arXiv:2011.01542v1 [cs.AI] 2 Nov 2020

Multi-Fidelity Multi-Objective Bayesian Optimization: An Output Space Entropy Search Approach

Syrine Belakaria, Aryan Deshwal, Janardhan Rao Doppa
School of EECS, Washington State University
{syrine.belakaria, aryan.deshwal, jana.doppa}@wsu.edu

Abstract

We study the novel problem of blackbox optimization of multiple objectives via multi-fidelity function evaluations that vary in the amount of resources consumed and their accuracy. The overall goal is to approximate the true Pareto set of solutions by minimizing the resources consumed for function evaluations. For example, in power system design optimization, we need to find designs that trade-off cost, size, efficiency, and thermal tolerance using multi-fidelity simulators for design evaluations. In this paper, we propose a novel approach referred as **M***ulti*-**F***idelity* **O***utput* **S***pace* **E***ntropy Search for* **M***ulti-objective* **O***ptimization (MF-OSEMO)* to solve this problem. The key idea is to select the sequence of candidate input and fidelity-vector pairs that maximize the information gained about the true Pareto front per unit resource cost. Our experiments on several synthetic and real-world benchmark problems show that MF-OSEMO, with both approximations, significantly improves over the state-of-the-art single-fidelity algorithms for multi-objective optimization.

1 Introduction

Multi-objective optimization of expensive black-box functions has many real-world applications. For example, creating hardware to optimize performance, reliability, and thermal objectives. There are two key challenges in solving these problems. First, the objective functions are unknown and we need to select from experiments of different fidelity to evaluate each candidate input. These multi-fidelity experiments vary in the amount of resources consumed and the accuracy of evaluation. Second, all the objectives cannot be optimized simultaneously due to their conflicting nature. Hence, we resort to finding the *Pareto optimal* set of solutions. A solution is called Pareto optimal if it cannot be improved in any of the objectives without compromising some other objective. The overall goal is to approximate the optimal Pareto set by minimizing the overall resource cost of function evaluations.

Bayesian optimization (BO) (Shahriari et al. 2016) is a popular framework for solving blackbox optimization problems. BO methods build a surrogate statistical model, e.g., Gaussian process, from the training data of function evaluations; employ a acquisition function (AF) that is parameterized by the model, e.g., upper-confidence bound, to score the utility of evaluating candidate inputs; and select the highest scoring input for evaluation in each iteration. Existing AFs can be broadly classified into two categories. First, *myopic* AFs rely on improving a "local" measure of utility (e.g., expected improvement). Second, *non-myopic* AFs measure the "global" utility of evaluating a candidate input for solving the blackbox optimization problem (e.g., predictive entropy search). Prior work has shown the advantages of non-myopic AFs over myopic AFs in terms of both theory and practice (Jiang et al. 2017; Hernández-Lobato, Hoffman, and Ghahramani 2014; Wang and Jegelka 2017; Hoffman and Ghahramani 2015).

In this paper, we propose a novel and principled approach referred as **M**ulti-**F**idelity **O**utput **S**pace **E**ntropy Search for **M**ulti-objective **O**ptimization (MF-OSEMO) to solve multi-objective optimization problems via multi-fidelity function evaluations. *To the best of our knowledge, this is the first work to study this problem within ML literature*. MF-OSEMO employs an output space entropy based non-myopic acquisition function to select the candidate inputs and fidelity vectors for evaluation. Output space entropy search has many advantages over other non-myopic AFs based on input space entropy (Hoffman and Ghahramani 2015; Wang and Jegelka 2017): a) allows much tighter approximation; b) significantly cheaper to compute; and c) naturally lends itself to robust optimization. We provide two qualitatively different approximations to efficiently compute the entropy, which is a key step for MF-OSEMO. These approximations make different trade-offs in terms of accuracy and computational-efficiency: one has a closed-form expression and another employs numerical integration.

Contributions. We make the following key contributions.

- Developing a principled approach referred as MF-OSEMO to solve multi-fidelity multi-objective blackbox optimization problems. MF-OSEMO employs an output space entropy based acquisition function to select the sequence of candidate inputs and fidelity vectors for evaluation. Providing two different approximations within MF-OSEMO.
- Experimental evaluation on synthetic and real-world benchmark problems to show the effectiveness of MF-OSEMO over state-of-the-art single-fidelity algorithms.

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

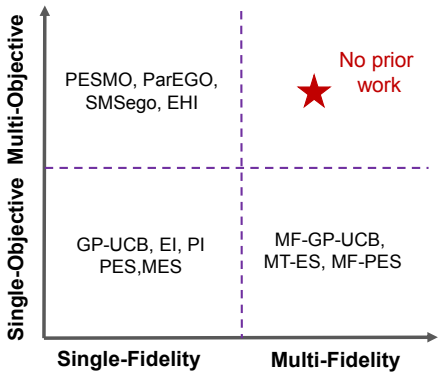

Figure 1: Current state of knowledge for generic BO methods.

2 Problem Setup

Let $\mathfrak{X} \subseteq \Re^d$ be an input space. In the multi-objective optimization problem, our goal is to minimize $K \geq 2$ *expensive* objective functions $f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_K(\mathbf{x})$. Evaluation of a candidate input $\mathbf{x} \in \mathfrak{X}$ produces a vector of K function values $\mathbf{y} = (y_1, y_2, \cdots, y_K)$, where $y_i = f_i(x)$ for all $i \in \{1, 2, \cdots, K\}$. A point $\mathbf{x}$ is said to *Pareto-dominate* another point $\mathbf{x}'$ if $f_i(\mathbf{x}) \leq f_i(\mathbf{x}')\ \forall i$ and there exists some $j \in \{1, 2, \cdots, K\}$ such that $f_j(\mathbf{x}) < f_j(\mathbf{x}')$. The optimal solution of MOO problem is a set of points $\mathcal{X}^* \subset \mathfrak{X}$ such that no point $\mathbf{x}' \in \mathfrak{X} \setminus \mathcal{X}^*$ Pareto-dominates a point $\mathbf{x} \in \mathcal{X}^*$. The solution set $\mathcal{X}^*$ is called the optimal *Pareto set* and the corresponding set of function values $\mathcal{Y}^*$ is called the optimal *Pareto front*. In the multi-fidelity version of MOO problem, we have access to M_i fidelities for each function f_i that vary in the amount of resources consumed and the accuracy of evaluation. Let $\lambda_i^{(m_i)}$ be the cost of evaluating i^{th} function f_i at $m_i \in [M_i]$ fidelity, where m_i=M_i corresponds to the highest fidelity for f_i. Evaluation of an input $\mathbf{x} \in \mathfrak{X}$ with fidelity vector $\mathbf{m} = [m_1, m_2, \cdots, m_K]$ produces an evaluation vector of K values denoted by $\mathbf{y}^{\mathbf{m}} \equiv [y_1^{(m_1)}, \cdots, y_K^{(m_K)}]$, where $y_i^{(m_i)} = f_i^{(m_i)}(x)$ for all $i \in \{1, 2, \cdots, K\}$, and normalized cost of evaluation is $\lambda^{(\mathbf{m})} \equiv \sum_{i=1}^{K} \left(\lambda_i^{(m_i)} / \lambda_i^{(M_i)}\right)$. We normalize the total cost since the cost units can be different for different objectives(e.g. cost unit for f_1 is computation time while cost unit for f_2 could be memory space size). Our goal is to approximate $\mathcal{X}^*$ by minimizing the overall cost of function evaluations. For the sake of reader, Table 1 contains all the mathematical notations used in this paper.

3 Related work

Multi-fidelity single-objective optimization. AFs for single-fidelity and single-objective BO has been extensively studied (Shahriari et al. 2016). Canonical examples of myopic AFs include expected improvement (EI) and upper-confidence bound (UCB). EI was extended to multi-fidelity setting (Huang et al. 2006; Picheny et al. 2013; Lam, Allaire, and Willcox 2015). The popular GP-UCB method (Srinivas et al. 2009) was also extended to multi-fidelity setting with discrete fidelities (Kandasamy et al. 2016) and continuous fidelities (Kandasamy et al. 2017).

Entropy based methods fall under the category of *non-myopic* AFs Some examples include entropy search (ES) (Hennig and Schuler 2012) and predictive entropy search (PES) (Hernández-Lobato, Hoffman, and Ghahramani 2014). Their multi-fidelity extensions include MT-ES (Swersky, Snoek, and Adams 2013; Klein et al. 2017) and MF-PES(Zhang et al. 2017; McLeod, Osborne, and Roberts 2017). Unfortunately, they inherit the computational difficulties of the original ES and PES. Max-value entropy search (MES) and output space predictive entropy search (Wang and Jegelka 2017; Hoffman and Ghahramani 2015) are recent approaches that rely on the principle of output space entropy (OSE) search. Prior work (Wang and Jegelka 2017) has shown advantages of OSE search in terms of compute-time, robustness, and accuracy over input space entropy search methods. A recent work (Takeno et al. 2019) extended MES to multi-fidelity setting and showed its effectiveness over MF-PES. Recent work (Song, Chen, and Yue 2019) proposed a general approach based on mutual information.

Single-fidelity multi-objective optimization. Multi-objective algorithms can be classified into three families. *Scalarization methods* are model-based algorithms that reduce the problem to single-objective optimization. ParEGO method (Knowles 2006) employs random scalarization for this purpose. ParEGO is simple and fast, but more advanced approaches often outperform it. *Pareto hypervolume optimization methods* optimize the Pareto hypervolume (PHV) metric (Emmerich and Klinkenberg 2008) that captures the quality of a candidate Pareto set. This is done by extending the standard acquisition functions to PHV objective, e.g., expected improvement in PHV (Emmerich and Klinkenberg 2008) and probability of improvement in PHV (Picheny 2015). Unfortunately, algorithms to optimize PHV based acquisition functions scale very poorly and are not feasible for more than three objectives. To improve scalability, methods to reduce the search space are also explored (Ponweiser et al. 2008). A common drawback of this family is that reduction to single-objective optimization can potentially lead to more exploitative behavior.

Uncertainty reduction methods like PAL (Zuluaga et al. 2013), PESMO (Hernández-Lobato et al. 2016) and the concurrent works USeMO (Belakaria et al. 2020) and MESMO (Belakaria, Deshwal, and Doppa 2019) are principled algorithms based on information theory. In each iteration, PAL selects the candidate input for evaluation towards the goal of minimizing the size of uncertain set. PAL provides theoretical guarantees, but it is only applicable for input space $\mathfrak{X}$ with finite set of discrete points. USeMO is a general framework that iteratively generates a cheap Pareto front using the surrogate models and then selects the point with highest uncertainty as the next query. PESMO relies on input space entropy search and iteratively selects the input that maximizes the information gained about the optimal Pareto set $\mathcal{X}^*$. Unfortunately, optimizing this acquisition function poses significant challenges: a) requires a series of approximations, which can be potentially sub-optimal; and b) optimization,

Notation	Meaning
$\mathbf{x}, \mathbf{y}, \mathbf{f}, \mathbf{m}$	bold notation represents vectors
$[n]$	set of first n natural numbers $\{1, 2, \cdots, n\}$
$f_1, f_2, \cdots, f_K$	true objective functions
$M_1, M_2, \cdots, M_K$	no. of fidelities for each function
$\mathbf{x}$	input vector
$\mathbf{m} = [m_1, m_2, \cdots, m_K]$	fidelity vector where each fidelity $m_j \in [M_j]$
$y_j^{m_j}$	jth function f_j evaluated at m_jth fidelity where $m_j \in [M_j]$
$\mathbf{y}^{\mathbf{m}}$	output vector equivalent to $[y_1^{(m_1)}, \cdots, y_K^{(m_K)}]$
$\mathcal{Y}^*$	true Pareto front of the highest fidelities $[y_1^{(M_1)}, y_2^{(M_2)}, \cdots, y_K^{(M_K)}]$
$\lambda_j^{(m_j)}$	cost of evaluating jth function f_j at m_jth fidelity
$\lambda^{(\mathbf{m})}$	total normalized cost $\lambda^{(\mathbf{m})} \equiv \sum_{j=1}^{K} \left(\lambda_j^{(m_j)} / \lambda_j^{(M_j)} \right)$
$\tilde{f}_j^{(m_j)}$	function sampled from jth gaussian process model at m_jth fidelity

Table 1: Table describing the mathematical notations used in this paper.

even after approximations, is expensive c) performance is strongly dependent on the number of Monte-Carlo samples. MESMO (Belakaria, Deshwal, and Doppa 2019) is a concurrent work that improves over PESMO by extending MES to the multi-objective setting.

Application-specific multi-fidelity multi-objective optimization. Prior work outside ML literature has considered domain-specific methods that employ single-fidelity multi-objective approaches in the context of multi-fidelity setting by using the lower fidelities *only as an initialization* (Kontogiannis et al. 2018; Ariyarit and Kanazaki 2017). Specifically, (Ariyarit and Kanazaki 2017) employs the single-fidelity algorithm based on expected hypervolume improvement acquisition function and (Kontogiannis et al. 2018) employs an algorithm that is very similar to SMSego. Additionally, both these methods model all fidelities with the same GP and assume that higher fidelity evaluation is a sum of lower-fidelity evaluation and offset error. These are strong assumptions and may not hold in general multi-fidelity settings including the problems we considered in our experimental evaluation.

4 MF-OSEMO Algorithm

In this section, we explain the technical details of our proposed MF-OSEMO algorithm.

4.1 Surrogate models

Let $D = \{(\mathbf{x}_i, \mathbf{y}_i^{(\mathbf{m})})\}_{i=1}^{t-1}$ be the training data from past $t-1$ function evaluations, where $\mathbf{x}_i \in \mathfrak{X}$ is an input and $\mathbf{y}_i^{(\mathbf{m})} = [y_1^{(m_1)}, y_2^{(m_2)}, \cdots, y_K^{(m_K)}]$ is the output vector resulting from evaluating functions $f_1^{(m_1)}, f_2^{(m_2)}, \cdots, f_K^{(m_k)}$ at $\mathbf{x}_i$.

Gaussian processes (GPs) are known to be effective surrogate models in prior work on single and multi-objective BO (Srinivas et al. 2009; Hernández-Lobato et al. 2016). We learn K surrogate models $\mathcal{GP}_1, \mathcal{GP}_2, \cdots, \mathcal{GP}_K$ from $\mathcal{D}$, where each $\mathcal{GP}_j$ corresponds to the jth function f_j. In our setting, each function has multiple fidelities. So one ideal property desired for the surrogate model of a single function is to take into account all the fidelities in a single model. Multi-fidelity GPs (MF-GP) are capable of modeling functions with multiple fidelities in a single model. Hence, each of our surrogate model $\mathcal{GP}_j$ is a multi-fidelity GP.

Specifically, we use the MF-GP model as proposed in (Kennedy and O'Hagan 2000; Takeno et al. 2019). We describe the complete details of the MF-GP model below. One key thing to note about MF-GP model is that the kernel function ($k((\mathbf{x_i}, m_i), (\mathbf{x_j}, m_j))$) is dependent on both the input and the fidelity. For a given input $\mathbf{x}$, the MF-GP model returns a *vector* (one for each fidelity) of predictive mean, a *vector* of predictive variance, and a matrix of predictive covariance. The MF-GP model has two advantages: 1) All fidelities are integrated into one single GP; and 2) Difference among fidelities are adaptively estimated without any additional feature representation for fidelities. It should be noted that we employ an independent multi-fidelity GP for each function.

Multi-fidelity Gaussian process model. We describe full details of a MF-GP model for one objective function f_j (without loss of generality) below:

Let $y_j^{(1)}(\mathbf{x}), \ldots, y_j^{(M_j)}(\mathbf{x})$ represent the values obtained by evaluating the function f_j at its 1st, 2nd, $\ldots$, M_jth fidelity respectively.

In a MF-GP model, each fidelity is represented by a gaussian process and the observation is modeled as

$$y_j^{(m_j)}(\mathbf{x}) = f_j^{(m_j)}(\mathbf{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_{\text{noise}}^2).$$

Let $f_j^{(1)} \sim GP(0, k_1(\mathbf{x}, \mathbf{x}'))$ be a gaussian process for the 1st fidelity i.e. $m_j = 1$, where $k_1 : \mathcal{R}^d \times \mathcal{R}^d \to \mathcal{R}$ is a suitable kernel. The output for successively fidelities $m_j = 2, \ldots, M_j$ is recursively defined as

$$f_j^{(m_j)}(\mathbf{x}) = f_j^{(m_j-1)}(\mathbf{x}) + f_{j_e}^{(m_j-1)}(\mathbf{x}), \tag{4.1}$$

where, $f_{j_e}^{(m_j-1)} \sim GP(0, k_e(\mathbf{x}, \mathbf{x}'))$ with $k_e : \mathcal{R}^d \times \mathcal{R}^d \to \mathcal{R}$. It is assumed that $f_{j_e}^{(m_j-1)}$ is conditionally independent from all fidelities lower than m_j. As a result, *the kernel for a pair of points evaluated at the same fidelity* becomes:

$$k_{m_j}(\mathbf{x}, \mathbf{x}') \equiv k_1(\mathbf{x}, \mathbf{x}') + (m_j - 1) k_e(\mathbf{x}, \mathbf{x}') \tag{4.2}$$

and as a result, the output for m_jth fidelity is also mode as a gaussian process:

$$f_j^{(m_j)} \sim GP(0, k_{m_j}(\mathbf{x}, \mathbf{x}')).$$

The kernel function for a pair of inputs evaluated at dij ent fidelities m_j and m'_j is given as:

$$k((\mathbf{x}, m_j), (\mathbf{x}', m'_j)) = \text{cov}\left(f_j^{(m_j)}(\mathbf{x}), f_j^{(m'_j)}(\mathbf{x}')\right) = k_{m_j}(\mathbf{x}, \mathbf{x}'),$$

where $m_j \leq m'_j$ and cov represents covariance. Using a nel matrix $K \in \mathcal{R}^{n \times n}$ in which the p, q element is defi by $k((\mathbf{x}, m_j^p), (\mathbf{x}', m_j^q))$, all fidelities $f_j^{(1)}, \ldots, f_j^{(M_j)}$ car integrated into one common gaussian process by which dictive mean and variance are obtained as

$$\mu^{(m_j)}(\mathbf{x}) = K + \sigma^2_{\text{noise}} I^{-1}\mathbf{Y}, \quad (4$$

$$\sigma^{2^{(m_j)}}(\mathbf{x}) = k((\mathbf{x}, m_j), (\mathbf{x}, m_j)) - k_n^{(m_j)}(\mathbf{x})^\top K + \sigma^2_{\text{noise}} I^{-1} k_n^{(m_j)}(\mathbf{x}), \quad (4.4)$$

where $\mathbf{Y} = (y_1^{(m_{j_1})}(\mathbf{x}_1), \ldots, y_n^{(m_{j_n})}(\mathbf{x}_n))^\top$ and $k_n^{(m_j)}(\mathbf{x}) \equiv (k((\mathbf{x}, m_j), (\mathbf{x}_1, m_{j_1})), \ldots, k((\mathbf{x}, m_j), (\mathbf{x}_n, m_{j_n})))^\top$.

We also define $\sigma^{2^{(m_j m'_j)}}(\mathbf{x})$ as the predictive covariance between $(\mathbf{x}, m_j)$ and $(\mathbf{x}, m'_j)$, i.e., covariance for identical $\mathbf{x}$ at different fidelities:

$$\sigma^{2(m_j m'_j)}(\mathbf{x}) = k((\mathbf{x}, m_j), (\mathbf{x}, m'_j)) - k_n^{(m_j)}(\mathbf{x})^\top K + \sigma^2_{\text{noise}} I^{-1} k_n^{(m'_j)}(\mathbf{x}). \quad (4.5)$$

4.2 Multi-fidelity output space entropy based acquisition function

We describe our proposed acquisition function for multi-fidelity multi-objective setting in this section. We leverage the information-theoretic principle of output space information gain to develop an efficient and robust acquisition function. The proposed method is applicable for the general case, where at each iteration the objective functions can be evaluated at different fidelities.

The key idea behind the proposed acquisition function is to find the pair $\{\mathbf{x}, \mathbf{m}\}$ that maximizes the information gain about the **Pareto front of the highest fidelities (denoted by $\mathcal{Y}^*$)** per unit cost, where $\{\mathbf{x}, \mathbf{m}\}$ represents a candidate input $\mathbf{x}$ evaluated at a vector of fidelities $\mathbf{m} = [m_1, m_2, \cdots, m_K]$.

This idea can be expressed mathematically as given below:

$$\alpha(\mathbf{x}, \mathbf{m}) = I(\{\mathbf{x}, \mathbf{y}^{(\mathbf{m})}\}, \mathcal{Y}^* \mid D)/\lambda^{(\mathbf{m})} \quad (4.6)$$

where $\lambda^{(\mathbf{m})}$ is the total *normalized* cost of evaluating the objective functions at $\mathbf{m}$ and D is the data collected so far. Figure 2 provides an overview of the MF-OSEMO algorithm. The information gain in equation 4.6 is defined as the expected reduction in entropy $H(.)$ of the posterior distribution $P(\mathcal{Y}^* \mid D)$ as a result of evaluating $\mathbf{x}$ at fidelity vector $\mathbf{m}$:

$$I(\{\mathbf{x}, \mathbf{y}^{(\mathbf{m})}\}, \mathcal{Y}^* \mid D)$$
$$= H(\mathcal{Y}^* \mid D) - \mathbb{E}_{y^{(\mathbf{m})}}[H(\mathcal{Y}^* \mid D \cup \{\mathbf{x}, \mathbf{y}^{(\mathbf{m})}\})] \quad (4.7)$$
$$= H(\mathbf{y}^{(\mathbf{m})} \mid D, \mathbf{x}) - \mathbb{E}_{\mathcal{Y}^*}[H(\mathbf{y}^{(\mathbf{m})} \mid D, \mathbf{x}, \mathcal{Y}^*)] \quad (4.8)$$

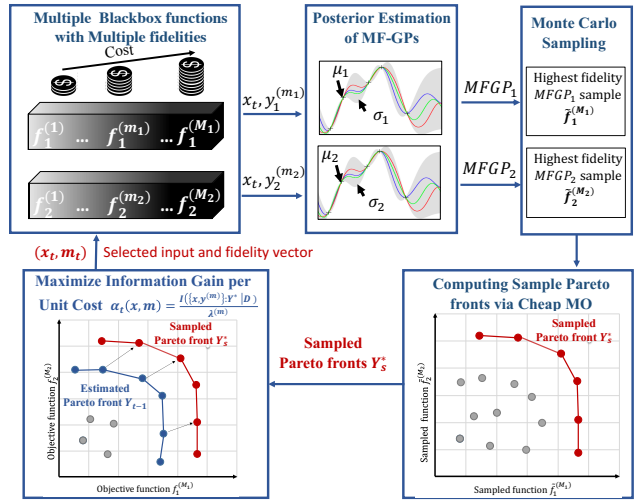

Figure 2: Overview of the MF-OSEMO algorithm for two objective functions (k=2). We build multi-fidelity statistical models $\mathcal{MFGP}_1$, $\mathcal{MFGP}_2$ for the two objective functions $f_1(x)$ and $f_2(x)$ with M_1 and M_2 fildelities respectively. First, we sample highest fidelity functions from the statistical models. We compute sample pareto fronts by solving a cheap MO problem over the sampled functions. Second, we select the best candidate input x_t and fidelity vector $m_t = (m_1, m_2)$ that maximizes the information gain per unit cost . Finally, we evaluate the functions for x_t at fidelities m_t to get $(y_1^{(m_1)}, y_2^{(m_2)})$ and update the statistical models using the new training example.

Equation 4.8 follows from equation 4.7 as a result of the symmetric property of information gain. The first term in the r.h.s of equation 4.8 is the entropy of a factorizable K-dimensional gaussian distribution $P(\mathbf{y}^{(\mathbf{m})} \mid D, \mathbf{x})$) which can be computed in closed form as shown below:

$$H(\mathbf{y}^{(\mathbf{m})} \mid D, \mathbf{x}) = \frac{K(1 + \ln(2\pi))}{2} + \sum_{j=1}^{K} \ln(\sigma_j^{(m_j)}(\mathbf{x})) \quad (4.9)$$

where $\sigma_j^{(m_j)}(\mathbf{x})$ is the predictive variance of jth surrogate model GP_j at input $\mathbf{x}$ and fidelity m_j. The second term in the r.h.s of equation 4.8 is an expectation over the Pareto front of the highest fidelities $\mathcal{Y}^*$. We can approximately compute this term via Monte-Carlo sampling as shown below:

$$\mathbb{E}_{\mathcal{Y}^*}[H(\mathbf{y}^{(\mathbf{m})} \mid D, \mathbf{x}, \mathcal{Y}^*)] \simeq \frac{1}{S} \sum_{s=1}^{S} [H(\mathbf{y}^{(\mathbf{m})} \mid D, \mathbf{x}, \mathcal{Y}_s^*)] \quad (4.10)$$

where S is the number of samples and $\mathcal{Y}_s^*$ denote a sample Pareto front obtained over the highest fidelity functions sample from K surrogate models. The main advantages of our acquisition function are: cost efficiency, computational-efficiency, and robustness to the number of samples. Our

experiments demonstrate these advantages over state-of-the-art single fidelity AFs for multi-objective optimization.

There are two key algorithmic steps to compute Equation 4.10: 1) Computing Pareto front samples $\mathcal{Y}_s^*$; and 2) Computing the entropy with respect to a given Pareto front sample $\mathcal{Y}_s^*$. We provide solutions for these two steps below.

1) Computing Pareto front samples via cheap multi-objective optimization. To compute a Pareto front sample $\mathcal{Y}_s^*$, we first sample highest fidelity functions from the posterior MF-GP models via random fourier features (Hernández-Lobato, Hoffman, and Ghahramani 2014; Rahimi and Recht 2008) and then solve a cheap multi-objective optimization over the K sampled high fidelity functions. It is important to note that we are sampling only the **highest fidelity function** from each MF-GP surrogate model.

Sampling functions from the posterior of MF-GP model. Similar to prior work (Hernández-Lobato, Hoffman, and Ghahramani 2014; Hernández-Lobato et al. 2016; Wang and Jegelka 2017), we employ random fourier features based sampling procedure. We approximate each GP prior of the highest fidelity as $\tilde{f}^{(M)} = \phi(\mathbf{x})^T\theta$, where $\theta \sim N(0, \mathbf{I})$. The key idea behind random fourier features is to construct each function sample $\tilde{f}^{(M)}(\mathbf{x})$ as a finitely parametrized approximation: $\phi(\mathbf{x})^T\theta$, where θ is sampled from its corresponding posterior distribution conditioned on the data D obtained from past function evaluations: $\theta|D \sim N(\mathbf{A^{-1}\Phi^T y}_n, \sigma^2\mathbf{A^{-1}})$, where $\mathbf{A} = \mathbf{\Phi^T\Phi} + \sigma^2\mathbf{I}$ and $\Phi^T = [\phi(\mathbf{x}_1), \cdots, \phi(\mathbf{x}_{t-1})]$.

Cheap MO solver. We sample $\tilde{f}_i^{(M_i)}$ from each surrogate model $\mathcal{MF} - \mathcal{GP}_i$ as described above. A *cheap* multi-objective optimization problem over the K sampled functions $\tilde{f}_1^{(M_1)}, \tilde{f}_2^{(M_2)}, \cdots, \tilde{f}_K^{(M_K)}$ is solved to compute the sample Pareto front $\mathcal{Y}_s^*$. This cheap multi-objective optimization also allows us to capture the interactions between different objectives. We employ the popular NSGA-II algorithm (Deb et al. 2002) to solve the MO problem with cheap objective functions noting that any other algorithm can be used.

2) Entropy computation with a sample Pareto front. Let $\mathcal{Y}_s^* = \{\mathbf{z}^1, \cdots, \mathbf{z}^l\}$ be the sample Pareto front, where l is the size of the Pareto front and each $\mathbf{z}^i = \{z_1^i, \cdots, z_K^i\}$ is a K-vector evaluated at the K sampled high fidelity functions. The following inequality holds for each component $y_j^{(m_j)}$ of the K-vector $\mathbf{y^{(m)}} = \{y_1^{(m_1)}, \cdots, y_K^{(m_k)}\}$ in the entropy term $H(\mathbf{y^{(m)}} \mid D, \mathbf{x}, \mathcal{Y}_s^*)$:

$$y_j^{(m_j)} \leq y_{j_s}^* \quad \forall j \in \{1, \cdots, K\} \tag{4.11}$$

where $y_{j_s}^* = \max\{z_j^1, \cdots z_j^l\}$. The inequality essentially says that the jth component of $\mathbf{y^m}$ (i.e., $y_j^{m_j}$) is upper-bounded by a value obtained by taking the maximum of jth components of all l vectors $\{\mathbf{z}^1, \cdots, \mathbf{z}^l\}$ in the Pareto front $\mathcal{Y}_s^*$. The proof of 4.11 can be divided into two cases:

Case I. If y_j is evaluated at its highest fidelity (i.e $m_j = M_j$), inequality 4.11 can be proven by a contradiction argument. Suppose there exists some component $y_j^{(M_j)}$ of $\mathbf{y^{(M)}}$ such that $y_j^{(M_j)} > y_{j_s}^*$. However, by definition, $\mathbf{y^{(M)}}$ is a non-dominated point because no point dominates it in the jth dimension. This results in $\mathbf{y^{(M)}} \in \mathcal{Y}_s^*$ which is a contradiction. Therefore, our hypothesis that $y_j^{(M_j)} > y_{j_s}^*$ is incorrect and inequality 4.11 holds.

Case II. If y_j is evaluated at one of its lower fidelities (i.e, $m_j \neq M_j$), the proof follows from the assumption that the value of lower fidelity of a objective is usually smaller than the corresponding higher fidelity, i.e., $y_j^{(m_j)} \leq y_j^{(M_j)} \leq y_{j_s}^*$. This is especially true for most real-world experiments. For example, in optimizing a neural network's accuracy with respect to its hyperparameters, a commonly employed fidelity is the number of data samples used for training. It is reasonable to believe that the accuracy is always higher for the higher fidelity (more data samples to train on) when compared to a lower fidelity (less data samples to train on).

By combining the inequality 4.11 and the fact that each function is modeled as an independent MF-GP, a common property of entropy measure allows us to decompose the entropy of a set of independent variables into a sum over entropies of individual variables (Cover and Thomas 2012):

$$H(\mathbf{y^{(m)}} \mid D, \mathbf{x}, \mathcal{Y}_s^*) \simeq \sum_{j=1}^{K} H(y_j^{(m_j)}|D, \mathbf{x}, y_{j_s}^*) \tag{4.12}$$

The computation of 4.12 requires the computation of the entropy of $p(y_j^{(m_j)}|D, \mathbf{x}, y_{j_s}^*)$. This is a conditional distribution that depends on the value of m_j and can be expressed as $H(y_j^{(m_j)}|D, \mathbf{x}, y_j^{(m_j)} \leq y_{j_s}^*)$. This entropy is dealt with in two cases:

First, for $\mathbf{m_j = M_j}$, the density function of this probability is a truncated Gaussian distribution and its entropy can be expressed as (Michalowicz, Nichols, and Bucholtz 2013):

$$H(y_j^{(M_j)}|D, \mathbf{x}, y_j^{(M_j)} \leq y_{j_s}^*) = \frac{(1 + \ln(2\pi))}{2} + \ln(\sigma_j^{(M_j)}(\mathbf{x})) + \ln \Phi(\gamma_s^{(M_j)}(\mathbf{x})) - \frac{\gamma_s^{(M_j)}(\mathbf{x})\phi(\gamma_s^{(M_j)}(\mathbf{x}))}{2\Phi(\gamma_s^{(M_j)}(\mathbf{x}))} \tag{4.13}$$

where $\gamma_s^{(M_j)}(\mathbf{x}) = \frac{y_{j_s}^* - \mu_j^{(M_j)}(\mathbf{x})}{\sigma_j^{(M_j)}(\mathbf{x})}$, and ϕ and Φ are the p.d.f and c.d.f of a standard normal distribution respectively.

Second, for $\mathbf{m_j \neq M_j}$, the density function of $p(y_j^{(m_j)}|D, \mathbf{x}, y_{j_s}^*)$ can be computed using two different approximations as described below:

Approximation 1 (MF-OSEMO-TG): As a consequence of **Case II**, which states that $y_j^{(m_j)} \leq y_{j_s}^*$ also holds for all lower fidelities, the entropy of $p(y_j^{(m_j)}|D, \mathbf{x}, y_{j_s}^*)$ can also be approximated by the entropy of a truncated gaussian distribution and expressed as follow:

$$H(y_j^{(m_j)}|D, \mathbf{x}, y_j^{(m_j)} \leq y_{j_s}^*) = \frac{(1 + \ln(2\pi))}{2} + \ln(\sigma_j^{(m_j)}(\mathbf{x})) + \ln \Phi(\gamma_s^{(m_j)}(\mathbf{x})) - \frac{\gamma_s^{(m_j)}(\mathbf{x})\phi(\gamma_s^{(m_j)}(\mathbf{x}))}{2\Phi(\gamma_s^{(m_j)}(\mathbf{x}))} \tag{4.14}$$

where $\gamma_s^{(m_j)}(\mathbf{x}) = \frac{y^*_{j_s} - \mu_j^{(m_j)}(\mathbf{x})}{\sigma_j^{(m_j)}(\mathbf{x})}$.

Approximation 2 (MF-OSEMO-NI): Although equation 4.14 is sufficient for computing the entropy for $m_j \neq M_j$, it can be improved by conditioning on a tighter inequality $y_j^{(M_j)} \leq y^*_{j_s}$ as compared to the general one, i.e., $y_j^{(m_j)} \leq y^*_{j_s}$. As we show below, this improvement comes at the expense of not obtaining a final closed-form expression, but it can be efficiently computed via numerical integration. We use the derivation of the entropy based on numerical integration, proposed in (Takeno et al. 2019).

Now, for calculating $H(y_j^{(m_j)}|D, \mathbf{x}, y_j^{(m_j)} \leq y^*_{j_s})$ by replacing $p(y_j^{(m_j)}|D, \mathbf{x}, y_j^{(m_j)} \leq y^*_{j_s})$ with $p(y_j^{(m_j)}|D, \mathbf{x}, y_j^{(M_j)} \leq y^*_{j_s})$ and using Bayes' theorem, we have:

$$p(y_j^{(m_j)}|D, \mathbf{x}, y_j^{(M_j)} \leq y^*_{j_s}) = \frac{p(y_j^{(M_j)} \leq y^*_{j_s}|y_j^{(m_j)}, D, \mathbf{x})p(y_j^{(m_j)}, D, \mathbf{x})}{p(y_j^{(M_j)} \leq y^*_{j_s}|D, \mathbf{x})} \quad (4.15)$$

Both the densities, $p(y_j^{(M_j)} \leq y^*_{j_s}|D, \mathbf{x})$ and $p(y_j^{(m_j)}, D, \mathbf{x})$ can be obtained from the predictive distribution of MF-GP model and is given as follows:

$$p(y_j^{(m_j)}, D, \mathbf{x}) = \frac{\phi(\gamma_j^{(m_j)}(\mathbf{x}))}{\sigma_j^{(m_j)}} \quad (4.16)$$

$$p(y_j^{(M_j)} \leq y^*_{j_s}|D, \mathbf{x}) = \Phi(\gamma_s^{(M_j)}(\mathbf{x}))) \quad (4.17)$$

where $\gamma_j^{(m_j)}(\mathbf{x}) = \frac{y_j^{(m_j)} - \mu_j^{(m_j)}(\mathbf{x})}{\sigma_j^{(m_j)}(\mathbf{x})}$.

Since MF-GP represents all fidelities as one unified Gaussian process, the joint marginal distribution $p(y_j^{(M_j)}, y_j^{(m_j)}|D, \mathbf{x})$ can be immediately obtained from the posterior distribution of the corresponding surrogate model $\mathcal{GP}_j$ as given below:

$$p(y_j^{(M_j)}|y_j^{(m_j)}, \mathbf{x}, D) \sim \mathcal{N}(\mu_j(\mathbf{x}), s_j^2(\mathbf{x})) \quad (4.18)$$

where $\mu_j(\mathbf{x}) = \frac{\sigma_j^{2^{(m_j M_j)}}(\mathbf{x})(y_j^{(m_j)} - \mu_j^{m_j}(\mathbf{x}))}{\sigma_j^{2^{(m_j)}}(\mathbf{x})}$

and $s_j^2(\mathbf{x}) = \sigma_j^{2^{(M_j)}}(\mathbf{x}) - \frac{(\sigma_j^{2^{(m_j M_j)}}(\mathbf{x}))^2}{\sigma_j^{2^{(m_j)}}(\mathbf{x})}$. As a result, $p(y_j^{(M_j)} \leq y^*_{j_s}|y_j^{(m_j)}, D, \mathbf{x})$ is expressed as the cumulative distribution of the Gaussian in 4.18:

$$p(y_j^{(M_j)} \leq y^*_{j_s}|y_j^{(m_j)}, D, \mathbf{x}) = \Phi(\frac{y^*_{j_s} - \mu_j(\mathbf{x})}{s_j(\mathbf{x})}) \quad (4.19)$$

By substituting 4.16, 4.17, and 4.19 into 4.15 we get:

$$H(y_j^{(m_j)}|D, \mathbf{x}, y_j^{(M_j)} \leq y^*_{j_s}) = -\int \Psi(y_j^{(m_j)}) \log(\Psi(y_j^{(m_j)}))dy_j^{(m_j)} \quad (4.20)$$

With $\Psi(y_j^{(m_j)}) = \Phi(\frac{y^*_{j_s} - \mu_j(\mathbf{x})}{s_j(\mathbf{x})})\frac{\Phi(\gamma_s^{(M_j)}(\mathbf{x})))\phi(\gamma_j^{(m_j)}(\mathbf{x}))}{\sigma_j^{(m_j)}}$

Since this integral is over one-dimension variable $y_j^{(m_j)}$, numerical integration can result in a tight approximation.

A complete description of the MF-OSEMO algorithm is given in Algorithm 1. The blue colored steps correspond to computation of our acquisition function via sampling.

5 Experiments and Results

In this section, we describe our experimental setup, and present results of MF-OSEMO and baseline methods.

5.1 Experimental Setup

Baselines. We compare MF-OSEMO with state-of-the-art single-fidelity MO algorithms: ParEGO (Knowles 2006), PESMO (Hernández-Lobato et al. 2016), SMSego (Ponweiser et al. 2008), EHI (Emmerich and Klinkenberg 2008), and SUR (Picheny 2015). We employ the code for these methods from the BO library Spearmint[1].

Statistical models. We use MF-GP models as described in section 4.1. We employ squared exponential (SE) kernels in all our experiments. The hyper-parameters are estimated after every 5 function evaluations. We initialize the MF-GP models for all functions by sampling initial points at random from a Sobol grid. We Initialise each of the lower fidelities with 5 points and the highest fidelity with only one point.

Synthetic benchmarks. We construct two synthetic benchmark problems using a combination of commonly employed benchmark functions for multi-fidelity and single-objective optimization [2], and two of the known general MO benchmarks (Habib, Singh, and Ray 2019). Their complete details are provided in Table 2.

Real-world benchmarks. We consider two challenging problems that are described below.

1) Rocket launching simulation. We consider the simulation study of a rocket (Hasbun 2012) being launched from the Earth's surface. Input variables for simulation are mass of fuel, launch height, and launch angle. Output objectives are the time taken to return to Earth's surface, the angular distance travelled with respect to the centre of the Earth, and the absolute difference between the launch angle and the radius at the point of launch. However, these simulations are computationally expensive and can take up to several hours. The simulator has a tolerance parameter that can be adjusted to perform multi-fidelity simulations: small tolerance means accurate simulations, but long runtime. We employ two tolerance parameter values to create two fidelities for each objective: cost of two fidelities are 0.05 minutes and 30 minutes respectively.

2) Network-on-chip (NOC) optimization. Designing good communication infrastructure is important to improve the quality of hardware designs. This is typically done using cycle-accurate simulators that imitate the real hardware. We consider a design space of NoC dataset consisting of 1024

[1] https://github.com/HIPS/Spearmint/tree/PESM

[2] https://www.sfu.ca/ ssurjano/optimization.html

Algorithm 1 MF-OSEMO Algorithm

Input: input space $\mathfrak{X}$; K blackbox objective functions where each function f_j has multiple fidelities M_j $\Big(\{f_1^{(1)}(\mathbf{x}), \cdots, f_1^{(M_1)}(\mathbf{x})\}, \cdots, \{f_K^{(1)}(\mathbf{x}), \cdots, f_K^{(M_K)}(\mathbf{x})\}\Big)$; and total cost budget λ_{Total}

Initialize multi-fidelity gaussian process surrogate models $\mathcal{GP}_1, \cdots, \mathcal{GP}_K$ by evaluating at initial points D
While $\lambda_t \leq \lambda_{total}$ **do**
  for each sample $s \in 1, \cdots, S$:
    Sample highest-fidelity functions $\tilde{f}_i^{(M_i)} \sim \mathcal{GP}_i, \quad \forall i \in \{1, \cdots, K\}$
    $\mathcal{Y}_s^* \leftarrow$ Pareto front of *cheap* multi-objective optimization over $(\tilde{f}_1^{(M_1)}, \cdots, \tilde{f}_K^{(M_K)})$
  Find the next point to evaluate: select $(\mathbf{x}_t, \mathbf{m}_t) \leftarrow \arg max_{\mathbf{x} \in \mathfrak{X}, \mathbf{m}} \ \alpha_t(\mathbf{x}, \mathbf{m}, \mathcal{Y}^*)$
  Update the total cost consumed: $\lambda_t \leftarrow \lambda_t + \lambda^{(\mathbf{m_t})}$
  Aggregate data: $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{x}_t, \mathbf{y}_t^{\mathbf{m}})\}$
  Update models $\mathcal{GP}_1, \cdots, \mathcal{GP}_K$
  $t \leftarrow t + 1$
end while
return Pareto front and Pareto set of $f_1(x), \cdots, f_K(x)$ based on $\mathcal{D}$
end
Procedure $\alpha_t(\mathbf{x}, \mathbf{m}, \mathcal{Y}_s^*)$
// Computes information gain (IG) about the posterior of true Pareto front $(\mathcal{Y}^*)$ per unit cost as a result of evaluating $\mathbf{x}$
// IG = H_1 - H_2; where H_1 = Entropy of $\mathbf{y}^{(\mathbf{m})}$ conditioned on D and $\mathbf{x}$
// and H_2 = Expected entropy of $\mathbf{y}^{(\mathbf{m})}$ conditioned on D, $\mathbf{x}$ and $(\mathcal{Y}^*)$
Set $H_1 = H(\mathbf{y}^{(\mathbf{m})} \mid D, \mathbf{x}) = K(1 + \ln(2\pi))/2 + \sum_{j=1}^{K} \ln(\sigma_j^{(m_j)}(\mathbf{x}))$ (entropy of K-factorizable Gaussian)
To compute $H_2 \simeq \frac{1}{S} \sum_{s=1}^{S} \sum_{j=1}^{K} H(y_j^{(m_j)} | D, \mathbf{x}, y_{j_s}^*)$, initialize H_2 = 0
for each sample $\mathcal{Y}_s^*$ **do**
  for $j \in 1 \cdots K$ **do**
    Set $y_{j_s}^*$ = maximum of jth component of all vectors in $\mathcal{Y}_s^*$
    If $m_j = M_j$ // if evaluating jth function at highest fidelity
      H_2 += $H(y_j^{(M_j)} | D, \mathbf{x}, y_j^{(M_j)} \leq y_{j_s}^*)$ (entropy of truncated Gaussian $p(y_j^{(M_j)} | D, \mathbf{x}, \underline{y_j^{(M_j)} \leq y_{j_s}^*})$)
    Else if $m_j \neq M_j$ // if evaluating jth function at lower fidelity
      // two approximations are provided
      If approximation = TG
        H_2 += $H(y_j^{(m_j)} | D, \mathbf{x}, y_j^{(m_j)} \leq y_{j_s}^*)$ (entropy of truncated Gaussian $p(y_j^{(M_j)} | D, \mathbf{x}, \underline{y_j^{(m_j)} \leq y_{j_s}^*})$)
      Else If approximation = NI
        H_2 += $H(y_j^{(m_j)} | D, \mathbf{x}, y_j^{(M_j)} \leq y_{j_s}^*)$ (entropy computed via numerical integration)
  end for
end for
Divide by number of samples: H_2 = H_2/S
return $(H_1 - H_2)/\lambda^{(\mathbf{m})}$

implementation of a network-on-chip (Che et al. 2009). Each configuration is defined by ten input variables (d=10). We optimize two objectives: latency and energy. This benchmark has two fidelities with costs 3 mins and 45 mins respectively.

Name	k	d	Benchmark functions	p	Costs
BC	2	2	Branin	2	$[1, 10]$
			Currin	2	$[1, 10]$
SPP	3	4	Shekel	3	$[0.1, 1, 10]$
			Park 1	2	$[1, 10]$
			Park 2	2	$[1, 10]$
ZDT3	2	6	Zitzler,Deb,Thiele	2^2	$[1, 10]^2$
DTLZ1	6	5	Deb,Thiele,Laumanns,Zitzler	3^6	$[0.1, 1, 10]^6$

Table 2: Details of synthetic benchmarks: Name, benchmark functions, no. of objectives k, input dimension d, number fidelities p, and costs of different fidelities for each function.

5.2 Results and Discussion

To evaluate the performance of MF-OSEMO, we employ a common multi-objective metric used in practice. The *Pareto hypervolume (PHV)* metric measures the quality of a given Pareto front (Zitzler 1999). PHV is defined as the volume between a reference point and the given Pareto front (set of non-dominated points). As a function of the cost of evaluations, we report the difference between the hypervolume of the ideal Pareto front $(\mathcal{Y}^*)$ and hypervolume of the best reached Pareto front estimated by optimizing the posterior mean of the models at the highest fidelities (Hernández-Lobato et al. 2016). The posterior means are optimized over a randomly generated grid of 10,000 points. We also provide the *cost reduction factor*, which is the ratio between the worst cost at which MF-OSEMO converges (worst case for MF-OSEMO), and the earliest cost for which any of the single-fidelity baselines converge (best case for baseline) after running all algorithms for very large costs. We run all experiments 10 times. The

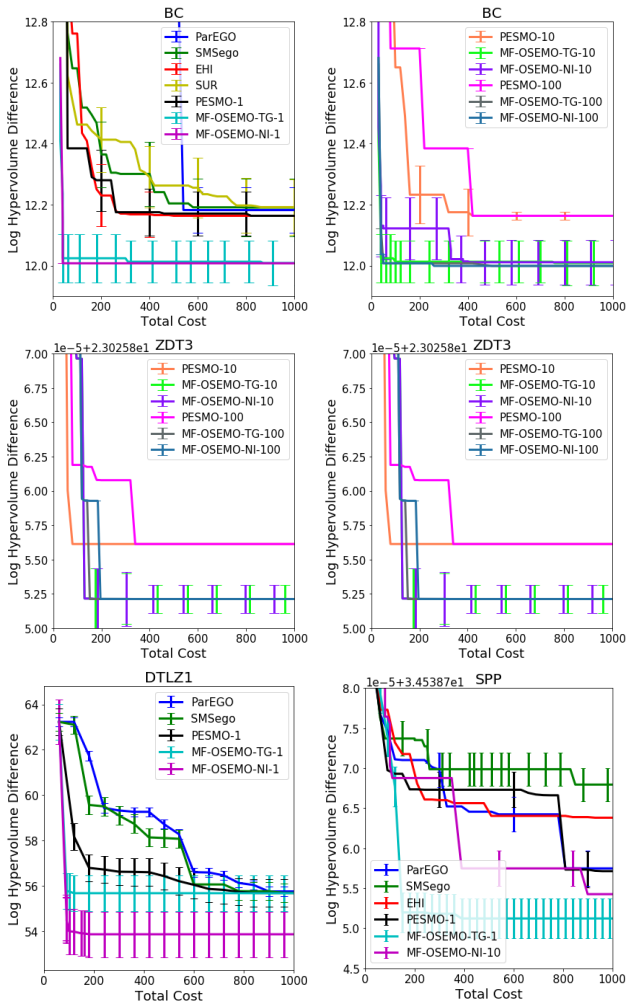

Figure 3: Results of MF-OSEMO and single-fidelity multi-objective BO algorithms on synthetic benchmarks. The log of the hypervolume difference is shown with varying cost.

mean and variance of the PHV metrics across different runs are reported as a function of the total cost consumed. Since in all our experiments, the costs of different functions are on the same scale, we plot results against the sum of the costs.

MF-OSEMO vs. State-of-the-art. We compare the performance of MF-OSEMO-TG and MF-OSEMO-NI with single-fidelity MO methods. Figure 3 and Figure 4 show the results of all multi-objective BO algorithms including MF-OSEMO for synthetic and real-world benchmarks respectively. We observe that: 1) MF-OSEMO consistently performs better than all baselines. Both the variants of MF-OSEMO converge at a much lower cost. 2) Rates of convergence of MF-OSEMO-TG and MF-OSEMO-NI are slightly different. However, in all cases, MF-OSEMO performs better than baseline methods. We notice that in few cases (e.g., both real-world benchmarks), MF-OSEMO-TG converges earlier than MF-OSEMO-NI. This demonstrates that even with loose approximation, using the MF-OSEMO-TG can provide consistently competitive results using less computation time.

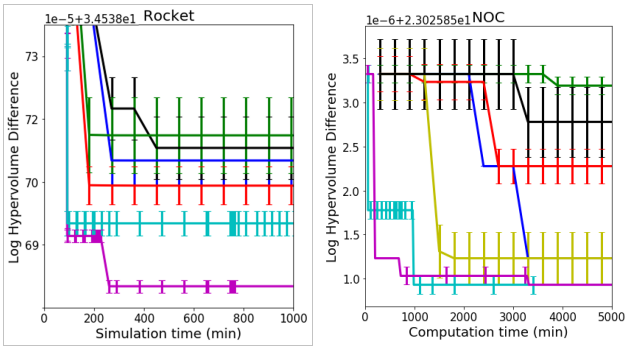

Figure 4: Results of MF-OSEMO and single-fidelity multi-objective BO algorithms on real-world problems. The log of the hypervolume difference is shown with varying cost.

Name	BC	SPP	ZDT3	DTLZ1	Rocket	NOC
λ	4.2	190	380	100	250	1200
λ_B	2000	1950	2000	800	4000	10000
Λ	99.79%	90.25%	81%	87.5%	93.75%	88%

Table 3: Convergence costs for MF-OSEMO and baselines, and cost reduction factor achieved by MF-OSEMO: *worst* convergence cost for MF-OSEMO λ, *best* convergence cost from all baselines methods λ_B, and cost reduction factor Λ.

Cost reduction factor. Some of the baselines will eventually converge if they are run for a much larger cost. In table 3, we provide the cost reduction factor to show the percentage of cost-gain achieved by using MF-OSEMO when compared to single-fidelity baselines. Although the metric gives advantage to baselines, the results in the table show a consistently high gain ranging from 81% to 99.8%.

Robustness of MF-OSEMO. We evaluate the performance of MF-OSEMO and PESMO with different number of Monte-Carlo samples (MCS). We provide results for two synthetic benchmarks BC and ZDT3 in figure 3 with 1, 10, and 100 MCS for PESMO, MF-OSEMO-TG, and MF-OSEMO-NI. For clarity of figures, we provided those results in two diffrent figures side by side. We notice that the convergence rate of PESMO is dramatically affected by the number of Monte-Carlo samples: 100 samples lead to better results than 10 and 1. However, MF-OSEMO-TG and MF-OSEMO-NI maintain a better performance consistently even with a single sample. These results strongly demonstrate that our proposed method is much more robust to the number of MCS.

6 Summary and Future Work

We introduced a novel and principled approach referred as MF-OSEMO to solve multi-fidelity multi-objective Bayesian optimization problems. The key idea is to employ an output space entropy based acquisition function to efficiently select inputs and fidelity vectors for evaluation. Our experimental results on both synthetic and real-world benchmarks showed that MF-OSEMO yields consistently better results than state-of-the-art single-fidelity methods. Immediate future work will

be to apply MF-OSEMO to novel real-world applications.

Acknowledgements. This research is supported by National Science Foundation grants IIS-1845922 and OAC-1910213.

References

[Ariyarit and Kanazaki 2017] Ariyarit, A., and Kanazaki, M. 2017. Multi-fidelity multi-objective efficient global optimization applied to airfoil design problems. *Applied Sciences* 7(12):1318.

[Belakaria et al. 2020] Belakaria, S.; Deshwal, A.; Jayakodi, N. K.; and Doppa, J. R. 2020. Uncertainty-aware search framework for multi-objective bayesian optimization. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[Belakaria, Deshwal, and Doppa 2019] Belakaria, S.; Deshwal, A.; and Doppa, J. 2019. Max-value entropy search for multi-objective bayesian optimization. In *Proceedings of International Conference on Neural Information Processing Systems (NeurIPS)*.

[Che et al. 2009] Che, S.; Boyer, M.; Meng, J.; Tarjan, D.; Sheaffer, J. W.; Lee, S.; and Skadron, K. 2009. Rodinia: A benchmark suite for heterogeneous computing. In *Proceedings of the 2009 IEEE International Symposium on Workload Characterization (IISWC)*, 44–54.

[Cover and Thomas 2012] Cover, T. M., and Thomas, J. A. 2012. *Elements of information theory*. John Wiley and Sons.

[Deb et al. 2002] Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T.; and Fast, A. 2002. Nsga-ii. *IEEE Transactions on Evolutionary Computation* 6(2):182–197.

[Emmerich and Klinkenberg 2008] Emmerich, M., and Klinkenberg, J.-w. 2008. The computation of the expected improvement in dominated hypervolume of pareto front approximations. *Technical Report, LU* 34.

[Habib, Singh, and Ray 2019] Habib, A.; Singh, H. K.; and Ray, T. 2019. A multiple surrogate assisted multi/many-objective multi-fidelity evolutionary algorithm. *Information Sciences*.

[Hasbun 2012] Hasbun, J. E. 2012. *Classical mechanics with MATLAB applications*. Jones & Bartlett Publishers.

[Hennig and Schuler 2012] Hennig, P., and Schuler, C. J. 2012. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research (JMLR)* 13(Jun):1809–1837.

[Hernández-Lobato et al. 2016] Hernández-Lobato, D.; Hernandez-Lobato, J.; Shah, A.; and Adams, R. 2016. Predictive entropy search for multi-objective bayesian optimization. In *Proceedings of International Conference on Machine Learning (ICML)*, 1492–1501.

[Hernández-Lobato, Hoffman, and Ghahramani 2014] Hernández-Lobato, J. M.; Hoffman, M. W.; and Ghahramani, Z. 2014. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, 918–926.

[Hoffman and Ghahramani 2015] Hoffman, M. W., and Ghahramani, Z. 2015. Output-space predictive entropy search for flexible global optimization. In *NIPS workshop on Bayesian Optimization*.

[Huang et al. 2006] Huang, D.; Allen, T. T.; Notz, W. I.; and Miller, R. A. 2006. Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization* 32(5):369–382.

[Jiang et al. 2017] Jiang, S.; Malkomes, G.; Converse, G.; Shofner, A.; Moseley, B.; and Garnett, R. 2017. Efficient nonmyopic active search. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1714–1723. JMLR. org.

[Kandasamy et al. 2016] Kandasamy, K.; Dasarathy, G.; Oliva, J. B.; Schneider, J.; and Póczos, B. 2016. Gaussian process bandit optimisation with multi-fidelity evaluations. In *Advances in Neural Information Processing Systems*, 992–1000.

[Kandasamy et al. 2017] Kandasamy, K.; Dasarathy, G.; Schneider, J.; and Poczos, B. 2017. Multi-fidelity bayesian optimisation with continuous approximations. *ICML*.

[Kennedy and O'Hagan 2000] Kennedy, M. C., and O'Hagan, A. 2000. Predicting the output from a complex computer code when fast approximations are available. *Biometrika* 87(1):1–13.

[Klein et al. 2017] Klein, A.; Falkner, S.; Bartels, S.; Hennig, P.; and Hutter, F. 2017. Fast bayesian optimization of machine learning hyperparameters on large datasets. In *Artificial Intelligence and Statistics*, 528–536.

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

[Kontogiannis et al. 2018] Kontogiannis, S. G.; Demange, J.; Kipouros, T.; and Savill, A. M. 2018. A comparison study of two multifidelity methods for aerodynamic optimization. In *Structures, Structural Dynamics, and Materials Conference*, 0415.

[Lam, Allaire, and Willcox 2015] Lam, R.; Allaire, D. L.; and Willcox, K. E. 2015. Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 0143.

[McLeod, Osborne, and Roberts 2017] McLeod, M.; Osborne, M. A.; and Roberts, S. J. 2017. Practical bayesian optimization for variable cost objectives. *arXiv preprint arXiv:1703.04335*.

[Michalowicz, Nichols, and Bucholtz 2013] Michalowicz, J. V.; Nichols, J. M.; and Bucholtz, F. 2013. *Handbook of differential entropy*. Chapman and Hall/CRC.

[Picheny et al. 2013] Picheny, V.; Ginsbourger, D.; Richet, Y.; and Caplin, G. 2013. Quantile-based optimization of noisy computer experiments with tunable precision. *Technometrics* 55(1):2–13.

[Picheny 2015] Picheny, V. 2015. Multi-objective optimization using gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing* 25(6):1265–1280.

[Ponweiser et al. 2008] Ponweiser, W.; Wagner, T.; Biermann, D.; and Vincze, M. 2008. Multiobjective optimization on a limited budget of evaluations using model-assisted s-metric

selection. In *International Conference on Parallel Problem Solving from Nature*, 784–794. Springer.

[Rahimi and Recht 2008] Rahimi, A., and Recht, B. 2008. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 1177–1184.

[Shahriari et al. 2016] Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P.; and De Freitas, N. 2016. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE* 104(1):148–175.

[Song, Chen, and Yue 2019] Song, J.; Chen, Y.; and Yue, Y. 2019. A general framework for multi-fidelity bayesian optimization with gaussian processes. *AISTATS*.

[Srinivas et al. 2009] Srinivas, N.; Krause, A.; Kakade, S. M.; and Seeger, M. 2009. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*.

[Swersky, Snoek, and Adams 2013] Swersky, K.; Snoek, J.; and Adams, R. P. 2013. Multi-task bayesian optimization. In *Advances in neural information processing systems*, 2004–2012.

[Takeno et al. 2019] Takeno, S.; Fukuoka, H.; Tsukada, Y.; Koyama, T.; Shiga, M.; Takeuchi, I.; and Karasuyama, M. 2019. Multi-fidelity bayesian optimization with max-value entropy search. *arXiv preprint arXiv:1901.08275*.

[Wang and Jegelka 2017] Wang, Z., and Jegelka, S. 2017. Max-value entropy search for efficient bayesian optimization. In *Proceedings of International Conference on Machine Learning (ICML)*.

[Zhang et al. 2017] Zhang, Y.; Hoang, T. N.; Low, B. K. H.; and Kankanhalli, M. 2017. Information-based multi-fidelity bayesian optimization. In *NIPS Workshop on Bayesian Optimization*.

[Zitzler 1999] Zitzler, E. 1999. *Evolutionary algorithms for multiobjective optimization: Methods and applications*, volume 63. Citeseer.

[Zuluaga et al. 2013] Zuluaga, M.; Sergent, G.; Krause, A.; and Püschel, M. 2013. Active learning for multi-objective optimization. In *Proceedings of International Conference on Machine Learning (ICML)*, 462–470.