Multi-objective Bayesian optimisation with preferences over objectives

Majid Abdolshah, Alistair Shilton, Santu Rana, Sunil Gupta, Svetla Venkatesh
The Applied Artificial Intelligence Institute (A²I²),
Deakin University, Australia
{majid,alistair.shilton,santu.rana,sunil.gupta,svetla.venkatesh}@deakin.edu.au

Abstract

We present a multi-objective Bayesian optimisation algorithm that allows the user to express preference-order constraints on the objectives of the type “objective A is more important than objective B”. These preferences are defined based on the stability of the obtained solutions with respect to preferred objective functions. Rather than attempting to find a representative subset of the complete Pareto front, our algorithm selects those Pareto-optimal points that satisfy these constraints. We formulate a new acquisition function based on expected improvement in dominated hypervolume (EHI) to ensure that the subset of Pareto front satisfying the constraints is thoroughly explored. The hypervolume calculation is weighted by the probability of a point satisfying the constraints from a gradient Gaussian Process model. We demonstrate our algorithm on both synthetic and real-world problems.

1 Introduction

In many real world problems, practitioners are required to sequentially evaluate a noisy black-box and expensive to evaluate function \( f \) with the goal of finding its optimum in some domain \( \mathbb{X} \). Bayesian optimisation is a well-known algorithm for such problems. There are a variety of studies such as hyperparameter tuning [27, 13, 12], expensive multi-objective optimisation for Robotics [2, 1], and experimentation optimisation in product design such as short polymer fiber materials [16].

Multi-objective Bayesian optimisation involves at least two conflicting, black-box, and expensive to evaluate objectives to be optimised simultaneously. Multi-objective optimisation usually assumes that all objectives are equally important, and solutions are found by seeking the Pareto front in the objective space [4, 5, 3]. However, in most cases, users can stipulate preferences over objectives. This information will impart on the relative importance on sections of the Pareto front. Thus using this information to preferentially sample the Pareto front will boost the efficiency of the optimiser, which is particularly advantageous when the objective functions are expensive.

In this study, preferences over objectives are stipulated based on the stability of the solutions with respect to a set of objective functions. As an example, there are scenarios when investment strategists are looking for Pareto optimal investment strategies that prefer stable solutions for return (objective 1) but more diverse solutions with respect to risk (objective 2) as they can later decide their appetite for risk. As can be inferred, the stability in one objective produces more diverse solutions for the other objectives. We believe in many real-world problems our proposed method can be useful in order to reduce the cost, and improve the safety of experimental design.

Whilst multi-objective Bayesian optimisation for sample efficient discovery of Pareto front is an established research track [9, 18, 8, 5], limited work has examined the incorporation of preferences. Recently, there has been a study [18] wherein given a user specified preferred region in objective space,
we need to make the weights corresponding to the gradient of \( f \) with respect to \( x \) (in terms of stability) than \( f_0 \).

To formalise the concept of preference-order constraints, we first note that a point is locally Pareto optimal from those that do not.

Bayesian optimiser toward Pareto optimal points that satisfy the preference-order constraint and away these posterior probabilities may be incorporated into the EHI acquisition function to steer the optimiser focuses its sampling to derive the Pareto front efficiently. However, such preferences are based on the assumption of having an accurate prior knowledge about objective space and the preferred region (generally a hyperbox) for Pareto front solutions. The main contribution of this study is formulating the concept of preference-order constraints and incorporating that into a multi-objective Bayesian optimisation framework to address the unavailability of prior knowledge and boosting the performance of optimisation in such scenarios.

We are formulating the preference-order constraints through ordering of derivatives and incorporating that into multi-objective optimisation using the geometry of the constraints space whilst needing no prior information about the functions. Formally, we find a representative set of Pareto-optimal solutions to the following multi-objective optimisation problem:

\[
\mathbb{D}^* \subset \mathbb{X}^* = \arg\max_{x \in \mathbb{X}} \mathbf{f}(x)
\]

subject to preference-order constraints - that is, assuming \( \mathbf{f} = [f_0, f_1, \ldots, f_m] \), \( f_0 \) is more important (in terms of stability) than \( f_1 \) and so on. Our algorithm aims to maximise the dominated hypervolume of the solution in a way that the solutions that meet the constraints are given more weights.

To formalise the concept of preference-order constraints, we first note that a point is locally Pareto optimal if any sufficiently small perturbation of a single design parameter of that point does not simultaneously increase (or decrease) all objectives. Thus, equivalently, a point is locally Pareto optimal if we can define a set of weight vectors such that, for each design parameter, the weighted sum of gradients of the objectives with respect to that design parameter is zero (see Figure 1a). Therefore, the weight vectors define the relative importance of each objective at that point. Figure 1b illustrates this concept where the blue box defines the region of stability for the function \( f_0 \). Since in this section the magnitude of partial derivative for \( f_0 \) is smaller compared to that of \( f_1 \), the weights required to satisfy Pareto optimality would need higher weight corresponding to the gradient of \( f_0 \) compared to that of \( f_1 \) (see Figure 1b). Conversely, in Figure 1c, the red box highlights the section of the Pareto front where solutions have high stability in \( f_1 \). To obtain samples from this section of the Pareto front, we need to make the weights corresponding to the gradient of \( f_0 \) to be smaller to that of the \( f_1 \).

Our solution is based on understanding the geometry of the constraints in the weight space. We show that preference order constraints gives rise to a polyhedral proper cone in this space. We show that for the pareto-optimality condition, it necessitates the gradients of the objectives at pareto-optimal points to lie in a perpendicular cone to that polyhedral. We then quantify the posterior probability that any point satisfies the preference-order constraints given a set of observations. We show how these posterior probabilities may be incorporated into the EHI acquisition function to steer the Bayesian optimiser toward Pareto optimal points that satisfy the preference-order constraint and away from those that do not.

Figure 1: (a) Local Pareto optimality for 2 objective example with 1D design space. Local optimality implies \( \partial f_0(x) \) and \( \partial f_1(x) \) have opposite signs since the weighted sum of gradients of the objectives with respect to \( x \) must be zero: \( s^T \partial f(x) = 0 \). In (b) we additionally require that \( || \partial f_1(x) || > || \partial f_0(x) || \), so perturbation of \( x \) will cause relatively more change in \( f_1 \) than \( f_0 \). Such solutions are (relatively) stable in objective \( f_0 \). (c) Shows the converse, namely \( || \partial f_0(x) || > || \partial f_1(x) || \) favoring solutions that are (relatively) stable in objective \( f_1 \) and diverse in \( f_0 \).
2 Notation

Sets are written $A, B, C, \ldots$ where $\mathbb{R}_+$ is the positive reals, $\mathbb{R}_+ = \mathbb{R}_+ \cup \{0\}$, $\mathbb{Z}_+ = \{1, 2, \ldots\}$, and $\mathbb{Z}_n = \{0, 1, \ldots, n - 1\}$. $|A|$ is the cardinality of the set $A$. Tuples (ordered sets) are denoted $\mathfrak{t}, \mathfrak{b}, \mathfrak{c}, \ldots$. Column vectors are bold lower case a, b, c, \ldots. Matrices bold upper case A, B, C, \ldots. Element $i$ of vector a is $a_i$, and element $i,j$ of matrix A is $A_{i,j}$ (all indexed $i,j = 0, 1, \ldots$). The transpose is denoted $a^T$, $A^T$. I is the identity matrix, I is a vector of 1s, 0 is a vector of 0s, and $e_i$ is a vector $e^{(i)} = \delta_{ij}$, where $\delta_{ij}$ is the Kronecker-Delta.

\[ \nabla x = \left[ \frac{\partial}{\partial x_0}, \frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_{n-1}} \right]^T, \text{ sgn}(x) \text{ is the sign of } x \text{ (where sgn}(0) = 0), \text{ and the indicator function is denoted as } I(a). \]

3 Background

3.1 Gaussian Processes

Let $X \subset \mathbb{R}^n$ be compact. A Gaussian process $\mathcal{GP}(\mu, K)$ is a distribution on the function space $f : X \rightarrow \mathbb{R}$ defined by mean $\mu : X \rightarrow \mathbb{R}$ (assumed zero without loss of generality) and kernel (covariance) $K : X \times X \rightarrow \mathbb{R}$. If $f(x) \sim \mathcal{GP}(0, K(x,x'))$ then the posterior of $f$ given $D = \{(x_{(j)}, y_{(j)}) \in \mathbb{R}^n \times \mathbb{R} | y_{(j)} = f(x_{(j)}) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2), j \in \mathbb{Z}_N\}, f(x)|D \sim \mathcal{N}(\mu_D(x), \sigma_D(x,x'))$, where:

\[
\begin{align*}
\mu_D(x) &= k^T(x) \left( K + \sigma^2 I \right)^{-1} y \\
\sigma_D(x,x') &= K(x,x') - k^T(x) \left( K + \sigma^2 I \right)^{-1} k(x') \quad (2)
\end{align*}
\]

and $y, k(x) \in \mathbb{R}^|D|$, $K \in \mathbb{R}^{|D| \times |D|}$, $k(x) = K(x, x_{(j)})$, $k_{jk} = K(x_{(j)}, x_{(k)})$.

Since differentiation is a linear operation, the derivative of a Gaussian process is also a Gaussian process [17][22]. The posterior of $\nabla_x f$ given $D$ is $\nabla_x f(x)|D \sim \mathcal{N}(\mu_D'(x), \sigma_D'(x,x'))$, where:

\[
\begin{align*}
\mu_D'(x) &= (\nabla_x k^T(x)) \left( K + \sigma^2 I \right)^{-1} y \\
\sigma_D'(x,x') &= \nabla_x \nabla_x^T K(x,x') - (\nabla_x k^T(x)) \left( K + \sigma^2 I \right)^{-1} \left( \nabla_x k^T(x') \right)^T \quad (3)
\end{align*}
\]

3.2 Multi-Objective Optimisation

A multi-objective optimisation problem has the form:

\[
\text{argmax}_{x \in \mathbb{X}} f(x)
\]

where the components of $f : \mathbb{X} \subset \mathbb{R}^n \rightarrow \mathbb{Y} \subset \mathbb{R}^m$ represent the $m$ distinct objectives $f_i : \mathbb{X} \rightarrow \mathbb{R}$. $\mathbb{X}$ and $\mathbb{Y}$ are called design space and objective space, respectively. A Pareto-optimal solution is a point $x^* \in \mathbb{X}$ for which it is not possible to find another solution $x \in \mathbb{X}$ such that $f_i(x) \geq f_i(x^*)$ for all objectives $f_0, f_1, \ldots, f_{m-1}$. The set of all Pareto optimal solutions is the Pareto set $\mathbb{X}^* = \{ x^* \in \mathbb{X} | \nexists x \in \mathbb{X} : f(x) \succ f(x^*) \}$ where $y \succ y'$ (y dominates $y'$) means $y \neq y'$, $y_i \geq y'_i \quad \forall i$, and $y \succeq y'$ means $y \succ y'$ or $y = y'$.

Given observations $D = \{(x_{(j)}, y_{(j)}) \in \mathbb{R}^n \times \mathbb{R}^m | y_{(j)} = f(x_{(j)}) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2)\}$ of $f$ the dominant subset $D^* = \{(x^*, y^*) \in D | \nexists (x, y) \in D : y \succeq y^*\}$ is the most optimal subset of $D$ (in the Pareto sense). The “goodness” of $D$ is often measured by the dominated hypervolume ($S$-metric, [31][10]) with respect to some reference point $z \in \mathbb{R}^m$:

\[
S(D) = \int_{y \succeq z} I(\{ y_{(j)} \in D | y_{(j)} \geq y \}) dy.
\]

Thus our aim is to find the set $D$ that maximises the hypervolume. Optimised algorithms exist for calculating hypervolume [29][23], $S(D)$, which is typically calculated by sorting the dominant observations along each axis in objective space to form a grid. Dominated hypervolume (with respect to $z$) is then the sum of the hypervolumes of the dominated cells $(c_k)$ - i.e. $S(D) = \sum_k \text{vol}(c_k)$.

3.3 Bayesian Multi-Objective Optimisation

In the multi-objective case one typically assumes that the components of $f$ are draws from independent Gaussian processes, i.e. $f_i(x) \sim \mathcal{GP}(0, K_{(i)}(x, x'))$, and $f_i$ and $f_i'$ are independent $\forall i \neq i'$. A
It is necessary to briefly consider the geometry of $S$ toward Pareto optimal points that satisfy the preference-order constraint. Before proceeding, however, probabilities may be incorporated into the EHI acquisition function to steer the Bayesian optimiser.

Constraints we then calculate the posterior probability that a recommended solution with the preference-order constraints. Having defined preference-order linear scalarisation optimisation conditions fail for Pareto points on non-convex regions [28].

It is important to note that this is not the same as the optimality conditions that may be derived from linear scalarisation, as the optimality conditions that arise from linear scalarisation additionally require that $s(0) = s(1) = \ldots = s(n-1)$. Moreover [7] applies to all Pareto-optimal points, whereas linear scalarisation optimisation conditions fail for Pareto points on non-convex regions [28].

Let $x^* \in \text{int}(X) \cap X^*$ be a Pareto-optimal point in the interior of $X$. Necessary (but not sufficient, local) Pareto optimality conditions require that, for all sufficiently small $\delta x \in \mathbb{R}^n$, $f(x^* + \delta x) \neq f(x)$, or, equivalently $(\delta x^T \nabla_x) f(x^*) \not\in \mathbb{R}^m$.

A necessary (again not sufficient) equivalent condition is that, for each axis $j \in Z_n$ in design space, sufficiently small changes in $x_j$ do not cause all objectives to simultaneously increase (and/or remain unchanged) or decrease (and/or remain unchanged). Failure of this condition would indicate that simply changing design parameter $x_j$ could improve all objectives, and hence that $x^*$ was not in fact Pareto optimal. In summary, local Pareto optimality requires that $\forall j \in Z_n$ there exists $s(j) \in \mathbb{R}^m \setminus \{0\}$ such that:

$$s(j) \frac{\partial}{\partial x_j} f(x) = 0 \quad (7)$$

It is important to note that this is not the same as the optimality conditions that may be derived from linear scalarisation, as the optimality conditions that arise from linear scalarisation additionally require that $s(0) = s(1) = \ldots = s(n-1)$. Moreover [7] applies to all Pareto-optimal points, whereas linear scalarisation optimisation conditions fail for Pareto points on non-convex regions [28].

**Definition 1 (Preference-Order Constraints)** Let $J = (i_0, i_1, \ldots, i_Q) | \{i_0, i_1, \ldots\} \subset Z_m, i_k \neq i_k \forall k \neq k'$ be an (ordered) preference tuple. A vector $x \in X$ satisfies the associated preference-order constraint if $\exists s(0), s(1), \ldots, s(n-1) \in S_3$ such that:

$$s(j) \frac{\partial}{\partial x_j} f(x) = 0 \forall j \in Z_n$$

where $S_3 \triangleq \{s \in \mathbb{R}^m \setminus \{0\} | s_0 \geq s_i \geq s_{i_2} \geq \ldots\}$. Further we define $X_3$ to be the set of all $x \in X$ satisfying the preference-order constraint. Equivalently:

$$X_3 = \{x \in X | \frac{\partial}{\partial x_j} f(x) \in S^+_3 \forall j \in Z_n\}$$

where $S^+_3 \triangleq \{x \in X | \exists s \in S_3, s^T x = 0\}$.

It is noteworthy to mention that [7] and Definition [1] are the key for calculating the compliance of a recommended solution with the preference-order constraints. Having defined preference-order constraints we then calculate the posterior probability that $x \in X_3$, and showing how these posterior probabilities may be incorporated into the EHI acquisition function to steer the Bayesian optimiser toward Pareto optimal points that satisfy the preference-order constraint. Before proceeding, however, it is necessary to briefly consider the geometry of $S_3$ and $S^+_3$.

4 Problem Formulation

Let $f : X \subset \mathbb{R}^n \rightarrow \mathbb{Y} \subset \mathbb{R}^m$ be a vector of $m$ independent draws $f_i \sim GP(0, K_{ij}(x, x))$ from zero-mean Gaussian processes. Assume that $f$ is expensive to evaluate. Our aim is to find a representative set of Pareto-optimal solutions to the following multi-objective optimisation problem:

$$\mathbb{D}^* \subset X^* = \underset{x \in X \subset X}{\text{argmax}} \ f(x) \quad (6)$$

subject to preference-order constraints. Specifically, we want to explore only that subset of solutions $X_3 \subset X$ that place more importance on one objective $f_{i_0}$ than objective $f_{i_1}$, and so on, as specified by the (ordered) preference tuple $J = (i_0, i_1, \ldots, i_Q) | \{i_0, i_1, \ldots\} \subset Z_m, i_k \neq i_k \forall k \neq k'$, where $Q \in Z_m$ is the number of defined preferences over objectives.

### 4.1 Preference-Order Constraints

Let $x^* \in \text{int}(X) \cap X^*$ be a Pareto-optimal point in the interior of $X$. Necessary (but not sufficient, local) Pareto optimality conditions require that, for all sufficiently small $\delta x \in \mathbb{R}^n$, $f(x^* + \delta x) \neq f(x)$, or, equivalently $(\delta x^T \nabla_x) f(x^*) \not\in \mathbb{R}^m$.

A necessary (again not sufficient) equivalent condition is that, for each axis $j \in Z_n$ in design space, sufficiently small changes in $x_j$ do not cause all objectives to simultaneously increase (and/or remain unchanged) or decrease (and/or remain unchanged). Failure of this condition would indicate that simply changing design parameter $x_j$ could improve all objectives, and hence that $x^*$ was not in fact Pareto optimal. In summary, local Pareto optimality requires that $\forall j \in Z_n$ there exists $s(j) \in \mathbb{R}^m \setminus \{0\}$ such that:

$$s(j) \frac{\partial}{\partial x_j} f(x) = 0 \quad (7)$$

It is important to note that this is not the same as the optimality conditions that may be derived from linear scalarisation, as the optimality conditions that arise from linear scalarisation additionally require that $s(0) = s(1) = \ldots = s(n-1)$. Moreover [7] applies to all Pareto-optimal points, whereas linear scalarisation optimisation conditions fail for Pareto points on non-convex regions [28].

**Definition 1 (Preference-Order Constraints)** Let $J = (i_0, i_1, \ldots, i_Q) | \{i_0, i_1, \ldots\} \subset Z_m, i_k \neq i_k \forall k \neq k'$ be an (ordered) preference tuple. A vector $x \in X$ satisfies the associated preference-order constraint if $\exists s(0), s(1), \ldots, s(n-1) \in S_3$ such that:

$$s(j) \frac{\partial}{\partial x_j} f(x) = 0 \forall j \in Z_n$$

where $S_3 \triangleq \{s \in \mathbb{R}^m \setminus \{0\} | s_0 \geq s_i \geq s_{i_2} \geq \ldots\}$. Further we define $X_3$ to be the set of all $x \in X$ satisfying the preference-order constraint. Equivalently:

$$X_3 = \{x \in X | \frac{\partial}{\partial x_j} f(x) \in S^+_3 \forall j \in Z_n\}$$

where $S^+_3 \triangleq \{x \in X | \exists s \in S_3, s^T x = 0\}$.

It is noteworthy to mention that [7] and Definition [1] are the key for calculating the compliance of a recommended solution with the preference-order constraints. Having defined preference-order constraints we then calculate the posterior probability that $x \in X_3$, and showing how these posterior probabilities may be incorporated into the EHI acquisition function to steer the Bayesian optimiser toward Pareto optimal points that satisfy the preference-order constraint. Before proceeding, however, it is necessary to briefly consider the geometry of $S_3$ and $S^+_3$.
We now define the preference-order constraints by assumption
\[ Q > ∈ v \]
where \( Q \) is a proper cone representing the preference-order constraints; \( S_\perp \) is the union of two sub-spaces. \( v \in S_\perp \) implies a solution complying with preference-order constraints. \( b_0 \) and \( b_1 \) are the projection of \( v \) over \( \tilde{a}(0) \) and \( \tilde{a}(1) \). In order to satisfy \( v \in S_\perp \), it is necessary that \( v_\in S_\perp \) s.t. \( v^T s = 0 \) or equivalently \( v = 0 \) or \( b_0 = \tilde{a}(0)^T v \) and \( b_1 = \tilde{a}(1)^T v \) have different signs.

4.2 The geometry of \( S_\perp \) and \( S_\perp^1 \)

In the following we assume, w.l.o.g., that the preference-order constraints follows the order of indices in objective functions (reorder, otherwise), and that there is at least one constraint.

We now define the preference-order constraints by assumption \( J = (0, 1, \ldots, Q | Q \in Z_m \setminus \{0\} ) \), where \( Q > 0 \). This defines the sets \( S_\perp \) and \( S_\perp^1 \), which in turn define the constraints that must be met by the gradients of \( f(x) \) - either \( s_i(0), s_i(1), \ldots, s_i(n-1) \in S_\perp \) such that \( s_i^T \frac{\partial f(x)}{\partial x_j} = 0 \) \( \forall j \in Z_n \) or, equivalently \( \frac{\partial f(x)}{\partial x_j} \in S_\perp \) \( \forall j \in Z_n \). Next, Theorem 1 defines the representation of \( S_\perp \).

**Theorem 1** Let \( J = (0, 1, \ldots, Q | Q \in Z_m \setminus \{0\} ) \) be an (ordered) preference tuple. Define \( S_\perp \) as per definition [4]. Then \( S_\perp \) is a polyhedral (finitely-generated) proper cone (excluding the origin) that may be represented using either a polyhedral representation:

\[
S_\perp = \left\{ s \in \mathbb{R}^m \mid a_i^T s \geq 0 \forall i \in Z_m \right\} \setminus \{0\} \tag{8}
\]

or a generative representation:

\[
S_\perp = \left\{ \sum_{i \in Z_m} c_i \tilde{a}(i) \mid c \in \mathbb{R}_+^m \right\} \setminus \{0\} \tag{9}
\]

where \( \forall i \in Z_m \):

\[
a(i) = \begin{cases} 
\frac{1}{\sqrt{Q}} (e_i - e_{i+1}) & \text{if } i \in Z_Q \\
e_i & \text{otherwise}
\end{cases}
\]

\[
\tilde{a}(i) = \begin{cases} 
\frac{1}{\sqrt{Q+1}} \sum_{l \in Z_{i+1}} e_l & \text{if } i \in Z_{Q+1} \\
e_i & \text{otherwise}
\end{cases}
\]

and \( e_0, e_1, \ldots, e_{m-1} \) are the Euclidean basis of \( \mathbb{R}^m \).

Proof of Theorem 1 is available in the supplementary material. To test if a point satisfies this requirement we need to understand the geometry of the set \( S_\perp \). The Theorem 1 shows that \( S_\perp \cup \{0\} \) is a polyhedral (finitely generated) proper cone, represented in terms of half-space constraints (polyhedral form) or as a positive span of extreme directions (generative representation). The geometrical intuition for this is given in Figure 2 for a simple, 2-objective case with a single preference order constraint.
Algorithm 1 Test if \( v \in S_+^2 \).

**Input:** Preference tuple \( J \)
Test vector \( v \in \mathbb{R}^m \).
**Output:** \( \mathbb{I}(v \in S_+^2) \).

// Calculate \( 1(v \in S_+^2) \).
Let \( b_j = \tilde{a}^T_j v \forall j \in \mathbb{Z}_m \).
if \( \exists i \neq k \in \mathbb{Z}_m : \text{sgn}(b_i) \neq \text{sgn}(b_k) \)
return TRUE
else if \( b = 0 \) return TRUE
else return FALSE.

Algorithm 2 Preference-Order Constrained Bayesian Optimisation (MOBO-PC).

**Input:** preference-order tuple \( J \).
Observations \( D = \{(x_{ij}, y_{ij}) \in \mathbb{X} \times \mathbb{Y} \} \).
for \( t = 0, 1, \ldots, T - 1 \) do
Select the test point:
\( x = \arg\max_{x \in \mathbb{X}} a_{t}^{\text{PEHI}}(x|D_t) \).
\( (a_{t}^{\text{PEHI}} \text{ is evaluated using algorithm 4}) \).
Perform Experiment \( y = f(x) + \epsilon \).
Update \( D_{t+1} := D_t \cup \{(x, y)\} \).
end for

Algorithm 3 Calculate \( \Pr(x \in X_J|D) \).

**Input:** Observations \( D = \{(x_{ij}, y_{ij}) \in \mathbb{X} \times \mathbb{Y} \} \).
Number of Monte Carlo samples \( R \).
Test vector \( x \in \mathbb{X} \).
**Output:** \( \Pr(x \in X_J|D) \).
Let \( q = 0 \).
for \( k = 0, 1, \ldots, R - 1 \) do
// Construct Samples
Sample \( u \sim \mathcal{N}(\mu_{x_i}(x), \sigma_{x_i}(x, x)) \) (see (3)).
Let \( v_{(i)} := u^T \).
end for
// Test if \( v_{(j)} \in S_+^2 \forall j \in \mathbb{Z}_m \).
Let \( q = q + \prod_{j \in \mathbb{Z}_m} \mathbb{I}(v_{(j)} \in S_+^2) \) (see algorithm 1).
Return \( q/R \).

Algorithm 4 Calculate \( a_{t}^{\text{PEHI}}(x|D) \).

**Input:** Observations \( D = \{(x_{ij}, y_{ij}) \in \mathbb{X} \times \mathbb{Y} \} \).
Number of Monte Carlo samples \( \tilde{R} \).
Test vector \( x \in \mathbb{X} \).
**Output:** \( a_{t}^{\text{PEHI}}(x|D) \).
Using algorithm 3 calculate:
\( s_x = \Pr(x \in X_J|D) \).
\( s_{(j)} = \Pr(x_{(j)} \in X_J|D) \forall (x_{(j)}, y_{(j)}) \in D \).
Let \( q = 0 \).
for \( k = 0, 1, \ldots, \tilde{R} - 1 \) do
Sample \( y_i \sim \mathcal{N}(\mu_{x_i}(x), \sigma_{x_i}(x)) \forall i \in \mathbb{Z}_m \) (see (2)).
Construct cells \( c_0, c_1, \ldots \) from \( \mathbb{X} \cup \{x, y\} \) by sorting along each axis in objective space to form a grid.
Calculate:
\( q = q + \sum_{k \geq c_k} \prod_{j \in \mathbb{Z}_m} \mathbb{I}(y_{(j)} \geq c_{k}) \).
end for
Return \( q/\tilde{R} \).

The subsequent corollary allows us to construct a simple algorithm (algorithm 1) to test if a vector \( v \) lies in the set \( S_+^2 \). We will use this algorithm to test if \( \delta_{x_i} f(x) \in S_+^2 \forall j \in \mathbb{Z}_m \) - that is, if \( x \) satisfies the preference-order constraints. The proof of corollary 1 is available in the supplementary material.

**Corollary 1** Let \( J = (0, 1, \ldots, Q|Q \in \mathbb{Z}_m\setminus\{0\}) \) be an (ordered) preference tuple. Define \( S_+^2 \) as per definition 1. Using the notation of Theorem 1, \( v \in S_+^2 \) if and only if \( v = 0 \) or \( \exists i \neq k \in \mathbb{Z}_m \) such that \( \text{sgn}(a_i^T v) \neq \text{sgn}(a_k^T v) \), where \( \text{sgn}(0) = 0 \).

### 5 Preference Constrained Bayesian Optimisation

In this section we do two things. First, we show how the Gaussian process models of the objectives \( f_i \) (and their derivatives) may be used to calculate the posterior probability that \( x \in X_J \) defined by \( J = (0, 1, \ldots, Q|Q \in \mathbb{Z}_m\setminus\{0\}) \). Second, we show how the EHI acquisition function may be modified and calculated to incorporate these probabilities and hence only reward points that satisfy the preference-order conditions. Finally, we give our algorithm using this acquisition function.

#### 5.1 Calculating Posterior Probabilities

Given that \( f_i \sim \mathcal{GP}(0, K_i(x, x)) \) are draws from independent Gaussian processes, and given observations \( D \), we wish to calculate the posterior probability that \( x \in X_J \).
i.e.: \( \Pr (x \in \mathcal{X}_3 | \mathbb{D}) = \Pr \left( \frac{\partial}{\partial x_j} f(x) \in \mathbb{S}^I \forall j \in \mathbb{Z}_n \right) \). As \( f_i \sim \mathcal{GP}(0, K_{i}(x, x)) \) it follows that \( \nabla x f_i(x) | \mathbb{D} \sim \mathcal{N}(\mu_{Di}(x), \sigma_{Di}(x, x')) \), as defined by [3]. Hence:

\[
\Pr (x \in \mathcal{X}_3 | \mathbb{D}) = \Pr \left( \begin{bmatrix} v(0)i \\ v(1)i \\ \vdots \\ v(n-1)i \end{bmatrix} \right) \sim \mathcal{N}(i) \]

where \( v \sim P(\nabla_x f | \mathbb{D}) \). We estimate it using Monte-Carlo [6] sampling as per algorithm 3.

### 5.2 Preference-Order Constrained Bayesian Optimisation Algorithm (MOBO-PC)

Our complete Bayesian optimisation algorithm with Preference-order constraints is given in algorithm 2. The acquisition function introduced in this algorithm gives higher importance to points satisfying the preference-order constraints. Unlike standard EHI, we take expectation over both the expected experimental outcomes \( f_i(x) \sim \mathcal{N}(\mu_{Di}(x), \sigma_{Di}(x, x)) \), \( \forall i \in \mathbb{Z}_m \), and the probability that points \( x_{(i)} \in \mathcal{X}_3 \) and \( x \in \mathcal{X}_3 \) satisfy the preference-order constraints. We define our preference-based EHI acquisition function as:

\[
a_t^{PEHI}(x|\mathbb{D}) = \mathbb{E}[S_3(\mathbb{D} \cup \{(x, f(x))\}) - S_3(\mathbb{D})] \quad (10)
\]

where \( S_3(\mathbb{D}) \) is the hypervolume dominated by the observations \((x, y) \in \mathbb{D}\) satisfying the preference-order constraints. The calculation of \( S_3(\mathbb{D}) \) is illustrated in the supplementary material. The expectation of \( S_3(\mathbb{D}) \) given \( \mathbb{D} \) is:

\[
\mathbb{E}[S_3(\mathbb{D})|\mathbb{D}] = \sum_k \text{vol}(c_k) \Pr(\exists (x, y) \in \mathbb{D} | y \succeq \tilde{y}_{c_k} \wedge \ldots x \in \mathcal{X}_3) \ldots
\]

where \( \tilde{y}_{c_k} \) is the dominant corner of cell \( c_k \), \( \text{vol}(c_k) \) is the hypervolume of cell \( c_k \), and the cells \( c_k \) are constructed by sorting \( \mathbb{D} \) along each axis in objective space. The posterior probabilities \( \Pr(x \in \mathcal{X}_3 | \mathbb{D}) \) are calculated using algorithm 3. It follows that:

\[
a_t^{PEHI}(x|\mathbb{D}) = \Pr (x \in \mathcal{X}_3 | \mathbb{D}) \mathbb{E}\left[ \sum_k \text{vol}(c_k) \prod_{j \in \mathbb{Z}_N : y_{(i)} \succeq \tilde{y}_{c_k}} (1 - \Pr(x_{(i)} \in \mathcal{X}_3 | \mathbb{D})) \right] y_i \sim \ldots \mathcal{N}(\mu_{Di}(x), \sigma_{Di}(x)) \forall i \in \mathbb{Z}_m
\]

where the cells \( c_k \) are constructed using the set \( \mathbb{D} \cup \{(x, y)\} \) by sorting along the axis in objective space. We estimate this acquisition function using Monte-Carlo simulation shown in algorithm 4.

### 6 Experiments

We conduct a series of experiments to test the empirical performance of our proposed method MOBO-PC and compare with other strategies. These experiments including synthetic data as well as optimizing the hyper-parameters of a feed-forward neural network. For Gaussian process, we use maximum likelihood estimation for setting hyperparameters [21].

#### 6.1 Baselines

To the best of our knowledge there are no studies aiming to solve our proposed problem, however we are using PESMO, SMSego, SUR, ParEGO and EHI [9,20,19,14,7] to confirm the validity of the obtained Pareto front solutions. The obtained Pareto front must be in the ground-truth whilst also satisfying the preference-order constraints. We compare our results with MOBO-RS [13] by suitably specifying bounding boxes in the objective space that can replicate a preference-order constraint.

#### 6.2 Synthetic Functions

We begin with a comparison on minimising synthetic function Schaffer function N. 1 with 2 conflicting objectives \( f_0, f_1 \) and 1-dimensional input. (see [24]). Figure 3a shows the ground-truth Pareto front.
Figure 3: Finding Pareto front which comply with the preference-order constraint. Figure 3a shows the full Pareto front solution (with no preferences). Figure 3b illustrates the Pareto front by assuming stability of first objective $f_0$ is similar to second objective $f_1$. In Figure 3c, stability of $f_1$ is preferred over $f_0$. Figure 3d shows more stable results for $f_0$ than $f_1$ ($s_0 > s_1$). Figure 3e and 3f shows the results obtained by MOBO-RS and the corresponding bounding boxes. The gradient color of the Pareto front points in Figure 3b, 3d indicates their degree of compliance with the constraints.

Figure 4 illustrates a special test case in which $s_0 > s_1$ and $s_2 > s_1$, yet no preferences specified over $f_2$ and $f_0$ while minimising Viennet function. The proposed complex preference-order constraint does not form a proper cone as elaborated in Theorem 1. However, $s_0 > s_1$ independently constructs a proper cone, likewise for $s_2 > s_1$. Figure 4a shows the results of processing these two independent constraints separately, merging their results and finding the Pareto front. Figure 4b implies more stable solutions for $f_0$ comparing to $f_1$. Figure 4c shows the Pareto front points comply with $s_2 > s_1$. For this function. To illustrate the behavior of our method, we impose distinct preferences. Three test cases are designed to illustrate the effects of imposing preference-order constraints on the objective functions for stability. Case (1): $s_0 \approx s_1$, Case (2): $s_0 < s_1$ and Case (3): $s_0 > s_1$. For our method it is only required to define the preference-order constraints, however for MOBO-RS, additional information as a bounding box is obligatory. Figure 3a (case 1), shows the results of preference-order constraints $\mathcal{S}_0 \triangleq \{ s \in \mathbb{R}^m \setminus \{0\} \mid s_0 \approx s_1 \}$ for our proposed method, where $s_0$ represents the importance of stability in minimising $f_0$ and $s_1$ is the importance of stability in minimising $f_1$. Due to same importance of both objectives, a balanced optimisation is expected. Higher weights are obtained for the Pareto front points in the middle region with highest stability for both objectives. Figure 3c (case 2) is based on the preference-order of $s_0 < s_1$ that implies the importance of stability in $f_1$ is more than $f_0$. The results show more stable Pareto points for $f_1$ than $f_0$. Figure 3d (case 3) shows the results of $s_0 > s_1$ preference-order constraint. As expected, we see more number of stable Pareto points for the important objective (i.e. $f_0$ in this case). We defined two bounding boxes for MOBO-RS approach which can represent the preference-order constraints in our approach (Figure 3e and 3f). There are infinite possible bounding boxes can serve as constraints on objectives in such problems, consequently, the instability of results is expected across the various definitions of bounding boxes. We believe our method can obtain more stable Pareto front solutions especially when prior information is sparse. Also, having extra information as the weight (importance) of the Pareto front points is another advantage.
6.3 Finding a Fast and Accurate Neural Network

Next, we train a neural network with two objectives of minimising both prediction error and prediction time, as per [9]. These are conflicting objectives because reducing the prediction error generally involves larger networks and consequently longer testing time. We are using MNIST dataset and the tuning parameters include number of hidden layers \((x_1 \in [1, 3])\), the number of hidden units per layer \((x_2 \in [50, 300])\), the learning rate \((x_3 \in (0, 0.2))\), amount of dropout \((x_3 \in [0.4, 0.8])\), and the level of \(l_1\) \((x_5 \in (0, 0.1))\) and \(l_2\) \((x_5 \in (0, 0.1))\) regularization. For this problem we assume stability of \(f_1(\text{time})\) in minimising procedure is more important than the \(f_0(\text{error})\). For MOBO-RS method, we selected \([0.02, 0], [0.03, 2]\) bounding box to represent an accurate prior knowledge (see Figure 5). The results were averaged over 5 independent runs. Figure 5 illustrates that one can simply ask for more stable solutions with respect to test time (without any prior knowledge) of a neural network while optimising the hyperparameters. As all the solutions found with MOBO-PC are in range of \((0, 5)\) test time. In addition, it seems the proposed method finds more number of Pareto front solutions in comparison with MOBO-RS.

7 Conclusion

In this paper we proposed a novel multi-objective Bayesian optimisation algorithm with preferences over objectives. We define objective preferences in terms of stability and formulate a common framework to focus on the sections of the Pareto front where preferred objectives are more stable, as is required. We evaluate our method on both synthetic and real-world problems and show that the obtained Pareto fronts comply with the preference-order constraints.

Acknowledgments

This research was partially funded by Australian Government through the Australian Research Council (ARC). Prof Venkatesh is the recipient of an ARC Australian Laureate Fellowship (FL170100006).
References


