# Uncertainty aware Search Framework for Multi-Objective Bayesian Optimization with Constraints

**Syrine Belakaria**                                          SYRINE.BELAKARIA@WSU.EDU
*Washington State University*

**Aryan Deshwal**                                          ARYAN.DESHWAL@WSU.EDU
*Washington State University*

**Janardhan Rao Doppa**                                          JANA.DOPPA@WSU.EDU
*Washington State University*

## Abstract

We consider the problem of constrained multi-objective (MO) blackbox optimization using expensive function evaluations, where the goal is to approximate the true Pareto set of solutions satisfying a set of constraints while minimizing the number of function evaluations. We propose a novel framework named Uncertainty-aware Search framework for Multi-Objective Optimization with Constraints (USeMOC) to efficiently select the sequence of inputs for evaluation to solve this problem. The selection method of USeMOC consists of solving a cheap constrained MO optimization problem via surrogate models of the true functions to identify the most promising candidates and picking the best candidate based on a measure of uncertainty. We applied this framework to optimize the design of a multi-output switched-capacitor voltage regulator via expensive simulations. Our experimental results show that USeMOC is able to achieve more than 90% reduction in the number of simulations needed to uncover optimized circuits.

## 1. Introduction

Many engineering and scientific applications involve making design choices to optimize multiple objectives. A representative example includes designing new materials to optimize strength, elasticity, and durability. There are three challenges in solving these kind of optimization problems: **1)** The objective functions are unknown and we need to perform expensive experiments to evaluate each candidate design. For example, performing physical lab experiments for material design application. **2)** The objectives are conflicting in nature and all of them cannot be optimized simultaneously. **3)** The designs that are part of the solution should satisfy a set of constraints. Therefore, we need to find the *Pareto optimal* set of solutions satisfying the constraints. 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 true Pareto set satisfying the constraints while minimizing the number of function evaluations.

Bayesian Optimization (BO) (Shahriari et al. (2016)) is an effective framework to solve blackbox optimization problems with expensive function evaluations. The key idea behind BO is to build a cheap surrogate model (e.g., Gaussian Process (Williams and Rasmussen (2006))) using the real experimental evaluations; and employ it to intelligently select the sequence of function evaluations using an acquisition function, e.g., expected improvement (EI). There is a large body of literature on single-objective BO algorithms (Shahriari et al. (2016); Deshwal et al. (2020)) and their applications including hyper-parameter tuning of machine learning methods (Snoek et al. (2012); Kotthoff et al. (2017)). However, there is

relatively less work on the more challenging problem of BO for multiple objectives (Knowles (2006); Emmerich and Klinkenberg (2008); Hernández-Lobato et al. (2016); Belakaria et al. (2019, 2020a)) and only *one* prior method to address constrained MO problems referred as PESMOC (Garrido-Merchán and Hernández-Lobato (2019)). PESMOC is a state-of-the-art information-theoretic approach that relies on the principle of input space entropy search. However, it is computationally expensive to optimize the acquisition function behind PESMOC. A series of approximations are performed to improve the efficiency potentially at the expense of accuracy.

In this paper, we propose a novel **U**ncertainty-aware **Se**arch framework for **M**ultiple **O**bjectives optimizing with **C**onstraints (USeMOC) to overcome the drawbacks of prior methods. The key insight is to improve uncertainty management via a two-stage search procedure to select candidate inputs for evaluation. First, it solves a cheap constrained MO optimization problem defined in terms of the acquisition functions (one for each unknown objective) to identify a list of promising candidates. Second, it selects the best candidate from this list based on a measure of uncertainty. We demonstrate the efficacy of USeMOC over prior methods using a real-world application from the domain of analog circuit design via expensive simulations.

## 2. Background and Problem Setup

**Bayesian Optimization Framework.** Let $\mathcal{X} \subseteq \Re^d$ be an input space. We assume an unknown real-valued objective function $F : \mathcal{X} \mapsto \Re$, which can evaluate each input $x \in \mathcal{X}$ to produce an evaluation $y = F(x)$. Each evaluation $F(x)$ is expensive in terms of the consumed resources. The main goal is to find an input $x^* \in \mathcal{X}$ that approximately optimizes $F$ via a limited number of function evaluations. BO algorithms learn a cheap surrogate model from training data obtained from past function evaluations. They intelligently select the next input for evaluation by trading-off exploration and exploitation to quickly direct the search towards optimal inputs. The three key elements of BO framework are:

**1) Statistical Model** of $F(x)$. *Gaussian Process (GP)* (Williams and Rasmussen (2006)) is the most commonly used model. A GP over a space $\mathcal{X}$ is a random process from $\mathcal{X}$ to $\Re$. It is characterized by a mean function $\mu : \mathcal{X} \times \mathcal{X} \mapsto \Re$ and a covariance or kernel function $\kappa$. If a function $F$ is sampled from GP($\mu$, $\kappa$), then $F(x)$ is distributed normally $\mathcal{N}(\mu(x), \kappa(x, x))$ for a finite set of inputs from $x \in \mathcal{X}$.

**2) Acquisition Function** (AF) to score the utility of evaluating a candidate input $x \in \mathcal{X}$ based on the statistical model. Some popular acquisition functions include expected improvement (EI), upper confidence bound (UCB) and lower confidence bound (LCB). For the sake of completeness, we formally define the acquisition functions employed in this work noting that any other acquisition function can be employed within USeMOC.

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

$$LCB(x) = \mu(x) - \beta^{1/2}\sigma(x) \tag{2}$$

$$EI(x) = \sigma(x)(\alpha\Phi(\alpha) + \phi(\alpha)), \ \alpha = \frac{\tau - \mu(x)}{\sigma(x)} \tag{3}$$

where $\mu(x)$ and $\sigma(x)$ correspond to the mean and standard deviation of the prediction from statistical model, $\beta$ is a parameter that balances exploration and exploitation; $\tau$ is the best uncovered input; and $\Phi$ and $\phi$ are the CDF and PDF of normal distribution respectively.

**3) Optimization Procedure** to select the best scoring candidate input according to AF via statistical model, e.g., DIRECT (Jones et al. (1993)).

**Multi-Objective Optimization (MOO) Problem with Constraints.** Without loss of generality, our goal is to minimize $k \geq 2$ real-valued objective functions $F_1(x), F_2(x), \cdots, F_k(x)$ while satisfying $L$ constraints $C_1(x), C_2(x), \cdots, C_L(x)$ over continuous space $X \subseteq \Re^d$. Each evaluation of an input $x \in \mathcal{X}$ produces a vector of objective values $Y = (y_1, y_2, \cdots, y_k)$ where $y_i = F_i(x)$ for all $i \in \{1, 2, \cdots, k\}$ and a vector of constraints values $C = (c_1, c_2, \cdots, c_L)$ where $c_i = C_i(x)$ for all $i \in \{1, 2, \cdots, L\}$. We say that a point $x$ *Pareto-dominates* another point $x'$ if $F_i(x) \leq F_i(x') \; \forall i$ and there exists some $j \in \{1, 2, \cdots, k\}$ such that $F_j(x) < F_j(x')$. The optimal solution of constrained MOO problem is a set of points $\mathcal{X}^* \subset \mathcal{X}$ such that no point $x' \in \mathcal{X} \setminus \mathcal{X}^*$ Pareto-dominates a point $x \in \mathcal{X}^*$ and all points in $\mathcal{X}^*$ satisfies the problem constraints. The solution set $\mathcal{X}^*$ is called the Pareto set and the corresponding set of function values is called the Pareto front. Our goal is to approximate $\mathcal{X}^*$ while minimizing the number of function evaluations.

## 3. Uncertainty-Aware Search Framework

In this section, we provide the details of USeMOC framework for solving constrained multi-objective optimization problems. First, we provide an overview of USeMOC followed by the details of its two main components.

### 3.1 Overview of USeMOC Framework

USeMOC is an iterative algorithm that involves four key steps. First, We build statistical models $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_k$ for each of the $k$ objective functions from the training data in the form of past function evaluations. Second, we select a set of promising candidate inputs $\mathcal{X}_p$ by solving a constrained cheap MO optimization problem defined using the statistical models. Specifically, multiple objectives of the cheap MO problem correspond to $\text{AF}(\mathcal{M}_1, x), \text{AF}(\mathcal{M}_2, x), \cdots, \text{AF}(\mathcal{M}_k, x)$ respectively. Any standard acquisition function AF from single-objective BO (e.g., EI, LCB) can be used for this purpose. Additionally, if the value of a constraint $C_i$ depends on the evaluation of one of the objectives $F_i(x)$, we use the predictive mean $\mu_i(x)$ instead. The Pareto set $\mathcal{X}_p$ corresponds to the inputs with different trade-offs in the utility space for $k$ unknown functions satisfying the constraints. Third, we select the best candidate input $x_s \in \mathcal{X}_p$ from the Pareto set that maximizes some form of uncertainty measure for evaluation. Fourth, the selected input $x_s$ is used for evaluation to get the corresponding function evaluations: $y_1 = F_1(x_s)$, $y_2 = F_2(x_s)$, $\cdots$, $y_k = F_k(x_s)$. Algorithm 1 provides the algorithmic pseudocode for USeMOC. Due to lack of space we present algorithm 1 in Appendix A.

**Advantages.** USeMOC has many advantages over prior methods. **1)** Provides flexibility to plug-in any acquisition function for single-objective BO. This allows us to leverage existing acquisition functions including EI and LCB. **2)** Computationally-efficient to solve constrained MO problems with many objectives.

### 3.2 Key Algorithmic Components of USeMO

The two main algorithmic components of USeMOC framework are: selecting most promising candidate inputs by solving a cheap constrained MO problem and picking the best candidate via uncertainty maximization. We describe their details below.

**Selection of promising candidates.** We employ the statistical models $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_k$ towards the goal of selecting promising candidate inputs as follows. Given an acquisition function $\mathrm{AF}$ (e.g., EI), we construct a constrained cheap multi-objective optimization problem with objectives $\mathrm{AF}(\mathcal{M}_1, x), \mathrm{AF}(\mathcal{M}_2, x), \cdots, \mathrm{AF}(\mathcal{M}_k, x)$, where $\mathcal{M}_i$ is the statistical model for unknown function $F_i$ and $C_1, C_2, \cdots, C_L$ are the problem constraints. Generally, there are two types of constraints: 1) $C_i$ is a function of the input $x$ and can be expressed declaratively. In this case, the constraint will take the same form in the cheap MO problem; and 2) $C_i$ is a function of both the input $x$ and a combination of the blackbox objective functions. Since verifying constraint $C_i$ depends on the evaluation of one (or more) objective $F_j$, we use the predictive mean(s) $\mu_i(x)$ instead. Since we present the framework as minimization, all AFs will be minimized. The Pareto set $\mathcal{X}_p$ obtained by solving this cheap constrained MO problem represents the most promising candidate inputs for evaluation.

$$\mathcal{X}_p \leftarrow \arg min_{x \in \mathcal{X}} \ (\mathrm{AF}(\mathcal{M}_1, x), \cdots, \mathrm{AF}(\mathcal{M}_k, x)) \tag{4}$$
$$\mathbf{s.t} \ C_0, \cdots, C_L$$

Each acquisition function $\mathrm{AF}(\mathcal{M}_i, x)$ is dependent on the corresponding surrogate model $\mathcal{M}_i$ of the unknown objective function $F_i$. Hence, each acquisition function will carry the information of its associated objective function. As iterations progress, using more training data, the models $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_k$ will better mimic the true objective functions $F_1, F_2, \cdots, F_k$. Therefore, the Pareto set of the acquisition function space (solution of Equation 4) becomes closer to the Pareto set of the true functions $\mathcal{X}^*$ with increasing iterations. Intuitively, the acquisition function $\mathrm{AF}(\mathcal{M}_i, x)$ corresponding to unknown objective function $F_i$ tells us the utility of a point $x$ for optimizing $F_i$. The input minimizing $\mathrm{AF}(\mathcal{M}_i, x)$ has the highest utility for $F_i$, but may have a lower utility for a different function $F_j$ ($j \neq i$). The utility of inputs for evaluation of $F_j$ is captured by its own acquisition function $\mathrm{AF}(\mathcal{M}_j, x)$. Therefore, there is a trade-off in the utility space for all $k$ different functions. The Pareto set $\mathcal{X}_p$ obtained by simultaneously optimizing acquisition functions for all $k$ unknown functions will capture this utility trade-off. As a result, each input $x \in \mathcal{X}_p$ is a promising candidate for evaluation towards the goal of solving MOO problem. USeMOC employs the same acquisition function for all $k$ objectives. The main reason is to give equivalent evaluation for all functions in the Pareto front (PF) at each iteration. If we use different AFs for different objectives, the sampling procedure would be different. Additionally, the values of various AFs can have considerably different ranges. Thus, this can result in an unbalanced trade-off between functions in the cheap PF leading to the same unbalance in our final PF.

**Cheap constrained MO solver.** We employ the constrained version of the popular NSGA-II algorithm (Deb et al. (2002)) to solve the MO problem with cheap objective functions and cheap constraints noting that any other algorithm can be used.

**Picking the best candidate input.** We need to select the best input from the Pareto set $\mathcal{X}_p$ obtained by solving the cheap MO problem. All inputs in $\mathcal{X}_p$ are promising in the sense that they represent the trade-offs in the utility space corresponding to different unknown functions. It is critical to select the input that will guide the overall search towards the goal of quickly approximating the true Pareto set $\mathcal{X}^*$. We employ a uncertainty measure defined in terms of the statistical models $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_k$ to select the most promising candidate input for evaluation. In single-objective optimization case, the learned model's uncertainty for an input can be defined in terms of the variance of the statistical model.

For multi-objective optimization case, we define the uncertainty measure as the volume of the uncertainty hyper-rectangle.

$$U_{\beta_t}(x) = VOL(\{(LCB(\mathcal{M}_i, x), UCB(\mathcal{M}_i, x)\}_{i=1}^k) \tag{5}$$

where $LCB(\mathcal{M}_i, x)$ and $UCB(\mathcal{M}_i, x)$ represent the lower confidence bound and upper confidence bound of the statistical model $\mathcal{M}_i$ for an input $x$ as defined in equations 2 and 3; and $\beta_t$ is the parameter value to trade-off exploitation and exploration at iteration $t$. We employ the adaptive rate recommended by (Srinivas et al. (2009)) to set the $\beta_t$ value depending on the iteration number $t$. We measure the uncertainty volume measure for all inputs $x \in \mathcal{X}_p$ and select the input with maximum uncertainty for function evaluation.

$$x_{t+1} = \arg max_{x \in \mathcal{X}_p} U_{\beta_t}(x) \tag{6}$$

## 4. Experiments and Results

In this section, we discuss the experimental evaluation of USeMOC and prior methods on a real-world analog circuit design optimization task.

**Analog circuit optimization domain.** We consider optimizing the design of a multi-output switched-capacitor voltage regulator via Cadence circuit simulator that imitates the real hardware Belakaria et al. (2020c). Each candidate circuit design is defined by 33 input variables ($d$=32). The first 24 variables are the width, length, and unit of the eight capacitors of the circuit $W_i, L_i, M_i\ \forall i \in 1 \cdots 8$. The remaining input variables are four output voltage references $V_{ref_i}\ \forall i \in 1 \cdots 4$ and four resistances $R_i\ \forall i \in 1 \cdots 4$. We optimize nine objectives: maximize efficiency $Eff$, maximize four output voltages $V_{o_1} \cdots V_{o_4}$, and minimize four output ripples $OR_1 \cdots OR_4$. Our problem has a total of nine constraints:

$$C_0 : Cp_{total} \simeq 20nF\ with\ Cp_{total} = \sum_{i=1}^{8}(1.955 W_i L_i + 0.54(W_i + L_i))M_i \tag{7}$$

$$C_1\ to\ C_4 : V_{o_i} \geq V_{ref_i}\ \forall \in 1 \cdots 4 \tag{8}$$

$$C_5\ to\ C_8\ :\ \ OR_{lb} \leq OR_i \leq OR_{ub}\ \forall i\ \in 1 \cdots 4 \tag{9}$$

$$C_9 : Eff \leq 100\% \tag{10}$$

where $OR_{lb}$ and $OR_{ub}$ are the predefined lower-bound and upper-bound of $OR_i$ respectively. $Cp_{total}$ is the total capacitance of the circuit.

**Multi-objective BO algorithms.** We compare USeMOC with the existing BO method PESMOC. Due to lack of BO approaches for constrained MO, we compare to known genetic algorithms (NSGA-II and MOEAD). However, they require large number of function evaluations to converge which is not practical for optimization of expensive functions.

**Evaluation Metrics.** To measure the performance of baselines and USeMOC, we employ two different metrics, one measuring the accuracy of solutions and another one measuring the efficiency in terms of the number of simulations. 1) Pareto hypervolume (PHV) is a commonly employed metric to measure the quality of a given Pareto front Zitzler (1999). After each iteration $t$ (or number of simulations), we measure the PHV for all algorithms. We evaluate all algorithms for 100 circuit simulations. 2) Percentage gain in simulations is the fraction of simulations our BO algorithm USeMOC is saving to reach the PHV accuracy of solutions at the convergence point of baseline algorithm employed for comparison.

**Results and Discussion.** We evaluate the performance of USeMOC with two different acquisition functions (EI and LCB) to show the generality and robustness of our approach. We also provide results for the percentage gain in simulations achieved by USeMOC when compared to each baseline method. Figure 1a shows the PHV metric achieved by different multi-objective methods including USeMOC as a function of the number of circuit simulations. We make the following observations: 1) USeMOC with both EI and LCB acquisition functions perform significantly better than all baseline methods. 2) USeMOC is able to uncover the best Pareto solutions from baselines using significantly less number of circuit simulations. This result shows the efficiency of our approach. Table 1 shows that USeMOC achieves percentage gain in simulations w.r.t baseline methods ranging from 90 to 92.5%.

The analog circuit is implemented in the industry-provided process design kit (PDK) and shows better efficiency and output ripples. Since MOEAD is the best performing baseline optimization method, we use it for the rest of the experimental analysis. Table 1b illustrates the simulated performance of circuit optimized by MOEAD (best baseline) and USeMOC-EI (best variant of our proposed algorithm). Results of both algorithms meet the voltage reference and ripple requirements (100mV) . Compared to MOEAD, the optimized circuit with USeMOC-EI can achieve a higher conversion efficiency of 76.2 % (5.25 % higher than MOEAD, highlighted in red color) with similar output ripples. The optimized circuits can generate the target output voltages within the range of 0.52V-0.61V (1/3x ratio) and 1.07V-1.12V (2/3x ratio) under the loads varying from 14 Ohms to 1697 Ohms (highlighted in black and green colors). Thus, the capability of USeMOC to optimize the parameters of circuit under different output voltage/current conditions is clearly validated. Future work includes improving USeMOC to solve more challenging problems Belakaria et al. (2020b).

| Method | MOEAD | NSGA-II | PESMOC |
|---|---|---|---|
| Gain in simulations | 90.7% | 93.3% | 92.5% |

Table 1: Percentage gain in simulations achieved by our USeMOC compared to baselines.
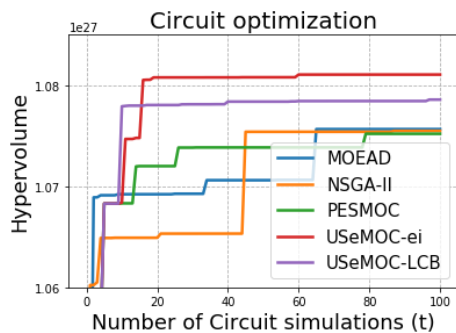


(a) Results of different multi-objective algorithms including USeMOC. The hypervolume metric is shown as a function of the number of circuit design simulations.

| SPECS | MOEAD | | | USEMOC-EI | | |
|---|---|---|---|---|---|---|
| $V_{ref1}$(V) | 0.53 | 0.6 | 0.6 | *0.52* | 0.53 | 0.56 |
| $V_{ref2}$(V) | 0.55 | 0.51 | 0.59 | *0.55* | 0.61 | 0.57 |
| $V_{ref3}$(V) | 1.14 | 1.06 | 1.07 | *1.07* | 1.12 | 1.11 |
| $V_{ref4}$(V) | 1.22 | 1.16 | 1.14 | *1.09* | 1.06 | 1.1 |
| $R_1$(Ohm) | 144 | 1668 | 1012 | **207** | **1198** | **619** |
| $R_2$(Ohm) | 758 | 620 | 559 | **306** | **1697** | **89** |
| $R_3$(Ohm) | 247 | 66 | 10 | **67** | **1379** | **70** |
| $R_4$(Ohm) | 222 | 144 | 1830 | **42** | **14** | **301** |
| $V_{o1}$(mV) | 551.5 | 702.18 | 775.01 | **677.10** | **760.60** | **656.9** |
| $V_{o2}$ (mV) | 612.2 | 671.01 | 912.22 | **690.70** | **725.70** | **569.4** |
| $V_{o3}$ (V) | 1.17 | 1.09 | 867.96 | *1.08* | **1.15** | **1.14** |
| $V_{o4}$ (V) | 1.117 | 1.12 | 1.12 | *1.08* | 0.99 | **1.13** |
| OR1(mV) | 52.69 | 11.39 | 0.496 | 2.50 | 4.20 | 11.30 |
| OR2(mV) | 9.32 | 4.52 | 0.984 | 3.50 | 5.00 | 15.90 |
| OR3(mV) | 64.96 | 96.57 | 28.199 | 58.9 | 87.1 | 75.7 |
| OR4(mV) | 4.75 | 4.80 | 1.05 | 80.7 | 25.1 | 74.3 |
| Eff (%) | 70.95 | 65.94 | 64.61 | *76.2003* | *74.82* | *73.71* |

(b) Comparison table of optimized circuit parameters obtained by MOEAD and USeMOC-EI (designs are selected from the Pareto set prioritized by efficiency)

# References

Syrine Belakaria, Aryan Deshwal, and Janardhan Doppa. Max-value entropy search for multi-objective Bayesian optimization. In *Proceedings of International Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Syrine Belakaria, Aryan Deshwal, Nitthilan Kannappan Jayakodi, and Janardhan Rao Doppa. Uncertainty-aware search framework for multi-objective Bayesian optimization. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 10044–10052, 2020a.

Syrine Belakaria, Derek Jackson, Yue Cao, Janardhan Rao Doppa, and Xiaonan Lu. Machine learning enabled fast multi-objective optimization for electrified aviation power system design. In *IEEE Energy Conversion Congress and Exposition (ECCE)*, 2020b.

Syrine Belakaria, Zhiyuan Zhou, Aryan Deshwal, Janardhan Rao Doppa, Partha Pande, and Deuk Heo. Design of multi-output switched-capacitor voltage regulator via machine learning. In *Proceedings of the Twenty-Third IEEE/ACM Design Automation and Test in Europe Conference (DATE)*, 2020c.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, T Meyarivan, and A Fast. Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

Aryan Deshwal, Syrine Belakaria, Janardhan Rao Doppa, and Alan Fern. Optimizing discrete spaces via expensive evaluations: A learning to search framework. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 3773–3780, 2020.

Michael Emmerich and Jan-willem Klinkenberg. The computation of the expected improvement in dominated hypervolume of pareto front approximations. *Technical Report, Leiden University*, 34, 2008.

Eduardo C Garrido-Merchán and Daniel Hernández-Lobato. Predictive entropy search for multi-objective bayesian optimization with constraints. *Neurocomputing*, 361:50–68, 2019.

Daniel Hernández-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. Predictive entropy search for multi-objective bayesian optimization. In *International Conference on Machine Learning*, pages 1492–1501, 2016.

Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of optimization Theory and Applications*, 79(1): 157–181, 1993.

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

Lars Kotthoff, Chris Thornton, Holger H Hoos, Frank Hutter, and Kevin Leyton-Brown. Auto-weka 2.0: Automatic model selection and hyperparameter optimization in weka. *The Journal of Machine Learning Research*, 18(1):826–830, 2017.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT Press Cambridge, MA, 2006.

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

# Appendix A. Additional technical details

---

**Algorithm 1** USeMOC Framework

---

**Input**: $\mathcal{X}$, input space; $F_1(x), F_2(x), \cdots, F_k(x)$, $k$ blackbox objective functions; $\mathrm{AF}$, acquisition function; and $T_{max}$, maximum no. of iterations

1: Initialize training data of function evaluations $\mathcal{D}$
2: Initialize statistical models $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_k$ from $\mathcal{D}$
3: **for** each iteration $t$=1 to $T_{max}$ **do**
4:    // Solve constrained cheap MO problem with objectives $\mathrm{AF}(\mathcal{M}_1, x), \cdots, \mathrm{AF}(\mathcal{M}_k, x)$ to get candidate inputs
5:    $\mathcal{X}_p \leftarrow \arg min_{x \in \mathcal{X}}(\mathrm{AF}(\mathcal{M}_1, x), \cdots, \mathrm{AF}(\mathcal{M}_k, x))$
             **s.t** $C_0, \cdots, C_L$
6:    // Pick the candidate input with maximum uncertainty
7:    Select $x_{t+1} \leftarrow \arg max_{x \in \mathcal{X}_p} U_{\beta_t}(x)$
8:    Evaluate $x_{t+1}$: $Y_{t+1} \leftarrow (F_1(x_{t+1}), \cdots, F_k(x_{t+1}))$
9:    Aggregate data: $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_{t+1}, Y_{t+1}, C_{t+1})\}$
10:   Update models $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_k$ using $\mathcal{D}$
11:   $t \leftarrow t + 1$
12: **end for**
13: **return** Pareto set and Pareto front of $\mathcal{D}$

---