vast majority have not.

Although not presented here, experiments with 'golden' points located on the extremities of each problem's Pareto Front revealed similar outcomes to those located within the bounds of the Pareto Front.

While the experiments outlined above focused on problems with two and three objectives, this approach can be used with problems comprising more objectives. However, a larger number of objectives does increase the cognitive demands on a DM when evaluating potential solutions. Increasing the number of decision variables on the other hand will result in increased computation cost of preference learning using BO. This may be mitigated with strategies specifically designed for high dimensional BO (for example see [45] and [60]), a topic for further work.

While a full parameter sensitivity analysis and experimentation with different kernels and acquisition functions may help illuminate the cause of BO's sub-optimal performance on the DTLZ1 problem, such investigation is beyond the scope of this work. The use of a single kernel and acquisition function, along with no manual parameter tuning, simplifies the evaluation and allows us to focus on the merit of using BO to learn preferences for multi-objective optimisation.

6 CONCLUSIONS

Eliciting preferences from a DM for a MOP while minimising the DM's cognitive burden is not a trivial task. While PWCs are an intuitive and relatively low burden method used to acquire preferences, many comparisons may be required to find preferred solutions to MOPs. The resultant increased fatigue associated with many queries is undesirable. This work shows that BO can find preferred locations in the objective space of MOPs with relatively few queries of the DM, and compared to a Non-BO technique, it has greater accuracy. The advantages of the BO approach are further enhanced by a reduced search space facilitated by integration with an MOEA, who's convergence toward preferred solutions can reduce the domain of decision variables. The BO approach integrated with a reference point-based MOEA can increase preferred, quality solutions with low DM burden. However, success remains problem dependant. Further work on ensemble or alternative BO kernels and acquisition functions has the potential to increase the robustness and accuracy of BO preference learning for MOO.

REFERENCES

[1] Majid Abdolshah, Alistair Shilton, Santu Rana, Sunil Gupta, and Svetha Venkatesh. 2019. Multi-objective Bayesian optimisation with preferences over objectives. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. D'Alch-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc., Vancouver, 12235—-12245.

[2] Raul Astudillo and Peter Frazier. 2020. Multi-attribute Bayesian optimization with interactive preference learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research)*, Silvia Chiappa and Roberto Calandra (Eds.), Vol. 108. PMLR, 4496–4507.

[3] Slim Bechikh, Marouane Kessentini, Lamjed Ben Said, and Khaled Ghédira. 2015. Preference Incorporation in Evolutionary Multiobjective Optimization: A Survey of the State-of-the-Art. In *Advances in Computers*, Ali R Hurson (Ed.). Advances in Computers, Vol. 98. Elsevier, 141–207.

[4] Valerie Belton, Jürgen Branke, Petri Eskelinen, Salvatore Greco, Julián Molina, Francisco Ruiz, and Roman Słowiński. 2008. *Interactive Multiobjective Optimization from a Learning Perspective*. Springer Berlin Heidelberg, Berlin, Heidelberg, 405–433.

[5] Alberto Bemporad and Dario Piga. 2021. Global optimization based on active preference learning with radial basis functions. *Machine Learning* 110, 2 (2021), 417–448.

[6] M. Binois, V. Picheny, P. Taillandier, and A. Habbal. 2020. The kalai-smorodinsky solution for many-objective bayesian optimization. *Journal of Machine Learning Research* 21 (2020), 1–42. arXiv:1902.06565

[7] Julian Blank and Kalyanmoy Deb. 2020. Pymoo: Multi-Objective Optimization in Python. (2020), 89497–89509 pages. arXiv:cs.NE/2002.04504

[8] Denis Bouyssou, Thierry Marchant, Marc Pirlot, Patrice Perny, Alexis Tsoukias, and Philippe Vincke. 2000. *Evaluation and Decision Models : a critical perspective*. Springer US, New York. 274 pages.

[9] Jurgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Słowinski (Eds.). 2008. *Multiobjective Optimization - Interactive and Evolutionary Approaches*. Vol. 5252 LNCS. Springer Berlin Heidelberg, Berlin Heidelberg.

[10] Eric Brochu. 2010. *Interactive Bayesian Optimization: Learning User Preferences for Graphics and Animation*. Doctor of Philosophy. University of British Columbia.

[11] Eric Brochu, Tyson Brochu, and Nandode Freitas. 2010. A Bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, Madrid, Spain, 103–112.

[12] Eric Brochu, Vlad M. Cora, and Nando de Freitas. 2010. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. (dec 2010). arXiv:1012.2599

[13] Eric Brochu, Nando De Freitas, and Abhijeet Ghosh. 2008. Active Preference Learning with Discrete Choice Data. In *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Kolle, Y. Singer, and S. T. Roweis (Eds.). Curran Associates, Inc., Vancouver, B.C., Canada, 409—-416.

[14] Olivier Chapelle and Lihong Li. 2011. An Empirical Evaluation of Thompson Sampling. In *Advances in Neural Information Processing Systems 24*, J Shawe-Taylor, R S Zemel, P L Bartlett, F Pereira, and K Q Weinberger (Eds.). Curran Associates, Inc., New York, 2249–2257.

[15] Wei Chu and Zoubin Ghahramani. 2005. Extensions of gaussian processes for ranking: semi-supervised and active learning. In *The NIPS 2005 Workshop on Learning to Rank*, Shivani Agarwal, Corinna Cortes, and Ralf Herbrich (Eds.). Whistler BC, 29.

[16] Wei Chu and Zoubin Ghahramani. 2005. Preference Learning with Gaussian Processes. In *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*. Association for Computing Machinery, New York, NY, USA, 137–144.

[17] Carlos a. Coello. 2000. An updated survey of GA-based multiobjective optimization techniques. *Comput. Surveys* 32, 2 (2000), 109–143.

[18] Carlos A Coello Coello, Gary B Lamont, and David a Van Veldhuizen. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems* (second ed.). Springer US, Boston. 800 pages.

[19] Vincent Conitzer. 2009. Eliciting Single-Peaked Preferences Using Comparison Queries. *Journal of Artificial Intelligence Research* 35, 1 (jun 2009), 161–191.

[20] Salvatore Corrente, Salvatore Greco, Miłosz Kadziński, and Roman Słowiński. 2013. Robust ordinal regression in preference learning and ranking. *Machine Learning* 93, 2-3 (2013), 381–422.

[21] D D Cox and S John. 1992. A statistical method for global optimization. In *International Conference on Systems, Man, and Cybernetics*, Vol. 2. IEEE, Chicago US, 1241–1246.

[22] Dennis D Cox and Susan John. 1997. SDO: A Statistical Method for Global Optimization. In *in Multidisciplinary Design Optimization: State-of-the-Art*. SIAM, Philadelphia, 315–329.

[23] Kalyanmoy. Deb. 2001. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, New York.

[24] Kalyanmoy Deb, Ankur Sinha, Pekka J. Korhonen, and Jyrki Wallenius. 2010. An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. *Trans. Evol. Comp* 14, 5 (oct 2010), 723–739.

[25] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. 2002. Scalable Multi-Objective Optimization Test Problems. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Vol. 1. IEEE, Honolulu, HI, USA, 825–830.

[26] David Kristjanson Duvenaud. 2014. *Automatic Model Construction with Gaussian Processes*. Doctor of Philosophy. University of Cambridge.

[27] Mark Ebden. 2015. Gaussian Processes: A Quick Introduction. *arXiv e-prints* (may 2015), arXiv:1505.02965. arXiv:math.ST/1505.02965

[28] Paul Feliot, Julien Bect, and Emmanuel Vazquez. 2019. User preferences in bayesian multi-objective optimization: The expected weighted hypervolume improvement criterion. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11331 LNCS (2019), 533–544. arXiv:1809.05450

[29] Peter I. Frazier. 2018. A Tutorial on Bayesian Optimization. (2018), 22 pages. arXiv:1807.02811

[30] Shengbo Guo and Scott Sanner. 2010. Real-time multiattribute Bayesian preference elicitation with pairwise comparison queries. *Journal of Machine Learning Research* 9 (2010), 289–296.

[31] Jussi Hakanen and Joshua D. Knowles. 2017. On Using Decision Maker Preferences with ParEGO. In *Evolutionary Multi-Criterion Optimization*, Heike Trautmann, G{\"u}nter Rudolph, Kathrin Klamroth, Oliver Sch{\"u}tze, Margaret Wiecek, Yaochu Jin, and Christian Grimme (Eds.), Vol. 10173. Springer-Verlag, Berlin, Heidelberg, 282—-297.

[32] Youwei He, Jinju Sun, Peng Song, Xuesong Wang, and Asif S. Usmani. 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 (2020), 105555.

[33] Donald R Jones, Matthias Schonlau, William J Welch, and W. J. Welch. 1998. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization* 13, 4 (1998), 455–492.

[34] 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, February 2020 (2020), 28.

[35] Kirthevasan Kandasamy, Karun Raju Vysyaraju, Willie Neiswanger, Biswajit Paria, Christopher R. Collins, Jeff Schneider, Barnabas Poczos, and Eric P. Xing. 2020. Tuning Hyperparameters without Grad Students: Scalable and Robust Bayesian Optimisation with Dragonfly. *Journal of Machine Learning Research* 21, 81 (2020), 1–27. arXiv:1903.06694

[36] Ralph L Keeney and Howard Raiffa. 1993. *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*. Cambridge University Press, Cambridge.

[37] H J Kushner. 1964. A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Basic Engineering* 86 (1964), 97–106.

[38] Ke Li, Renzhi Chen, Dragan Savic, and Xin Yao. 2019. Interactive decomposition multiobjective optimization via progressively learned value functions. *IEEE Transactions on Fuzzy Systems* 27, 5 (2019), 849–860. arXiv:1801.00609

[39] Ke Li, Kalyanmoy Deb, Erik Goodman, Carlos A Coello, Coello Kathrin, David Hutchison, and Ke Li. 2019. Progressive Preference Learning: Proof-of-Principle Results in MOEA/D. In *Evolutionary Multi-Criterion Optimization*, Kalyanmoy Deb, Erik Goodman, Carlos A Coello Coello, Kathrin Klamroth, Kaisa Miettinen, Sanaz Mostaghim, and Patrick Reed (Eds.), Vol. 11411 LNCS. Springer International Publishing, Cham, 631–643.

[40] Ruochen Liu, Ruinan Wang, Wen E N Feng, Junjun Huang, and Licheng Jiao. 2016. Interactive Reference Region Based Multi-Objective Evolutionary Algorithm Through Decomposition. *IEEE Access* 4 (2016), 7331–7346.

[41] Mariano Luque, Kaisa Miettinen, Petri Eskelinen, and Francisco Ruiz. 2009. Incorporating preference information in interactive reference point methods for multiobjective optimization. *Omega* 37, 2 (2009), 450–462.

[42] Kaisa Miettinen, Francisco Ruiz, and Andrzej P. Wierzbicki. 2008. Introduction to multiobjective optimization: Interactive approaches. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Berlin Heidelberg, Berlin Heidelberg, 27–57.

[43] Jonas Mockus. 1974. On Bayesian Methods for Seeking the Extremum. In *Proceedings of the IFIP Technical Conference*. Springer-Verlag, Berlin, Heidelberg, 400–404.

[44] Kevin P Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press, Cambridge, Massachusetts, USA.

[45] Santu Rana, Cheng Li, Sunil Gupta, Vu Nguyen, and Svetha Venkatesh. 2017. High dimensional Bayesian optimization with elastic Gaussian process. *34th International Conference on Machine Learning, ICML 2017* 6 (2017), 4407–4415.

[46] Carl E. Rasmussen and Christopher K. I. Williams. 2006. *Gaussian processes for machine learning*. MIT Press, Cambridge, Massachusetts, USA.

[47] Eric Schulz, José Miguel Hernández-lobato, and Samuel J Gershman. 2016. Quantifying mismatch in Bayesian optimization. *NIPS, BayesOpt workshop* 1, 3 (2016), 1–5.

[48] Burr Settles. 2012. *Active Learning*. Morgan & Claypool.

[49] 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 (2016), 148–175.

[50] Ankur Sinha, Kalyanmoy Deb, Pekka Korhonen, and Jyrki Wallenius. 2010. Progressively interactive evolutionary multi-objective optimization method using generalized polynomial value functions. In *IEEE Congress on Evolutionary Computation*. IEEE, 1–8.

[51] Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Kalyanmoy Deb. 2014. An interactive evolutionary multi-objective optimization algorithm with a limited number of decision maker calls. *European Journal of Operational Research* 233, 3 (mar 2014), 674–688.

[52] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. 2010. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*. Omnipress, Madison, WI, USA, 1015–1022. arXiv:arXiv:0912.3995v4

[53] Ryoji Tanabe and Hisao Ishibuchi. 2020. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing Journal* 89 (2020), 1–21.

[54] William R Thompson. 1933. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika* 25, 3/4 (1933), 285–294.

[55] L L Thurstone. 1927. A law of comparative judgment. *Psychological Review* 34, 4 (1927), 273–286.

[56] Micha K. Tomczyk and Miosz Kadzinski. 2020. On the elicitation of indirect preferences in interactive evolutionary multiple objective optimization. In *GECCO 2020 - Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. Association for Computing Machinery, New York, NY, USA, 569–577.

[57] Koert Van Ittersum, Joost M.E. Pennings, Brian Wansink, and Hans C.M. van Trijp. 2007. The validity of attribute-importance measurement: A review. *Journal of Business Research* 60, 11 (2007), 1177–1190.

[58] Yash Vesikar, Kalyanmoy Deb, and Julian Blank. 2019. Reference Point Based NSGA-III for Preferred Solutions. In *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018*. IEEE, New York, NY, USA, 1587–1594.

[59] Andrzej P. Wierzbicki. 1999. Reference Point Approaches. In *Multicriteria Decision Making. International Series in Operations Research & Management Science*, Gal T, Stewart T.J, and Hanne T (Eds.). Springer, Boston, MA, 237–275.

[60] Miao Zhang, Huiqi Li, and Steven Su. 2019. High dimensional Bayesian optimization via supervised dimension reduction. *IJCAI International Joint Conference on Artificial Intelligence* 2019-Augus (2019), 4292–4298. arXiv:1907.08953

[61] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. 2011. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation* 1, 1 (2011), 32–49.

[62] Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, E.Zitzler, Kalyanmoy Deb, and Lothar Thiele. 2000. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation* 8, 2 (jun 2000), 173–195.