Mono-surrogate vs Multi-surrogate in Multi-objective Bayesian Optimisation

Tinkle Chugh
t.chugh@exeter.ac.uk
Department of Computer Science
University of Exeter
Exeter, United Kingdom

ABSTRACT

Bayesian optimisation (BO) has been widely used to solve problems with expensive function evaluations. In multi-objective optimisation problems, BO aims to find a set of approximated Pareto optimal solutions. There are typically two ways to build surrogates in multi-objective BO: One surrogate by aggregating objective functions (by using a scalarising function, also called mono-surrogate approach) and multiple surrogates (for each objective function, also called multi-surrogate approach). In both approaches, an acquisition function (AF) is used to guide the search process. Mono-surrogate has the advantage that only one model is used, however, the approach has two major limitations. Firstly, the fitness landscape of the scalarising function and the objective functions may not be similar. Secondly, the approach assumes that the scalarising function distribution is Gaussian, and thus a closed-form expression of the AF can be used. In this work, we overcome these limitations by building a surrogate model for each objective function and show that the scalarising function distribution is not Gaussian. We approximate the distribution using Generalised extreme value distribution. The results and comparison with existing approaches on standard benchmark and real-world optimisation problems show the potential of the multi-surrogate approach.

KEYWORDS

Bayesian optimisation, Surrogate modelling, Gaussian process, Approximate inference, Bayesian Optimisation, Uncertainty

1 INTRODUCTION

Many real-world optimisation problems involve multiple conflicting objectives to be achieved. These problems are called multi-objective optimisation problems (MOPs). There is no single solution to such problems because of the conflicting nature between the objectives. The solutions to such problems are known as Pareto optimal solutions, and represent the trade-off between objectives [16]. We define a multi-objective optimisation problem (MOP) as:

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

with $m \geq 2$ objective functions $f_i(\mathbf{x}): S \rightarrow \mathbb{R}$. The (nonempty) feasible space S is a subset of the decision space $\mathbb{R}^n$ and consists of decision vectors $\mathbf{x} = (x_1, \ldots, x_n)^T$. In solving such problems, usually, the aim is to find an approximated set of Pareto optimal solutions.

In many real-world MOPs, the objective functions rely on computationally expensive evaluations. Such problems are usually black-box optimisation problems without any closed-form for the objective functions. Bayesian optimisation (BO) can be used to alleviate the computational cost and to find an approximated set of Pareto optimal solutions in the least number of function evaluations. These methods rely on a Bayesian model as the surrogate (or metamodel) of the objective functions and find promising decision vectors by optimising an acquisition function. The Bayesian model is usually a Gaussian process because it provides a meaningful quantification of uncertainty, which is then used in optimising the acquisition function. The acquisition function balances both exploration and exploitation in guiding the search process. A single objective BO with expected improvement (EI) as the acquisition function was first used in [17]. Emmerich et al. [6–9, 28] extended the EI to expected hypervolume improvement (EHVI) in multi-objective Bayesian optimisation. Recently, a closed-form expression of EHVI was proposed for batch multi-objective Bayesian optimisation [3].

In multi-objective BO, there are typically two different approaches to build a Bayesian model. In the first one, the models are built for each objective function and an acquisition function utilising these models is then used to find promising decision vectors. This approach is called multi-surrogate approach. The multi-objective BO with EHVI is a multi-surrogate approach. In the second one, a single Bayesian model is built after aggregating the objective functions. This approach is called mono-surrogate approach. The well-known ParEGO [12] algorithm comes under the second category. The second approach reduces the number of objectives from m to one. Moreover, a single objective acquisition function can be used in the mono-surrogate approach. The computational complexity of the first approach is at most $O(mN^3)$ and of the second approach is at most $O(N^3)$, where N is the size of the data set.

The ParEGO converts the multiple objectives into a single objective by utilising the augmented weighted Tchebycheff (TCH) [27] as the scalarising function. A Gaussian process model is then built on the scalarising function, which is then used in optimising the EI to find the next promising decision vector. Although the mono-surrogate approach is appealing, it has two major limitations. The first one is that the fitness landscape of the scalarising function and the objective functions may not be similar. In other words, a promising decision vector by using the surrogate on the scalarising function may not be promising for the underlying objective functions. The second limitation is that the approach assumes that the resulting scalarising function is Gaussian and thus a closed-form expression of the EI can be used.

We show that the distribution of the scalarising function after building individual models for each objective function is not Gaussian. Therefore, we approximate the distribution of the scalarising function using the Generalised extreme value (GEV) distribution [4, 26]. Particularly, we use the type I family of distributions in

the generalised extreme value theory to approximate the distribution. The GEV distribution is then used in optimising the expected improvement to find the next promising decision vector.

We compare mono-surrogate EI (as in ParEGO), multi-surrogate EI (this work) and multi-surrogate EHVI (as in EHVI-EGO [28]) approaches on standard benchmark and real-world optimisation problems with a different number of objectives and decision variables. The results on benchmark and real-world multi-objective optimisation problems clearly show the potential of the proposed work. To be summarised, the contributions of this work are as follows:

(1) We show that the weighted Tchebycheff function (TCH) as the scalarising function after building independent models for each objective function is not Gaussian.
(2) We approximate the distribution of the TCH with the type I family of the Generalised extreme value theory.
(3) We solve a real-world multi-objective free-radical polymerisation optimisation problem by using the multi-surrogate EI approach.

The rest of the paper is structured as follows. In Section 2, we provide a background to the Bayesian optimisation. In Section 3, we provide the main details of the proposed approach by comparing it with a mono-surrogate approach. In Section 4, we conduct numerical experiments and show the results on benchmark and real-world multi-objective optimisation problems. Finally, we conclude and mention the future research direction in Section 5.

2 BAYESIAN OPTIMISATION

In multi-objective BO, the input is the data set $\{(\mathbf{x}_i, \mathbf{f}(\mathbf{x}_i))\}_{i=1}^N$ of size N. This data set can be obtained with a design of experiment technique e.g. Latin Hypercube sampling [15]. As mentioned in the introduction, there are typically two approaches to build surrogate models on the data set: A single Bayesian model (mono-surrogate approach) or multiple Bayesian models (multi-surrogate approach). The Gaussian process models are the most commonly used Bayesian models in BO. They are non-parametric and provide uncertainties in predictions, which makes them different from other non-Bayesian and parametric models. The uncertainty is then used in the acquisition function in finding the promising decision vector. A Gaussian process ($\mathcal{GP}$) can be defined with a multivariate normal distribution [23]:

$$\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}, K),$$

where $\boldsymbol{\mu}$ is the mean vector and K is the covariance matrix. Without loss of generality, we use a prior of zero mean. A covariance function (or kernel) is used to get the covariance matrix. In this work, we used a Gaussian (or RBF or squared exponential) kernel with automatic relevant determination [21, 24]:

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

where $\Theta = (\sigma_f, l_1, \ldots, l_n, \sigma_n)$ is the set of hyperparameters and $\delta_{\mathbf{x}\mathbf{x}'}$ is the Kronecker delta function. The notation $|x_j - x_j'|$ represents the Euclidean distance between x_j and x_j'. The hyperparameters can be estimated by maximising the marginal likelihood function

Algorithm 1 Bayesian optimisation

Input: Data Set $D = \{(\mathbf{x}_i, \mathbf{f}_i)\}_{i=1}^N$
Output: Evaluated solutions
1: **while** Termination criterion is not met **do**
2: Train the $\mathcal{GP}$ models on the data set
3: Optimise the acquisition function i.e. $\mathbf{x}^* \leftarrow \text{argmax}_{\mathbf{x}}\, EI(x)$
4: Evaluate $\mathbf{x}^*$ and add to the data set

[23]:

$$p(\mathbf{f}|X, \boldsymbol{\Theta}) = \frac{1}{|2\pi K|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\mathbf{f}^T K^{-1}\mathbf{f}\right\}.$$

The posterior predictive distribution at new point $\mathbf{x}^*$after training the model is also Gaussian:

$$p\left(f^*|\mathbf{x}^*, X, \mathbf{f}, \boldsymbol{\Theta}\right) = \mathcal{N}\Big(\boldsymbol{\kappa}(\mathbf{x}^*, X)K^{-1}\mathbf{f}, \kappa(\mathbf{x}^*, \mathbf{x}^*) - \boldsymbol{\kappa}(\mathbf{x}^*, X)^T K^{-1}\boldsymbol{\kappa}(X, \mathbf{x}^*)\Big).$$

The acquisition function determines the next decision vector to be evaluated. The expected improvement (EI) is one of the well-known and widely used acquisition functions. It measures the amount of improvement over the current best objective value and balances both exploitation and exploration. For a minimisation problem, the improvement over the best evaluated function value $f'(\mathbf{x})$ is:

$$I(\mathbf{x}) = \max(0, f'(\mathbf{x}) - f).$$

The expected improvement can then be estimated as:

$$\alpha_{EI}(\mathbf{x}) = \int_{-\infty}^{f'(\mathbf{x})} I(x) df$$

As the posterior is Gaussian, the expected improvement has a closed-form expression:

$$\alpha_{EI}(\mathbf{x}) = (f'(\mathbf{x}) - \mu(\mathbf{x}))\Phi\Big(\frac{f'(\mathbf{x}) - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\Big) + \sigma(\mathbf{x})\phi\Big(\frac{f'(\mathbf{x}) - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\Big),$$

where $\mu(x)$ and $\sigma(x)$ is the posterior mean and standard deviation, respectively and $\Phi(\cdot)$ and $\phi(\cdot)$ are cumulative and probability distribution function of standard normal distribution, respectively.

In mono-surrogate multi-objective BO, this formulation of EI can be used as the objective functions are aggregated into a single function. In the multi-surrogate approach, EHVI, which is an extension of EI for multiple objectives can be used. In this work, we focus on EI after using multiple surrogates. The EI is usually a multimodal function and therefore, a suitable optimiser e.g. an evolutionary algorithm is often used to optimise it to find the next decision vector, which is then evaluated with expensive objective function and added to the data set. This process continues until a termination criterion is met. Algorithm 1 outlines these steps.

3 MONO-SURROGATE VS MULTI-SURROGATE

The mono-surrogate approach is one of the well-known approaches in Bayesian optimisation. We start by introducing the weighted Tchebycheff (TCH) function [1, 16, 27] in a widely used mono-surrogate approach (the ParEGO algorithm). The TCH function is

defined as:

$$g = \max_i (w_i(f_i - z_i)),$$

where $\mathbf{w}$ is the weight vector and $\mathbf{z}$ is the ideal objective vector (or minimum of objective function values). In ParEGO, an augmented Tchebycheff formulation was used to obtain properly Pareto optimal solutions [16]. If the objective function values at the current iteration are normalised between 0 and 1, then $\mathbf{z}$ is a vector of zeros. Given a data set with decision variable and objective function values, the TCH converts the multiple objective function values into a single value. A Gaussian process model is then built on the resulting data set, which is then used in optimising the expected improvement. Recently, three other scalarising functions called hypervolume improvement, dominance ranking and sign distance were proposed in [22] and used in the framework of the mono-surrogate approach.

In this work, we use a multi-surrogate approach by building independent models on each objective function. These independent models are then used to build a probabilistic model for the TCH function, which is then used in optimising the EI. We start by providing a simple example to compare both approaches with TCH as the scalarising function.

Consider two objective functions f_1 and f_2 shown in Figure 1. We assume that these functions are black box and their analytical forms are not available. However, we have some data set (with decision variable and objective functions values) shown as + in the figure. The true or underlying TCH function is also shown in the figure. As mentioned, there are two ways to build surrogate models: mono-surrogate and multi-surrogate. In the mono-surrogate approach, we get the TCH values after aggregating two objective values (right plot in Figure 1). We then build a Gaussian process model on it as shown in Figure 2. We then maximise the EI to get the next decision variable value i.e. $x^* = \text{argmax EI}$. The location of x^* can be seen in Figure 2. We then evaluate the x^* with the objective functions. The resulting objective function values are shown as square in the right in Figure 2. In this way, the mono-surrogate approach tries to find a potential decision vector for the scalarising function and not for the underlying objective functions. To address this concern, we can build independent models for objective functions.

The predictions with uncertainty estimates after building independent models are shown in Figure 3. The challenge is how to use these models to find the distribution of the TCH as the scalarising function. We start to address this challenge by showing that the resulting scalarising function after building independent models on objective functions is not Gaussian. The weighted Tchebycheff function is defined as:

$$g = \max_i (w_i(f_i - z_i)),$$

where f_i is Gaussian i.e. $f_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$; μ_i and σ_i are the posterior predictive means and standard deviations, respectively. The above formulation can be written as [13, 19]:

$$g \sim \max_i \mathcal{N}\Big(w_i(\mu_i - z_i), w_i^2\sigma_i^2\Big) \tag{1}$$

After some rearrangements, the distribution shown above can be written in the following closed form expression [2, 14, 20]:

$$p(g) = \sum_{i=1}^{m} \frac{1}{w_i\sigma_i} \times \frac{\phi\Big(\frac{g - w_i(\mu_i - z_i)}{w_i\sigma_i}\Big)}{\Phi\Big(\frac{g - w_i(\mu_i - z_i)}{w_i\sigma_i}\Big)} \times \prod_{i=1}^{m} \Phi\Big(\frac{g - w_i(\mu_i - z_i)}{w_i\sigma_i}\Big), \tag{2}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ represent the probability density and cumulative density functions of the standard normal distribution, respectively. The distribution in Equation (2) has a closed-form expression but it is not Gaussian distributed. Therefore, a closed-form expression of acquisition functions, which rely on the Gaussian assumption of the function cannot be used. A possible solution to this problem is to approximate the distribution with Generalised extreme value theory [4, 26]. The distribution can be described with Type 1 distribution in Generalised value distributions. Specifically, we use a Gumbel distribution [25] for approximating the scalarising function:

$$g \sim \text{Gumbel}\ (\alpha, \beta)$$

where α and β are location and scale parameters, respectively. The probability density function is:

$$p(g_i|\alpha, \beta) = \frac{1}{\beta} e^{-(t_i + e^{-t_i})} \quad \text{for} \quad i = 1, \ldots, N$$

where $t_i = \frac{g_i - \alpha}{\beta}$. We can estimate the parameters by maximising the following log-likelihood function

$$LL(\alpha, \beta) = N \log \frac{1}{\beta} - \sum_{i=1}^{N} t_i - \sum_{i=1}^{N} e^{-t_i},$$

where N is the number of samples drawn from Equation (1) or (2). The estimated parameters are:

$$\alpha, \beta = \underset{\alpha,\beta}{\text{argmax}} \prod_{i=1}^{N} p(g_i|\alpha, \beta) \ \text{ Or}$$

$$\alpha, \beta = \underset{\alpha,\beta}{\text{argmax}} \left(N \log \frac{1}{\beta} - \sum_{i=1}^{N} t_i - \sum_{i=1}^{N} e^{-t_i}\right)$$

After taking the partial derivatives, the parameters can be estimated by solving the following two equations:

$$\beta = \bar{g} - \frac{\sum_{i=1}^{N} g_i e^{-\frac{g_i}{\beta}}}{\sum_{i=1}^{N} e^{-\frac{g_i}{\beta}}}$$

$$\alpha = -\beta \log\left(\frac{1}{N}\sum_{i=1}^{N} e^{\frac{-g_i}{\beta}}\right)$$

Once the parameters are known, we can use the approximated distribution in estimating the EI:

$$\alpha_{EI} = \int_{-\infty}^{g'(\mathbf{x})} \max(0, g'(x) - g)\, dg$$

One major drawback is that the EI does not have a closed-form expression. Therefore, we use Monte Carlo for estimating the EI. To summarise, we first approximate the distribution of TCH using Gumbel distribution and then draw samples from the distribution to estimate EI. For the two-objective example, the resulting TCH predictions and uncertainty estimates after building independent

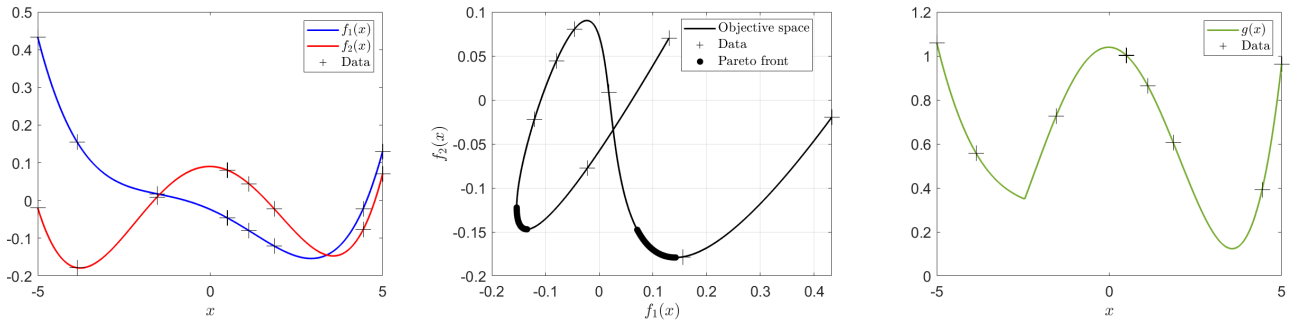

Figure 1: A bi-objective optimisation problem. Both objectives are to be minimised. The data is shown in '+' in the left figure. The Pareto front and the objective space are shown in the middle figure. The resulting scalarising function with the data set is shown in the right figure.

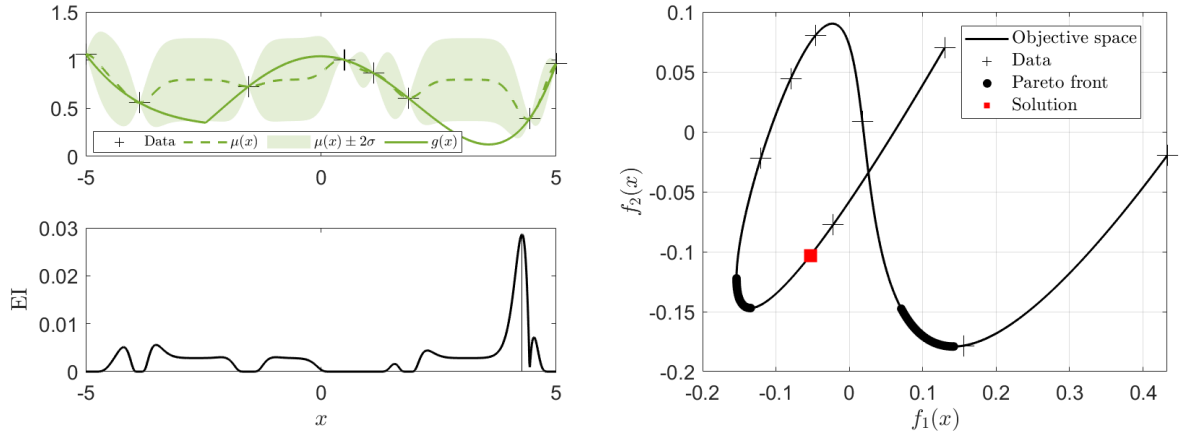

Figure 2: A Gaussian process model and the landscape of EI on the weighted Tchebycheff function (left figure). The evaluated solution (shown as the square marker) in the objective space after maximising the EI (right figure).

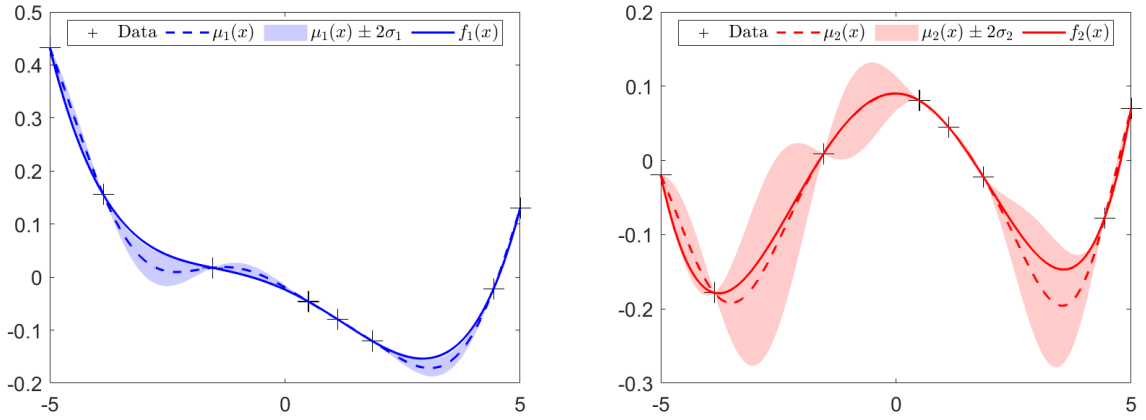

Figure 3: A Gaussian process model for each objective function.

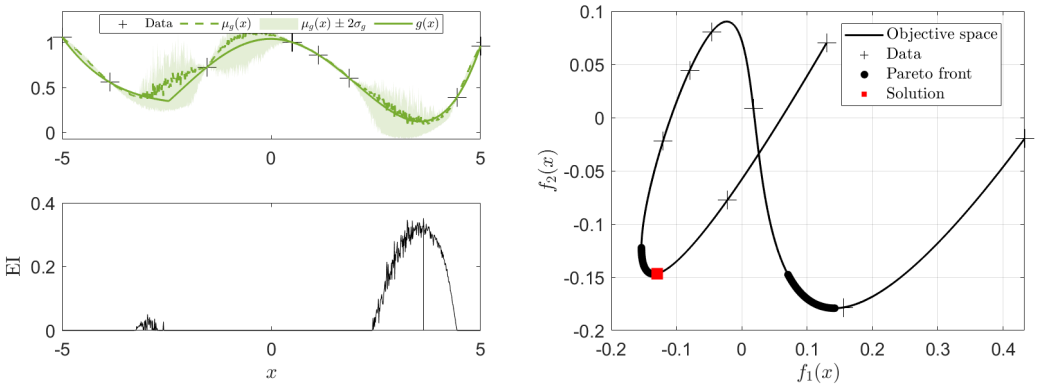

Figure 4: A Gaussian process model and the landscape of EI on the weighted Tchebycheff function after building independent models for each objective function (left figure). The evaluated solution (shown as the square marker) in the objective space after maximising the EI (right figure).

Gaussian process models on objective functions are shown in Figure 4. We also show the landscape of the expected improvement in the figure and can see that the optimal location is different from the mono-surrogate approach. The resulting decision variable is then evaluated with underlying objective functions and as can be seen, the objective function values lie on the Pareto front. This demonstration on an easy one-dimensional two objective optimisation problem shows that the multi-surrogate approach is better and finding an appropriate distribution of the scalarising function is important.

4 NUMERICAL EXPERIMENTS

We investigate the performance of the multi-surrogate approach on standard DTLZ [5] benchmark and a real-world Free-Radical Polymerisation problem [18]. We compare it to the ParEGO as the mono-surrogate approach and multi-surrogate expected hypervolume improvement (EHVI) [7, 28]. We implemented all different approaches and used the same settings wherever possible for a fair comparison. The numerical settings used are as follows:

- Problems: DTLZ (2, 5 and 7), Free-Radical Polymerisation
- Number of objectives: 2 and 3
- Number of decision variables: 4 and 5
- Number of independent runs = 11
- Size of the initial data set: 10 × number of decision variables
- Maximum number of function evaluations: 30 × number of decision variables
- Kernel: Squared exponential (or Radial basis function, Gaussian) with automatic relevant determination
- Optimiser to maximise acquisition functions: Genetic Algorithm
- Optimiser to maximise marginal likelihood in Gaussian process: BFGS with 10 restarts
- Performance indicator: Hypervolume

The hypervolume with the number of function evaluations for DTLZ problems with five decision variables and two objectives are shown in Figure 5. As can be seen, the multi-surrogate approach outperformed the other two approaches. The performance of the mono-surrogate approach can be explained with our analysis in the previous section. In the multi-surrogate EHVI approach, the hypervolume contributions of solutions dominated by the current approximated Pareto front is zero, which can make it difficult for the algorithm to converge. The same concern in using EHVI was also mentioned in [22]. The multi-surrogate approach did not perform well on the DTLZ7 problem. The problem has a disconnected Pareto front and separable objective functions. The performance of the multi-surrogate approach can be related to the shape of the Pareto front and is worth exploring as the future research direction. The approximated Pareto fronts with three different approaches of the run with the median hypervolume are shown in Figure 6.

The real-world problem is Free-Radical Polymerisation problem. The problem is the manufacturing of Polyvinyl Acetate polymer. The polymer is manufactured in a batch reactor and the process can be modelled with a series of ordinary differential equations. These equations are stiff and therefore solving these equations can be computationally expensive. The computation time varies from 30 seconds to 20 minutes for one evaluation. The problem involves four decision variables: monomer concentration, initiator concentration, the temperature of the rector and polymerisation time and three objectives: maximise weight average molecular weight (MW), number average molecular weight (MN) and minimise polydispersity index (PDI). The hypervolume and approximated Pareto fronts after running mono-surrogate and multi-surrogate approaches are shown in Figure 7. As can be seen, the multi-surrogate approach outperformed the mono-surrogate approach. We also show the two dimensional scatter plots between different objectives in Figure 8.

The multi-surrogate approach relies on approximations and Monte Carlo simulations when estimating EI, which makes the optimisation process slower. The computation time on the DTLZ2 problem of one run with two objectives and five variables of both approaches is shown in Figure 9. As can be seen, the multi-surrogate approach is slower than the mono-surrogate approach. However, this computation time may not be significant in many real-world applications and may be negligible compared to the objective function evaluation.

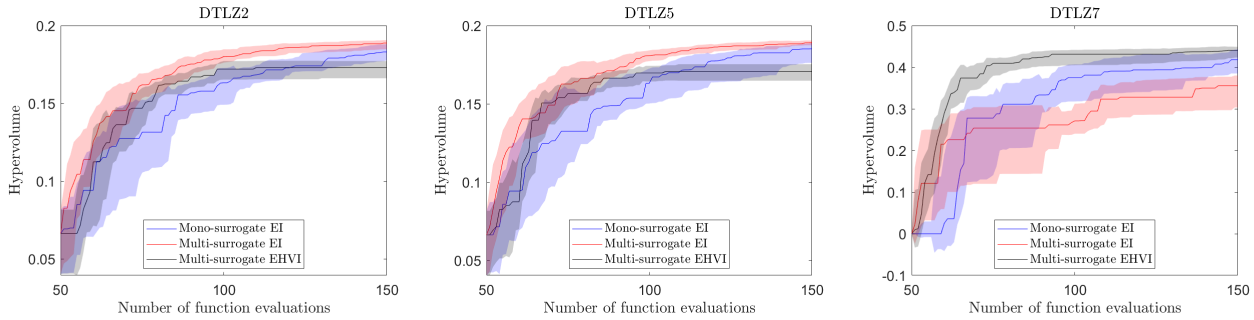

Figure 5: Hypervolume with the number of function evaluations on DTLZ problems with two objectives and five decision variables. The solid line is the median and shaded region is 95% confidence interval.

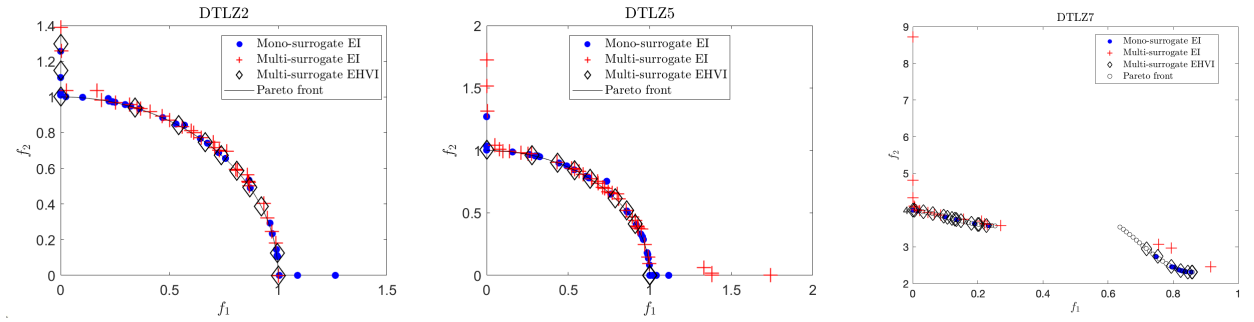

Figure 6: Approximated Pareto fronts of the run with the median hypervolume value.

5 CONCLUSIONS

In this work, we presented a multi-surrogate approach in multi-objective Bayesian optimisation. In comparison to the mono-surrogate approach, the proposed approach did not use the surrogate model built on the scalarised function and did not assume the scalarised function distribution is Gaussian. We built independent Gaussian process models on the objective functions and showed that the weighted Tchebycheff as the scalarising function after building independent models for each objective function was not Gaussian. We approximated the scalarising function distribution with Type 1 of the Generalised value extreme value distributions and used Monte Carlo to estimate the acquisition function. For the demonstration of the potential of the multi-surrogate approach, we tested the approach on benchmark and real-world problems. The proofs and analysis of the results indicated that one should use an appropriate distribution of the resulting scalarising function in multi-objective Bayesian optimisation.

Future work will include testing on other scalarising functions e.g. hypervolume improvement, dominance rank, penalty boundary intersection and scalarising functions used in the evolutionary multi-objective optimisation algorithms. As mentioned, the multi-surrogate approach is more computationally expensive than the mono-surrogate approach. An alternative to alleviate the computation cost is to use the Laplace approximation for the distribution of the scalarising function. We provided initial calculations of the Laplace approximation in the Appendix. We plan to work on using these calculations in the future. Testing on a wide range of benchmark and real-world problems will also be in our future works.

ACKNOWLEDGMENTS

The author would like to thank Dr De Ath George at the University of Exeter, UK for his feedback on the paper.

APPENDIX

Laplace approximation calculations for bi-objective optimisation

Let us denote $mi = w_i(\mu_i - z_i)$ and $si = w_i\sigma_i$, where μ_i, σ_i, z_i and w_i are the i^{th} element of the posterior mean vector, the standard deviation vector, the ideal objective vector and the weight vector, respectively. The distribution of g mentioned in Equation (2) can also be written as:

$$p(g) = \frac{1}{4}\mathrm{Erfc}\left[-\frac{g-\mathrm{m1}}{\sqrt{2}\mathrm{s1}}\right]\left(\frac{e^{-\frac{(g-\mathrm{m1})^2}{2\mathrm{s1}^2}}\sqrt{\frac{2}{\pi}}}{\mathrm{s1}\mathrm{Erfc}\left[-\frac{g-\mathrm{m1}}{\sqrt{2}\mathrm{s1}}\right]} + \frac{e^{-\frac{(g-\mathrm{m2})^2}{2\mathrm{s2}^2}}\sqrt{\frac{2}{\pi}}}{\mathrm{s2}\mathrm{Erfc}\left[-\frac{g-\mathrm{m2}}{\sqrt{2}\mathrm{s2}}\right]}\right)\mathrm{Erfc}\left[-\frac{g-\mathrm{m2}}{\sqrt{2}\mathrm{s2}}\right] \quad (3)$$

The mode, g_0 can be calculated by taking the derivative of the $\log g$ i.e. $\frac{d\log g}{dg}$ =0:

$$\frac{4\mathrm{s1}^2\mathrm{s2}^2 + \sqrt{2\pi}\left(-e^{-\frac{(g-\mathrm{m1})^2}{2\mathrm{s1}^2}}(g-\mathrm{m2})\mathrm{s1}^3\mathrm{Erfc}\left[\frac{-g+\mathrm{m1}}{\sqrt{2}\mathrm{s1}}\right] - e^{-\frac{(g-\mathrm{m2})^2}{2\mathrm{s2}^2}}(g-\mathrm{m1})\mathrm{s2}^3\mathrm{Erfc}\left[\frac{-g+\mathrm{m2}}{\sqrt{2}\mathrm{s2}}\right]\right)}{\sqrt{2\pi}\mathrm{s1}^2\mathrm{s2}^2\left(e^{-\frac{(g-\mathrm{m1})^2}{2\mathrm{s1}^2}}\mathrm{s1}\mathrm{Erfc}\left[\frac{-g+\mathrm{m1}}{\sqrt{2}\mathrm{s1}}\right] + e^{-\frac{(g-\mathrm{m2})^2}{2\mathrm{s2}^2}}\mathrm{s2}\mathrm{Erfc}\left[\frac{-g+\mathrm{m2}}{\sqrt{2}\mathrm{s2}}\right]\right)} = 0 \quad (4)$$

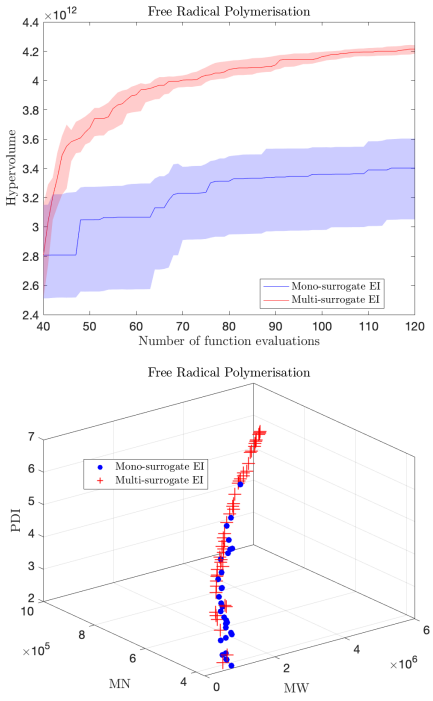

Figure 7: Hypervolume with the number of function evaluations on the Free-Radical Polymerisation problem (top figure). The approximated Pareto fronts, MW and MN are to be maximised and PDI is to be minimised.

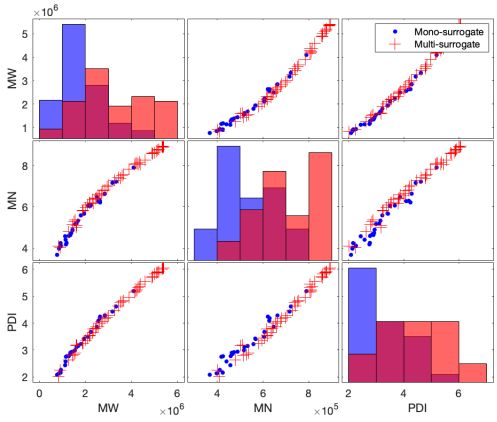

Figure 8: Two dimensional scatter plots between different objectives in the Free-Radical Polymerisation problem.

The equation does not have a closed form expression for estimating the mode. Therefore, the mode can be estimated by maximising a posteriori with an iterative method e.g. Newton's method.

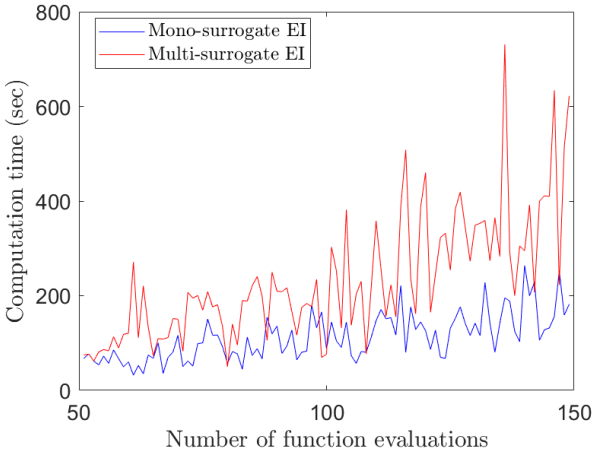

Figure 9: Computation time of mono- and multi-surrogate approaches on the DTLZ2 problem with two objectives and five decision variables

Once the mode is known, we can estimate the second derivative at the mode:

$$
\begin{aligned}
A &= -\left(\frac{\mathrm{d}^2 \log g}{\mathrm{d}g^2}\right)_{g=g_0} \\
&= \left(-e^{\frac{(g-\mathrm{m1})^2}{\mathrm{s1}^2}} \pi \mathrm{s1}^5 \mathrm{s2} \mathrm{Erfc}\left[\frac{-g+\mathrm{m1}}{\sqrt{2}\mathrm{s1}}\right]^2\right. \\
&\quad + e^{\frac{(g-\mathrm{m1})^2}{2\mathrm{s1}^2}} \mathrm{Erfc}\left[\frac{-g+\mathrm{m1}}{\sqrt{2}\mathrm{s1}}\right] \\
&\quad \left(\sqrt{2\pi}\mathrm{s1}^2\mathrm{s2}\left((g-\mathrm{m2})\mathrm{s1}^2 + 3(-g+\mathrm{m1})\mathrm{s2}^2\right)\right. \\
&\quad + e^{\frac{(g-\mathrm{m2})^2}{2\mathrm{s2}^2}} \pi \left((g-\mathrm{m2})^2\mathrm{s1}^4 - \mathrm{s1}^2\left(2(g-\mathrm{m1})(g-\mathrm{m2}) + \mathrm{s1}^2\right)\mathrm{s2}^2 + (g-\mathrm{m1}-\mathrm{s1})(g-\mathrm{m1}+\mathrm{s1})\mathrm{s2}^4\right) \\
&\quad \left.\mathrm{Erfc}\left[\frac{-g+\mathrm{m2}}{\sqrt{2}\mathrm{s2}}\right]\right) \\
&\quad - \mathrm{s1}\mathrm{s2}^2\left(8\mathrm{s1}^2\mathrm{s2} + e^{\frac{(g-\mathrm{m2})^2}{2\mathrm{s2}^2}} \sqrt{2\pi}\left(3(g-\mathrm{m2})\mathrm{s1}^2 + (-g+\mathrm{m1})\mathrm{s2}^2\right)\right. \\
&\quad \mathrm{Erfc}\left[\frac{-g+\mathrm{m2}}{\sqrt{2}\mathrm{s2}}\right] \\
&\quad \left.\left. + e^{\frac{(g-\mathrm{m2})^2}{\mathrm{s2}^2}} \pi \mathrm{s2}^3 \mathrm{Erfc}\left[\frac{-g+\mathrm{m2}}{\sqrt{2}\mathrm{s2}}\right]^2\right)\right) \\
&\quad / \left(\pi \mathrm{s1}^3 \mathrm{s2}^3\left(e^{\frac{(g-\mathrm{m1})^2}{2\mathrm{s1}^2}} \mathrm{s1}\mathrm{Erfc}\left[\frac{-g+\mathrm{m1}}{\sqrt{2}\mathrm{s1}}\right]\right.\right. \\
&\quad \left.\left. + e^{\frac{(g-\mathrm{m2})^2}{2\mathrm{s2}^2}} \mathrm{s2}\mathrm{Erfc}\left[\frac{-g+\mathrm{m2}}{\sqrt{2}\mathrm{s2}}\right]\right)^2\right)
\end{aligned}
\tag{5}
$$

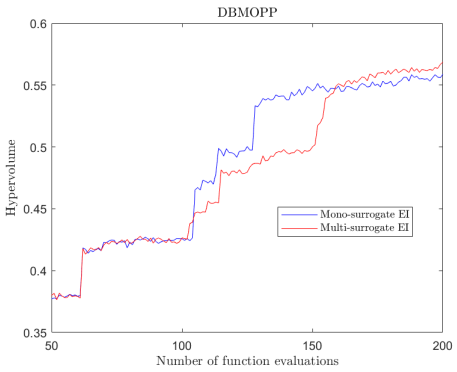

Figure 10: Hypervolume with the number of function evaluations on DBMOPP problems with five objectives and 10 decision variables.

After estimating the mode and the second derivative at the mode, the distribution of g can be written as:

$$g \sim \mathcal{N}(g_0, A^{-1}),$$

where g_0 is the mean and A^{-1} is the variance. These calculations allow us to use the closed form expression of acquisition functions e.g. expected improvement and probability of improvement.

Results on DBMOPP

In addition to DTLZ, we tested the proposed approach on a single instance of DBMOPP [10, 11]. The results of hypervolume with five objectives and 10 decision variables are shown in Figure 10. We did not see a significant difference in the results of the two approaches. As DBMOPP problems are visualisable, the solutions can be projected to two dimensions. We plotted the solutions with the number of function evaluations in Figure 11. The solutions on or inside the convex hull (shown in red) represent the Pareto optimal solutions. As can be seen, both approaches converged closer to the Pareto front. This problem is relatively easier and does not have any complexity like local Pareto fronts, varying density and dominance resistance regions. It will be worth testing the multi-surrogate approach on complex DBMOPP problems.

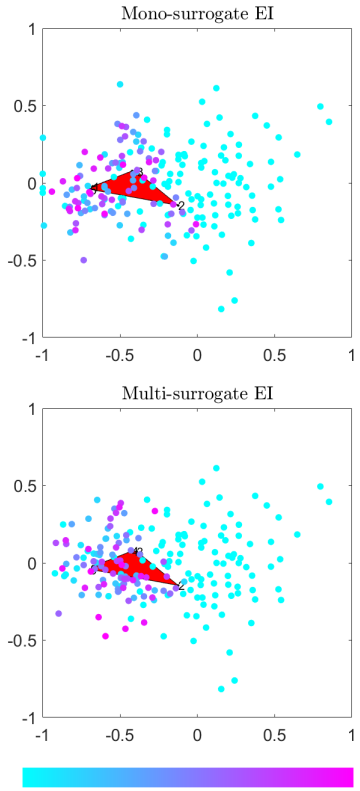

Figure 11: Trace iteration plots for a DBMOPP problem with five objectives and 10 decision variables. The colormap represents the iteration counter between 0 - 200.

REFERENCES

[1] George De Ath, Tinkle Chugh, and Alma AM Rahat. 2022. Multi-objective Bayesian Optimisation by Density-Ratio Estimation. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. https://doi.org/10.1145/3512290.3528769

[2] A. Azzalini. 1985. A Class of Distributions Which Includes the Normal Ones. *Scandinavian Journal of Statistics* 12, 2 (1985), 171–178.

[3] 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

[4] Laurens de Haan and Ana Ferreira. 2006. *Extreme Value Theory*. Springer New York. https://doi.org/10.1007/0-387-34471-3

[5] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. 2005. *Scalable Test Problems for Evolutionary Multiobjective Optimization*. Springer London, London, 105–145.

[6] M. Emmerich, A.H. Deutz, and J.W. Klinkenberg. 2011. Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE, 2147–2154.

[7] M. Emmerich, K. Giannakoglou, and B. Naujoks. 2006. Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation* 10 (2006), 421–439.

[8] M. Emmerich and B. Naujoks. 2004. Metamodel assisted multiobjective optimisation strategies and their application in airfoil design. In *Adaptive Computing in Design and Manufacture VI*, I. C. Parmee (Ed.). Springer London, 249–260.

[9] Michael T. M. Emmerich. 2005. Single- and multi-objective evolutionary design optimization assisted by gaussian random field metamodels. (2005). https://doi.org/10.17877/DE290R-57

[10] J.E. Fieldsend, T. Chugh, R. Allmendinger, and K. Miettinen. 2022. A Feature Rich Distance-Based Many-Objective Visualisable Test Problem Generator. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. ACM, 541–549.

[11] J. E. Fieldsend, T. Chugh, R. Allmendinger, and K. Miettinen. 2022. A Visualizable Test Problem Generator for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation* 26 (2022), 1–11. https://doi.org/10.1109/tevc.2021.3084119

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

[13] Don Lemons. 2002. *An introduction to stochastic processes in physics : containing "On the theory of Brownian motion" by Paul Langevin, translated by Anthony Gythiel*. Johns Hopkins University Press, Baltimore.

[14] Atanu Mazumdar, Tinkle Chugh, Jussi Hakanen, and Kaisa Miettinen. 2022. Probabilistic Selection Approaches in Decomposition-based Evolutionary Algorithms for Offline Data-Driven Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation* (2022), 1–1. https://doi.org/10.1109/TEVC.2022.3154231

[15] M.D. Mckay, R.J. Beckman, and W.J. Conover. 2000. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 42 (2000), 55–61.

[16] K. Miettinen. 1999. *Nonlinear multiobjective optimization.* Kluwer, Boston, MA.

[17] J. Močkus. 1975. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, G. I. Marchuk (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 400–404.

[18] A. Mogilicharla, T. Chugh, S. Majumder, and K. Mitra. 2014. Multi-objective optimization of bulk Vinyl Acetate polymerization with branching. *Materials and Manufacturing Processes* 29 (2014), 210–217.

[19] Alexander Mood. 1974. *Introduction to the theory of statistics.* McGraw-Hill, Auckland.

[20] Saralees Nadarajah and Samuel Kotz. 2008. Exact Distribution of the Max/Min of Two Gaussian Random Variables. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 16, 2 (2008), 210–212.

[21] P. S. Palar, L. R. Zuhal, T. Chugh, and A. Rahat. 2020. On the Impact of Covariance Functions in Multi-Objective Bayesian Optimization for Engineering Design. In *AIAA Scitech 2020 Forum.* https://doi.org/10.2514/6.2020-1867 arXiv:https://arc.aiaa.org/doi/pdf/10.2514/6.2020-1867

[22] 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* (Berlin, Germany) *(GECCO '17)*. ACM, New York, NY, USA, 873–880. https://doi.org/10.1145/3071178.3071276

[23] C. E. Rasmussen and C. K. I. Williams. 2006. *Gaussian processes for machine learning.* The MIT Press.

[24] T.Chugh, A. Rahat, and P.S. Palar. 2019. Trading-off Data Fit and Complexity in Training Gaussian Processes with Multiple Kernels. In *Proceedings of the International Conference on Machine Learning, Optimization, and Data Science (LOD) (Lecture Notes in Computer Science (LNCS))*. Springer.

[25] Zi Wang and Stefanie Jegelka. 2017. Max-Value Entropy Search for Efficient Bayesian Optimization. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (Sydney, NSW, Australia) *(ICML'17)*. JMLR.org, 3627–3635.

[26] Eric W Weisstein. [n. d.]. Weisstein, Eric W. "Extreme Value Distribution". mathworld.wolfram.com..

[27] A.P. Wierzbicki. 1980. The use of reference objectives in multiobjective optimization. In *Proceedings of the Multiple Criteria Decision Making Theory and Application*, G. Fandel and T. Gal (Eds.). Springer, 468–486.

[28] Kaifeng Yang, Michael Emmerich, André 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