Maximising Hypervolume and Minimising ϵ-Indicators using Bayesian Optimisation over Sets

Tinkle Chugh
University of Exeter
Exeter, UK
t.chugh@exeter.ac.uk

Manuel López-Ibáñez
University of Málaga
Málaga, Spain
Alliance Manchester Business School
Manchester, UK
manuel.lopez-ibanez@uma.es

ABSTRACT

Bayesian optimisation methods have been widely used to solve problems with computationally expensive objective functions. In the multi-objective case, these methods have been successfully applied to maximise the expected hypervolume improvement of individual solutions. However, the hypervolume, and other unary quality indicators such as multiplicative ϵ-indicator, measure the quality of an approximation set and the overall goal is to find the set with the best indicator value. Unfortunately, the literature on Bayesian optimisation over sets is scarce. This work uses a recent set-based kernel in Gaussian processes and applies it to maximise hypervolume and minimise ϵ-indicators in Bayesian optimisation over sets. The results on benchmark problems show that maximising hypervolume using Bayesian optimisation over sets gives a similar performance than non-set based methods. The performance of using ϵ indicator in Bayesian optimisation over sets needs to be investigated further. The set-based method is computationally more expensive than the non-set-based ones, but the overall time may be still negligible in practice compared to the expensive objective functions.

CCS CONCEPTS

• **Mathematics of computing** → **Probabilistic algorithms**; • **Computing methodologies** → **Supervised learning by regression**.

KEYWORDS

Surrogate, Gaussian Processes, Pareto optimality, Set based optimisation, Quality indicators

ACM Reference Format:
Tinkle Chugh and Manuel López-Ibáñez. 2021. Maximising Hypervolume and Minimising ϵ-Indicators using Bayesian Optimisation over Sets. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3449726.3463178

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3463178

1 INTRODUCTION

Many real-world optimisation problems have conflicting objectives and use computationally expensive models or costly experiments. Optimisation of expensive optimisation problems, including single and multi-objective, has seen significant advances in the last decade thanks to the adoption of techniques from Bayesian Optimisation [22]. These algorithms have seen successful applications in simulation-based and data-driven [12] problems and are being increasingly adopted by practitioners. We define a multi-objective optimisation problem (MOP) as:

$$\text{minimise} \quad (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \qquad \text{subject to} \quad \mathbf{x} \in S \qquad (1)$$

with $m \geq 2$ objective functions $f_i(\mathbf{x}) \colon S \to \mathbb{R}$. The vector of objective function values is denoted by $f(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T$. The (nonempty) feasible space S is a subset of the decision space $\mathbb{R}^n$ and consists of decision vectors $\mathbf{x} = (x_1, \ldots, x_n)^T$ that satisfy all the constraints.

When no information is available about the preferences of a decision-maker, the goal is to approximate the Pareto front. Evaluating the quality of approximation sets is a difficult task and many evaluation metrics have been proposed in the literature. The hypervolume metric is among the few Pareto-compliant metrics that never contradict Pareto optimality. In addition, the hypervolume is the only (known) unary metric that is able to report that one approximation set is better than another for every case in which this is true in terms of Pareto optimality [27]. Moreover, the hypervolume measures qualitative aspects that include closeness to the Pareto front and diversity along the Pareto front. It has also been used as the selection criterion in Evolutionary Multi-Objective Optimisation Algorithms (EMOAs), with the most prominent examples being SMS-EMOA [1] and IBEA [26]. Detailed benchmarks show that both SMS-EMOA and IBEA are among the best performing MOEAs, often outperforming other more popular algorithms [2].

Another widespread quality metric is the multiplicative unary ϵ-indicator [27]. The ϵ-indicator measures the factor by which the objective vectors of a given approximation set must be multiplied in order to weakly dominate a reference set. Lower values are preferred. Values smaller than 1 mean that the approximation set is better than the reference set. Although the ϵ-indicator is also Pareto-compliant, it is weaker than hypervolume because two approximation fronts may have the same ϵ-indicator value even if one is clearly better than the other in terms of Pareto optimality. Nevertheless, there is empirical evidence that the search landscapes

of hypervolume and ϵ are significantly different [18] and optimising them leads to different approximations of the Pareto front [11]. Therefore, we decided to consider both in this study.

Bayesian optimisation (BO) using expected improvement as acquisition function was first used in [19] for single-objective optimisation problems. Later, Emmerich et al. [7] extended it to expected hypervolume improvement to handle problems with expensive multiple objectives. Later works in the literature have increased the efficiency of this acquisition function by finding its gradients and adapting it for parallel evaluations [5]. A different formulation of the hypervolume improvement as the scalarising function was used in [20]. All of these works predict the improvement of a solution to the hypervolume of a reference set (i.e., hypervolume contribution) and not the exact hypervolume of a set. This is due to the reason that hypervolume relies on sets and Bayesian optimisation using sets has not received much attention.

A recent proposal has extended the use of Gaussian processes ($\mathcal{GP}$s) over sets and applied them in Bayesian optimisation [14]. In this paper, we apply this proposal for the set-based optimisation of the hypervolume and ϵ indicators and evaluate its performance against non set-based Bayesian optimisation methods. Our results show that Bayesian optimisation over sets with hypervolume obtained similar results to existing methods and Bayesian optimisation over sets with ϵ-indicator did not perform well on some problems.

This paper is structured as follows. In the next section, we provide an overview of Bayesian optimisation. In Section 3, we explain the use of Gaussian processes over sets. In Section 4, we explain the approach of generating sets for training the $\mathcal{GP}$ model and use them in optimising indicators with Bayesian optimisation. The results are discussed in Section 5. Finally, we conclude and mention the future research directions in Section 6.

2 BAYESIAN OPTIMISATION

The input to Bayesian optimisation is the data set $D = \{(X, Y) \mid X \in \mathbb{R}^{N\times n}, Y \in \mathbb{R}^{N\times m}\}$, having n decision variables and m objective function values and N is the size of the data set. We assume this data is either available or can be obtained with some design of experiment technique. Next we build $\mathcal{GP}$ model(s) on the data set. There are typically two different ways to build GP models in multi-objective optimisation problems. One is to build a model for each objective function [7] and second is to build a single model after scalarising the objective functions [3], i.e., $\mathbf{g} = g(Y) \in \mathbb{R}^{N\times 1}$. The second method reduces the number of objectives from m to one. The computational complexity of the first method is at most $O(mN^3)$ and of the second method is at most $O(N^3)$.

After building the $\mathcal{GP}$ model(s), we optimise an acquisition function (or infill criterion) to find potential decision vector. Both exploitation and exploration are usually combined within the infill criterion. Most of the methods in the literature on Bayesian optimisation focus on developing an efficient infill criterion. Some of them are expected improvement [13, 19], probability of improvement [16], and expected hypervolume improvement [7]. Maximising the infill criterion often requires the use of an optimisation algorithm. For instance, BFGS [8], CMA-ES [9], or some evolutionary algorithm [25]. The selected decision vector (more than one if in batch mode) is then evaluated on the true objective function (e.g. via simulator) and added to the data set. All the solutions evaluated with the expensive model become the final solutions after a termination criterion is met (usually maximum number of expensive evaluations). In the multi-objective case, the final approximation to the Pareto front returned is the mutually non-dominated set of solutions from all solutions ever evaluated.

Algorithm 1: Bayesian optimisation

Input: Data Set, $D = (X, Y)$
Output: Evaluated solutions
repeat
  $M \leftarrow$ Train $\mathcal{GP}$ model(s) on the the data set D
  // Get a sample by maximising the acquisition function:
  $\mathbf{x}^* \leftarrow \arg\max_{\mathbf{x}} \ \alpha(M, D)$
  Evaluate $\mathbf{x}^*$ and add to the data set D
until *termination criterion is not met*

3 GAUSSIAN PROCESSES OVER SETS

Gaussian processes ($\mathcal{GP}$s) have advantages over other regression methods because of their ability to provide uncertainty information in addition to the point predictions [21]. This uncertainty can be used in selecting efficient samples by maximising the acquisition function and in efficient decision-making [17]. Let us denote the function values with $\mathbf{y}$ which can be $\mathbf{f}_i$ for $i = 1, \ldots, m$ or the scalarising function values $\mathbf{g}$.

A $\mathcal{GP}$ is described with a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix K:

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, K) \tag{2}$$

For simplicity in calculations, we assume a mean of zero. For details about other mean functions, refer to [21]. The covariance matrix elements are calculated using covariance or a kernel function, $k(\mathbf{x}, \mathbf{x}')$. For instance, a Gaussian kernel (or squared exponential kernel) is defined as:

$$k(\mathbf{x}, \mathbf{x}', \boldsymbol{\Theta}) = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{j=1}^{n}\frac{|x_j - x_j'|^2}{l_j^2}\right) + \sigma_t^2 \delta_{\mathbf{x}\mathbf{x}'}, \tag{3}$$

where $\Theta = (\sigma_f, l_1, \ldots, l_n, \sigma_t)$ is the set of parameters and $\delta_{\mathbf{x}\mathbf{x}'}$ is the Kronecker delta function. The $|x_j - x_j'|$ represents the Euclidean distance between x_j and x_j'. The parameters σ_f, l_j and σ_t represent the amplitude, length scale of j^{th} variable and noise in the data, respectively. These parameters can be estimated by maximising (e.g. with some gradient based algorithm) the likelihood function:

$$p(\mathbf{y} \mid X, \boldsymbol{\Theta}) = \frac{1}{\sqrt{|2\pi K|}} \exp\left(-\tfrac{1}{2}\mathbf{y}^T K^{-1}\mathbf{y}\right) \ . \tag{4}$$

The model built after estimating the parameters is used for approximating a posterior predictive distribution (also Gaussian):

$$\begin{aligned} p\left(y^* | \mathbf{x}^*, X, \mathbf{y}, \boldsymbol{\Theta}\right) = &\mathcal{N}\big(\mathbf{k}(\mathbf{x}^*, X)K^{-1}\mathbf{y}, \\ &k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*, X)^T K^{-1}\mathbf{k}(X, \mathbf{x}^*)\big). \end{aligned} \tag{5}$$

where the posterior mean is $\mathbf{k}(\mathbf{x}^*, X)K^{-1}\mathbf{y}$ and the variance representing the uncertainty is $k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*, X)^T K^{-1}\mathbf{k}(X, \mathbf{x}^*)$.

The $\mathcal{GP}$ detailed above relies on the correlation between two decision vectors $\mathbf{x}$ and $\mathbf{x}'$. Therefore, it is straightforward to use it in traditional Bayesian optimisation for single or multi-objective optimisation problems. In a recent work [14], the correlation was extended to sets and a set kernel was proposed. Given a kernel $k(\mathbf{x}, \mathbf{x}')$ between two decision vectors, the set kernel between two sets X and X' is defined as:

$$k_{\text{set}}(X, X') = \frac{1}{|X||X'|} \sum_{\mathbf{x} \in X} \sum_{\mathbf{x}' \in X'} k(\mathbf{x}, \mathbf{x}') \ , \tag{6}$$

where $|X|$ represent the cardinality of the set X. One of the important features of this kernel is that the resulting covariance is symmetric and positive-semi-definite. Moreover, the kernel is not affected by the ordering of the decision vectors in the set. The updated kernel can be used in model building by maximising the likelihood function in (4), where $\mathcal{X} = [X_1, \ldots, X_L], \mathbf{y} \in \mathbb{R}^{L \times 1}$ and L is the number of sets. The posterior predictive distribution of a new set X^* is:

$$\begin{aligned} p\left(y^* | X^*, X, \mathbf{y}, \Theta\right) = & \mathcal{N}\big(\mathbf{k}_{\text{set}}(X^*, X) K^{-1} \mathbf{y}, \\ & k_{\text{set}}(X^*, X^*) - \mathbf{k}_{\text{set}}(X^*, X)^T K^{-1} \mathbf{k}_{\text{set}}(X, X^*)\big). \end{aligned} \tag{7}$$

It is worthy to mention that the computational complexity of the set kernel is $O(N_s^2 |X|^2 n)$, which makes it computationally expensive compared to the traditional kernel. In [14], the authors proposed an approximation of the set kernel to alleviate the computational cost. In this work, we do not use such approximation and assume that the computational cost of building models is significantly lower than expensive objective evaluation.

4 OPTIMISING QUALITY INDICATORS WITH BAYESIAN OPTIMISATION OVER SETS

We explain next how to use Bayesian optimisation over sets to tackle expensive multi-objective black-box problems. The main idea is to build a $\mathcal{GP}$ model that, given a set of decision vectors, predicts the corresponding value of a unary quality indicator, such as hypervolume or multiplicative-ϵ, for the whole set, instead of individual decision vectors. Thus, Bayesian optimisation using acquisition function searches for a candidate set of decision vectors. Once identified, decision vectors in this set are evaluated to find their corresponding expensive objective function values, which in turn can be used to evaluate the true unary indicator value of the set (or any subset). There are two crucial components in this algorithm: (1) given a data set of decision vectors and their corresponding objective function values, how to generate *training sets* for building the $\mathcal{GP}$ model over sets, and (2) given the trained $\mathcal{GP}$ model over sets, how to identify a new candidate set for evaluation.

In this work, when computing the hypervolume, we use a predefined reference point that remains constant throughout the algorithm. In our experiments, the reference point is set after generating the initial data set, by adding 1 to the worst value within the data set for each objective. In the case of ϵ-indicator, the reference front corresponds to the last (worst) front after applying nondominated sorting to the data set.

4.1 Generation of subsets for training

When using a $\mathcal{GP}$ model over sets, we first need to create a list of sets for training the model. We use Algorithm 2 for this purpose starting from a data set of solutions that stores both the decision $\mathbf{x}$ and its objective vectors $\mathbf{f}(\mathbf{x})$. Given the desired cardinality N_s of each generated training set and the data set A^0, Algorithm 2 returns a list of subsets of A_0, each of size N_s (the algorithm may internally update N_s if necessary as explained in the following).

At each iteration i, we construct a subset $S_i \subseteq A^0$ and delete those solutions from A, which starts as a copy of A^0. Each iteration begins by considering only the nondominated solutions that remain in A. If there are at least N_s nondominated solution, the call to $\text{SubsetSelect}(S_i, N_s)$ selects exactly N_s of them by removing one-by-one the solution that contributes the least to the hypervolume of the set. Otherwise, we have too few nondominated solutions and we need to add more solutions to S_i. In the first iteration ($i = 1$), we simply reduce the value of N_s and accept the subset S_1 as it is. In subsequent iterations, we search in the original data set A^0 (not in A, which may be empty at this point) for solutions that do not dominate the ones in the current subset and we select from those solutions the best with respect to the hypervolume contribution (i.e., using again SubsetSelect). In this manner, we try to add a solution that appears to be of high-quality when considered in isolation, yet it is dominated by the current subset S_i and, thus, it has no effect in the indicator value of S_i. We keep adding solutions until we have N_s or we cannot find a solution in A_0 that does not dominate any in S_i. In the latter case, we complete the subset by randomly selecting the remaining solutions from the previously generated subset (S_{i-1}), which means that some solutions in the current subset S_i may become dominated.

Once a subset S_i is completed, i.e., it contains exactly N_s solutions, those solutions are removed from A. The algorithm stops when A is empty, which means that every element of A^0 appears in at least one S_i.

4.2 Optimising indicators in Bayesian optimisation using sets

After creating the training sets and calculating their corresponding indicator values (hypervolume or ϵ), we build a $\mathcal{GP}$ model over sets as explained in Section 3. We use a genetic algorithm to estimate the parameters of the model by maximising the marginal likelihood. This model is then used in optimising the acquisition function to find a potential new set. In particular, we use the following expected improvement as the acquisition function:

$$\alpha_{EI}(X) = (I(X) - I_{\max})\Phi\big(\frac{I(X) - I_{\max}}{\sigma(X)}\big) + \sigma(X)\phi\big(\frac{I(X) - I_{\max}}{\sigma(X)}\big), \tag{8}$$

where $\Phi(Z)$ and $\phi(Z)$ are cumulative and probability distribution function of standard normal distribution. In the above equation, I is the posterior mean of the $\mathcal{GP}$ model at X, $I_{\max}$ is the maximum hypervolume or negative ϵ-indicator value at the current iteration and σ is the standard deviation from the $\mathcal{GP}$ model.

We apply the CMA-ES algorithm to maximise the acquisition function. For simplicity, we fixed the cardinality of the set to N_s, thus a set X is represented within CMA-ES as a vector of $N_s \times n$ decision

Algorithm 2: GenerateSets: Generate training sets.

Input: A^0: data set of evaluated solutions (decision and objective vectors), N_s: set size
Output: $(S_1, \ldots, S_i)$: list of sets

$i \leftarrow 1, A \leftarrow A^0$ // Create a copy
repeat
  $S_i \leftarrow$ filter dominated solutions in A
  if $|S_i| \geq N_s$ **then**
    $S_i \leftarrow \text{SubsetSelect}(S_i, N_s)$
  else if $i > 1$ **then**
    repeat
      // Find solutions in A^0 that do not dominate S_i and select the one that contributes the most to the hypervolume
      $A' \leftarrow \{f(\mathbf{x}') \in A^0 \mid \forall f(\mathbf{x}) \in S_i, f(\mathbf{x}') \not\preceq f(\mathbf{x})\}$
      $A' \leftarrow \text{SubsetSelect}(A', 1)$
      $S_i \leftarrow S_i \cup A'$
    until $|S_i| = N_s \vee A'$ *is empty*
    if $|S_i| < N_s$ **then**
      $S_i \leftarrow S_i \cup \text{SelectRandom}(S_{i-1}, N_s - |S_i|)$
  else
    $N_s \leftarrow |S_i|$
  $A \leftarrow A \setminus S_i$
  $i \leftarrow i + 1$
until *A is empty*
return $(S_1, \ldots, S_i)$

variables. CMA-ES returns a single set, which represents N_s decision vectors of the expensive problem. These decision vectors are then evaluated with expensive objective functions and the resulting solutions are added to the data set. The next iteration creates new training sets from the updated data set. The algorithm is stopped after a maximum number of expensive evaluations. The steps of the algorithm explained above are shown in Algorithm 3.

Algorithm 3: Optimising unary indicators using Bayesian optimisation over sets

Input: $D = (X, Y)$: data set, N_s: size of training sets
Output: Evaluated solutions

repeat
  $\mathcal{X} = (X_1, \ldots, X_N) \leftarrow \text{GenerateSets}(D, N_s)$
  // Compute the indicator value of each training set
  $\mathbf{I} = (I(X_1), \ldots, I(X_N))^T$
  $M \leftarrow$ Train $\mathcal{GP}$ model over sets on $(\mathcal{X}, \mathbf{I})$
  // Get a set by maximising the acquisition function:
  $X^* \leftarrow \arg\max_X \ \alpha(M, \mathcal{X}, \mathbf{I})$
  Evaluate solutions in X^* and add them to D
until *maximum number of expensive evaluations reached*

5 RESULTS AND DISCUSSION

The resulting algorithm was applied to DTLZ problems [6] with 2 and 3 objectives and $n = 3$ decision variables. We compare the results with those obtained by ParEGO [15] and EHVI-EGO [7, 24]. ParEGO scalarises the multiobjective optimisation problem into a single-objective one using weighted Chebyshev [23] and builds a $\mathcal{GP}$ model on it. It then uses expected improvement to find the next promising decision vector. The EHVI-EGO builds models for each objective function and maximises the expected hypervolume improvement to find a new promising decision vector. We used the implementation of EHVI available at https://liacs.leidenuniv.nl/~csmoda/index.php?page=code. In all algorithms, we used the following parameter values: the maximum number of function evaluations is 100, the size of the initial data set is 30 , and the maximum size of the sets (N_s) is 5 (only for Bayesian optimisation over sets). We replicated each run 11 times with different random seeds.

Figures 1 and 2 show the evolution of the hypervolume ratio (ratio of hypervolume of the nondominated solution at the current iteration to the hypervolume of the Pareto front) of the data set over number of expensive function evaluations for problems with 2 and 3 objectives respectively. We also show the IGD+ [10] with number of exepensive function evaluations in Figures 3 and 4. In most cases, BO over sets with hypervolume obtains similar results to ParEGO and EHVI-EGO. On the other hand, BO over sets with ϵ-indicator did not perform well on DTLZ6 and DTLZ7. The similar findings can also be observed in Figures 5 and 6, which show the nondominated solutions in the final data set of the different algorithms.

DTLZ2 is an easy problem and all algorithms converge close to the Pareto front. No algorithm performed well on DTLZ4. The reason is that DTLZ4 has a dense set of solutions near the f_m–f_1 plane resulting in variable density of solutions, which makes finding promising decision vectors challenging for an algorithm using $\mathcal{GP}$ models. The DTLZ5 problem has a degenerated Pareto front with a natural bias for solutions close to the Pareto front [4], which makes the problem easy to solve and, as a result, all algorithms converged close the Pareto front in DTLZ5. DTLZ6 is a difficult problem to solve because of many-to-one mapping. All algorithms found it difficult to solve DTLZ6 and BO over sets with ϵ-indicator performed the worst. DTLZ7 has a disconnected Pareto front but all the objectives are separable. This makes the problem easy to solve especially when the models are built for each objective function as in EHVI-EGO. The performance of the ϵ indicator needs to be investigated further.

The training time of the $\mathcal{GP}$ model with set based and non-set based kernel (as in ParEGO) is shown in Figure 7 (top). In this work, we used the Gaussian kernel in both set based and non-set based kernels. As mentioned in Section 3, the computational complexity is more than the traditional non-set based kernel, which can increase the training time of $\mathcal{GP}$ model. The training time can also be affected by the cardinality of the sets and we plan to investigate the sensitivity of cardinality in the performance and training time in the future. Further, the time to generate sets for training the $\mathcal{GP}$ model is shown in Figure 7 (bottom). As mentioned in Section 4.1, we used the hypervolume contribution to generate sets, thus it is expected that time increases with size of the data

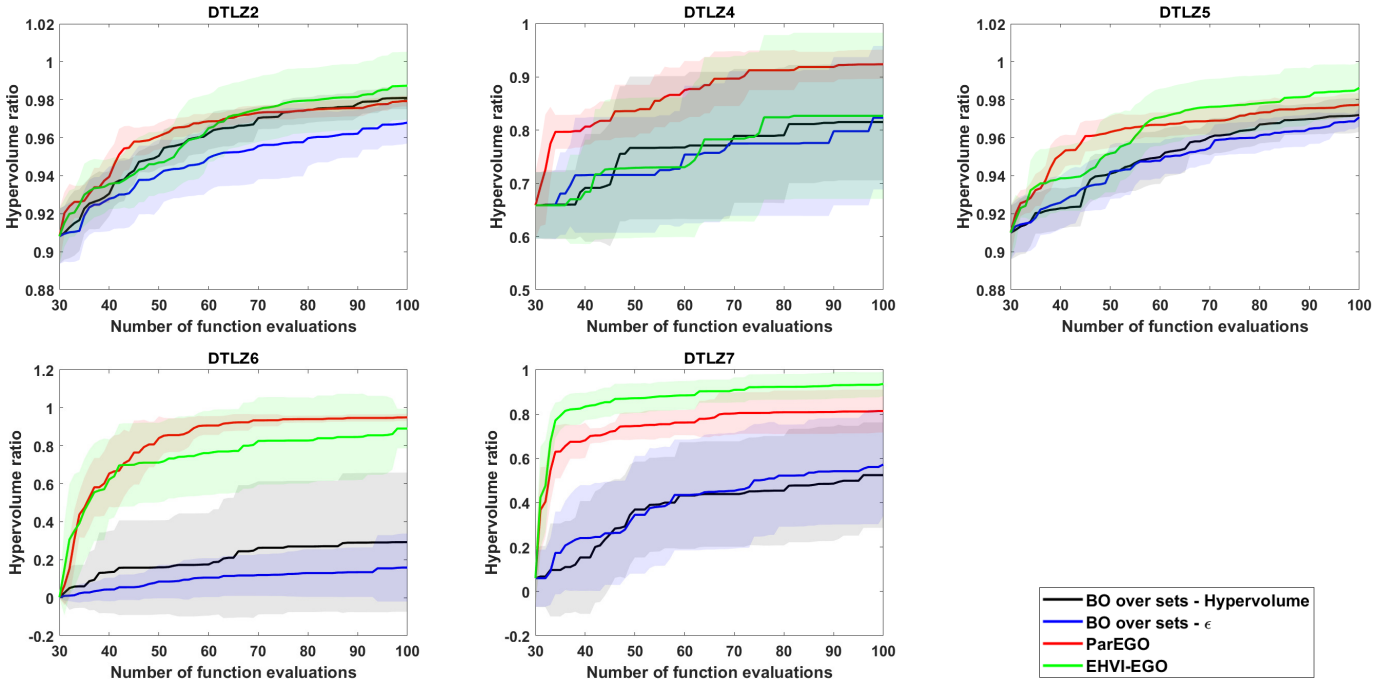

Figure 1: Performance of BO over sets with Hypervolume and ϵ-indicators, ParEGO and EHVI-EGO on DTLZ problems with two objectives

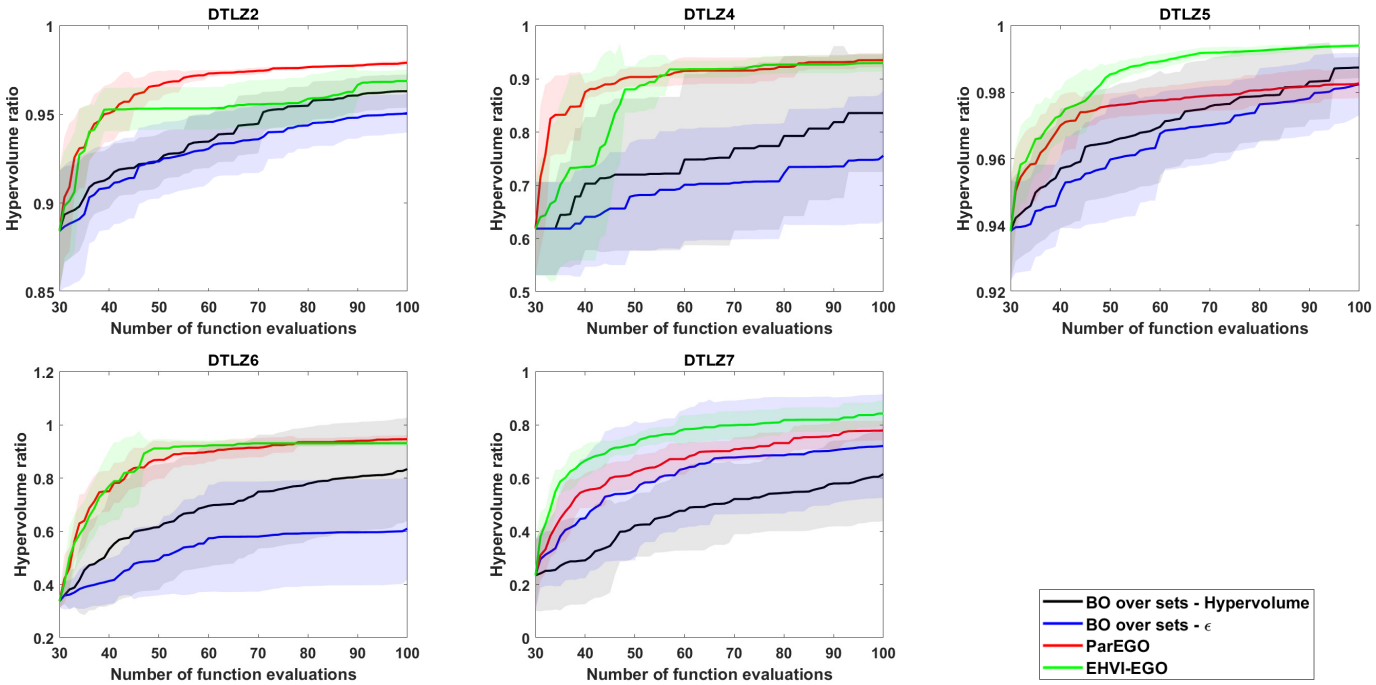

Figure 2: Performance of BO over sets with Hypervolume and ϵ-indicators, ParEGO and EHVI-EGO on DTLZ problems with three objectives

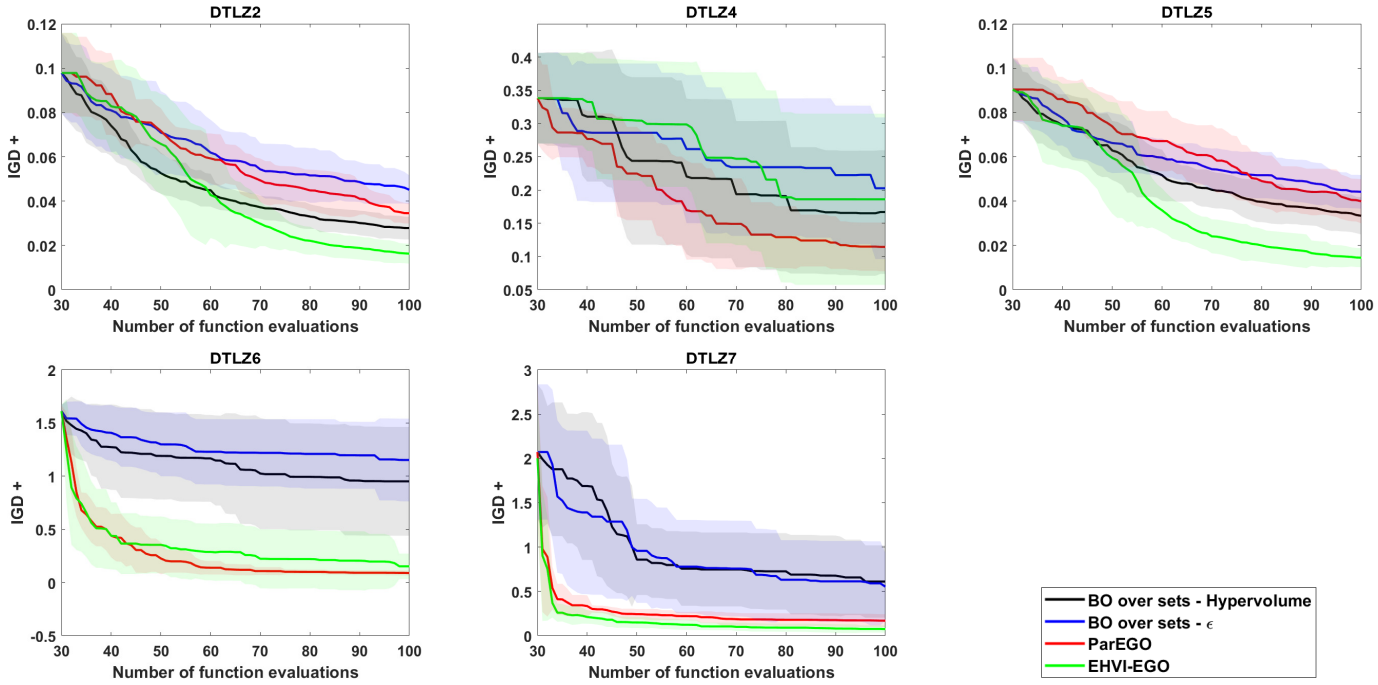

Figure 3: Performance of BO over sets with Hypervolume and ϵ-indicators, ParEGO and EHVI-EGO on DTLZ problems with two objectives

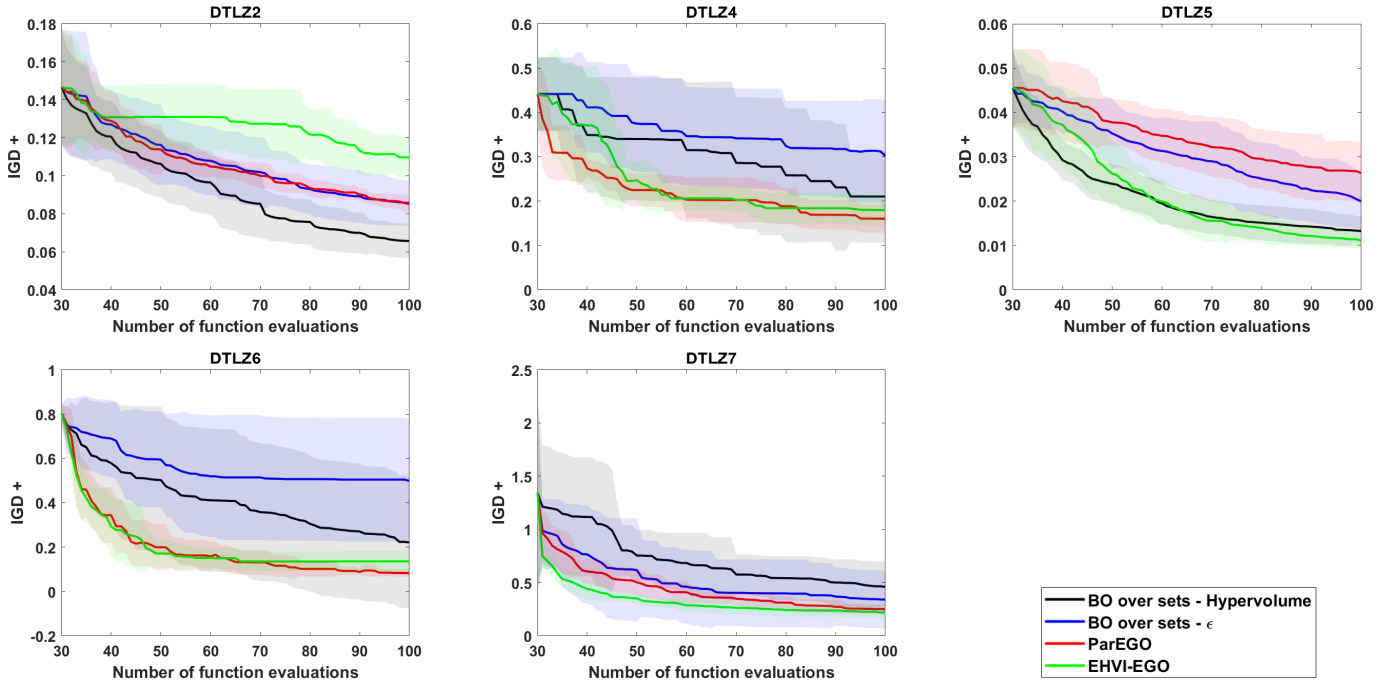

Figure 4: Performance of BO over sets with Hypervolume and ϵ-indicators, ParEGO and EHVI-EGO on DTLZ problems with three objectives

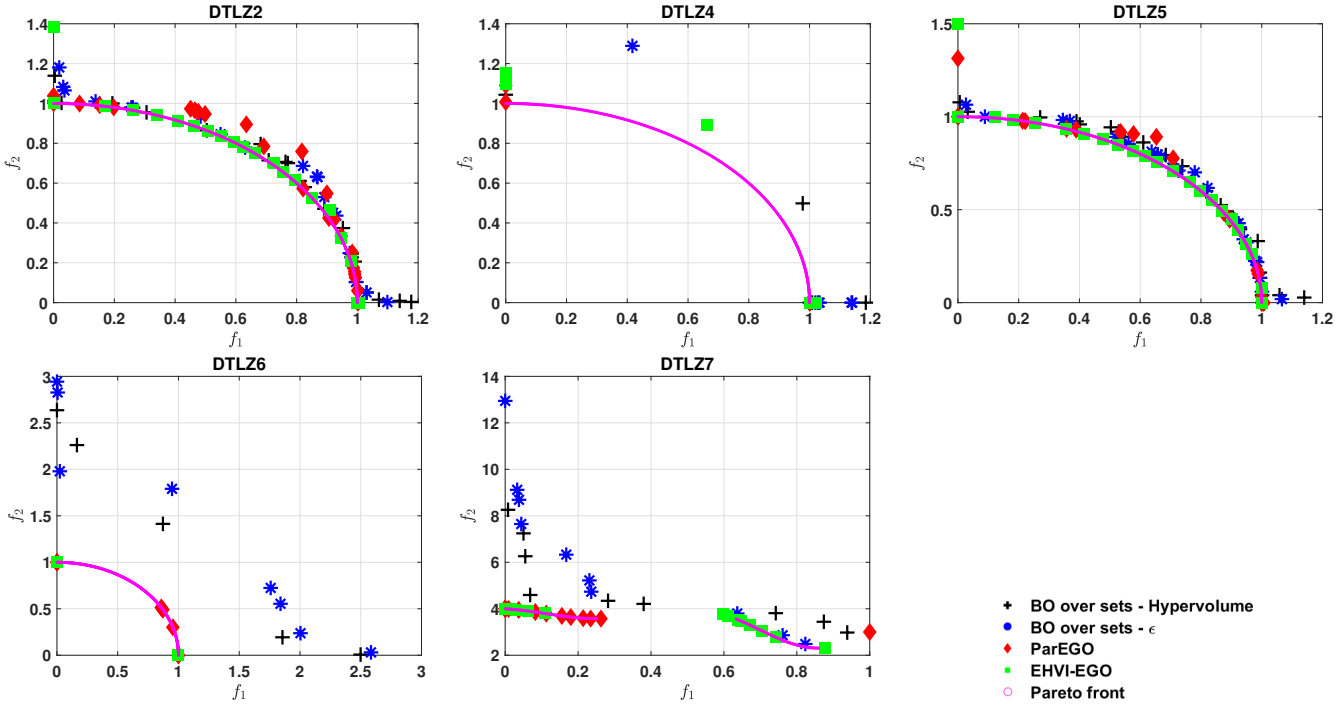

Figure 5: Nondominated solutions from BO over sets with Hypervolume and ϵ-indicators, ParEGO and EHVI-EGO on DTLZ problems with two objectives

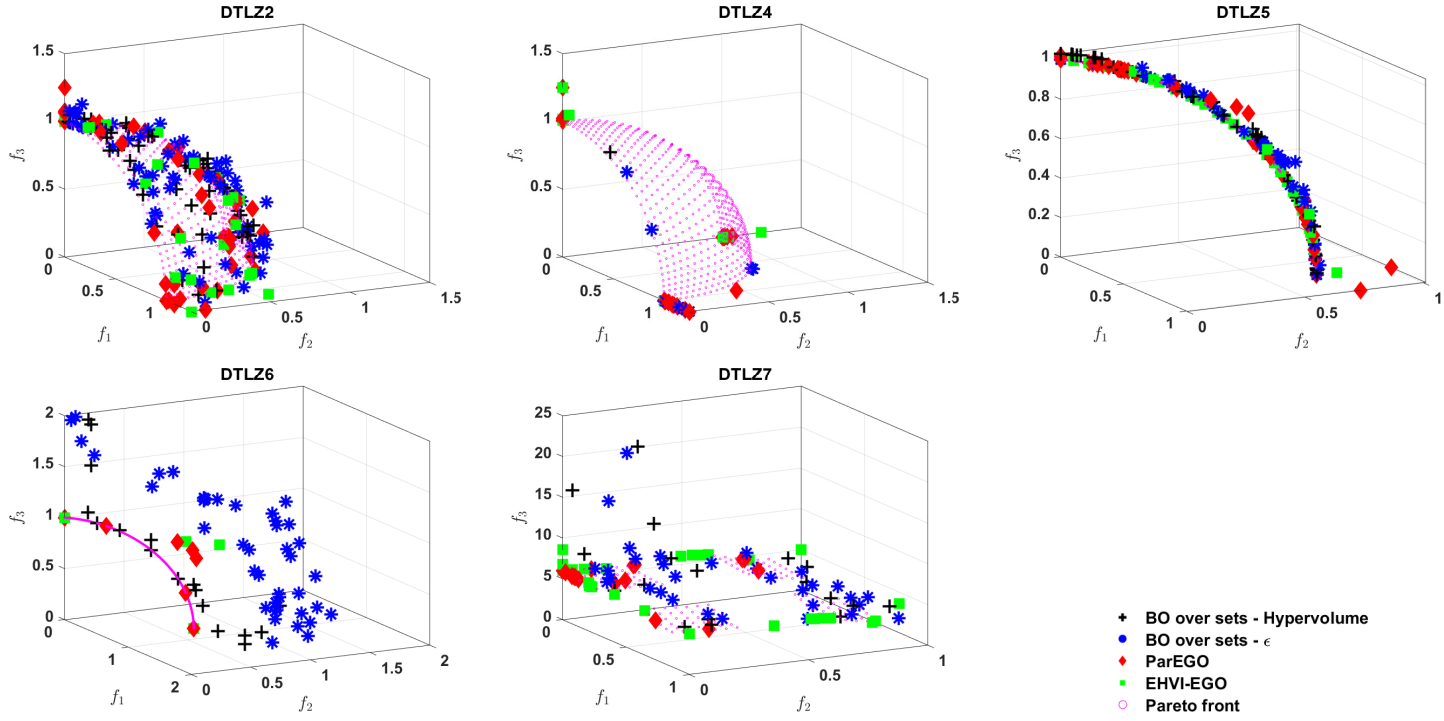

Figure 6: Nondominated solutions from BO over sets with Hypervolume and ϵ-indicators, ParEGO and EHVI-EGO on DTLZ problems with three objectives

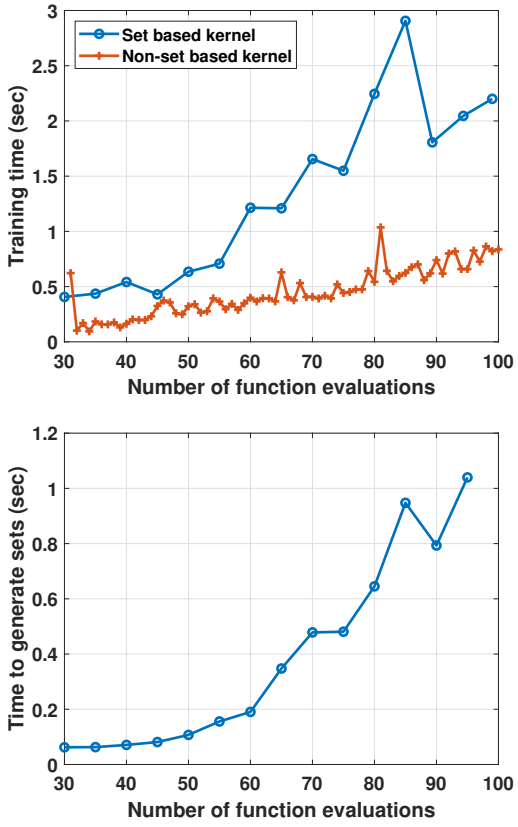

Figure 7: Training time of the $\mathcal{GP}$ model with set and non-set based kernel (top) and time to generate sets for training the set based $\mathcal{GP}$ (bottom). The cardinality of the sets in these figures is fixed to five.

set (or number of function evaluations). However, it is important to mention that the computation time of training the model and generating sets is assumed to be significantly lower than evaluating an expensive objective function.

6 CONCLUSION

In this work, we study the suitability of a set-based kernel in Gaussian processes to optimise unary quality indicators in multi-objective optimisation with expensive objective functions. We focused here on the hypervolume and multiplicative-ϵ indicators.

Our results show that the results of Bayesian optimisation over sets using the hypervolume indicator are comparable to those of exisiting Bayesian multiobjective optimisation methods with non-set based kernel, such as ParEGO and EHVI-EGO, with no method being the worst nor the best in all problems. On the other hand, Bayesian optimisation over sets using the ϵ-indicator did not perform well in general, being typically the worst method in all problems. This may be due to its weaker ability, compared to the hypervolume, to differentiate among sets. Moreover, in the computation of ϵ-indicator, we used as a reference set, the worst nondominated set in the data set, instead of the best nondominated set as it is usual in the computation of the unary ϵ-indicator. The performance of the ϵ indicator needs to be investigated further.

A set-based approach for Bayesian multi-objective optimisation has several advantages. First, the overall goal in multi-objective optimisation is to find a good approximation *set* of the Pareto front, thus a set-based approach more closely matches this goal. Second, in principle, there are more ways of generating training sets than the number of individual decision vectors in the data set, thus, in principle, one can generate more training data for a set-based model than for a non-set based one with the same number of expensive evaluations. Third, at each iteration, the set-based approach generates a new candidate set, whose elements may be evaluated in parallel, thus the proposed approach is suitable for batch evaluations.

Although the experiments here are preliminary and limited in scope, they suggest that Bayesian optimisation over sets may be a promising research direction in the context of expensive multi-objective optimisation. There are several possible improvements to the ideas presented here. First, our proposal for generating training sets is limited to a fixed and relatively small maximum cardinality, which is a user-defined parameter. Improvements in computational complexity are possible by approximating the set-based kernel, which would allow us to consider larger set cardinalities. The use of a variable set cardinality is also worth studying to remove the dependency on this user-defined parameter. Second, alternative approaches for generating the training sets are clearly possible and they are likely to strongly affect the performance of Bayesian optimisation over sets. Third, extending the approach to other quality indicators over sets (or combinations thereof) would also be interesting. Finally, a more extensive empirical analysis considering other benchmarks and larger number of objectives and decisions variables would help to understand the scalability of Bayesian optimisation over sets.

Reproducibility. Source code and data sets required to reproduce the results presented in this paper are available at doi:10.5281/zenodo.4675569 .

ACKNOWLEDGMENTS

M. López-Ibáñez is a "Beatriz Galindo" Senior Distinguished Researcher (BEAGAL 18/00053) funded by the Ministry of Science and Innovation of the Spanish Government.

REFERENCES

[1] Nicola Beume, Boris Naujoks, and Michael T. M. Emmerich. 2007. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* 181, 3 (2007), 1653–1669. https://doi.org/10.1016/j.ejor.2006.08.008

[2] Leonardo C. T. Bezerra, Manuel López-Ibáñez, and Thomas Stützle. 2018. A Large-Scale Experimental Evaluation of High-Performing Multi- and Many-Objective Evolutionary Algorithms. *Evolutionary Computation* 26, 4 (2018), 621–656. https://doi.org/10.1162/evco_a_00217

[3] Tinkle Chugh. 2020. Scalarizing Functions in Bayesian Multiobjective Optimization. In *Proceedings of the 2020 Congress on Evolutionary Computation (CEC 2020)*. IEEE Press, Piscataway, NJ, 1–8. https://doi.org/10.1109/CEC48606.2020.9185706

[4] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, NY.

[5] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. 2020. Differentiable Expected Hypervolume Improvement for Parallel Multi-Objective Bayesian Optimization. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 9851–9864. https://proceedings.neurips.cc/paper/2020/file/6fec24eac8f18ed793f5eaad3dd7977c-Paper.pdf

[6] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. 2005. Scalable Test Problems for Evolutionary Multiobjective Optimization. In *Evolutionary Multiobjective Optimization*, Ajith Abraham, Lakhmi Jain, and Robert Goldberg (Eds.). Springer, London, UK, 105–145.

[7] Michael T. M. Emmerich, K. C. Giannakoglou, and Boris Naujoks. 2006. Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation* 10, 4 (2006), 421–439. https://doi.org/10.1109/TEVC.2005.859463

[8] R. Fletcher. 1987. *Practical methods of optimization*. John Wiley & Sons, New York, NY.

[9] Nikolaus Hansen and A. Ostermeier. 2001. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation* 9, 2 (2001), 159–195. https://doi.org/10.1162/106365601750190398

[10] Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. 2015. Modified Distance Calculation in Generational Distance and Inverted Generational Distance. In *Evolutionary Multi-Criterion Optimization*, António Gaspar-Cunha, Carlos Henggeler Antunes, and Carlos Coello Coello (Eds.). Springer International Publishing, Cham, 110–125.

[11] S. Jiang, Y. S. Ong, J. Zhang, and L. Feng. 2014. Consistencies and Contradictions of Performance Metrics in Multiobjective Optimization. *IEEE Transactions on Cybernetics* 44, 12 (2014), 2391–2404.

[12] Yaochu Jin, Handing Wang, Tinkle Chugh, Dan Guo, and Kaisa Miettinen. 2019. Data-Driven Evolutionary Optimization: An Overview and Case Studies. *IEEE Transactions on Evolutionary Computation* 23, 3 (June 2019), 442–458. https://doi.org/10.1109/tevc.2018.2869001

[13] D. R. Jones, M. Schonlau, and W. J. Welch. 1998. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization* 13, 4 (1998), 455–492.

[14] Jungtaek Kim, Michael McCourt, Tackgeun You, Saehoon Kim, and Seungjin Choi. 2021. Bayesian Optimization with Approximate Set Kernels. *Machine Learning* (2021). https://doi.org/10.1007/s10994-021-05949-0

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

[16] H. J. Kushner. 1964. A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Basic Engineering* 86, 1 (March 1964), 97–106. https://doi.org/10.1115/1.3653121

[17] R. M. Lark and D. J. Lapworth. 2014. A new statistic to express the uncertainty of kriging predictions for purposes of survey planning. In *EGU General Assembly Conference Abstracts*. Article 2183. https://ui.adsabs.harvard.edu/abs/2014EGUGA..16.2183L

[18] Arnaud Liefooghe, Manuel López-Ibáñez, Luís Paquete, and Sébastien Verel. 2018. Dominance, Epsilon, and Hypervolume Local Optimal Sets in Multi-objective Optimization, and How to Tell the Difference. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018*, Hernán E. Aguirre and Keiki Takadama (Eds.). ACM Press, New York, NY, 324–331. https://doi.org/10.1145/3205455.3205572

[19] Jonas Močkus. 1975. On Bayesian Methods for Seeking the Extremum. In *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, G. I. Marchuk (Ed.). Lecture Notes in Computer Science, Vol. 27. Springer, Heidelberg, Germany, Berlin, Heidelberg, 400–404. https://doi.org/10.1007/3-540-07165-2_55

[20] Alma A. M. Rahat, Richard M. Everson, and Jonathan E. Fieldsend. 2017. Alternative infill strategies for expensive multi-objective optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017*, Peter A. N. Bosman (Ed.). ACM Press, New York, NY, 873–880.

[21] Carl Edward Rasmussen and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA.

[22] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and Nando de Freitas. 2016. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* 104, 1 (2016), 148–175.

[23] A. P. Wierzbicki. 1980. The Use of Reference Objectives in Multiobjective Optimisation. In *MCDM theory and Application, Proceedings, Hagen*, G. Fandel and T. Gal (Eds.). Number 177 in Lecture Notes in Economics and Mathematical Systems. Springer, Heidelberg, Germany, 468–486.

[24] Kaifeng Yang, Michael T. M. Emmerich, André H. Deutz, and Thomas Bäck. 2019. Multi-Objective Bayesian Global Optimization using Expected Hypervolume Improvement Gradient. *Swarm and Evolutionary Computation* 44 (Feb. 2019), 945–956. https://doi.org/10.1016/j.swevo.2018.10.007

[25] Carlos Yasojima, Tiago Araújo, Bianchi Meiguins, Nelson Neto, and Jefferson Morais. 2019. A Comparison of Genetic Algorithms and Particle Swarm Optimization to Estimate Cluster-Based Kriging Parameters. In *Progress in Artificial Intelligence*, Paulo Moura Oliveira, Paulo Novais, and Luís Paulo Reis (Eds.). Springer International Publishing, Cham, Switzerland, 750–761.

[26] Eckart Zitzler and Simon Künzli. 2004. Indicator-based Selection in Multiobjective Search. In *Proceedings of PPSN-VIII, Eigth International Conference on Parallel Problem Solving from Nature*, Xin Yao et al. (Eds.). Lecture Notes in Computer Science, Vol. 3242. Springer, Heidelberg, Germany, 832–842.

[27] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert da Fonseca. 2003. Performance Assessment of Multiobjective Optimizers: an Analysis and Review. *IEEE Transactions on Evolutionary Computation* 7, 2 (2003), 117–132. https://doi.org/10.1109/TEVC.2003.810758