IOP Publishing Mach. Learn.: Sci. Technol. 5 (2024) 015056 https://doi.org/10.1088/2632-2153/ad35a4

MACHINE LEARNING
Science and Technology

PAPER

Leveraging trust for joint multi-objective and multi-fidelity optimization

Faran Irshad, Stefan Karsch and Andreas Döpp*

Faculty of Physics, Ludwig Maximilian University of Munich, Am Coulombwall 1, 85748 Garching, Germany
* Author to whom any correspondence should be addressed.

E-mail: a.doepp@lmu.de

Keywords: Bayesian optimization, multi-objective optimization, computational physics

OPEN ACCESS

RECEIVED
27 June 2023

REVISED
2 February 2024

ACCEPTED FOR PUBLICATION
19 March 2024

PUBLISHED
27 March 2024

Abstract

In the pursuit of efficient optimization of expensive-to-evaluate systems, this paper investigates a novel approach to Bayesian multi-objective and multi-fidelity (MOMF) optimization. Traditional optimization methods, while effective, often encounter prohibitively high costs in multi-dimensional optimizations of one or more objectives. Multi-fidelity approaches offer potential remedies by utilizing multiple, less costly information sources, such as low-resolution approximations in numerical simulations. However, integrating these two strategies presents a significant challenge. We propose the innovative use of a trust metric to facilitate the joint optimization of multiple objectives and data sources. Our methodology introduces a modified multi-objective (MO) optimization policy incorporating the trust gain per evaluation cost as one of the objectives of a Pareto optimization problem. This modification enables simultaneous MOMF optimization, which proves effective in establishing the Pareto set and front at a fraction of the cost. Two specific methods of MOMF optimization are presented and compared: a holistic approach selecting both the input parameters and the fidelity parameter jointly, and a sequential approach for benchmarking. Through benchmarks on synthetic test functions, our novel approach is shown to yield significant cost reductions—up to an order of magnitude compared to pure MO optimization. Furthermore, we find that joint optimization of the trust and objective domains outperforms sequentially addressing them. We validate our findings with the specific use case of optimizing particle-in-cell simulations of laser-plasma acceleration, highlighting the practical potential of our method in the Pareto optimization of highly expensive black-box functions. Implementation of the methods in existing Bayesian optimization frameworks is straightforward, with immediate extensions e.g. to batch optimization possible. Given their ability to handle various continuous or discrete fidelity dimensions, these techniques have wide-ranging applicability in tackling simulation challenges across various scientific computing fields such as plasma physics and fluid dynamics.

1. Introduction

Black-box functions are systems where only the input-output relationship is considered without knowledge of the internal workings. Optimization of such systems is crucial in various fields such as computer science, engineering, and physics [1–6].

There exist various properties that can make black-box functions very difficult to optimize, such as the well-known 'curse of dimensionality,' which makes optimization increasingly difficult as the number of input dimensions grows. Moreover, many optimization problems involve multiple objectives, which can be challenging to define or even conflicting in nature. These kind of multi-objective (MO) optimization problems arise in diverse research domains [7] including, but not restricted to, the design of photonic crystal filters [8], aerospace design problems [9] and robotics [10]. MO problems can be solved by directly examining different trade-offs between various objectives and selecting an optimal combination that leads to

the desired outcome. This concept is embodied by Pareto efficiency, visualized as the Pareto front, which represents a set of optimal trade-offs between competing objectives. Finding this Pareto front is usually referred to as *MO* optimization and various techniques have been developed to tackle them [11]. Among the most popular are techniques based on evolutionary algorithms [12] and, increasingly in recent years, Bayesian optimization [13]. While evolutionary algorithms focus on the efficient combinations of successful ('fit') parameters, Bayesian techniques try to approximate the unknown black box function with fast surrogate models for its mean and variance. Incorporating multiple information sources into the optimization process can further improve efficiency, especially when dealing with costly black-box functions. Such information sources may have different levels of 'fidelity', an example being simulations with different numerical resolutions that approximate underlying physical processes. *Multi-fidelity* (MF) optimization can leverage information from different sources, each with varying levels of fidelity, accuracy, or computational cost [14]. Using this information, MF optimization can provide more informed decision-making while reducing the overall cost.

In this article, we detail an approach that expands the scope of MO optimization to encompass varied fidelity levels, allowing us to build the Pareto front more economically using cost-efficient approximations of the core black box function. Central to our method is the inclusion of an objective that embodies the information content-termed 'trust'-which can be seamlessly optimized alongside other objectives using proven numerical strategies from the realm of MO optimization. The manuscript is structured as follows: after an introduction to Bayesian optimization via Gaussian process regression (2.1), we discuss common acquisition functions used for single-objective, single-fidelity optimization (2.2). We then outline how these policies are generalized to either MF (2.3) or MO (2.4) optimization. In section 3 we use these concepts to devise policies to perform combined *multi-objective, multi-fidelity* (MOMF) optimization, either in a joint fashion via trust (3.1) or in a sequential manner (3.2). We benchmark our results for different test functions to understand the behavior of proposed multi-objective MF algorithms (3.3). Lastly, we apply our trust-based MOMF (Trust-MOMF) algorithm to a real-world application. The example is taken from computational physics where simulations are optimized (3.4). In section 4 we summarize our results and give an outlook on potential applications.

2. Background and related work

In this section, we provide an overview of the key components of Bayesian optimization, including Gaussian processes (GPs) and acquisition functions, and discuss their extension to multi-objective or MF optimization [15]. The models underlying most Bayesian optimization are GPs, which explicitly define a probability distribution over possible function values across the input space and describe a black-box objective function as a distribution over functions in a continuous domain [16]. GPs consist of a mean function, which represents the average value of the objective function, and a kernel function, which captures the relationship between different points in the parameter space [17]. Prior information about the function can be incorporated either in the mean function or the kernel [18]. In many cases the prior mean is assumed to be zero throughout the parameter space and all the prior information is encoded in the kernel function. As the black-box function is evaluated at specific positions, the GP is updated based on these evaluations. In essence, the GP combines the prior information about the objective function with the evaluated data resulting in a so-called posterior distribution that also defines a probability distribution over function values. A function usually denoted as an acquisition function uses this posterior distribution to predict the next possible best position to evaluate the objective function. By optimizing the acquisition function, which is computationally efficient, the problem of optimizing the expensive black-box function is transformed into optimizing the acquisition function [19].

2.1. Acquisition functions for single-objective, single-fidelity optimization

Acquisition functions encompass a broad class of functions that can be constructed using the posterior distribution of the GP model. The effectiveness of an acquisition function is assessed based on its ability to converge to the global optimum, with minimal evaluations of the objective function. This can be best illustrated through a series of diagrams shown in figure 1, depicting the iterations of Bayesian optimization loop starting from three initial points. After constructing a GP model with a mean (solid blue line) and variance (shaded region), an acquisition function is derived (bottom plot). This acquisition function is then optimized, often using gradient methods, to identify its maximum, which corresponds to the next evaluation point for the black-box objective function. The choice of the acquisition function significantly impacts the convergence of Bayesian optimization towards the global optimum. This section provides an overview of some widely recognized acquisition function policies discussed in the existing literature. We would like to

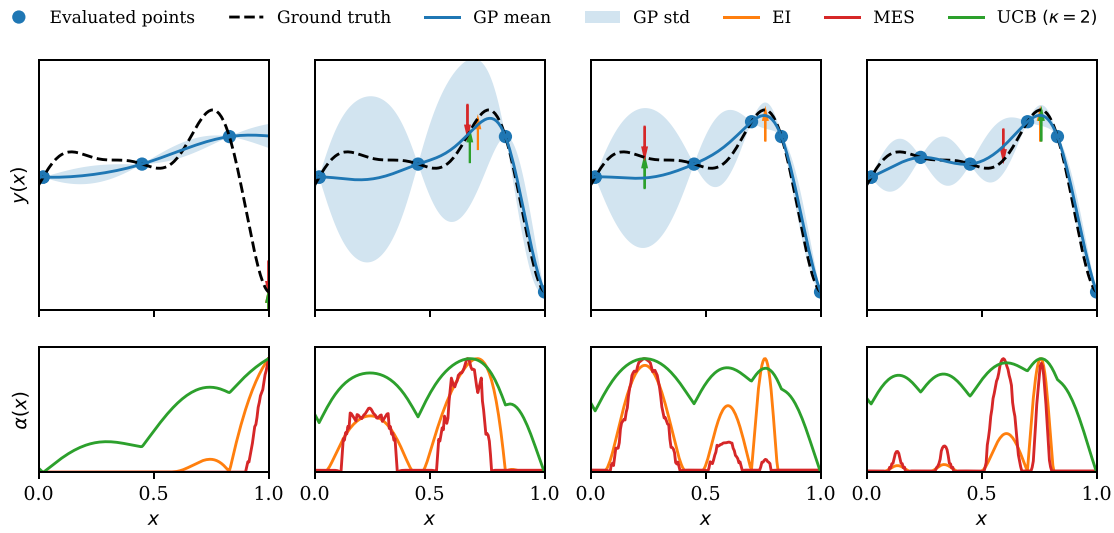

Figure 1. Consecutive iterations of Bayesian optimization (maximization) of the Forrester test function defined in appendix. *Top*: the dashed black line represents the true function that is being optimized. The blue line is the GP regression mean with a shaded region around it representing the standard deviation of the probability distribution for the function value. The blue dots are the points that were evaluated. Arrows indicate the next measurement point suggested by the respective acquisition functions from the bottom plot, showing the different prioritization towards exploration and exploitation. *Bottom*: evaluations of three different metrics, expected improvement (EI, equation (2)), max-value entropy search (MES, equation (5).) and upper confidence bound (UCB, equation (1) with $\kappa = 2$) for each step in the upper plot.

note that the development of acquisition functions is still an active area of research, with some recent contributions using for instance distance correlation [20] or deep neural networks [21].

Assuming that the objective function has been evaluated n times generating a data set $D_n = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$, these acquisition functions propose the optimal choice for the subsequent evaluation point, denoted as $\boldsymbol{x}_{n+1}$. Please note that $\boldsymbol{x}$ is a vector of input parameters while the function output y is a single scalar value for this section. This selection is achieved through the optimization of a derived metric, taking into account the information from previously evaluated points. Specifically, the next evaluation point is given by $\boldsymbol{x}_{n+1} = \arg\max_{\boldsymbol{x}\in\mathcal{X}} A(\boldsymbol{x}|D_n)$, where $\mathcal{X}$ is the input parameter domain and $A(\boldsymbol{x}|D_n)$ represents one of the acquisition functions given the dataset D_n detailed below.

One widely used acquisition function is the upper confidence bound (UCB) policy [22], which is alternatively referred to as the lower confidence bound for minimization tasks. The UCB acquisition function is expressed as:

$$\mathrm{UCB}\left(\boldsymbol{x}|D_n\right) = \mu_n\left(\boldsymbol{x}\right) + \kappa\sigma_n\left(\boldsymbol{x}\right), \tag{1}$$

where $\mu_n(\boldsymbol{x})$ and $\sigma_n(\boldsymbol{x})$ denote the mean and the standard deviation of the function value under the posterior distribution $p(y|\boldsymbol{x}, D_n)$, respectively, derived from GPs. The hyperparameter κ is used to balance exploration and exploitation. UCB is popular due to its simplicity and effectiveness in practice.

Another well-known acquisition function in Bayesian optimization is expected improvement (EI) [23], which suggests the next evaluation point based on the EI over the current optimal objective value y^*. The EI acquisition function is expressed as:

$$\mathrm{EI}\left(\boldsymbol{x}|D_n\right) = \mathbb{E}\left[\max\left(f_{n+1}\left(\boldsymbol{x}\right) - y^*, 0\right)\right], \tag{2}$$

where, $\max(f_{n+1}(\boldsymbol{x})$ is drawn from the posterior distribution and $\mathbb{E}$ denotes the expectation symbol, representing an average over the probability distribution defined by the Gaussian process (GP) model. A variation of EI is the knowledge gradient (KG) acquisition function [23, 24], which relies entirely on the posterior model. KG selects the next evaluation point based not on the best observable value but on the best value of the posterior mean. The KG acquisition function is given by:

$$\mathrm{KG}\left(\boldsymbol{x}|D_n\right) = \mathbb{E}\left[\max_{\boldsymbol{x}'\in\mathcal{X}}\left(\mu_{n+1}\left(\boldsymbol{x}'\right)\right)\right] - \max_{\boldsymbol{x}'\in\mathcal{X}}\left(\mu_n\left(\boldsymbol{x}'\right)\right), \tag{3}$$

where the terms $\mu_{n+1}(\boldsymbol{x})$ and $\mu_n(\boldsymbol{x})$ represent the posterior mean of the GP after $n+1$ and n evaluations, respectively. The term $\mu_{n+1}(\boldsymbol{x})$ represents the posterior mean of the GP after including an additional data point $(\boldsymbol{x}, f(\boldsymbol{x}))$, where $f(\boldsymbol{x})$ is the realization of the stochastic process at location $\boldsymbol{x}$. This captures the updated

belief about the function based on the new data point. The term $\mu_n(\boldsymbol{x})$ represents the prior belief before the new data point is included. KG is more exploratory than EI as it is influenced by posterior changes throughout the domain.

Information-theoretic acquisition functions utilize the mutual information $I(\boldsymbol{x};\boldsymbol{x}^*)$ between a specific location in the parameter domain and the observed data set. One such function is Entropy Search (ES) [25], represented mathematically as:

$$\mathrm{ES}(\boldsymbol{x}|D_n) = H(p(\boldsymbol{x}^*|D_n)) - \mathbb{E}(H(p(\boldsymbol{x}^*|D_n,\boldsymbol{x},f(\boldsymbol{x})))), \tag{4}$$

where H denotes Shannon's entropy and $p(\boldsymbol{x}^*|D_n)$ refers to the probability distribution for the position of the maximum given currently observed data. Moreover, $p(\boldsymbol{x}^*|D_n,\boldsymbol{x},f(\boldsymbol{x}))$ is the probability distribution for the position of the maximum given data D_n as well as $\boldsymbol{x}$ and $f(\boldsymbol{x})$, where $f(\boldsymbol{x})$ is drawn from the GP trained on D_n. The left term in the equation represents the entropy of the posterior distribution of the maximizing location $\boldsymbol{x}^*$, while the right term depicts an expectation over the entropy of the posterior after an additional sample. In higher-dimensional input spaces, evaluating the mutual information between the point to be queried and the maximizing location $\boldsymbol{x}^*$ becomes challenging. To address this, computationally more efficient variations have been introduced. MES [26] utilizes the mutual information between the maximum value y^* rather than $\boldsymbol{x}^*$. The MES acquisition function is formulated as:

$$\mathrm{MES}(\boldsymbol{x}|D_n) = H(p(\boldsymbol{y}^*|D_n)) - \mathbb{E}(H(p(\boldsymbol{y}^*|D_n,\boldsymbol{x},f(\boldsymbol{x})))). \tag{5}$$

MES offers comparable or even better performance than ES while being significantly faster to compute [26]. It should be noted that MES may perform sub-optimally in certain conditions since it assumes noiseless observations only [27].

2.2. MO, single-fidelity optimization

As we outlined in the introduction, there exist many use cases in which the objective consists of multiple sub-goals. One can think of these sub-goals as a vector of solutions, which has to be reduced to a scalar number to be compatible with conventional numerical optimization schemes. This reduction, or scalarization, is often not straightforward and the weights given to each sub-goal are usually found empirically. A potent strategy for such use cases is MO optimization, where one tries to increase the diversity of solutions. In this case, the function that is being maximized can be expressed as a vector of functions

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} f^{(1)}(\boldsymbol{x}) \\ f^{(2)}(\boldsymbol{x}) \\ \dots \end{pmatrix}$$

that are evaluated to yield output vectors

$$\boldsymbol{y}(\boldsymbol{x}) = \begin{pmatrix} y^{(1)}(\boldsymbol{x}) \\ y^{(2)}(\boldsymbol{x}) \\ \dots \end{pmatrix}.$$

The optimum solution in this case consists of a solution vector. A useful concept describing this situation of not just a single optimum but a set of optimal points in the multi-dimensional objective space is the *Pareto efficiency* [11], which is visualized as the *Pareto front* ($\mathcal{P}$). To describe the Pareto front, the notion of *domination* is outlined in definition 1. The Pareto front is composed of a set of *non-dominated* points in the output space as shown in figure 2.

One of the simplest algorithms for MO optimization is Pareto Efficient Global Optimization (ParEGO) proposed by Knowles [30]. ParEGO is a scalarization-based approach in which a single-objective optimizer is employed to find the Pareto front. In this algorithm, scalarization is achieved using a weighted sum of the objective functions with random weights generated at each iteration. The main advantage of ParEGO over other scalarization methods is that the random weights allow for exploring the Pareto front for a diverse set of solutions without predefining the weights or relying on user preferences. By iteratively running this algorithm with new weights, it becomes possible to approximate the entire Pareto optimal set.

Alternatively, one can reframe the MO problem as a single-objective optimization by using a new objective that implicitly increases solution diversity. One such criterion is the Hypervolume (HV) Indicator. HV is defined as the n-dimensional volume of the output subspace covered from a reference point, always taken to be zero in this work, to a set of points in the objective space. Using this definition of HV, we can then proceed to define Hypervolume Improvement (HVI) as

$$\mathrm{HVI}(\mathcal{P},\boldsymbol{y}) = \mathrm{HV}(\mathcal{P}\cup\boldsymbol{y}) - \mathrm{HV}(\mathcal{P}). \tag{6}$$

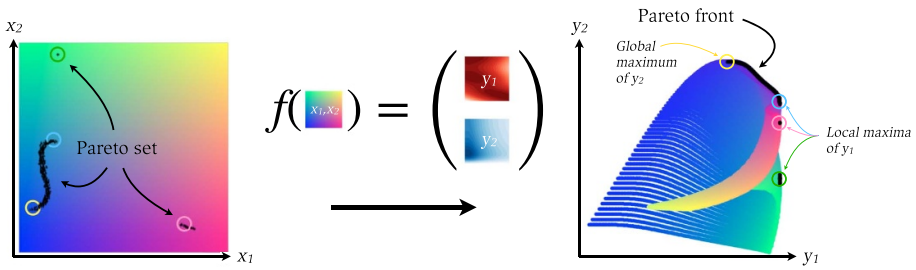

Figure 2. Pareto front. Illustration of how a multi-objective function $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{y}$ acts on a two-dimensional input space $\boldsymbol{x} = (x_1, x_2)$ and transforms it to the objective space $\boldsymbol{y} = (y_1, y_2)$ on the right. The entirety of possible input positions is uniquely color-coded on the left and the resulting position in the objective space is shown in the same color on the right. The Pareto front is the ensemble of points that dominate others, meaning points that give the highest combination of y_1 and y_2. The corresponding set of coordinates in the input space is called the Pareto set. Note that both the Pareto front and the Pareto set may be continuously defined locally, but can also contain discontinuities when local maxima get involved. In this example, f is a modified version of the Branin–Currin function from [28, 29] that exhibits a single, global maximum in y_2 but multiple local maxima in y_1, see also illustration in the center.

Equation (6) describes the difference between the current HV and one with an additional output point $\boldsymbol{y}$ [31]. If the set of points making up $\mathcal{P}$ already dominate $\boldsymbol{y}$ then HVI = 0, because there is no HV gained by adding the point $\boldsymbol{y}$.

HVI can be used to generalize the EI policy described in section 1.2 to the MO scenario. First proposed by Emmerich *et al* [32], this method is called *Expected Hypervolume Improvement (EHVI)*. Following the definition from Yang *et al* [31], we can write this as

$$\mathrm{EHVI}(\boldsymbol{x}|D_n) = \mathbb{E}(\mathrm{HVI}(\mathcal{P}, \boldsymbol{y})) = \int \mathrm{HVI}(\mathcal{P}, \boldsymbol{y})\; p(\boldsymbol{y}|\boldsymbol{x}, D_n)\; \mathrm{d}y. \tag{7}$$

This infill criterion has been demonstrated to achieve a good convergence to the true Pareto front [33–35].

A common criticism of the EHVI acquisition function has been the time complexity involved in calculating it. A first closed-form calculation of EHVI was implemented by Emmerich *et al* [36] with a computational complexity $\mathcal{O}(n^3 \log n)$ for a 2D output space. Over the years with efforts by Hupkens *et al* [37], Emmerich *et al* [38] and Yang *et al* [39] the time complexity for 2D and 3D case has been reduced to $\mathcal{O}(n \log n)$. In this work an implementation of EHVI available on BoTorch based on estimating gradients using auto-differentiation is used as described by Daulton *et al* [40]. This exploits the high number of cores that are available with modern GPUs to make EHVI optimization fast and applicable to real-world scenarios. While we have focused our discussion on EHVI, it should noted that information theoretic acquisition functions such as entropy search can also be adapted to MO scenarios [41, 42].

2.3. Single-Objective, MF optimization

In many real-world optimization problems, multiple information sources with varying degrees of fidelity are available. These sources can provide different levels of accuracy, typically with a trade-off between data fidelity and cost. Integrating such MF data into single-objective optimization algorithms is an essential way to improve the optimization process [14, 43–47].

Let us denote the input search parameters as $\boldsymbol{x} \in \mathcal{X}$ and the fidelity parameters as $\boldsymbol{s} \in \mathcal{S}$ where $\mathcal{X}$ and $\mathcal{S}$ are the input and fidelity spaces respectively. Importantly, in our GP model, fidelity $\boldsymbol{s}$ is treated as a regular input dimension alongside $\boldsymbol{x}$. This means that fidelity $\boldsymbol{s}$ is accounted for in the kernel of the GP, allowing us to make probabilistic inferences conditioned on both $\boldsymbol{x}$ and $\boldsymbol{s}$. For more details on the choice of GP kernels, the reader is referred to appendix. The goal is to build a surrogate model that incorporates information from $f(\boldsymbol{x}, \boldsymbol{s})$. The challenge in MF optimization is to effectively balance the trade-off between information and cost while ultimately finding a global maximum at the target fidelity. This target fidelity usually corresponds to the highest fidelity which is also the most expensive information source.

One intuitive solution to this problem is the two-step approach proposed by Lam *et al* [48]. In this approach, the selection of the next point to probe in $\boldsymbol{x}$ is done separately from the fidelity choice $\boldsymbol{s}$. To achieve this, Lam *et al* use an EI policy that is evaluated at the target fidelity only. Lower fidelity measurements are implicitly incorporated into this process, as they affect the surrogate model at the highest fidelity. Once a suitable position has been identified in $\boldsymbol{x}$, the ideal fidelity for probing this point is chosen by comparing the predicted reduction in uncertainty or gain in knowledge with the computational cost involved.

An alternative approach is to combine both the selection of the next point and the weighting by the expected knowledge gain per unit cost in a single step. Notably, MES and KG acquisition functions can be

adapted with minor changes for this kind of MF optimization. For KG [49, 50], the acquisition function is conditioned on the best value of the posterior mean at the target fidelity $\boldsymbol{s}^*$ ($\max_{\boldsymbol{x}\in\mathcal{X}}(f(\boldsymbol{x},\boldsymbol{s}^*))$) rather than the best value of the posterior mean. In the case of MES, the mutual information between the maximum value y^* at the highest fidelity and the data set is maximized. This gain of information is then divided by the computational cost that is a function of fidelity $\boldsymbol{s}$ [51]. The resulting MF MES acquisition function is then given by a simple modification to equation (5):

$$\mathrm{MF{-}MES}\left(\boldsymbol{x},\boldsymbol{s}|D_n\right)=\frac{\mathrm{MES}\left(\boldsymbol{x}|D_n\right)}{\mathrm{cost}\left(\boldsymbol{s}\right)}. \tag{8}$$

Similar MF policies can be developed for other exploratory acquisition functions, such as the UCB [52, 53].

3. Bayesian MOMF optimization

So far we have reviewed established techniques for MOMF optimization. However, these methods are usually applied in isolation and, as we have seen, MO optimization seeks to solve problems where the objective consists of several sub-objectives, whereas MF optimization leverages data from multiple information sources with varying fidelity to optimize a single objective function. Both techniques offer unique benefits in their respective areas, but there are many real-world problems where the integration of both techniques would be beneficial. This leads us to the idea of joint MOMF optimization, a method that incorporates both MOMF aspects into a single optimization framework. Several compelling reasons motivate the pursuit of such an integrated optimization approach:

(i) **Efficiency:** joint MOMF optimization efficiently leverages the strengths of both MOMF optimization. The MO aspect ensures diverse solution coverage, thoroughly exploring different areas of the solution space. Concurrently, MF optimization minimizes the necessity of costly high-fidelity evaluations by using cheaper, lower-fidelity data sources where possible. This dual approach leads to a more effective balance between exploration and exploitation, thereby improving the overall optimization process.

(ii) **Feasibility:** the complexity and cost of evaluating multiple objectives can often limit the feasibility of MO optimization, particularly in real-world scenarios where such evaluations may involve physical experiments or high-fidelity simulations. By incorporating MF optimization into the process, the MOMF approach leverages lower-fidelity, less costly models during the exploration phases. This considerably reduces the need for expensive high-fidelity evaluations, making the optimization process more manageable and feasible, even for scenarios where traditional MO optimization would be prohibitively expensive or time-consuming.

(iii) **Robustness:** the joint approach enhances the robustness of the optimization process by mitigating the risk of over-reliance on low-fidelity data. This risk is prevalent in single-objective MF optimization and could introduce bias if the lower-fidelity models are inaccurate. By incorporating multiple objectives into the optimization process, the search strategy becomes more diversified, leading to a more robust and reliable optimization outcome.

(iv) **Improved Decision Making:** MOMF optimization offers a more comprehensive understanding of the trade-offs involved in optimization tasks by allowing decision-makers to analyze the interplay between different objectives at varying levels of fidelity. This in-depth understanding facilitates more informed and holistic decision-making, ultimately leading to better optimization outcomes.

Despite these apparent benefits, MOMF optimization remains an emerging area of research. A first paper reporting such an approach for discrete fidelity levels was recently published by Belakaria *et al* [54]. Their MES for MO Bayesian optimization is based on the maximization of mutual information between the Pareto front and the search domain. Being mostly motivated by applications in neural network training, the approach from this paper is however limited to scenarios where higher fidelities yield higher objective values. The assumption that the objective values at lower fidelities are always upper-bounded by the values at the highest fidelity does not hold true for many use cases, such as numerical simulations of physical systems. Here we present two new approaches for Bayesian MOMF optimization that remedy these issues and are furthermore much simpler to implement for practitioners. We first introduce a holistic approach to combine MOMF optimization using trust. This method can select both the input parameters x and the fidelity parameter s jointly. To benchmark this approach, we also introduce a second method that selects the input parameters x and the fidelity parameter s sequentially.

Algorithm 1. Trust-based Multi-Objective Multi-Fidelity Optimization (Trust-MOMF).

Inputs:
- Probed dataset D_n
- Gaussian process models $GP_1, \ldots, GP_k$
- Fidelity function $\boldsymbol{s}$, cost function $C(\boldsymbol{s})$
- Total computational budget C_{total}

Outputs: Optimal Pareto front $\mathcal{P}^*$

Initialization:
Generate n_{init} initial data points D_n
Fit Gaussian process models $GP_1, \ldots, GP_k$
Initialize Pareto front $\mathcal{P} = \emptyset$
Initialize spent budget $C_{\text{spent}} = C_{\text{init}}$

While $C_{\text{spent}} < C_{\text{total}}$ **do**

Optimization step:
Optimize the Trust-MOMF acquisition function, equation (10)
$(\boldsymbol{x}_{n+1}, \boldsymbol{s}_{n+1}) \leftarrow \arg\max_{\boldsymbol{x},\boldsymbol{s}} \left[\frac{\text{EHVI}(\boldsymbol{x},\boldsymbol{s})|D)}{C(\boldsymbol{s})}\right]$

Probe problem at selected position and fidelity:
$\boldsymbol{y}_{n+1} \leftarrow \text{Problem}(\boldsymbol{x}_{n+1}, \boldsymbol{s}_{n+1})$

Data and model updates:
Update dataset $D_{n+1} \leftarrow D_n \cup \{(\boldsymbol{x}_{n+1}, \boldsymbol{s}_{n+1}, \boldsymbol{y}_{n+1})\}$
Re-fit Gaussian process models $GP_1, \ldots, GP_k$
Update Pareto front $\mathcal{P}$
Increment C_{spent} by $C(\boldsymbol{s}_{n+1})$ and n by 1

Terminate

3.1. Trust-based optimization

Our proposed optimization scheme hinges on the *joint* optimization of a function and our *trust* in the information source that yields the results. We use 'trust' to express our confidence level in the outcomes generated by the information source at various levels of fidelity. This formulation allows us to recast MF optimization as a MO problem, with trust $\theta(\boldsymbol{s})$ and output objectives $f(\boldsymbol{x})$ forming the two poles of optimization. Hence, we aim to optimize the following function output:

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{s}) = \begin{pmatrix} \boldsymbol{f}(\boldsymbol{x}) \\ \theta(\boldsymbol{s}) \end{pmatrix}. \tag{9}$$

The trust objective $\theta(\boldsymbol{s})$ is central to the proposed optimization scheme. High fidelity sources, usually more costly, are intrinsically more trustworthy than their low fidelity counterparts due to their higher accuracy and reliability, and thus, trust is a function that grows monotonically with fidelity. In the simplest case, one can assume that there is a constant increase in trust as the fidelity parameter is raised and correspondingly, the trust would be equal to the fidelity itself, $\theta(s) = s$. A more rigorous approach may involve an actual measure of trust, e.g. linking the notion of trust to concepts such as mutual information. In these scenarios, the trust objective can be seen as an approximation that reflects the average mutual information shared across fidelities. For numerous circumstances, like simulations where the outputs at increasing fidelity converge, an appropriate trust curve follows approximately $\theta(s) \approx \tanh(s)$. We discuss the influence of the trust objective on the optimizer in more detail in section 3.3.3.

As discussed in section 2.2, we can optimize a problem of the form equation 9. using, for instance, EHVI. This acquisition function strives to increase the joint HV encapsulated within the Pareto front of $f(\boldsymbol{x}, \boldsymbol{s})$ and $\theta(\boldsymbol{s})$. Due to the ascending nature of trust value, the optimizer is predisposed to probe points with the highest trust. In a MF context where computational or experimental costs are not uniform, the acquisition function can be normalized by incorporating a cost penalty, specifically by dividing the value of the acquisition function by the corresponding cost (section 2.3). This adjustment to the acquisition function ensures that the optimization process selectively probes the point that maximizes the ratio of EHVI to the associated computational or experimental cost. As a result, the acquisition function optimizes our knowledge of the Pareto front and Pareto set on a per-unit-cost basis. The final acquisition function is then written as

$$\text{Trust-MOMF}\,(\boldsymbol{x}, \boldsymbol{s}|D_n) = \frac{\text{EHVI}\,(\boldsymbol{x}, s|D_n)}{\text{cost}\,(\boldsymbol{s})}. \tag{10}$$

where $\text{EHVI}(\boldsymbol{x}, \boldsymbol{s})$ signifies that the complete objective comprises both the normal output objectives and the trust objective. The entire structure of this approach for Trust-MOMF optimization is outlined in algorithm 1.

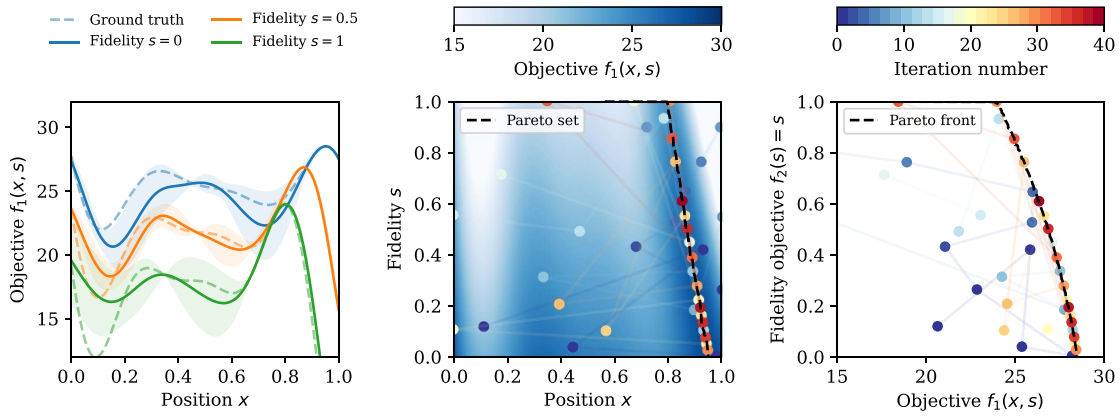

Figure 3. Multi-fidelity optimization via HV improvement of a modified Forrester function. *Left:* mean and variance (shaded curve) of the fitted GP after 40 iterations of the optimizer. The true values are indicated as dashed lines. Note the small variance close to the maximum at all fidelity values. *Center:* map of the objective values $f_1(x, s)$ and sampled points colored according to the iteration number. This illustrates how the optimizer first explores at low fidelity and then moves along the Pareto set. *Right:* same optimization in the objective space, showing how the optimizer tries to increase the HV, which is simply the area in 2D, below the Pareto front.

The algorithm starts with a random initialization of n points that result in a data set $D_n = \{(\boldsymbol{x}_1, \boldsymbol{s}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{s}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_n, \boldsymbol{s}_n, \boldsymbol{y}_n)\}$. With this data set, k number of GP models are fitted corresponding to k output objectives. The initial cost C_{init} utilized until this point is the cost undertaken to generate the D_n. After the initialization, the acquisition function is optimized to yield both the new input and fidelity position where the problem is then probed. This generates a new data sample $(\boldsymbol{x}_{n+1}, \boldsymbol{s}_{n+1}, \boldsymbol{y}_{n+1})$ that is used to update the data set and re-fit the GP models. The last step is to update the spent cost C_{spent} and check whether we exceeded our computational budget C_{total}. The process is terminated when the spent cost becomes greater or equal to our available computational budget.

Figure 3 exemplifies the optimization of a 1D Forrester function by integrating information from lower fidelity data that incur reduced cost. In this test case, trust is assumed linear ($\theta(s) = s$) and the cost is modeled as $C(s) = \exp[a \cdot s]$, with $a = 5$, rendering an evaluation at maximum fidelity ($s = 1$) approximately 150 times costlier than at minimum fidelity ($s = 0$). This cost function can be substituted with any other monotonically increasing function. A key assumption of the Trust-MOMF method is that the Pareto set, which underlies the Pareto front, belongs to a similar region in the search space, thereby facilitating efficient translation of information across fidelities. It is important to note that this method, akin to all MF approaches that leverage knowledge transfer, becomes less efficient if the Pareto set experiences significant shifts between fidelities.

Finally, we would like to point out that this method can be expanded to include multiple fidelity dimensions $s^{(m)}$ where m represents the number of fidelity dimensions. This is especially useful for multi-dimensional numerical models with individual resolution parameters for each dimension. Depending on the problem, these individual fidelity dimensions can either be managed via a single, unified trust objective or treated as separate dimensions within the optimization problem. However, it should be noted that the EHVI algorithm does not scale efficiently to many output dimensions, and the sequential MOMF method introduced in the next section may be a more suitable alternative when numerous fidelity dimensions need to be optimized individually. The different fidelity dimensions can also be discrete or categorical in nature where they take on finite values. The trust-MOMF method still applies but the optimization of the acquisition functions needs to be adapted. One example is mixed acquisition function optimization as performed in botorch for input spaces that include both continuous and discrete input parameters.

3.2. Sequential optimization

While we have so far discussed the benefits and implementation of joint Trust-MOMF optimization, an alternative approach worth exploring is a sequential version of the MOMF optimization. This sequential optimization scheme, inspired by the work of Lam *et al* [48] on single-objective optimization, separates the selection of the next position to probe and the fidelity choice into two distinct steps, thus bringing a different perspective to MF optimization. In the following, we will present and discuss this method, outlining its main principles, operational procedures, and potential benefits. This *sequential* MOMF approach will also serve as a benchmark to evaluate the extent to which our problems benefit from a joint optimization strategy.

Algorithm 2. Sequential MOMF optimization (Seq. MOMF).

Inputs:
- Probed dataset D_n
- Gaussian process models $GP_1, \ldots, GP_k$
- Fidelity function $\boldsymbol{s}$, cost function $C(\boldsymbol{s})$
- Total computational budget C_{total}

Outputs: optimal Pareto front $\mathcal{P}$

1: Initialization:
Generate n_{init} initial data points D_n
Fit Gaussian Pprocess models $GP_1, \ldots, GP_k$
Initialize Pareto front $\mathcal{P} = \emptyset$
Initialize spent budget $C_{\text{spent}} = C_{\text{init}}$

2: while $C_{\text{spent}} < C_{\text{total}}$ **do**

3: Position selection step:
Perform expected hypervolume improvement, equation (7), at maximum fidelity:
$\boldsymbol{x}_{n+1} \leftarrow \arg\max_{\boldsymbol{x}\in\mathcal{X}}[\text{EHVI}(\boldsymbol{f}(\boldsymbol{x})|D_n, s=1)]$

4: Fidelity selection step:
Normalize Data $\boldsymbol{y} \rightarrow \boldsymbol{y}'$
Scalarize output objectives: $y_{\text{scalar}} = \sum_{i=1}^{k} a_i \cdot y_i'$
Fit a Gaussian process model GP_{scalar} on scalarized output objective
Perform max-value entropy search, equation (8), on GP_{scalar} at selected position $\boldsymbol{x}_{n+1}$
$\boldsymbol{s}_{n+1} \leftarrow \arg\max_{s\in\mathcal{S}}[\text{MF-MES}(\boldsymbol{x}_{n+1})]$

5: Probe problem at selected position and fidelity:
$\boldsymbol{y}_{n+1} \leftarrow \text{Problem}(\boldsymbol{x}_{n+1}, \boldsymbol{s}_{n+1})$

6: Data and model updates:
Update dataset $D_{n+1} \leftarrow D_n \cup \{(\boldsymbol{x}_{n+1}, \boldsymbol{s}_{n+1}, \boldsymbol{y}_{n+1})\}$
Re-fit Gaussian process models $GP_1, \ldots, GP_k$
Update Pareto front $\mathcal{P}$
Increment C_{spent} by $C(\boldsymbol{s}_{n+1})$ and n by 1

7: Terminate

In this scheme, the first step after initialization is to get a candidate position in the input space $\boldsymbol{x}_{n+1}$. This position selection step is identical to conventional MO optimization, see section 2.2. In algorithm 2 and the subsequent examples, we use EHVI for this task. Once the candidate point is identified, we can move on to fidelity selection. For this step, the GP first needs to be scalarized at this particular position. Here this is done by summing all the objectives with equal weights. This is done to avoid any preference between objectives and is motivated by the fact that all the objective functions change at a similar scale (due to a normalization step in the data processing that scales outputs of all objectives to the range $[0, 1]$). Alternatively, one could employ other scalarization schemes such as the weighted Chebyshev method or penalty boundary intersection [55]. Once the GP returns a scalar output, we can apply a MF acquisition function (see section 2.3) to select a suitable fidelity setting s_{n+1}. Here we use the MF-MES method that maximizes information gain versus cost, but other methods such as those based on KG would also be suitable. After probing the problem at the selected position and fidelity (x_{n+1}, s_{n+1}), the dataset and GPs are updated and the optimization loop restarts until the allotted computational budget is met.

As we will discuss in section 3.3, this relatively straightforward implementation of a MOMF policy already provides a significant speedup compared to pure MO optimization. An advantage of this scheme is the negligible computational overhead compared to pure MO optimization, as the information gain across fidelities only needs to be calculated at the already selected candidate point.

3.3. Comparison and benchmark

In this section, we will describe the results of our proposed Trust-based and sequential MOMF algorithms on synthetic test functions. All benchmarking is performed by modifying existing implementations of MES and EHVI in the BoTorch package [56] to the multi-objective, MF problem.

Test functions. To assess the performance of the methods and estimate the cost reduction factors, we use MF modifications of the popular maximizing Branin–Currin (2D) and Park (4D) test functions. The function definitions for the Branin–Currin and Park are given in the appendix. The cost function for the different fidelities is modeled as an exponential function of the form $C(s) = \exp[a \cdot s]$ with $a = 4.7$ to result in a ratio of about 120:1 between the highest ($s = 1$) and lowest ($s = 0$) fidelity. The number of iterations for the MOMF algorithms was fixed to 120 while the MO single-fidelity optimization referred to as MO ran for 80 iterations. For the MO optimization, the total cost was 9600 while the MOMF algorithms stopped at variable costs ranging from 1500 to 4000.

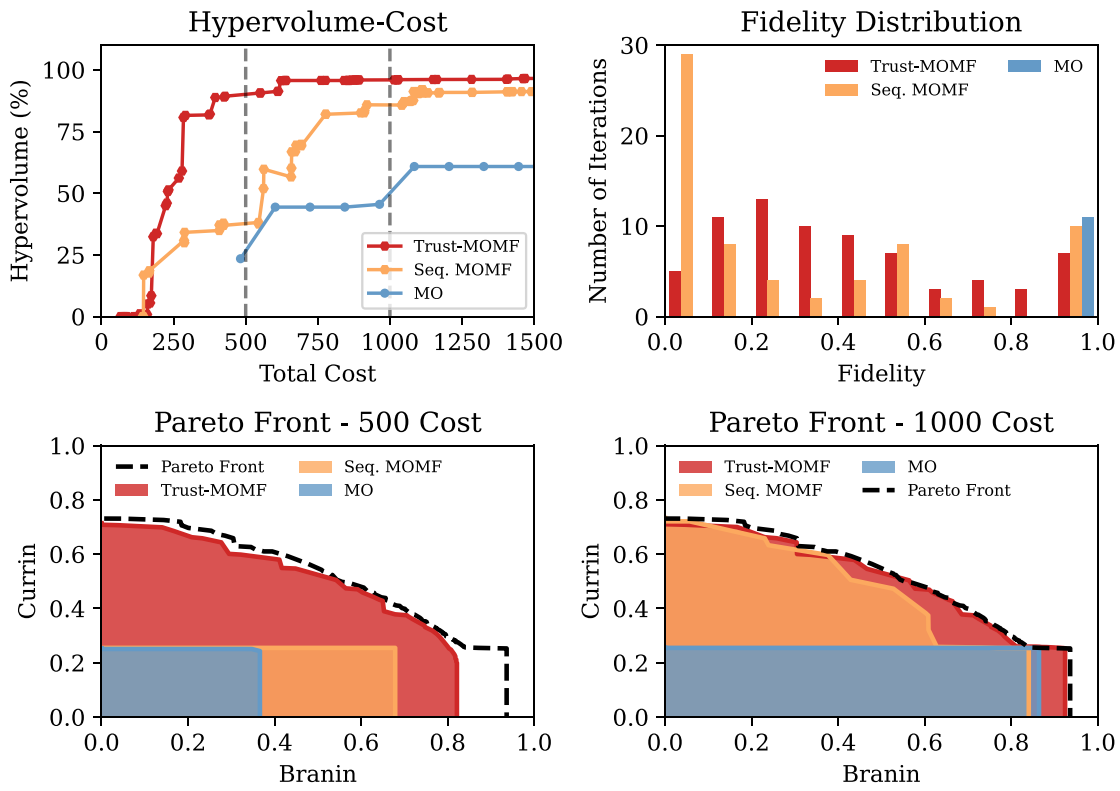

Figure 4. Benchmark with 2D-Branin-Currin problem. *Top Left:* hypervolume of a single representative trial expressed as a percentage of the total hypervolume versus total cost for both MOMF versions and single-fidelity MO. *Top Right:* the number of points taken at different fidelites for the representative trials is shown on the top left. The Trust-MOMF method takes more iterations at intermediate fidelities when compared with the sequential MOMF method which has relatively high peaks at fidelity 0 and 1. *Bottom Left:* the Pareto front for each of the three algorithms for the same trial at a cost of 500 (indicated by the dashed line in the figure on the top left). The dashed black line represents an estimated Pareto front calculated from 10 000 random points. The Trust-MOMF optimization has already found values in the trade-off region and close to the individual maxima of the Branin and Currin functions. The sequential MOMF method is approaching the maximum of the Branin function but both sequential MOMF optimization and MO algorithms have yet to explore the trade-off points. *Bottom Right:* the Pareto front for a cost of 1000 cost for a single representative trial, clearly shows how the Trust-MOMF method has reached an accurate representation of the estimated Pareto front while the conventional, single-fidelity MO algorithm has only discovered a small region of the Pareto front at this cost.

Initialization. Both MOMF algorithms are initialized with five starting points that are randomly distributed in the input search space. The single-fidelity MO optimizer is initially configured with a single point at the highest fidelity $s = 1$. Thus, it starts at an initial cost of $C(1) \approx 120$. Meanwhile, the fidelity of the initial points for the MOMF optimizers is drawn from a cost-aware probability distribution of the form $p(s) \propto 1/C(s)$, which results, on average, in a five-times lower initialization cost. Given the influence of initial points on the performance of optimization, we conduct each optimization run 10 times with different initializations to acquire more robust statistics on algorithm convergence.

HV calculation. After an optimization run is completed, the HV shown in benchmarks is calculated. For this calculation, the points taken during the optimization run are used as training inputs for a GP model while a random input sample of 10 000 points at the highest fidelity are taken as test inputs. From these 10 000 points, we determine a set of non-dominated points, which are then used to calculate the HV at each iteration. It is important to note that this HV estimate is based on expectation value instead of samples from the GP. We do not incorporate model confidence, as done for instance in EHVI. Because of this, the estimation may change or even decrease as the model is updated with new points, see for instance sequential MOMF method at cost ∼600 in figure 4. This effect is minimized as the GP model becomes more certain of its predictions. Note that in the first two iterations due to a lack of training points for the GP model we obtain a large set of non-dominated points making this calculation computationally expensive, thus without any loss of information this calculation is performed after four iterations. This accounts for the graph starting from a cost of 480 for MO optimizations in the subsequent figures.

3.3.1. Results

Branin-Currin test function. In figure 4 an optimization of the MF versions of Branin and Currin [28, 29] functions is shown. A single representative trial that was close to the mean HV as a function of cost is

depicted on the top left. The regular steps along the cost axis for MO optimization indicate a fixed cost at the highest fidelity, while for both the MOMF optimizations it can be seen that step sizes are irregular. The MOMF algorithms take a few points at intermediate fidelities before taking a high fidelity point. This can also be seen in the top right of the figure where the distribution of selected fidelities is shown for each algorithm. Interestingly, we observe a different behavior between the two MOMF algorithms. The sequential MOMF method takes considerably more points at the lowest fidelity, while the Trust-MOMF optimization takes much more intermediate fidelity points, possibly because of the joint optimization of both input and fidelity space.

The bottom part of figure 4 depicts the behavior of the Pareto front at two different costs. The black-dashed line represents the true Pareto front calculated using 10 000 random points. Here it can be seen that the Trust-MOMF method already has a good coverage of the trade-off region and the maximum of the Currin function. The MO and sequential MOMF algorithms at a cost of 500 have much less HV coverage. At a cost of 1000, the MO optimization has optimized the Branin function but still has not explored the trade-off region. The sequential MOMF algorithm at a cost of 1000, is in a similar state as that of Trust-MOMF optimization at a cost of 500. Meanwhile, the Trust-MOMF optimization has reached near 97% HV coverage.

For the estimation of the cost advantage, we use the average HV cost curves from ten runs (shown at the left in figure 6). Taking 90% convergence as a threshold, there is an order of magnitude cost advantage for the Trust-MOMF optimization over the MO optimization. The MO algorithm converged to 90% at about a cost of 6000 while the Trust-MOMF method reached the same HV at a cost of 530, resulting in a cost reduction factor of ~ 11. Moreover, the Trust-MOMF method converged to a higher HV percentage (99%) when compared with the final convergence of 94% for MO optimization.

Park test function. figure 5 depicts a representative trial run results of the same three algorithms for the optimization run of modified Park functions [57]. The top left sub-figure shows the HV percentage covered versus the total cost for the MOMF and MO optimization runs. The Park problem shows a similar behavior regarding the fidelity distribution, i.e. the sequential MOMF method concentrates on points at minimum ($s = 0$) and maximum($s = 1$) fidelity, while the Trust-MOMF method takes more intermediate fidelity points. On the bottom two plots of figure 5, we see the evolution of the Pareto fronts for the 3 optimizers. The Trust-MOMF method already at a cost of 500 has converged to a HV coverage of 94%, hence its Pareto front at a cost of 750 is close to the true Pareto front. The sequential MOMF method also has started exploring the trade-off region and consequently pushes out the Pareto front slightly up to a cost of 1500, as shown on the right. The MO optimization at the cost of 750 has found points close to Park 2 maximum and thus explores the trade-off region at a cost of 1500. The cost advantage estimation is again done using the mean HV versus total cost curves generated using 10 trials as shown in figure 6. The MO algorithm converged to a HV of 89.8% at a cost of about 7600, whereas the Trust-MOMF optimization reached 90% HV coverage at a cost of about 560. This results in a cost reduction factor of about 13, similar to the Branin–Currin problem.

3.3.2. Discussion

For both test functions considered we have observed significant cost reduction in finding the Pareto front of each problem using either the joint trust-based or sequential MOMF optimization. The joint optimization generally outperforms sequential optimization, hinting at its better use of available information within the joint search domain. It should be noted that the cost reduction is intrinsically linked to the cost ratio between the lowest and highest fidelity. In our examples, this ratio was 1:120, and thus, the highest possible cost reduction by taking only the lowest fidelity data points would be 120. Averaging the fidelity of the Trust-MOMF optimizations over 10 trials and 150 iterations for both Branin–Currin and Park functions yields $\bar{s} \simeq 0.30$ and $\bar{s} \simeq 0.32$, respectively. This results in an average cost of 4.3 and 4.6 per iteration, while the conventional MO optimization is fixed to a cost of 120. Thus, a cost reduction of 26 is the maximum that could be achieved if the information gain were the same at all fidelity levels. However, this is generally not the case and low-cost approximations do normally not carry as much information as the highest fidelity. In our case, we see a cost reduction of about half this best possible value, i.e. a cost reduction factor of 13 in both considered test cases. The cost reduction factor can be even higher when the ratio between the cost at the highest and the lowest fidelity is increased. For instance, at a maximum cost ratio of 1200:1 for the Branin–Currin problem we measure a cost reduction factor of 44 for convergence to 90% of the HV.

3.3.3. Understanding the trust objective

The trust objective in the optimizations of the Branin–Currin and the Park functions was defined as a linear function of the fidelity parameter. This trust objective serves to quantify the amount of information that can

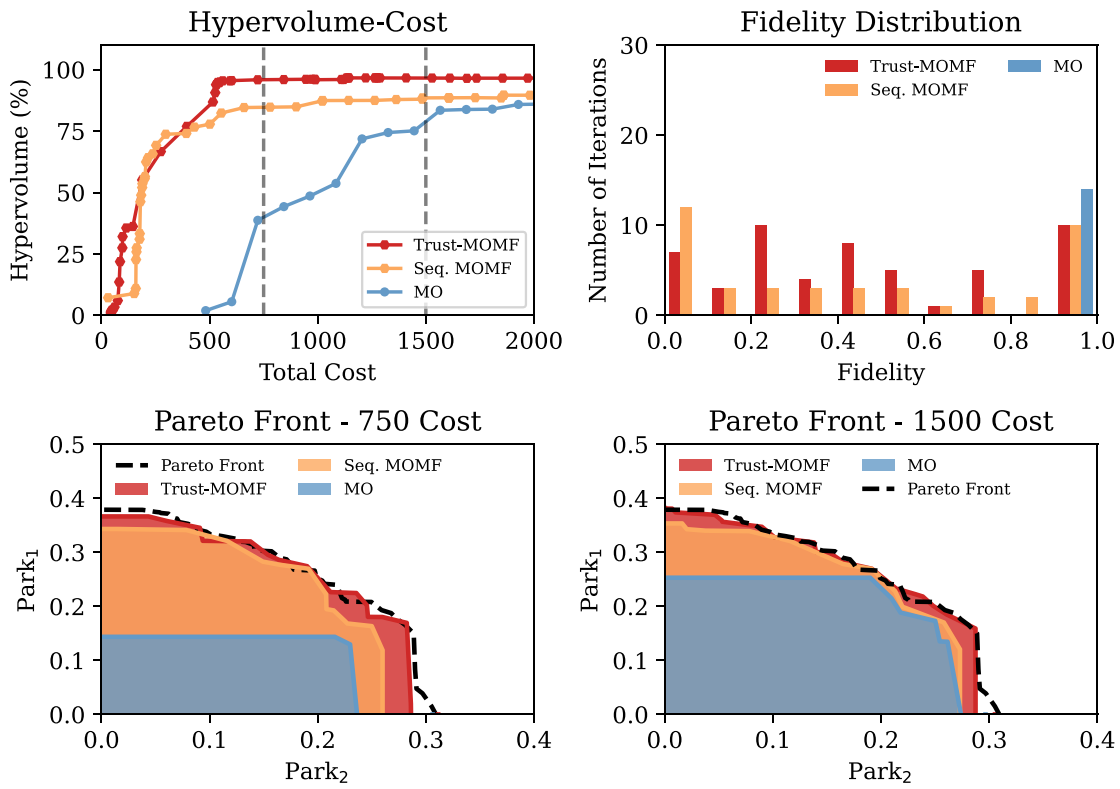

Figure 5. Benchmark with 4D-Park$_{1,2}$ problem. *Top Left:* hypervolume of a single representative trial expressed as a percentage of the total hypervolume versus total cost for both MOMF versions and single-fidelity MO. *Top Right:* the number of points taken at different fidelites for the representative trials shown on the left. The Trust-MOMF method in this case has a higher number of points taken at the highest fidelity. This is because once it has converged it takes 6 points at the highest fidelity to increase hypervolume. The sequential MOMF optimization as seen in the Branin–Currin case takes fewer intermediate fidelity points when compared to the Trust-MOMF optimization. *Bottom Left:* the Pareto front for each of the three algorithms for the same trial at a cost of 750 (indicated by the dashed line in the figure on the top left). The area represents the amount of Pareto front covered by each algorithm. The Trust-MOMF method has already converged to almost 95% of the total hypervolume. The sequential MOMF optimization also converged but to a lower overall hypervolume. The MO optimization at this cost has only found a maximum of Park 2 function. *Bottom Right:* the Pareto front for a cost of 1500 shows little changes in both MOMF Pareto fronts, but a better coverage for the MO Pareto front. At this cost, the MO optimization still has not reached the hypervolume that the Trust-MOMF optimization reached at a cost of 750.

be obtained through probing different fidelities. A linear function implies a constant increase in information as the fidelity parameter is incrementally raised. Combining a linear function $\theta(s) = s$ for the trust objective and an exponential cost function $\mathrm{cost}(s) = \exp(-5s)$, we get atrust-versus-cost trade-off in the Trust-MOMF objective of the form $s \cdot \exp(-5s)$. This trade-off reaches its analytical maximum at $s = 0.2$, which is close to the mean fidelity adopted by the Trust-MOMF method during optimization. However, the mean fidelity during optimization was higher because the objective functions depend on and increase with the fidelity.

We can also use the trust objective to encapsulate the convergence behavior of the optimization problem. As an example, we could select a converging trust function such as $\theta(s) = s\exp(-s)$ defined over the domain $s \in [0, 1]$. This function is monotonically increasing but with diminishing returns. Note that the combination with a cost function of the form $\mathrm{cost}(s) = \exp(-4s)$ yields the same trust-versus-cost trade-off and thus, the same optimum at $s = 0.2$. In all cases, this optimum serves as an estimate for the potential speedup attainable when employing Trust-MOMF with a specific trust objective function and cost function. In appendix, we have shown tests conducted at different cost ratios and different cost functions that shift this analytical maximum. A lower cost ratio leads to a higher analytical maximum and thus a higher mean fidelity. This results in an overall higher cost and a lower speedup when using the Trust-MOMF method. In the case of a linear cost function as depicted in appendix, there is no analytical maximum and hence the cost ratio does not affect the outcome of the optimization as can be seen from the two different cost ratios.

3.4. Application to particle-in-cell (PIC) simulations of laser-plasma acceleration

In this section, we present an application from computational physics that makes full use of the new joint MOMF acquisition function to speed up optimization.

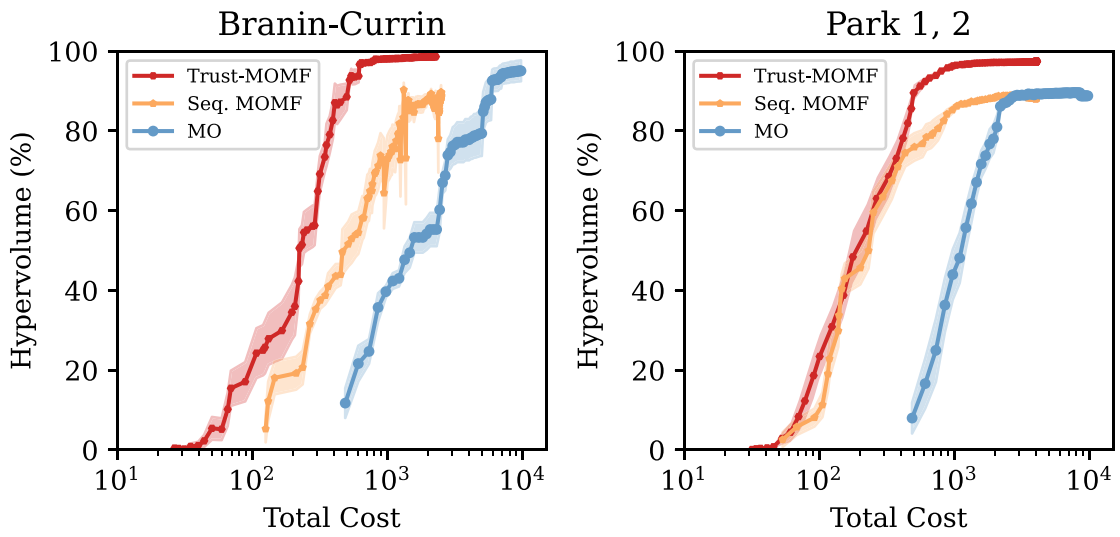

Figure 6. Mean Hypervolume with the shaded region indicating the standard deviation of 10 trials for Branin–Currin and Park functions. *Left:* mean Hypervolume as a percentage of the total for 10 trials of Branin–Currin optimization is illustrated. The total cost is in the log scale to depict the large differences in the cost. The dips seen in the curve around 900 are due to the method of calculating the hypervolume described earlier. When considering convergence to 90% hypervolume an order of magnitude cost difference is seen between Trust-MOMF optimization and MO optimizations. *Right:* the mean hypervolume of 10 trials for the Park functions is shown. Here again, considering a convergence to 90% almost an order of magnitude advantage is seen between the Trust-MOMF optimization and MO optimizations.

PIC Simulation code. One promising application area for MOMF optimization is computational physics. Here we are going to discuss an example from the realm of laser-plasma interactions, which are governed by the Vlasov–Maxwell equations. This kind of system is often simulated using the PIC method [58], where the Maxwell equations are solved on a numerical grid, and the Vlasov equation is solved continuously using a set of macroparticles. The need to compute the trajectories of millions of simulated particles and their fields makes PIC simulations computationally quite costly. For our example, we used the FBPIC which is an open-source Python code frequently used for simulations of laser or plasma wakefield accelerators [59]. FBPIC is a GPU-based code that uses several approximations, making it a relatively 'lightweight' PIC code, with typical runtimes on the order of tens of minutes for test runs and several hours or days for production runs on a single high-end GPU. Extensive scans at production run resolution are hence prohibitively expensive, making this an ideal test case for the use of a MF optimization.

Optimization Parameters. In our example, the laser wakefield accelerator generates quasi-mono energetic electron beams with GeV-level energies and charge of up to a few hundred picocoulomb (pC), similar to the conditions reported in Götzfried *et al* [60]. The optimization goal was to produce beams near a target energy of 300 MeV with low bandwidth and high charge. The input parameters that were controlled during this optimization were the laser focus, the plateau plasma electron density, as well as the shape of the plasma profile (upramp and downramp length). The output objectives are the total charge in the beam, the distance of mean energy to the target energy and the standard deviation of the energy spectrum. The distance to the target energy and the standard deviation are minimization objectives while the charge is to be maximized. This makes the optimization a 4×3 D optimization, making it difficult to depict the Pareto surface. Since these are expensive simulations we also only use the Trust-MOMF method to optimize and compare it with MO optimization.

Cost function and resolution. The cost function for the FBPIC simulations considered here scales as an exponential of the form $C(s) = \exp[a \cdot s]$ with $a = 5$ resulting in a ratio of about 150:1. This cost function was chosen as it approximates the cost increase as a result of increasing the resolution in our FBPIC simulations. For FBPIC simulations that use cylindrical coordinates, defining $\chi = 1 + 3s$, each of these is scaled by a factor of 4 between low and high fidelity, i.e. $\Delta r = \Delta r_0/\chi$, $\Delta z = \Delta z_0/\chi$, $p_\theta = 5\chi$, where Δr and Δz is the mesh size in the radial and axial direction, while the p_θ is the macro-particle number in the azimuth direction. The ranges are chosen based on experience from convergence studies. The total computational budget for each run was capped at 30 h.

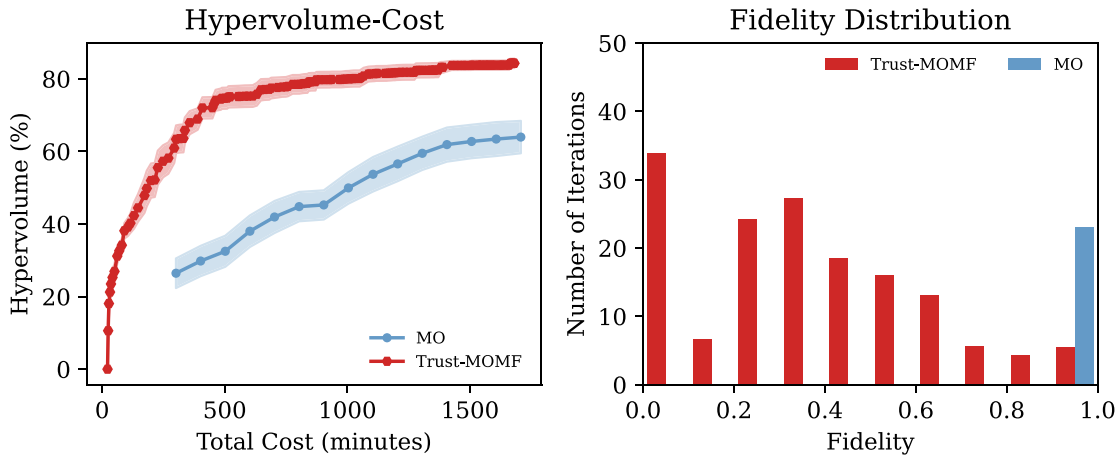

Figure 7. Benchmark with PIC simulations of laser wakefield accelerator using a 4D input space. *Left:* mean hypervolume from 10 trials for the Trust-MOMF method and the MO optimization expressed as a percentage vs the time taken in minutes. The shaded region indicates the standard deviation calculated from the 10 trials. For each run, the computational budget was 30 hours. The sequential MOMF algorithm was excluded because of limited computational and time resources when optimizing with simulations.*Right:* the number of points taken at different fidelities for both the Trust-MOMF optimization and the MO optimization for a single trial.

HV reference. For FBPIC simulations the true HV is unknown, hence the estimation of the HV fraction is based on an ideal 'utopia' point and a reference 'nadir' point for the worst performance. In this instance, we choose:

- Utopia point: 1 nC charge, 0 MeV bandwidth, 0 MeV distance to target energy
- Nadir point: 0 nC charge, 1 GeV bandwidth, 1 GeV distance to target energy.

The maximum volume from this would be 1 nC·GeV^2. We note that this value itself has little meaning since it can be changed by either changing the Utopia or the Nadir point. The more meaningful criteria used for comparing the MO and the Trust-MOMF optimization is the fractional HV gained.

Results. In figure 7 we see that the Trust-MOMF algorithm outperforms the MO optimization by converging to a HV of about 85% while the MO optimization in this budget converged to a HV of about 63%. The MO optimization was run for 8 more hours but still it failed to reach a HV of more than 73%. Also shown in figure 7 is the fidelity distribution which shows that the Trust-MOMF method takes a large number of simulations with fidelity of less than 0.5, thus efficiently making use of these faster, low-resolution simulations. Comparing the costs we see that the Trust-MOMF method in this case provides a cost reduction factor of about 6, which is similar to the gain we previously estimated with test functions. These encouraging results confirm the usefulness of joint MOMF for expensive numerical simulations in physics.

4. Summary and outlook

In conclusion, we have successfully demonstrated the practical implementation of simultaneous MF and MO optimization, achieving substantial cost reductions—over an order of magnitude—for test problems with a cost ratio of 1:120. Our Trust-MOMF method showed superior convergence towards a higher HV compared to single-fidelity MO optimization approaches. We furthermore observed that the joint Trust-MOMF policy consistently outperforms the sequential MOMF strategy, indicating the beneficial use of joint information across objectives and fidelities. We also discussed the theoretical upper limit for the cost reduction factor, which can be amplified further by widening the cost ratio between the highest and lowest fidelity levels.

The significance of these findings lies in the substantial cost-efficiency MOMF optimization offers, particularly for applications aiming to optimize multiple objectives where access to lower fidelity data is available. Our approach, which builds upon existing acquisition functions, is easily implementable within optimized Bayesian optimization frameworks such as BoTorch [56]. It can be seamlessly extended, for instance, to batch optimization.

Our work contributes a key optimization tool for complex problems in fields such as physics and engineering, where simulations with varying degrees of accuracy are commonly used. Moreover, it holds

potential for any domain where there is an advantage in optimizing different objectives while also having access to less costly evaluations. The flexibility of our approach is underscored by its adaptability to any other MO, single-fidelity acquisition function—a trust objective can be introduced and the acquisition value penalized by cost. Additionally, our MOMF methods can be extended to include multiple fidelity dimensions, offering the prospect of efficient optimization over different resolution parameters in numerical simulations.

Code samples for the benchmark cases are available online for further exploration and experimentation. As we look to the future, we anticipate the increasingly broad application and continued evolution of MF and MO optimization techniques.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/PULSE-ML/MOMFBO-Algorithm.

Acknowledgments

This work was supported by the DFG through the Cluster of Excellence Munich-Centre for Advanced Photonics (MAP EXC 158), TR-18 funding schemes and the Max Planck Society. It was also supported by the Independent Junior Research Group 'Characterization and control of high-intensity laser pulses for particle acceleration', DFG Project No. 453619281. F I is part of the Max Planck School of Photonics supported by BMBF, Max Planck Society, and Fraunhofer Society. The authors thank C Eberle for helpful discussions.

Appendix

A.1. Notation

Notation	Description
$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{f}, \boldsymbol{s}$	bold notation for vectors
$f^{(1)}, f^{(2)}, \ldots, f^{(k)}$	index k of objective functions
$\boldsymbol{x}, \boldsymbol{y}$	input and output vector
$\boldsymbol{f}, \boldsymbol{s}$	vector of objective functions and fidelity vector

A.2. Definition

Definition 1. A point $\mathbf{p}_1 = (y_1, y_2, \ldots, y_n)$ in an n-dimensional output space is defined as *non-dominated* if there does not exist another point $\mathbf{p}_2' = (y_1', y_2', \ldots, y_n')$ such that $\mathbf{p}_2'$ satisfies both:

$$\bigwedge_{i=1}^{n} (y_i' \geqslant y_i) \qquad \text{and} \qquad \bigvee_{i=1}^{n} (y_i' > y_i),$$

where $\bigwedge_{i=1}^{n}$ represents the logical AND operation applied across all n conditions, indicating that all inequalities $y_i' \geqslant y_i$ must hold simultaneously for $i = 1, 2, \ldots, n$. The $\bigvee_{i=1}^{n}$ represents the logical OR operation, meaning that at least one of these inequalities must be strict.

A.3. Additional benchmarks

In this section, we study the effect of different cost functions and cost ratios on the Trust-MOMF acquisition function. All of the tests conducted in this section are on the modified MF MO Branin–Currin functions. In figure A1, on the left side the ratio between the maximum and minimum cost was kept to 120 as in the main benchmarks while for the graph on the right, this ratio was reduced to 10. The Trust-MOMF method still reaches a higher HV in less cost but the advantage is decreased from a higher cost ratio. In figure A2, the cost function is no longer an exponential but a linear function with two different cost ratios between the maximum and minimum cost. The difference between the cost ratios with a linear cost function is not as pronounced as in the case of exponential functions. This effect is already explained in detail in section 3.3.3.

A.4. GP Kernel choice

Many studies motivated by applications such as neural network training use an exponential decay kernel and a zero mean prior by default. In this paper, we have k number of $GP_1, GP_2, \ldots, GP_k$ models modelling k

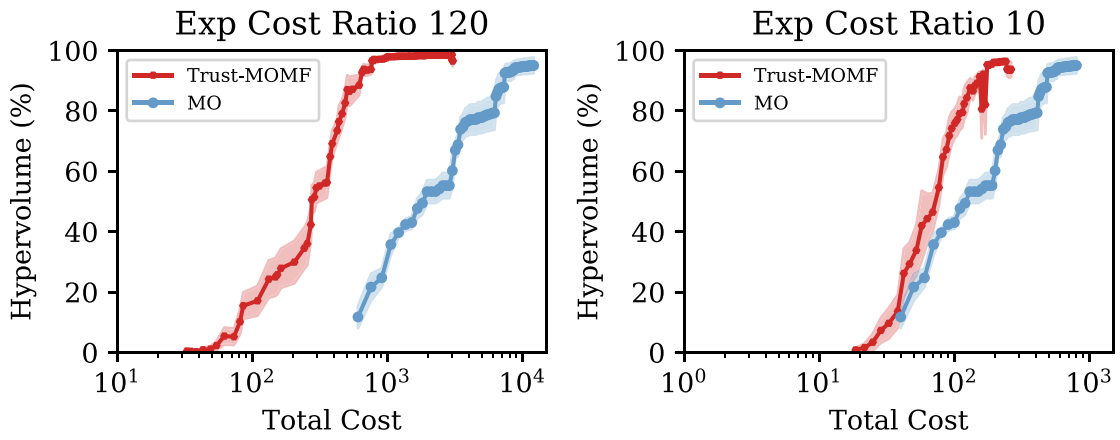

Figure A1. Benchmark with Branin–Currin functions with two different cost ratios of an exponential cost function.

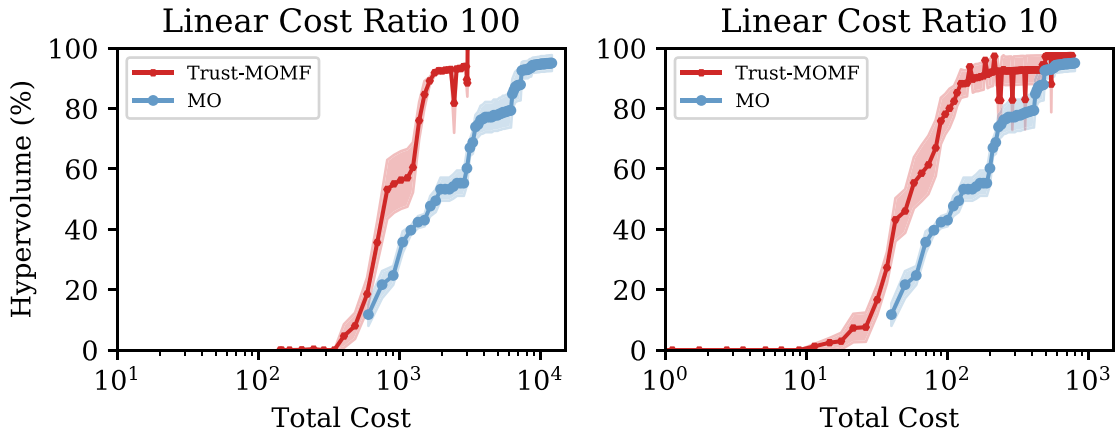

Figure A2. Benchmark with Branin–Currin functions with two different cost ratios of a linear cost function.

objective functions $f_1, f_2, \ldots, f_k(\boldsymbol{x})$. Each of these functions also has an input fidelity parameter s that implies greater accuracy with higher values. We have chosen a Matern 5/2 kernel to model the fidelity dependence, which has shown significantly better performance on our test and application cases than the exponential decay kernel.

A.5. Test functions

In this section, we describe the analytical functions used to benchmark the performance of our algorithm. As this is one of the first studies on combined MOMF optimization, we could not directly use test functions from the literature but had to modify them to incorporate an additional fidelity input dimension and exhibit trade-off behavior between the objectives.

A.5.1. Modified MF forrester function

The original Forrester function is

$$f(x) = (6x-2)^2 \cdot \sin(12x-4) + 7.025$$

and we use the following modified version as a MF reference case:

$$f(x,s) = D(s) \cdot [E - g(x,s)]$$

where

$$g(x,s) = A(s) \cdot f[x - 0.2(1 - x \cdot s)] + B(s) \cdot (x - 0.5) - C(s)$$

with $A = 0.5 + 0.5s$, $B = 2 - 2s$, $C = 5s - 5$, $D = 1.5 - 0.5s$ and $E = 25$. The most important differences to other MF versions of this function are that it contains a fidelity- and position-dependent shift

$\tilde{x}(x,s) = x - 0.2(1 - x \cdot s)$, is inverted for maximization ($E - g(x,s)$ term), the value of the maxima along the fidelity is decreasing (D(s) term) and the maxima are continuously connected from at low to high fidelity.

A.5.2. Modified MF, MO branin-currin function

Two popular functions used for optimization benchmarking are Branin–Currin functions which were also modified. The usual form of the Branin function is

$$B(\boldsymbol{x}) = a\left(x_2 - bx_1^2 + cx_1 - r\right)^2 + p(1-t)\cos(x_1) + p,$$

where values of the constants were taken to be $a = 1$, $b = 5.1/(4\pi^2)$, $c = 5/\pi$, $r = 6$, $p = 10$, $t = 1/(8\pi)$ and the form of Currin function is

$$C(\boldsymbol{x}) = \left[1 - \exp\left(-\frac{1}{2x_2}\right)\right]\frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20}.$$

Both of these functions were modified to make the range of input and output values to be between 0 and 1, i.e. $x_i, y_i \in [0,1]\,\forall i = [1,2]$. The modified form that was used for the Branin is

$$B(\boldsymbol{x},s) = -\left[a\left(x_{22} - b(s)x_{11}^2 + c(s)x_{11} - r\right)^2 + p(1-t(s))\cos(x_{11}) + p\right]$$

where $x_{11} = 15x_1 - 5$, $x_{22} = 15x_2$, $a = 1$, $b(s) = 5.1/(4\pi^2) - 0.01(1-s)$, $c(s) = 5/\pi - 0.1(1-s)$, $r = 6$, $p = 10$, $t(s) = (1/(8\pi)) + 0.05(1-s)$ was used. The x_{11} and x_{22} are used to scale the original domain of the Branin to [0,1]. The main difference is the addition of fidelity parameter s which for a value of 1 yields the original Branin function and since we are maximizing the problem a minus sign is added. The modified form for the Currin function is

$$C(\boldsymbol{x},s) = -\left[\left[1 - (0.1)(1-s)\exp\left(-\frac{1}{2x_2}\right)\right]\frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20}\right]$$

where again the main difference is the addition of fidelity term $1 - s$ and the addition of a minus sign for maximization.

A.5.3. Modified MF MO park functions

For a benchmark of the MO MF problem in higher dimensions, MF versions of Park 1 and Park 2 functions were used. The original form of the Park functions is

$$P_1(\boldsymbol{x}) = \frac{x_1}{2}\left[\sqrt{1 + (x_2 + x_3^2)\frac{x_4}{x_1^2}} - 1\right] + (x_1 + 3x_4)\exp[1 + \sin(x_3)]$$

$$P_2(x) = \frac{2}{3}\exp(x_1 + x_2) - x_4\sin(x_3) + x_3.$$

We modified the above two Park functions adding a fidelity dimension (s). To achieve a reasonable Pareto front for optimization, the two functions were also slightly modified. The location of the Pareto set was also modified to not have all the optimizing points in the corners of the 4D hypercube. A last modification is shifting the Pareto front of the Park functions by subtraction to place a higher importance on the trade-off region. The final form of the two modified Park functions is

$$P_1(\boldsymbol{x},s) = A(s)[T_1 + T_2 - B(s)]/22 - 0.8$$

$$T_1 = \left[\frac{x_1 + 0.001(1-s)}{2}\right].\left[\sqrt{1 + (x_2 + x_3^2)\frac{x_4}{x_1^2}}\right]$$

$$T_2 = (x_1 + 3x_4)\exp[1 + \sin(x_3)]$$

$$P_2(\boldsymbol{x},s) = A(s)\left[5 - \frac{2}{3}\exp(x_1 + x_2) - (x_4)\sin(x_3)A(s) + x_3 - B(s)\right]/4 - 0.7$$

where $A(s) = (0.9 + 0.1s)$ and $B(s) = 0.1 * (1-s)$. Both Park functions now contain a fidelity parameter s. These Park functions are evaluated on a transformed input space

$$[x_1, x_2, x_3, x_4] \rightarrow \left[1 - 2(x_1 - 0.6)^2, x_2, 1 - 3(x_3 - 0.5)^2, 1 - (x_4 - 0.8)^2\right].$$

ORCID iD

Andreas Döpp https://orcid.org/0000-0003-2913-5729

References

[1] Snoek J, Larochelle H and Adams R P 2012 Practical bayesian optimization of machine learning algorithms *Advances in Neural Information Processing Systems* vol 25
[2] Mockus J 1994 Application of bayesian approach to numerical methods of global and stochastic optimization *J. Glob. Opt.* **4** 347–65
[3] Jones D R, Schonlau M and Welch W J 1998 Efficient global optimization of expensive black-box functions *J. Glob. Opt.* **13** 455
[4] Packwood D *et al* 2017 *Bayesian Optimization for Materials Science* (Springer)
[5] Shoemaker C A, Regis R G and Fleming R C 2007 Watershed calibration using multistart local optimization and evolutionary optimization with radial basis function approximation *Hydrol. Sci. J.* **52** 450–65
[6] Döpp A, Eberle C, Howard S, Irshad F, Lin J and Streeter M 2023 Data-driven science and machine learning methods in laser-plasma physics *High Power Laser Sci. Eng.* **11** 1–50
[7] Sharma S and Kumar V 2022 A comprehensive review on multi-objective optimization techniques: past, present and future *Arch. Comput. Methods Eng.* **29** 5605–33
[8] Mohammad Mirjalili S M, Merikhi B, Zahra Mirjalili S Z, Zoghi M and Mirjalili S 2017 Multi-objective versus single-objective optimization frameworks for designing photonic crystal filters *Appl. Opt.* **56** 9444–51
[9] Arias-Montano A, Coello Coello C A and Mezura-Montes E 2012 Multiobjective evolutionary algorithms in aeronautical and aerospace engineering *IEEE Trans. Evol. Comput.* **16** 662–94
[10] Avder A, Sahin I, Dörterler M 2019 Multi-objective design optimization of the robot grippers with spea2 *Int. J. Intell. Syst. Appl. Eng.* **7** 83–87
[11] Branke J, Branke J, Deb K, Miettinen K and Slowiński R 2008 *Multiobjective Optimization: Interactive and Evolutionary Approaches* vol 5252 (Springer)
[12] Deb K 2011 Multi-objective optimisation using evolutionary algorithms: an introduction *Multi-Objective Evolutionary Optimisation for Product Design and Manufacturing* (Springer) pp 3–34
[13] Feliot P, Bect J and Vazquez E 2017 A bayesian approach to constrained single-and multi-objective optimization *J. Glob. Opt.* **67** 97–133
[14] Klein A, Falkner S, Bartels S, Hennig P and Hutter F 2017 Fast bayesian optimization of machine learning hyperparameters on large datasets *Artificial Intelligence and Statistics* (PMLR) pp 528–36
[15] Frazier P I 2018 A tutorial on bayesian optimization (arXiv:1807.02811)
[16] Edward Rasmussen C 2003 Gaussian processes in machine learning *Summer School on Machine Learning* (Springer) pp 63–71
[17] Edward Rasmussen C and Williams C K I 2006 *Gaussian Processes for Machine Learning* vol 1 (Springer)
[18] Pfingstl S and Zimmermann M 2022 On integrating prior knowledge into gaussian processes for prognostic health monitoring *Mech. Syst. Signal Process.* **171** 108917
[19] Wilson J, Hutter F and Deisenroth M 2018 Maximizing acquisition functions for bayesian optimization *Advances in Neural Information Processing Systems* vol 31
[20] Kanazawa T 2021 Using distance correlation for efficient bayesian optimization (arXiv:2102.08993)
[21] Shibo Li, Xing W, Kirby R and Zhe S 2020 Multi-fidelity bayesian optimization via deep neural networks *Advances in Neural Information Processing Systems* vol 33, ed H Larochelle, M Ranzato, R Hadsell, M F Balcan and H Lin (Curran Associates, Inc) pp 8521–31
[22] Cox D D and John S 1992 A statistical method for global optimization *Proc. 1992 IEEE Int. Conf. on Systems, Man and Cybernetics* (IEEE) pp 1241–6
[23] Frazier P I, Powell W B and Dayanik S 2008 A knowledge-gradient policy for sequential information collection *SIAM J. Control Optim.* **47** 2410–39
[24] Scott W, Frazier P and Powell W 2011 The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression *SIAM J. Optim.* **21** 996–1026
[25] Hennig P and Schuler C J 2012 Entropy search for information-efficient global optimization *J. Mach. Learn. Res.* **13**
[26] Wang Z and Jegelka S 2017 Max-value entropy search for efficient bayesian optimization *Int. Conf. on Machine Learning* (PMLR) pp 3627–35
[27] Phong Nguyen Q, Kian Hsiang Low B and Jaillet P 2022 Rectified max-value entropy search for bayesian optimization (arXiv:2202.13597)
[28] Dixon L C W 1978 The global optimization problem. An introduction *Toward Glob. Opt.* **2** 1–15
[29] Currin C, Mitchell T, Morris M and Ylvisaker D 1991 Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments *J. Am. Stat. Assoc.* **86** 953–63
[30] Knowles J 2006 Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems *IEEE Trans. Evol. Comput.* **10** 50–66
[31] Yang K, Emmerich M, Deutz A and Bäck T 2019 Multi-objective bayesian global optimization using expected hypervolume improvement gradient *Swarm Evol. Comput.* **44** 945–56
[32] Emmerich M T M, Giannakoglou K C and Naujoks B 2006 Single-and multiobjective evolutionary optimization assisted by gaussian random field metamodels *IEEE Trans. Evol. Comput.* **10** 421–39
[33] Couckuyt I, Deschrijver D and Dhaene T 2014 Fast calculation of multiobjective probability of improvement and expected improvement criteria for pareto optimization *J. Glob. Opt.* **60** 575–94
[34] Luo C, Shimoyama K and Obayashi S 2014 Kriging model based many-objective optimization with efficient calculation of expected hypervolume improvement *2014 IEEE Congress on Evolutionary Computation (CEC)* (IEEE) pp 1187–94
[35] Shimoyama K, Jeong S and Obayashi S 2013 Kriging-surrogate-based optimization considering expected hypervolume improvement in non-constrained many-objective test problems *2013 IEEE Congress on Evolutionary Computation* (IEEE) pp 658–65
[36] Emmerich M T M, Deutz A H and Willem Klinkenberg J 2011 Hypervolume-based expected improvement: monotonicity properties and exact computation *2011 IEEE Congress of Evolutionary Computation (CEC)* (IEEE) pp 2147–54

[37] Hupkens I, Deutz A, Yang K and Emmerich M 2015 Faster exact algorithms for computing expected hypervolume improvement *Int. Conf. on Evolutionary Multi-Criterion Optimization* (Springer) pp 65–79
[38] Emmerich M, Yang K, Deutz A, Wang H and Fonseca C M 2016 A multicriteria generalization of Bayesian global optimization *Advances in Stochastic and Deterministic Global Optimization* (Springer) pp 229–42
[39] Yang K, Emmerich M, Deutz A and Fonseca C M 2017 Computing 3-d expected hypervolume improvement and related integrals in asymptotically optimal time *Int. Conf. on Evolutionary Multi-Criterion Optimization* (Springer) pp 685–700
[40] Daulton S, Balandat M and Bakshy E 2020 Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization (arXiv:2006.05078)
[41] Suzuki S, Takeno S, Tamura T, Shitara K and Karasuyama M 2020 Multi-objective bayesian optimization using pareto-frontier entropy *Int. Conf. on Machine Learning* (PMLR) pp 9279–88
[42] Tu B, Gandy A, Kantas N and Shafei B 2022 Joint entropy search for multi-objective bayesian optimization *Advances in Neural Information Processing Systems* vol 35, ed S Koyejo, S Mohamed, A Agarwal, D Belgrave, K Cho and A Oh (Curran Associates, Inc) pp 9922–38
[43] Huang D, Allen T T, Notz W I and Allen Miller R 2006 Sequential kriging optimization using multiple-fidelity evaluations *Struct. Multidiscip. Opt.* **32** 369–82
[44] Picheny V, Ginsbourger D, Richet Y and Caplin G 2013 Quantile-based optimization of noisy computer experiments with tunable precision *Technometrics* **55** 2–13
[45] Swersky K, Snoek J and Adams R P 2013 Multi-task bayesian optimization *Advances in Neural Information Processing Systems* vol 26, ed C J C Burges, L Bottou, M Welling, Z Ghahramani and K Q Weinberger (Curran Associates, Inc)
[46] McLeod M, Osborne M A and Roberts S J 2017 Practical bayesian optimization for variable cost objectives (arXiv:1703.04335)
[47] Zhang Y, Nghia Hoang T, Kian Hsiang Low B and Kankanhalli M 2017 Information-based multi-fidelity bayesian optimization *Nips Workshop on Bayesian Optimization*
[48] Lam R, Allaire D L and Willcox K E 2015 Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conf.* p 0143
[49] Jian W and Frazier P I 2017 Continuous-fidelity bayesian optimization with knowledge gradient *NIPS Workshop on Bayesian Optimization*
[50] Jian W, Toscano-Palmerin S, Frazier P I and Gordon Wilson A 2020 Practical multi-fidelity bayesian optimization for hyperparameter tuning *Uncertainty in Artificial Intelligence* (PMLR) pp 788–98
[51] Takeno S, Fukuoka H, Tsukada Y, Koyama T, Shiga M, Takeuchi I and Karasuyama M 2020 Multi-fidelity bayesian optimization with max-value entropy search and its parallelization *Int. Conf. on Machine Learning* (PMLR) pp 9334–45
[52] Kandasamy K, Dasarathy G, Poczos B and Schneider J 2016 The multi-fidelity multi-armed bandit *Advances in Neural Information Processing Systems* vol **29** pp 1777–85
[53] Kandasamy K, Dasarathy G, Schneider J and Póczos B 2017 Multi-fidelity bayesian optimisation with continuous approximations *Int. Conf. on Machine Learning* (PMLR) pp 1799–808
[54] Belakaria S, Deshwal A and Rao Doppa J 2020 Multi-fidelity multi-objective bayesian optimization: an output space entropy search approach *Proc. 34th AAAI Conf. on Artificial Intelligence* pp 10035–43
[55] Chugh T 2020 Scalarizing functions in bayesian multiobjective optimization *2020 IEEE Congress on Evolutionary Computation (CEC)* pp 1–8
[56] Balandat M, Karrer B, Jiang D, Daulton S, Letham B, Gordon Wilson A and Bakshy E 2020 Botorch: a framework for efficient monte-carlo bayesian optimization *Advances in Neural Information Processing Systems (NeurIPS)*
[57] Soo Park J Tuning complex computer codes to data and optimal designs *PhD Thesis* University of Illinois at Urbana-Champaign
[58] Vay J-L and Lehe R 2016 Simulations for Plasma and Laser Acceleration *Rev. Accel. Sci. Technol.* **09** 165–86
[59] Lehe R, Kirchen M, Andriyash I A, Godfrey B B and Vay J-L 2016 A spectral, quasi-cylindrical and dispersion-free Particle-In-Cell algorithm *Comput. Phys. Commun.* **203** 66–82
[60] Götzfried J, Döpp A, Gilljohann M F, Foerster F M, Ding H, Schindler S, Schilling G, Buck A, Veisz L and Karsch S 2020 Physics of high-charge electron beams in laser-plasma wakefields *Phys. Rev.* X **10** 041015