A Flexible Framework for Multi-Objective Bayesian Optimization using Random Scalarizations

Biswajit Paria
MLD, Carnegie Mellon University
bparia@cs.cmu.edu

Kirthevasan Kandasamy*
EECS, UC Berkeley
kandasamy@eecs.berkeley.edu

Barnabás Póczos
MLD, Carnegie Mellon University
bapoczos@cs.cmu.edu

Abstract

Many real world applications can be framed as multi-objective optimization problems, where we wish to simultaneously optimize for multiple criteria. Bayesian optimization techniques for the multi-objective setting are pertinent when the evaluation of the functions in question are expensive. Traditional methods for multi-objective optimization, both Bayesian and otherwise, are aimed at recovering the Pareto front of these objectives. However, in certain cases a practitioner might desire to identify Pareto optimal points only in a subset of the Pareto front due to external considerations. In this work, we propose a strategy based on random scalarizations of the objectives that addresses this problem. Our approach is able to flexibly sample from desired regions of the Pareto front and, computationally, is considerably cheaper than most approaches for MOO. We also study a notion of regret in the multi-objective setting and show that our strategy achieves sublinear regret. We experiment with both synthetic and real-life problems, and demonstrate superior performance of our proposed algorithm in terms of the flexibility and regret.

1 INTRODUCTION

Bayesian optimization (BO) is a popular recipe for optimizing expensive black-box functions where the goal is to find a global maximizer of the function. Bayesian optimization has been used for a variety of practical optimization tasks such as hyperparameter tuning for machine learning algorithms, experiment design, online advertising, and scientific discovery (Snoek et al., 2012; Hernández-Lobato et al., 2017; Martinez-Cantin et al., 2007; González et al., 2015; Kandasamy et al., 2017).

In many practical applications however, we are required to optimize multiple objectives, and moreover, these objectives tend to be competing in nature. For instance, consider drug discovery, where each evaluation of the functions is an in-vitro experiment and as the output of the experiment, we measure the solubility, toxicity and potency of a candidate example. A chemist wishes to find a molecule that has high solubility and potency, but low toxicity. This is an archetypal example for Bayesian optimization as the lab experiment is expensive. Further, drugs that are very potent are also likely to be toxic, so these two objectives are typically competing. Other problems include creating fast but accurate neural networks. While smaller neural networks are faster to evaluate, they suffer in terms of accuracy.

Due to their conflicting nature, all the objectives cannot be optimized simultaneously. As a result, most multi-objective optimization (MOO) approaches aim to recover the Pareto front, defined as the set of Pareto optimal points. A point is Pareto optimal if it cannot be improved in any of the objectives without degrading some other objective. More formally, given K objectives $f(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_K(\boldsymbol{x})) : \mathcal{X} \to \mathbb{R}^K$ over a compact domain $\mathcal{X} \subset \mathbb{R}^d$, a point $\boldsymbol{x}_1 \in \mathcal{X}$ is Pareto dominated by another point $\boldsymbol{x}_2 \in \mathcal{X}$ iff $f_k(\boldsymbol{x}_1) \leq f_k(\boldsymbol{x}_2) \,\forall k \in [K]$ and $\exists k \in [K]$ s.t. $f_k(\boldsymbol{x}_1) < f_k(\boldsymbol{x}_2)$, where we use the notation $[K]$ throughout the paper to denote the set $\{1, \ldots, K\}$. We denote this by $f(\boldsymbol{x}_1) \prec f(\boldsymbol{x}_2)$. A point is Pareto optimal if it is not Pareto dominated by any other point. We use $\mathcal{X}_f^\star$ to denote the Pareto front for a multi-objective function f, and $f(\mathcal{X}_f^\star)$ to denote the set of Pareto optimal values, where $f(\boldsymbol{X}) = \{f(\boldsymbol{x}) \mid \boldsymbol{x} \in \boldsymbol{X}\}$ for any $\boldsymbol{X} \subseteq \mathcal{X}$. The traditional goal in the MOO optimization regime is to approximate the set of Pareto optimal points (Hernández-Lobato et al., 2016; Knowles, 2006; Ponweiser et al., 2008; Zuluaga et al., 2013).

---

*This work was done when KK was at CMU

However, in certain scenarios, it is preferable to explore only a part of the Pareto front. For example, consider the drug discovery application described above. A method which aims to find the Pareto front, might also invest its budget to discover drugs that are potent, but too toxic to administer to a human. Such scenarios arise commonly in many practical applications. Therefore, we need flexible methods for MOO that can steer the sampling strategy towards regions of the Pareto front that a domain expert may be interested in. Towards this end, we propose a Bayesian approach based on random-scalarizations in which the practitioner encodes their preferences as a prior on a set of scalarization functions.

A common approach to multi-objective optimization is to use *scalarization functions*[1] $\mathfrak{s}_{\boldsymbol{\lambda}}(\boldsymbol{y}) : \mathbb{R}^K \to \mathbb{R}$ (Roijers et al., 2013), parameterized by $\boldsymbol{\lambda}$ belonging to a set Λ, and $\boldsymbol{y} \in \mathbb{R}^K$ denoting K-dimensional objective values. Scalarizations are often used to convert multi-objective values to scalars, and standard Bayesian optimization methods for scalar functions are applied. Since our goal is to sample points from the Pareto front, we need additional assumptions to ensure that the utility functions are maximized for $\boldsymbol{y} \in f(\mathcal{X}_f^{\star})$. Following Roijers et al. (2013) and Zintgraf et al. (2015) we assume that $\mathfrak{s}_{\boldsymbol{\lambda}}(\boldsymbol{y})$ are monotonically increasing in all coordinates. Optimizing for a single fixed scalarization amounts to the following maximization problem, which returns a single optimal point lying on the Pareto front.

$$\boldsymbol{x}_{\boldsymbol{\lambda}}^{\star} = \underset{\boldsymbol{x} \in \mathcal{X}}{\operatorname{argmax}} \ \mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x})) \tag{1}$$

One can verify that $\boldsymbol{x}_{\boldsymbol{\lambda}}^{\star} \in \mathcal{P}_f$ follows from the monotonicity of the scalarization. In this work, we are interested in a set of points $\boldsymbol{X}^{\star} = \{\boldsymbol{x}_i^{\star}\}_{i=1}^T$ of size at most T, spanning a specified region of the Pareto front rather than a single point. To achieve this we take a Bayesian approach and assume a prior $p(\boldsymbol{\lambda})$ with support on Λ, which intuitively translates to a prior on the set of scalarizations $\mathcal{S}_{\Lambda} = \{\mathfrak{s}_{\boldsymbol{\lambda}} \mid \boldsymbol{\lambda} \in \Lambda\}$. Thus, in place of optimizing a single scalarization, we aim to optimize over a set of scalarizations weighted by the prior $p(\boldsymbol{\lambda})$. Each $\boldsymbol{\lambda} \in \Lambda$ maps to a pareto optimal value $f(\boldsymbol{x}_{\boldsymbol{\lambda}}^{\star}) \in f(\mathcal{X}_f^{\star})$. Thus, the prior $p(\boldsymbol{\lambda})$ defines a probability distribution over the set of Pareto optimal values, and hence encodes user preference, which is depicted in Figure 1.

In this paper, we propose to minimize a Bayes regret which incorporates user preference through the prior and scalarization specified by the user. We propose multi-objective extensions of classical BO algorithms: upper

confidence bound (UCB) (Auer, 2002), and Thompson sampling (TS) (Thompson, 1933) to minimize our proposed regret. At each step the algorithm computes the next point to evaluate by randomly sampling a scalarization $\mathfrak{s}_{\boldsymbol{\lambda}}$ using the prior $p(\boldsymbol{\lambda})$, and optimizes it to get $\boldsymbol{x}_{\boldsymbol{\lambda}}^{\star}$. Our algorithm is fully amenable to changing priors in an interactive setting, and hence can also be used with other interactive strategies in the literature. The complete algorithm is presented in Algorithm 1 and discussed in detail in Section 3. While random scalarizations have been previously explored by Knowles (2006) and Zhang et al. (2010), our approach is different in terms of the underlying algorithm. Furthermore, we study a more general class of scalarizations and also prove regret bounds. As we shall see, this formulation fortunately gives rise to an extremely flexible framework that is much simpler than the existing work for MOO and computationally less expensive. Our contributions can summarized as follows:

1. We propose a flexible framework for MOO using the notion of random scalarizations. Our algorithm is flexible enough to sample from the entire Pareto front or an arbitrary region specified by the user. It is also naturally capable of sampling from non-convex regions of the Pareto front. While other competing approaches can be modified to sample from such complex regions, this seamlessly fits into our framework. In contrast to the prior work on MOBO, we consider more general scalarizations that are only required to be Lipschitz and monotonic.

2. We prove sublinear regret bounds making only assumptions of Lipschitzness and monotonicity of the scalarization function. To our knowledge the only prior work discussing theoretical guarantees for MOO algorithms is Pareto Active Learning (Zuluaga et al., 2013) with sample complexity bounds.

3. We compare our algorithm to other existing MOO approaches on synthetic and real-life tasks. We demonstrate that our algorithm achieves the said flexibility and superior performance in terms of the proposed regret, while being computationally inexpensive.

Related Work

Most multi-objective bayesian optimization approaches aim at approximating the whole Pareto front. Predictive Entropy Search (PESMO) by Hernández-Lobato et al. (2016) is based on reducing the posterior entropy of the Pareto front. SMSego by Ponweiser et al. (2008) uses an optimistic estimate of the function in an UCB fashion, and chooses the point with the maximum hypervolume improvement. Pareto Active Learning (PAL) (Zuluaga et al., 2013) and ε-PAL (Zuluaga et al., 2016) are similar to SMSego, and with theoretical guarantees. Campigotto

---

[1]Also known as utility functions in decision theory literature. To avoid confusion with acquisition functions which are sometimes referred to as utility functions in BO, we use the term scalarization function.

et al. (2014) introduce another active learning approach that approximates the surface of the Pareto front. Expected hypervolume improvement (EHI) (Emmerich and Klinkenberg, 2008) and Sequential uncertainty reduction (SUR) (Picheny, 2015) are two similar approaches based on maximizing the expected hypervolume. Computing the expected hypervolume is an expensive process that renders EHI and SUR computationally intractable in practice when there are several objectives.

The idea of random scalarizations has been previously explored in the following works aimed at recovering the whole Pareto front: ParEGO (Knowles, 2006) which uses random scalarizations to explore the whole Pareto front; MOEA/D (Zhang and Li, 2007), an evolutionary computing approach to MOO; and MOEA/D-EGO (Zhang et al., 2010), an extension of MOEA/D using Gaussian processes that evaluates batches of points at a time instead of a single point. At each iteration, both ParEGO and MOEA/D-EGO sample a weight vector uniformly from the $K-1$ simplex, which is used to compute a scalar objective. The next candidate point is chosen by maximizing an *off-the-shelf* acquisition function over the GP fitted on the scalar objective. Our algorithm on the other hand, maintains K different GPs, one for each objective. Furthermore, our approach *necessitates* using acquisitions specially designed for the multi-objective setting for any general scalarization; more specifically, they are generalizations of single-objective acquisitions for multiple objectives (see Table 1). These differences with ParEGO are not merely superficial – our approach gives rise to a theoretical regret bound, while no such bound exists for the above methods.

Another line of work involving scalarizations include utility function based approaches. Roijers et al. (2013); Zintgraf et al. (2015) propose scalar utility functions as an evaluation criteria. Zintgraf et al. (2018); Roijers et al. (2018, 2017) propose interactive strategies to maximize an unknown utility. In contrast to our approach the utility in these works is assumed to be fixed.

While there has been ample work on incorporating preferences in multi-objective optimization using evolutionary techniques (Deb and Sundar, 2006; Thiele et al., 2009; Kim et al., 2012; Branke and Deb, 2005; Branke, 2008), there has been fewer on using preferences for optimization, when using surrogate functions. Surrogate functions are essential for expensive black-box optimization. PESC (Garrido-Merchán and Hernández-Lobato, 2016) is an extension of PESM allowing to specify preferences as constraints. Hakanen and Knowles (2017) propose an extension of ParEGO in an interactive setting, where users provide feedback on the observations by specifying constraints on the objectives in an online fashion. Yang et al. (2016) propose another way to take preferences into account by using truncated functions. An interesting idea proposed by Sato et al. (2007) uses a modified notion of Pareto dominance to prevent one or more objectives from being too small. The survey by Ishibuchi et al. (2008) on evolutionary approaches to MOO can be referred for a more extensive review.

When compared to existing work for MOO, our approach enjoys the following advantages.

1. *Flexibility:* Our approach allows the flexibility to specify any region of the Pareto front including non-connected regions of the Pareto front, which is not an advantage enjoyed by other methods. Furthermore, the approach is flexible enough to recover the entire Pareto front when necessary. Our approach is not restricted to linear scalarization and extends to a much larger class of scalarizations.

2. *Theoretical guarantees:* Our approach seamlessly lends itself to analysis using our proposed notion of regret, and achieves sub-linear regret bounds.

3. *Computational simplicity:* The computational complexity of our approach scales linearly with the number of objectives K. This is in contrast to EHI and SUR, whose complexity scales exponentially with K. Our method is also computationally cheaper than other entropy based methods such as PESMO.

2 BACKGROUND

Most BO approaches make use of a probabilistic model acting as a surrogate to the unknown function. Gaussian processes (GPs) Rasmussen and Williams (2006) are a popular choice for their ability to model well calibrated uncertainty at unknown points. We will begin with a brief review of GPs and single objective BO.

Gaussian Processes. A Gaussian process (GP) defines a prior distribution over functions defined on some input space $\mathcal{X}$. GPs are characterized by a mean function $\mu : \mathcal{X} \mapsto \mathbb{R}$ and a kernel $\kappa : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$. For any function $f \sim \mathcal{GP}(\mu, \kappa)$ and some finite set of points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$, the function values $f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)$ follow a multivariate Gaussian distribution with mean μ and covariance Σ given by $\mu_i = \mu(\boldsymbol{x}_i)$, $\Sigma_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \; \forall 1 \leq i, j \leq n$. Examples of popular kernels include the squared exponential and the Matérn kernel. The mean function is often assumed to be 0 without any loss of generality. The posterior process, given observations $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{t-1}$ where $y_i = f(\boldsymbol{x}_i) + \epsilon_i \in \mathbb{R}$, $\epsilon_i \sim \mathcal{N}(\mu, \sigma^2)$, is also a GP with the mean and kernel

function given by

$$\mu_t(\boldsymbol{x}) = k^T(\Sigma + \sigma^2 I)^{-1}Y,$$
$$\kappa_t(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\boldsymbol{x}, \boldsymbol{x}') - k^T(\Sigma + \sigma^2 I)^{-1}k'. \quad (2)$$

where $Y = [y_i]_{i=1}^t$ is the vector of observed values, $\Sigma = [\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^t$ is the Gram matrix, $k = [\kappa(\boldsymbol{x}, \boldsymbol{x}_i)]_{i=1}^t$, and $k' = [\kappa(\boldsymbol{x}', \boldsymbol{x}_i)]_{i=1}^t$. Further details on GPs can be found in Rasmussen and Williams (2006).

Bayesian Optimization. BO procedures operate sequentially, using past observations $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{t-1}$ to determine the next point $\boldsymbol{x}_t$. Given $t-1$ observations Thompson Sampling (TS) (Thompson, 1933) draws a sample g_t from the posterior GP. The next candidate $\boldsymbol{x}_t$ is choosen as $\boldsymbol{x}_t = \operatorname{argmax} g_t(\boldsymbol{x})$. Gaussian Process UCB (Srinivas et al., 2010) constructs an upper confidence bound U_t as $U_t(\boldsymbol{x}) = \mu_{t-1}(\boldsymbol{x}) + \sqrt{\beta_t}\sigma_{t-1}(\boldsymbol{x})$. Here μ_{t-1} and σ_{t-1} are the posterior mean and covariances according to equations 2. β_t is a function of t and the dimensionality of the input domain $\mathcal{X}$. GP-UCB stipulates that we choose $\boldsymbol{x}_t = \operatorname{argmax}_{\boldsymbol{x} \in \mathcal{X}} U_t(\boldsymbol{x})$.

In this paper, we assume that the K objectives $f_1, \ldots, f_K$ are sampled from known GP priors $\mathcal{GP}(0, \kappa_k)$, $(k \in [K])$, with a common compact domain $\mathcal{X} \subset \mathbb{R}^d$. Without loss of generality, we assume $\mathcal{X} \subseteq [0,1]^d$. The feasible region is defined as $f(\mathcal{X})$. We further assume that the observations are noisy, that is, $y_k = f_k(\boldsymbol{x}) + \varepsilon_k$, where $\varepsilon_k \sim \mathcal{N}(0, \sigma_k^2)$, $\forall k \in [K]$.

3 OUR APPROACH

We first provide a formal description of random scalarizations, then we formulate a regret minimization problem, and finally propose multi-objective extensions of the classical UCB and TS algorithms to optimize it.

3.1 Random Scalarizations

As discussed earlier in Section 1 in this paper we consider a set of scalarizations $\mathfrak{s}_{\boldsymbol{\lambda}}$ parameterized by $\boldsymbol{\lambda} \in \Lambda$. We assume a prior $p(\boldsymbol{\lambda})$ with support Λ. We further assumed that, for all $\boldsymbol{\lambda} \in \Lambda$, $\mathfrak{s}_{\boldsymbol{\lambda}}$ is $L_{\boldsymbol{\lambda}}$-Lipschitz in the ℓ_1-norm and monotonically increasing in all the coordinates. More formally,

$$\mathfrak{s}_{\boldsymbol{\lambda}}(\boldsymbol{y}_1) - \mathfrak{s}_{\boldsymbol{\lambda}}(\boldsymbol{y}_2) \le L_{\boldsymbol{\lambda}}\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_1,$$
$$\forall \boldsymbol{\lambda} \in \Lambda, \ \boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{R}^d,$$
$$\text{and,} \quad \mathfrak{s}_{\boldsymbol{\lambda}}(\boldsymbol{y}_1) < \mathfrak{s}_{\boldsymbol{\lambda}}(\boldsymbol{y}_2) \text{ whenever } \boldsymbol{y}_1 \prec \boldsymbol{y}_2. \quad (3)$$

The Lipschitz condition can also be generalized to ℓ_p-norms using the fact that $\|\boldsymbol{y}\|_1 \ge K^{1-\frac{1}{p}}\|\boldsymbol{y}\|_p$ for any
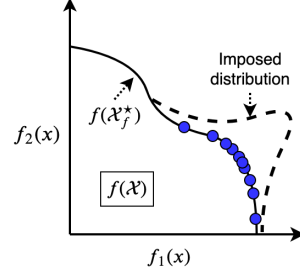

Figure 1: A prior $p(\boldsymbol{\lambda})$ imposes a distribution on the set of Pareto optimal values. The imposed probability density is illustrated using the dotted lines. The imposed distribution leads to a concentration of the sampled values (blue circles) in the high probability region.

$p \in [1, \infty]$ and $\boldsymbol{y} \in \mathbb{R}^K$. Monotonicity ensures that

$$\boldsymbol{x}_{\boldsymbol{\lambda}}^\star = \operatorname*{argmax}_{\boldsymbol{x} \in \mathcal{X}} \ \mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x})) \in \mathcal{X}_f^\star,$$

since otherwise, if $f(\boldsymbol{x}_{\boldsymbol{\lambda}}^\star) \prec f(\boldsymbol{x})$ for some $\boldsymbol{x} \ne \boldsymbol{x}_{\boldsymbol{\lambda}}^\star$, then we have $\mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x}_{\boldsymbol{\lambda}}^\star)) < \mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x}))$, leading to a contradiction. Each $\boldsymbol{\lambda} \in \Lambda$ maps to an $\boldsymbol{x}_{\boldsymbol{\lambda}}^\star \in \mathcal{X}_f^\star$ and a $\boldsymbol{y}^\star = f(\boldsymbol{x}_{\boldsymbol{\lambda}}^\star) \in f(\mathcal{X}_f^\star)$. Assuming the required measure theoretic regularity conditions hold, the prior $p(\boldsymbol{\lambda})$ imposes a probability distribution on $f(\mathcal{X}_f^\star)$ through the above mapping as depicted in Figure 1.

3.2 Bayes Regret

In contrast to (1), which returns a single optimal point, in this work, we aim to return a set of points from the user specified region. Our goal is to compute a subset $\boldsymbol{X} \subset \mathcal{X}$ such that $f(\boldsymbol{X})$ spans the high probability region of $f(\mathcal{X}_f^\star)$. This can be achieved by minimizing the following *Bayes regret* denoted by $\mathcal{R}_B$,

$$\mathcal{R}_B(\boldsymbol{X}) = \mathbb{E}_{\boldsymbol{\lambda} \sim p(\boldsymbol{\lambda})}\Big(\underbrace{\max_{\boldsymbol{x} \in \mathcal{X}} \mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x})) - \max_{\boldsymbol{x} \in \boldsymbol{X}} \mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x}))}_{\text{Pointwise regret}} \Big),$$
$$\boldsymbol{X}^\star = \operatorname*{argmin}_{\boldsymbol{X} \subset \mathcal{X}, |\boldsymbol{X}| \le T} \mathcal{R}_B(\boldsymbol{X}) \quad (4)$$

We now elaborate on the above expression. The pointwise regret $\max_{\boldsymbol{x} \in \mathcal{X}} \mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x})) - \max_{\boldsymbol{x} \in \boldsymbol{X}} \mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x}))$ quantifies the regret for a particular $\boldsymbol{\lambda}$ and is analogous to the simple regret in the standard bandit setting (Bubeck et al., 2012). $\mathcal{R}_B(T)$ similarly corresponds to the Bayes simple regret in a bandit setting. The pointwise is minimized when $\boldsymbol{x}_{\boldsymbol{\lambda}}^\star = \operatorname{argmax}_{\boldsymbol{x} \in \mathcal{X}} \ \mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x}))$ belongs to $\boldsymbol{X}$. Since $\boldsymbol{X}$ is finite, the minimum may not be achieved for all $\boldsymbol{\lambda}$, as the set of optimal points can be potentially infinite. However, the regret can be small when $\exists \boldsymbol{x} \in \boldsymbol{X}$ such that $f(\boldsymbol{x})$, $f(\boldsymbol{x}_{\boldsymbol{\lambda}}^\star)$ are close,
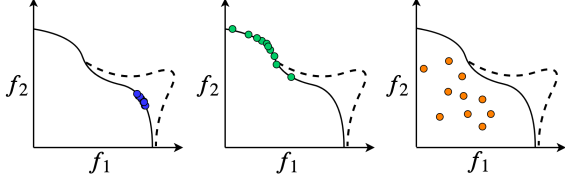
Figure 2: Three scenarios which incur a high regret: (1) The points are clustered in a small region. (2) The points are not from the desired distribution. (3) The points are not Pareto optimal.

from which it follows using the Lipschitz assumption that $\mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x}_{\boldsymbol{\lambda}}^{\star})) - \mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x}))$ is small. Therefore, roughly speaking, the Bayes regret is minimized when the points in $\boldsymbol{X}$ are Pareto optimal and $f(\boldsymbol{X})$ well approximates the high probability regions of $f(\mathcal{X}_f^{\star})$. In this case, $\mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x}_{\boldsymbol{\lambda}}^{\star})) - \mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x}))$ is small for $\boldsymbol{\lambda}$s with high probabilities. Even though the rest of the regions are not well approximated, it does not affect the Bayes regret since those regions do not dominate the expectation by virtue of their low probability. This is what was desired from the beginning, that is, to compute a set of points with the majority of them spanning the desired region of interest. This is also illustrated in Figure 2 showing three scenarios which can incur a high regret.

It is interesting to ask, why cannot one simply maximize

$$\max_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}_{\boldsymbol{\lambda} \sim p(\boldsymbol{\lambda})} \left[\mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x})) \right].$$

The above expression can be maximized using a single point $\boldsymbol{x}$ which is not the purpose of our approach. On the other hand, our proposed Bayes regret is not minimized by a single point or multiple points clustered in a small region of the Pareto front. Minimizing the pointwise regret for a single $\boldsymbol{\lambda}$ does not minimize the Bayes regret, as illustrated in Figure 2. Our proposed regret has some resemblance to the expected utility metric in Zintgraf et al. (2015). However, the authors present it as an evaluation criteria, whereas we propose an optimization algorithm for minimizing it and also prove regret bounds on it.

3.3 Scalarized Thompson Sampling and UCB

In this section we introduce Thompson Sampling and UCB based algorithms for minimizing the Bayes regret. In contrast to other methods based on random scalarizations (Knowles, 2006; Nakayama et al., 2009), our algorithm does not convert each observation to a scalar value and fit a GP on them, but instead models them separately by maintaining a GP for each objective separately. In each iteration, we first fit a GP for each objective using the previous observations. Then we sample a $\boldsymbol{\lambda} \sim p(\boldsymbol{\lambda})$,

which is used to compute a multi-objective acquisition function based on the scalarization $\mathfrak{s}_{\boldsymbol{\lambda}}$. The next candidate point is chosen to be the maximizer of the acquisition function. The complete algorithm is presented in Algorithm 1 and the acquisition functions are presented in Table 1. The acquisition function for UCB is a scalarization of the individual upper bounds of each of the objectives. Similarly, the acquisition function for TS is a scalarization of posterior samples of the K objectives.

The intuition behind our approach is to choose the $\boldsymbol{x}_t$ that minimizes the pointwise regret for the particular $\boldsymbol{\lambda}_t$ sampled in that iteration. Looking at the expression of the Bayes regret, at a high level, it seems that it can be minimized by sampling a $\boldsymbol{\lambda}$ from the prior and choosing an $\boldsymbol{x}_t$ that minimizes the regret for the sampled $\boldsymbol{\lambda}$. We prove regret bounds for both TS and UCB in Section 4 and show that this idea is indeed true.

Practical Considerations. In practice, our method requires the prior and class of scalarization functions to be specified by the user. These would typically be domain dependent. In practice, a user would also interactively update their prior based on the observations, as done in Hakanen and Knowles (2017); Roijers et al. (2017, 2018). Our approach is fully amenable to changing the prior interactively, and changing regions of interest. In this paper we do not propose any general methods for choosing or updating the prior, as it is not possible to do so for any general class of scalarizations. The interested readers can refer to the literature on interactive methods for MOBO. However, for the sake of demonstration we propose a simple heuristic in the experimental section.

3.4 Computational Complexity

At each step all algorithms incur a cost of at most $\mathcal{O}(KT^3)$, for fitting K GPs, except for ParEGO, which fits a single GP at each time step with a cost of $\mathcal{O}(T^3)$. The next step of maximizing the acquisition function differs widely across the algorithms. Computing the acquisition function at each point $\boldsymbol{x}$ costs $\mathcal{O}(T)$ for ParEGO, and $\mathcal{O}(KT)$ for our approach. The additional factor K is the price one must pay when maintaining K GPs.

Apart from fitting the K GPs, SMSEgo requires computing the expected hypervolume gain at each point which is much more expensive than computing the acquisitions for UCB or TS. Computing the expected hypervolume improvement in EHI is expensive and grows exponentially with K. PESM has a cost that is linear in K. However the computation involves performing expensive steps of expectation-propagation and MC estimates, which results in a large constant factor.

Algorithm 1 MOBO using Random Scalarizations (MOBO-RS)

Init $\mathcal{D}^{(0)} \leftarrow \emptyset$, $\mathcal{GP}_k^{(0)} \leftarrow \mathcal{GP}(\mathbf{0}, \kappa)$, $\forall k \in [K]$
for $t = 1 \rightarrow T$ **do**
    Sample $\boldsymbol{\lambda}_t \sim p(\boldsymbol{\lambda})$
    $\boldsymbol{x}_t \leftarrow \operatorname{argmax}_{\boldsymbol{x} \in \mathcal{X}} \operatorname{acq}(\boldsymbol{x}, \boldsymbol{\lambda}_t)$
              {See Table 1 for acquisition functions}
    Evaluate $\boldsymbol{y} = f(\boldsymbol{x}_t)$
    $\mathcal{D}^{(t)} = \mathcal{D}^{(t-1)} \cup \{(\boldsymbol{x}_t, \boldsymbol{y})\}$
    $\mathcal{GP}_k^{(t)} \leftarrow \operatorname{post}\left(\mathcal{GP}_k^{(t-1)} \mid (\boldsymbol{x}_t, \boldsymbol{y}_k)\right), \forall k \in [K]$
end for
return $\mathcal{D}^{(T)}$

Table 1: Acquisition functions used in Algorithm 1. $\mu^{(t)}(\boldsymbol{x})$, $\sigma^{(t)}(\boldsymbol{x})$ are K dimensional vectors denoting the posterior means and variances at $\boldsymbol{x}$ of the K objectives respectively, in step t. c is a hyperparameter and d is dimension of the input space $\mathcal{X}$. f_k' is randomly sampled from the posterior of the kth objective function.

	$\operatorname{acq}(\boldsymbol{x}, \boldsymbol{\lambda})$
UCB	$\mathfrak{s}_{\boldsymbol{\lambda}}\left(\mu^{(t)}(\boldsymbol{x}) + \sqrt{\beta_t}\sigma^{(t)}(\boldsymbol{x})\right), \beta_t = cd \ln t$
TS	$\mathfrak{s}_{\boldsymbol{\lambda}}(f'(\boldsymbol{x}))$, where $f_k' \sim \mathcal{GP}_k^{(t)}$, $k \in [K]$

4 REGRET BOUNDS

In this section we provide formal guarantees to prove upper bounds on the Bayes regret $\mathcal{R}_B$ which goes to zero as $T \rightarrow \infty$. We also show that our upper bound is able to recover regret bounds for single objectives when $K = 1$.

Analogous to the notion of regret in the single-objective setting (Bubeck et al., 2012), we first define the instantaneous and cumulative regrets for the multi-objective optimization. The *instantaneous regret* incurred by our algorithm in step t is defined as,

$$r(\boldsymbol{x}_t, \boldsymbol{\lambda}_t) = \max_{\boldsymbol{x} \in \mathcal{X}} \mathfrak{s}_{\boldsymbol{\lambda}_t}(f(\boldsymbol{x})) - \mathfrak{s}_{\boldsymbol{\lambda}_t}(f(\boldsymbol{x}_t)), \quad (5)$$

where $\boldsymbol{\lambda}_t$ and $\boldsymbol{x}_t$ are the same as in Algorithm 1. The *cumulative regret* till step T is defined as,

$$\mathcal{R}_C(T) = \sum_{t=1}^{T} r(\boldsymbol{x}_t, \boldsymbol{\lambda}_t). \quad (6)$$

For convenience, we do not explicitly mention the dependency of $\mathcal{R}_C(T)$ on $\{\boldsymbol{x}_t\}_{t=1}^{T}$ and $\{\boldsymbol{\lambda}_t\}_{t=1}^{T}$. Next, we will make a slight abuse of notation here and define $\mathcal{R}_B(T)$, the Bayes regret incurred till step T, as $\mathcal{R}_B(\boldsymbol{X}_T)$ (See Eqn. 4), where $\boldsymbol{X}_T = \{\boldsymbol{x}_t\}_{t=1}^{T}$.

We further define the *expected Bayes regret* as $\mathbb{E}\mathcal{R}_B(T)$, where the expectation is taken over the random process f, noise ε and any other randomness occurring in the algorithm. Similarly, we also define the *expected cumulative regret* as $\mathbb{E}\mathcal{R}_C(T)$, where the expectation is taken over all the aforementioned random variables and additionally $\{\boldsymbol{\lambda}_t\}_{t=1}^{T}$. We will show that the expected Bayes regret can be upper bounded by the expected cumulative regret, which can be further upper bounded using the maximum information gain.

Maximum Information Gain. The *maximum information gain (MIG)* captures the notion of information gained about a random process f given a set of observations. For any subset $A \subset \mathcal{X}$ define $\boldsymbol{y}_A = \{y_a = f(a) + \varepsilon_a | a \in A\}$. The reduction in uncertainty about a random process can be quantified using the notion of information gain given by $\mathbf{I}(\boldsymbol{y}_A; f) = \mathbf{H}(\boldsymbol{y}_A) - \mathbf{H}(\boldsymbol{y}_A | f)$, where $\mathbf{H}$ denotes the Shannon entropy. The maximum information gain after T observations is defined as

$$\gamma_T = \max_{A \subset \mathcal{X}: |A| = T} \mathrm{I}(\boldsymbol{y}_A; f). \quad (7)$$

Regret Bounds. We assume that $\forall k \in [K]$, $t \in [T]$, $\boldsymbol{x} \in \mathcal{X}$, $f_k(\boldsymbol{x})$ follows a Gaussian distribution with marginal variances upper bounded by 1, and the observation noise $\varepsilon_{tk} \sim \mathcal{N}(0, \sigma_k^2)$ is drawn independently of everything else. Assume upper bounds $L_{\boldsymbol{\lambda}} \leq L$, $\sigma_k^2 \leq \sigma^2$, $\gamma_{Tk} \leq \gamma_T$, where γ_{Tk} is the MIG for the k th objective. When $\mathcal{X} \subseteq [0, 1]^d$, the cumulative regret after T observations can be bounded as follows.

Theorem 1. *The expected cumulative regret for MOBO-RS after T observations can be upper bounded for both UCB and TS as,*

$$\mathbb{E}\mathcal{R}_C(T) = \mathcal{O}\left(L\left[\frac{K^2 T d \gamma_T \ln T}{\ln(1 + \sigma^{-2})}\right]^{1/2}\right). \quad (8)$$

The proof follows from Theorem 2 in the appendix. The bound for single-objective BO can be recovered by setting $K = 1$, which matches the bound of $\mathcal{O}(\sqrt{T d \gamma_T \ln T})$ shown in Russo and Van Roy (2014); Srinivas et al. (2010). The proof is build on ideas for single objective analyses for TS and UCB (Russo and Van Roy, 2014; Kandasamy et al., 2018).

Under further assumption of the space Λ being a bounded subset of a normed linear space, and the scalarizations $\mathfrak{s}_{\boldsymbol{\lambda}}$ being Lipschitz in $\boldsymbol{\lambda}$, it can be shown that $\mathbb{E}\mathcal{R}_B(T) \leq \frac{1}{T}\mathbb{E}\mathcal{R}_C(T) + o(1)$, which combined with Theorem 1 shows that the Bayes regret converges to zero as $T \rightarrow \infty$. A complete proof can be found in Appendix B.3.

5   EXPERIMENTAL RESULTS

We experiment with both synthetic and real world problems. We compare our methods to the other existing MOO approaches in the literature: PESM, EHI, SM-Sego, ParEGO, and MOEA/D-EGO. EHI being computationally expensive is not feasible for more than two objectives. Other than visually comparing the results for three or lesser objectives we also compare them in terms of the Bayes regret defined in Eqn. 4.

While our method is valid for any scalarization satisfying the Lipschitz and monotonicity conditions, we demonstrate the performance of our algorithm on two commonly used scalarizations, the linear and the Tchebyshev scalarizations (Nakayama et al., 2009) defined as,

$$\mathfrak{s}_{\boldsymbol{\lambda}}^{\mathrm{lin}}(\boldsymbol{y}) = \sum_{k=1}^{K} \boldsymbol{\lambda}_k \boldsymbol{y}_k,$$

$$\mathfrak{s}_{\boldsymbol{\lambda}}^{\mathrm{tch}}(\boldsymbol{y}) = \min_{k=1}^{K} \boldsymbol{\lambda}_k (\boldsymbol{y}_k - \boldsymbol{z}_k), \tag{9}$$

where $\boldsymbol{z}$ is some reference point. In both cases, $\Lambda = \{\boldsymbol{\lambda} \succ \mathbf{0} \mid \|\boldsymbol{\lambda}\|_1 = 1\}$. It can be verified that the Lipschitz constant in both cases is upper bounded by 1.

Choosing the weight distribution $p(\boldsymbol{\lambda})$. While the user has the liberty to choose any distribution best suited for the application at hand, for demonstration we show one possible way. A popular way of specifying user preferences is by using bounding boxes (Hakanen and Knowles, 2017), where the goal is to satisfy $f_k(\boldsymbol{x}) \in [a_k, b_k], \forall 1 \le k \le K$. We convert bounding boxes to a weight distribution using a heuristic described below.

For the linear scalarization, it can be verified that the regret is minimized when $\boldsymbol{y}$ is pareto optimal, and the normal vector at the surface of the Pareto front at $\boldsymbol{y}$ has the same direction as $\boldsymbol{\lambda}$. This is illustrated using a simple example in Figure 3. Consider two simple objectives $f_1(x, y) = xy, f_2(x, y) = y\sqrt{1 - x^2}$. Sampling $\boldsymbol{\lambda} = \left[\frac{u}{u+1}, \frac{1}{u+1}\right]$ where $u \sim \mathrm{Unif}(0, 0.3)$, results in the first figure. In this example we have $\boldsymbol{\lambda}_1$ smaller than $\boldsymbol{\lambda}_2$, resulting in exploration of the region where $f_2(x, y)$ is high. Whereas sampling $\boldsymbol{\lambda} = \left[\frac{u}{u+v}, \frac{v}{u+v}\right]$ where $u, v \sim \mathrm{Unif}(0.5, 0.7)$ results in the second figure since both components of $\boldsymbol{\lambda}$ have similar magnitudes. This idea leads to the following heuristic to convert bounding boxes to a sampling strategy. We sample as $\boldsymbol{\lambda} = \boldsymbol{u}/\|\boldsymbol{u}\|_1$ where $\boldsymbol{u}_k \sim \mathrm{Unif}(a_k, b_k), \ k \in [K]$. The intuition behind this approach is shown in Figure 4. Such a weight distribution roughly samples points from inside the bounding box.

For the Tchebychev scalarization, at the optimum, $\boldsymbol{y} - \boldsymbol{z}$
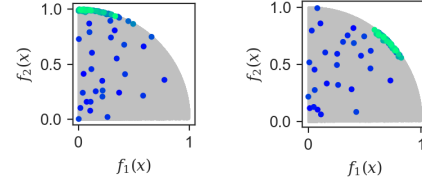

Figure 3: The feasible region is shown in grey. The color of the sampled points corresponds to the iteration they were sampled in, with brighter colors being sampled in the later iterations.
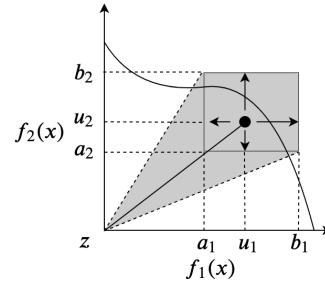

Figure 4: Weight distribution from bounding box.

is inversely proportional to $\boldsymbol{\lambda}$. For the purpose of demonstration and comparison we would like both the scalarization to obtain similar objective values. Therefore, we reuse the $\boldsymbol{\lambda}$ sampled for the linear scalarization to get $\boldsymbol{\lambda}_{\mathrm{tch}} = \boldsymbol{\lambda}'/\|\boldsymbol{\lambda}'\|_1$ where $\boldsymbol{\lambda}' = (1/\boldsymbol{\lambda}_1, \dots, 1/\boldsymbol{\lambda}_K)$. We have normalized the vector so that it lies in Λ.

In order to explore the whole Pareto front, one can also specify a *flat* distribution. For instance consider the Dirichlet distribution on the simplex $\{\boldsymbol{x} \in \mathbb{R}^K \mid \sum_{k=1}^{K} \boldsymbol{x}_k = 1, \boldsymbol{x} \succ 0\}$. One can sample from the Dirichlet distribution as $\boldsymbol{\lambda} \sim \mathrm{Dir}(1, \dots, 1)$, which roughly provides equal weight to all the objectives leading to exploration of the whole Pareto front. Other strategies include $\boldsymbol{\lambda} = |\boldsymbol{\lambda}'|/\|\boldsymbol{\lambda}'\|_1$ where $\boldsymbol{\lambda}' \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

Other possible ways of choosing the weight vector includes learning the distribution of the weight vector from interactive user feedback. In fact, our framework also allows us to perform a joint posterior inference on the GP model and the weight distribution, thus learning the weight distribution in a more principled manner. While we leave these methods to future work, this demonstrates the flexibility of our framework.

Experimental Setup. For all our experiments, we use the squared exponential function as the GP kernel (in practice, this is a hyperparameter that must be specified by the user), given by $\kappa(\boldsymbol{x}_1, \boldsymbol{x}_2) = s \exp\left(-\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2/(2\sigma^2)\right)$, where s and σ are parameters that are estimated during optimization. We perform

experiments with both TS and UCB using both kinds of scalarizations. In Eqn. 4, we observe that the term $\mathbb{E}_{\boldsymbol{\lambda}} \max_{\boldsymbol{x} \in \mathcal{X}} \mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x}))$ is independent of the algorithm, hence it is sufficient to plot $-\mathbb{E}_{\boldsymbol{\lambda}} \max_{\boldsymbol{x} \in \boldsymbol{X}} \mathfrak{s}_{\boldsymbol{\lambda}}(f(\boldsymbol{x}))$. In all our experiments, we plot this expression, thus avoiding computing the global maximum of an unknown function. For the purposes of computing the Bayes simple regret, we linearly map the objective values to $[0, 1]$ so that the values are of reasonable magnitude. This however is not a requirement of our algorithm. Further experimental details can be found in the Appendix. The implementation can be found in Dragonfly[2], a publicly available python library for scalable Bayesian optimization (Kandasamy et al., 2019).

Synthetic two-objective function. We construct a synthetic two-objective optimization problem using the Branin-4 and CurrinExp-4 functions as the two objectives respectively. These are the 4-dimensional counterparts of the Branin and CurrinExp functions (Lizotte, 2008), each mapping $[0, 1]^4 \rightarrow \mathbb{R}$. For this experiment we specify the bounding boxes $[(a_1, b_1), (a_2, b_2)]$. We sample from three different regions, which we label as *top*: $[(-110, -95), (23, 27)]$, *mid*:$[(-80, -70), (16, 22)]$, and *flat*: where we sample from a flat distribution. We also sample from a mixture of the *top* and *mid* distributions denoted by *top/mid*, thus demonstrating sampling from non-connected regions in the Pareto front. Figure 5 shows a scatter plot of the sampled values for the various methods. The simple regret plots are shown in Figure 6.

Synthetic six-objective function. To show the viability of our method in high-dimensions, we sample six random functions $f_k : \mathbb{R}^6 \rightarrow \mathbb{R}$, $f_k \sim \mathcal{GP}(\mathbf{0}, \kappa)$, $k \in [6]$ where κ is the squared exponential kernel. Devoid of any domain knowledge about this random function, we linearly transform the objectives values to $[0, 1]$ for simplicity. We specify the bounding box as $[a_k, b_k] = [2/3, 1]$, $\forall k \in [6]$ and denote it as the *mid* region, as the weight samples are of similar magnitude. The simple regret plot for this experiment is shown in Figure 7.

Locality Sensitive Hashing. Locality Sensitive Hashing (LSH) (Andoni et al., 2015) is a randomized algorithm for computing the k-nearest neighbours. LSH involves a number of tunable parameters: the number of hash tables, number of hash bits, and the number of probes to make for each query. The parameters affect the average query time, precision and memory usage. While increasing the number of hash tables results in smaller query times, it leads to an increase in the memory footprint. Similarly, while increasing the number of probes leads to a higher precision, it increases the query time. We
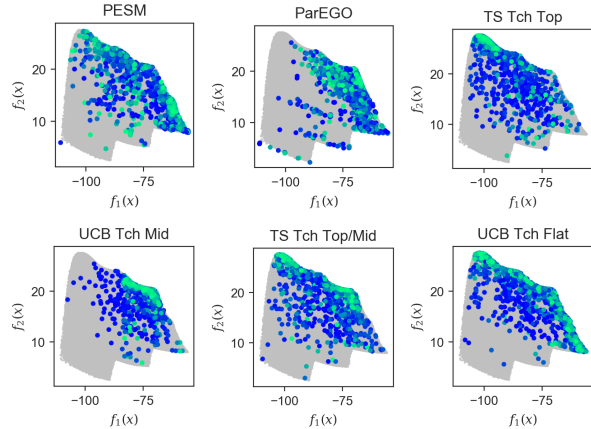
Figure 5: The feasible region is shown in grey. The color of the sampled points corresponds to the iteration they were sampled in, with brighter colors being sampled in the later iterations. The figure titles denote the method used and the region sampled. A complete set of results is presented in the Appendix.

explore the trade-offs between these three objectives.

We run LSH using the publicly available FALCONN library[3] on Glove word embeddings (Pennington et al., 2014). We use the Glove Wikipedia-Gigaword dataset trained on 6B tokens with a vocabulary size of $400K$ and 300-d embeddings. Given a word embedding, finding the nearest word embedding from a dictionary of word embeddings is a common task in NLP applications. We consider the following three objectives to minimize with their respective bounding boxes: Time (s) $[0.0, 0.65]$, $1-$Precision $[0.0, 0.35]$, and the Memory (MB) $[0, 1600]$. The SR plots are shown in Figure 7 and the sampled objective values in Figure 8.

Viola Jones. The Viola Jones algorithm (Viola and Jones, 2001) is a fast stagewise face detection algorithm. At each stage a simple feature detector is run over the image producing a real value. If the value is smaller than a threshold the algorithm exits with a decision, otherwise the image is processed by the next stage and so on. The Viola Jones pipeline has 27 tunable thresholds. We treat these thresholds as inputs and optimize for Sensitivity, Specificity, and the Time per query. We consider the following three objectives to minimize with their bounding boxes: $1-$Sensitivity $[0, 0.3]$, $1-$Specificity $[0, 0.13]$, and Time per query $[0, 0.07]$. Figure 7 shows the regret plot for this experiment.

Results and Discussion. Figures 5 and 8 show the sampling patterns of our proposed approach for the synthetic 2-d and the LSH glove experiment. We observe that our
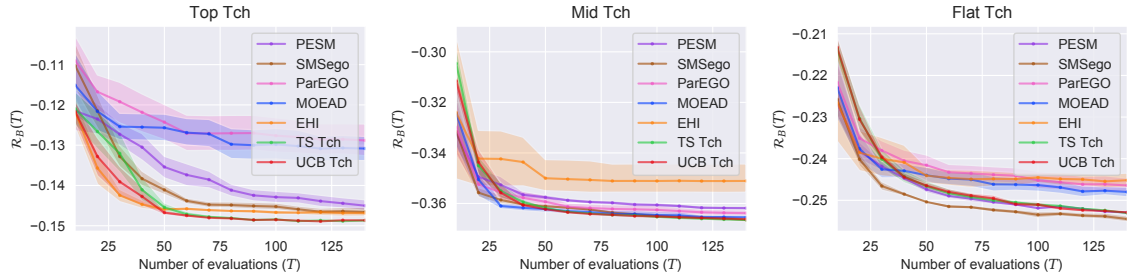
Figure 6: Bayes regret plots for the synthetic two-objective function. The mean and the 90% confidence interval were computed over 10 independent runs. The figure titles denote the sampling region and the scalarization used. We refer the reader to the appendix for results on linear scalarization.
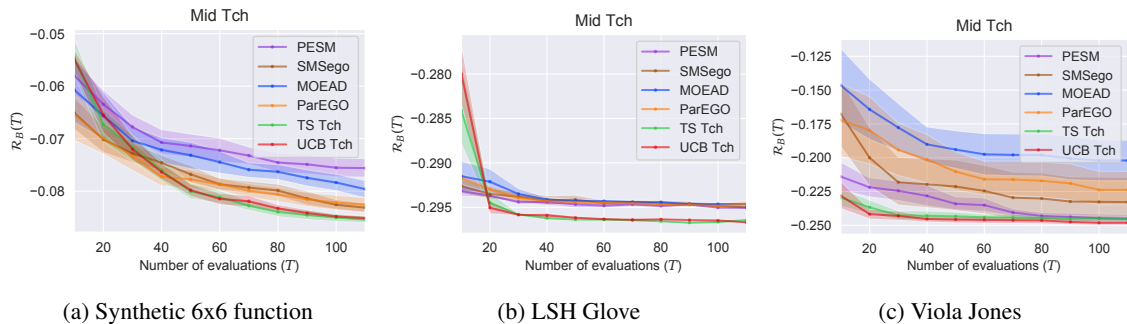

(a) Synthetic 6x6 function   (b) LSH Glove   (c) Viola Jones

Figure 7: Bayes regret plots. The mean and the 90% confidence interval were computed over 5 runs. The figure titles denote the region sampled and the scalarization used. A complete set of plots can be found in the appendix.
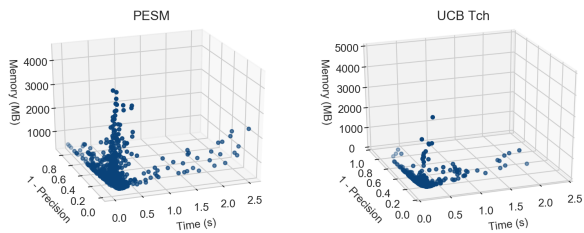

Figure 8: Sampled values for the LSH-Glove experiment over 5 independent runs. The figure titles denote the method used. A complete set of plots can be found in the appendix.

approach successfully samples from the specified region after some initial exploration, leading to a high concentration of points in the desired part of the Pareto front in the later iterations.

In Figures 6 and 7 we observe that the proposed approach achieves a smaller or comparable regret compared to the other baselines. We notice that the improvement is most significant for the high dimensional experiments. A plausible explanation for this could be that learning high dimensional surfaces have a much higher sample complexity. However, our since our approach learns only a part

of the Pareto front, it is able to achieve a small regret in a few number of samples, thus demonstrating the effectiveness of our approach.

6 CONCLUSION

In this paper we proposed a MOBO algorithm for efficient exploration of specific parts of the Pareto front. We experimentally showed that our algorithm can successfully sample from a specified region of the Pareto front as is required in many applications, but is still flexible enough to sample from the whole Pareto front. Furthermore, our algorithm is computationally cheap and scales linearly with the number of objectives.

Our approach also lends itself to a notion of regret in the MO setting that also captures user preferences; with the regret being high if not sampled from the specified region or sampled outside of it. We provided a theoretical proof of the fact that our algorithm achieves a zero regret in the limit under necessary regularity assumptions. We experimentally showed that our approach leads to a smaller or comparable regret compared to the baselines.

References

Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. Practical and optimal LSH for angular distance. In *Advances in Neural Information Processing Systems*, 2015.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 2002.

Jürgen Branke. Consideration of partial user preferences in evolutionary multiobjective optimization. In *Multiobjective optimization*. Springer, 2008.

Jürgen Branke and Kalyanmoy Deb. Integrating user preferences into evolutionary multi-objective optimization. In *Knowledge incorporation in evolutionary computation*. Springer, 2005.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 2012.

Paolo Campigotto, Andrea Passerini, and Roberto Battiti. Active learning of Pareto fronts. *IEEE transactions on neural networks and learning systems*, 2014.

Guillermo Canas and Lorenzo Rosasco. Learning probability measures with respect to optimal transport metrics. In *Advances in Neural Information Processing Systems*, 2012.

Kalyanmoy Deb and J Sundar. Reference point based multi-objective optimization using evolutionary algorithms. In *Genetic and evolutionary computation*. ACM, 2006.

Michael Emmerich and Jan-willem Klinkenberg. The computation of the expected improvement in dominated hypervolume of Pareto front approximations. *Rapport technique, Leiden University*, 2008.

Eduardo C Garrido-Merchán and Daniel Hernández-Lobato. Predictive entropy search for multi-objective Bayesian optimization with constraints. *arXiv preprint arXiv:1609.01051*, 2016.

Subhashis Ghosal and Anindya Roy. Posterior consistency of Gaussian process prior for nonparametric binary regression. *The Annals of Statistics*, 2006.

Javier González, Joseph Longworth, David C James, and Neil D Lawrence. Bayesian optimization for synthetic gene design. *arXiv preprint arXiv:1505.01627*, 2015.

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

Daniel Hernández-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. Predictive entropy search for multi-objective Bayesian optimization. In *International Conference on Machine Learning*, 2016.

José Miguel Hernández-Lobato, James Requeima, Edward O Pyzer-Knapp, and Alán Aspuru-Guzik. Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space. In *International Conference on Machine Learning*. JMLR, 2017.

Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization: A short review. In *IEEE Congress on Evolutionary Computation*. IEEE, 2008.

Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of optimization Theory and Applications*, 1993.

Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning*, 2015.

Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. Query efficient posterior estimation in scientific experiments via Bayesian active learning. *Artificial Intelligence*, 2017.

Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabás Póczos. Parallelised Bayesian optimisation via Thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, 2018.

Kirthevasan Kandasamy, Karun Raju Vysyaraju, Willie Neiswanger, Biswajit Paria, Christopher R Collins, Jeff Schneider, Barnabas Poczos, and Eric P Xing. Tuning hyperparameters without grad students: Scalable and robust Bayesian optimisation with dragonfly. *arXiv preprint arXiv:1903.06694*, 2019.

Jong-Hwan Kim, Ji-Hyeong Han, Ye-Hoon Kim, Seung-Hwan Choi, and Eun-Soo Kim. Preference-based solution selection algorithm for evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2012.

Joshua Knowles. ParEGO: A hybrid algorithm with online landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 2006.

Daniel James Lizotte. *Practical Bayesian optimization*. University of Alberta, 2008.

Ruben Martinez-Cantin, Nando de Freitas, Arnaud Doucet, and José A Castellanos. Active policy learning for robot planning and exploration under uncertainty. In *Robotics: Science and Systems*, 2007.

Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie royale des sciences de Paris*, 1781.

Hirotaka Nakayama, Yeboon Yun, and Min Yoon. *Sequential approximate multiobjective optimization using computational intelligence*. Springer Science & Business Media, 2009.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *empirical methods in natural language processing (EMNLP)*, 2014.

Victor Picheny. Multiobjective optimization using Gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, 2015.

Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted $\mathcal{S}$-metric selection. In *International Conference on Parallel Problem Solving from Nature*. Springer, 2008.

Carl Edward Rasmussen and Christopher KI Williams. Gaussian processes for machine learning. 2006. *The MIT Press, Cambridge, MA, USA*, 2006.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 2013.

Diederik M Roijers, Luisa M Zintgraf, and Ann Nowé. Interactive Thompson sampling for multi-objective multi-armed bandits. In *International Conference on Algorithmic DecisionTheory*. Springer, 2017.

Diederik M Roijers, Luisa M Zintgraf, Pieter Libin, and Ann Nowé. Interactive multi-objective reinforcement learning in multi-armed bandits for any utility function. In *ALA workshop at FAIM*, 2018.

Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 2014.

Hiroyuki Sato, Hernán E Aguirre, and Kiyoshi Tanaka. Controlling dominance area of solutions and its impact on the performance of moeas. In *International conference on evolutionary multi-criterion optimization*. Springer, 2007.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, 2012.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *International Conference on Machine Learning*. JMLR, 2010.

Lothar Thiele, Kaisa Miettinen, Pekka J Korhonen, and Julian Molina. A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary computation*, 2009.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 1933.

Cédric Villani. *Optimal transport: old and new*. Springer Science & Business Media, 2008.

Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition*. IEEE, 2001.

Kaifeng Yang, Longmei Li, André Deutz, Thomas Back, and Michael Emmerich. Preference-based multiobjective optimization using truncated expected hypervolume improvement. In *Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. IEEE, 2016.

Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 2007.

Qingfu Zhang, Wudong Liu, Edward Tsang, and Botond Virginas. Expensive multiobjective optimization by moea/d with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 2010.

Luisa M Zintgraf, Timon V Kanters, Diederik M Roijers, Frans A Oliehoek, and Philipp Beau. Quality assessment of MORL algorithms: A utility-based approach. In *Belgian-Dutch Conference on Machine Learning*, 2015.

Luisa M Zintgraf, Diederik M Roijers, Sjoerd Linders, Catholijn M Jonker, and Ann Nowé. Ordered preference elicitation strategies for supporting multi-objective decision making. In *International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Marcela Zuluaga, Guillaume Sergent, Andreas Krause, and Markus Püschel. Active learning for multi-objective optimization. In *International Conference on Machine Learning*, 2013.

Marcela Zuluaga, Andreas Krause, and Markus Püschel. ε-pal: an active learning approach to the multi-objective optimization problem. *Journal of Machine Learning Research*, 2016.