Multi-Objective Bayesian Optimization for Engineering Simulation

Joachim van der Herten, Nicolas Knudde, Ivo Couckuyt and Tom Dhaene

Abstract Rather than optimizing expensive objective functions such as complex engineering simulations directly, Bayesian optimization methodologies fit a surrogate model (typically Kriging or a Gaussian Process) on evaluations of the objective function(s). To determine the next evaluation, an acquisition function is optimized (also referred to as infill criterion or sampling policy) which incorporates the model prediction and uncertainty and balances exploration and exploitation. Therefore, Bayesian optimization methodologies replace a single optimization of the objective function by a sequence of optimization problems: this makes sense as the acquisition function is cheap-to-evaluate whereas the objective is not. Depending on the goal different acquisition functions are available: multi-objective acquisition functions are relatively new and this chapter gives a state-of-the-art overview and illustrates some approaches based on hypervolume improvement. It is shown that the quality of the model is crucial for the performance of Bayesian optimization and illustrate this by using the more flexible Student-t processes as surrogate models.

1 Introduction

Over the past decades the use of computer simulations became an important part of the design process of complex systems, acting as an abstraction layer of the real world. The ability to perform experiments virtually allows to significantly reduce the number of required physical prototypes resulting in a cost reduction and a shorter

Joachim van der Herten · Nicolas Knudde · Ivo Couckuyt · Tom Dhaene
Ghent University - imec,
IDLab
iGent Tower - Department of Electronics and Information Systems
Technologiepark-Zwijnaarde 15, B-9052 Ghent, Belgium
e-mail: {joachim.vanderherten, nicolas.knudde, ivo.couckuyt, tom.dhaene}@ugent.be

time-to-market. Furthermore, these virtual experiments are easier to replicate as the environment can usually be controlled without additional effort.

The accuracy of these simulations has increased at the cost of higher computational requirements. Some simulations can take up to days, weeks or even months to perform a single evaluation (Goethals et al., 2012). Confronted with several parameters, evaluating a grid to perform tasks such as optimization, design space exploration or visualization requires massive computational resources and takes a lot of time. As this option quickly becomes infeasible, an extra layer of abstraction was proposed and referred to as *surrogate model*, *metamodel* or *response surface model*. The surrogate model itself is a simple mathematical expression which is cheap to evaluate and can then be used instead of the simulator to accomplish a variety of goals. For visualization and design space exploration scenarios the surrogate model should accurately approximate the response of the simulator. For other tasks including optimization this is an option as well, but more efficient methods are available. Several approaches exist to obtain this mathematical expression:

- Model order reduction: using specific properties of a complex system and mathematical approximations, the complex (differential) equations of the real simulation are simplified. Model order reduction approaches are application specific and require the (mostly manual) process to be restarted for a new system.
- Data-driven: this approach considers the complex system as a black box. A set of combinations of the input parameters (samples or data points) are simulated. Using the samples and the obtained evaluations the response surface can then be approximated using regression (or classification) techniques. Although all information about the system properties is lost as it is assumed to be unavailable, data-driven approaches are very generic and can be applied to wide ranges of problems.
- Hybrid approaches: overlap between the model- and data-driven approaches. Usually a set of data is augmented with prior knowledge about the properties of the complex systems to obtain a specific model approach (which may be applicable to all applications of a certain type).

In this chapter we focus on data-driven surrogate models for optimization. A naive approach is sampling the domain defined by the input parameters and obtaining all responses, then construct the surrogate model. If it is sufficiently accurate a standard algorithm such as multi-start gradient-descent can be applied to identify the optimum. The latter step will require many evaluations which is acceptable as evaluating the surrogate model is not expensive as opposed to the simulator itself.

Although this approach is more efficient in comparison to direct application evaluation-intensive optimization procedures on the simulator, obtaining good accuracy over the entire domain is not required for the task of optimization: it is intuitive that regions which are not optimal can be approximated more roughly. This led to the development of Surrogate-Based Optimization (SBO) which apply the surrogate model as a tool to guide the search for optimality, but are not necessarily accurate over the entire domain. The Efficient Global Optimization (EGO) method (Jones et al., 1998) is probably the most famous and widely used method in the context of

single-objective optimization of engineering simulations. It has also been applied for optimization of hyperparameters (Frohlich and Zell, 2005; Snoek et al., 2012): in this context it is often referred to as Bayesian optimization. A key property of these methods is that the data set is constructed sequentially: each iteration a new point for evaluation is selected by optimizing a sampling policy referred to as *acquisition* function. This function maps the information of the model to a score which guides the optimization to promising regions.

Often, optimization problems in engineering do not permit optimality to be described by a single scalar value. Rather, several potential conflicting objectives are relevant to describe the performance of a system for which a trade-off must be found. This is referred to as multi-objective optimization problems which are often solved with Multi-Objective Evolutionary Algorithms (MOEAs). Unfortunately MOEAs rely on a lot of evaluations of the objectives, making their application to expensive engineering simulations problematic. This resulted in the development of Bayesian optimization strategies for multi-objective problems which is the subject of this chapter. In Section 2 we revisit the definition of Bayesian optimization and review Gaussian Processes (GPs) and Kriging which are the most popular surrogate models for these methodologies. The section concludes with a brief overview of some acquisition functions for single-objective optimization. In Section 3 approaches for multi-objective Bayesian optimization are discussed, as well as illustrated on a set of test functions. It is shown how the model quality is crucial in order to obtain good results, which is also the basis for Section 4 where Student-t processes for multi-objective Bayesian optimization are studied.

2 Bayesian Optimization

Confronted with the following global optimization problem:

$$\arg\min_{\mathbf{x}\in\mathscr{X}} f(\mathbf{x}). \tag{1}$$

for an unknown function $f : \mathscr{X} \to \mathbb{R}^p$ corresponding to a simulator mapping points from a d-dimensional bounded input domain $\mathscr{X} \subset \mathbb{R}^d$ to a p-dimensional output space. The input space spans all possible input combinations of the simulator parameters, whereas the output space is represented by the optimization objectives. For each input $\mathbf{x}$ a corresponding observation $\mathbf{y}$ in the output space can be observed by evaluating f. Assuming the observation is not exact and subject to uncertainty, the distribution of $\mathbf{y}$ is centered around the true response $\mathbf{f}$ with variance given by σ_n^2. In the context of engineering typically deterministic simulations are assumed, hence we assume $\sigma_n^2 = 0$. We adopt the deterministic property in this chapter.

The goal of Bayesian optimization is to come up with a sequence of N decisions $\mathbf{x}_i$ with $i = 0, ..., N-1$ (N being the total amount of allowed evaluations) using a sampling policy (often referred to as *acquisition* function) such that the probability of identifying the optimal solution to Eq. (1) is maximal. Because of the assumption

that evaluating f is expensive, additional computational effort to determine these decisions is justified. Formally, the next decision $\mathbf{x}_n$ is selected by solving the following optimization problem:

$$\mathbf{x}_n = \arg\max_{\mathbf{x}\in\mathscr{X}} \alpha(\mathbf{x}).$$

The information obtained on f after each evaluation is represented using a model (or multiple models). The acquisition function then uses this knowledge to *exploit* promising regions more, but also incorporates an *exploration* component to assure the input space is covered and optima are not missed. This section proceeds with an in-depth explanation of GPs, as well as their use as part of Kriging models, which is the model type included in the popular EGO algorithm (Jones et al., 1998). Finally, some widely used acquisition functions for single-objective Bayesian optimization are briefly reviewed.

2.1 Gaussian Processes

A GP is essentially a generalization of a multivariate Gaussian distribution to an infinite number of dimensions. Drawing a sample from a GP results in a random function (like drawing a sample from n-dimensional Gaussian distribution results in a n-dimensional vector). The analogy continues as a GP is defined by mean and covariance functions $\gamma(\mathbf{x})$ and $k(\mathbf{x},\mathbf{x}')$ respectively, similar to a multivariate Gaussian distribution which is defined by its mean vector γ and covariance matrix $\mathbf{K}$. Formally we can define GPs as follows:

Definition 1 (Gaussian Process). A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution (Rasmussen and Williams, 2006). The GP can be used as a non-parametric prior over a latent function f:

$$\begin{aligned} f &\sim GP\left(\gamma(\mathbf{x}), k(\mathbf{x},\mathbf{x}')\right), \\ \gamma(\mathbf{x}) &= \mathbb{E}\left[f(\mathbf{x})\right], \\ k(\mathbf{x},\mathbf{x}') &= \mathbb{E}\left[\left(f(\mathbf{x})-\gamma(\mathbf{x})\right)\left(f(\mathbf{x}')-\gamma(\mathbf{x}')\right)\right]. \end{aligned}$$

The typical choice for the mean function is $\gamma(\mathbf{x}) = 0$: this is also the convention for this section. This may seem counter-intuitive at first, but can be achieved by shifting the training data prior to constructing a GP, or by constructing a hierarchical k and adding a bias kernel. The covariance function is chosen upfront and defines some properties such as smoothness, periodicity, trends or bias. It is usually parametrized by a set of kernel hyperparameters θ_k. Some popular choices for the covariance function are discussed in Section 2.2.

In a modeling scenario, an inherently infinite model specifies a finite Gaussian distribution due to the limited set of training data. This assumes the remainder of $\mathscr{X}$ is marginalized. The GP prior is conditioned on the training data, which results in a

posterior distribution over $\mathbf{F}$ that "*fits*" the data. Denoting the training input data $\mathbf{X}$ and observations collected in $\mathbf{F} \in \mathbb{R}^{n \times p}$, this can be written as

$$p(\mathbf{F}|\mathbf{X}, \theta_k) = \prod_{i=1}^{p} p(\mathbf{f}^{(i)}|\mathbf{X}, \theta_k).$$

In the remainder of this section we assume $p = 1$ and omit the column index. However, it is clear that a GPs can also be used for multi-output functions f as the likelihood can be obtained by multiplication over the dimensions. The model now specifies a finite Gaussian distribution for the training data:

$$\begin{aligned} \mathbf{f}|\mathbf{X}, \theta_k &\sim \mathcal{N}(0, \mathbf{K}_{ff}), \\ p(\mathbf{f}|\mathbf{X}, \theta_k) &= (2\pi)^{-\frac{n}{2}} |\mathbf{K}_{ff}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\mathbf{f}^T \mathbf{K}_{ff}^{-1} \mathbf{f}\right), \end{aligned}$$

with the square covariance matrix $\mathbf{K}_{ff} \in \mathbb{R}^{n \times n}$ constructed by evaluating the covariance function k on the samples $\mathbf{X}_i$:

$$\mathbf{K}_{ff} = \begin{bmatrix} k(\mathbf{x}_0, \mathbf{x}_0) & \dots & k(\mathbf{x}_0, \mathbf{x}_{n-1}) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_{n-1}, \mathbf{x}_0) & \dots & k(\mathbf{x}_{n-1}, \mathbf{x}_{n-1}) \end{bmatrix}.$$

If $\mathbf{f}$ can be observed directly (and no noise is present) the expression for $p(\mathbf{f}|\mathbf{X})$ has no latent variables, but still depends on the kernel hyperparameters. Before further discussing these parameters, we first extend the GP formalism to incorporate observation noise. Formally the noise corruption is assigned a Gaussian prior:

$$\begin{aligned} y &= f(\mathbf{x}) + \varepsilon, \\ \varepsilon &\sim \mathcal{N}(0, \sigma_n^2). \end{aligned}$$

This turns $\mathbf{f}$ into a latent variable, for which the posterior distribution $p(\mathbf{f}|\mathbf{X}, \mathbf{y}, \theta)$ can be computed with Bayes rule. Here, $\theta = (\theta_k, \sigma_n^2)$. By specifying the likelihood distribution

$$\mathbf{y}|\mathbf{f}, \sigma_n^2 \sim \mathcal{N}(\mathbf{f}, \sigma_n^2 \mathbf{I}),$$

marginalizing $\mathbf{f}$ is tractable and results in an analytical expression for the marginal likelihood. This is obtained by integrating the product of the prior on $\mathbf{f}$ (conditioned on $\mathbf{X}$) and the likelihood[1]:

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \theta) &= \int p(\mathbf{y}|\mathbf{f}, \sigma_n^2)\ p(\mathbf{f}|\mathbf{X}, \theta_k)\, \mathrm{d}\mathbf{f}, \\ &= (2\pi)^{-\frac{n}{2}} |\mathbf{K}_{ff} + \sigma_n^2 \mathbf{I}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\mathbf{y}^T (\mathbf{K}_{ff} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}\right). \end{aligned} \tag{2}$$

[1] Marginal likelihood as in: marginalized over $\mathbf{f}$.

Note that the marginal likelihood incorporates the bias-variance trade-off: the determinant term restricts model complexity and reduces variance, whereas the exponential term promotes fitting the data. We now further explore the role of the hyperparameters and how they should be handled.

Ideally, in order to obtain predictions we are able to marginalize the hyperparameters and obtain the posterior $p(\theta|\mathbf{y},\mathbf{X})$ analytically according to Bayes rule:

$$p(\theta|\mathbf{y},\mathbf{X}) = \frac{p(\mathbf{y}|\mathbf{X},\theta)\ p(\theta)}{p(\mathbf{y}|\mathbf{X})}. \tag{3}$$

Hereafter, conditioning on $\mathbf{y}$ and $\mathbf{X}$ will be denoted by D. Under this setting, the posterior distribution on $f(\mathbf{X})$ of the *marginal GP* would be computed by marginalizing θ:

$$p(f(\mathbf{x}^\star)|\mathbf{x}^\star,D) = \int p(f(\mathbf{x}_\star)|\mathbf{x}_\star,D,\theta)\ p(\theta|D)\,\mathrm{d}\theta\,. \tag{4}$$

This formulation for the posterior predictive distribution is completely (hyper-) parameter free and hence does not require any further optimization. The first term is not problematic. Under the definition of GPs, the *posterior predictive distribution* is joint-Gaussian with the distribution over the observations:

$$\begin{bmatrix}\mathbf{y}\\ f(\mathbf{x}_\star)\end{bmatrix}\Bigg|\theta \sim \mathcal{N}\left(0, \begin{bmatrix}\mathbf{K}_{ff}+\sigma_n^2\mathbf{I} & \mathbf{K}_{f\star}\\ \mathbf{K}_{\star f} & \mathbf{K}_{\star\star}\end{bmatrix}\right).$$

In this expression $\mathbf{K}_{f\star} = \mathbf{K}_{\star f}^T$ represents the cross-covariance between $\mathbf{x}_\star$ and the training samples, and $\mathbf{K}_{\star\star} = k(\mathbf{x}_\star,\mathbf{x}_\star)$. Obtaining the posterior for $f(\mathbf{x}_\star)$ requires obtaining the conditional distribution given $\mathbf{y}$. This is straightforward as both random vectors are jointly Gaussian, and results in another (Gaussian) distribution:

$$\begin{aligned} f(\mathbf{x}_\star)|\mathbf{x}_\star,D,\theta &\sim \mathcal{N}\left(\mu(\mathbf{x}_\star|\theta), s^2(\mathbf{x}_\star|\theta)\right),\\ \mu(\mathbf{x}_\star|\theta) &= \mathbf{K}_{\star f}\left(\mathbf{K}_{ff}+\sigma_n^2\mathbf{I}\right)^{-1}\mathbf{y}, &\text{(5a)}\\ s^2(\mathbf{x}_\star|\theta) &= \mathbf{K}_{\star\star}-\mathbf{K}_{\star f}\left(\mathbf{K}_{ff}+\sigma_n^2\mathbf{I}\right)^{-1}\mathbf{K}_{f\star}. &\text{(5b)}\end{aligned}$$

Unfortunately, the second density of the integral in Eq. (4) is often problematic to compute. The denominator of Eq. (3) is the root of the problem as the hyperparameters typically occur non-linearly in kernel functions, making marginalization of θ intractable in most cases. An excellent description of the difficulty of propagating distributions through non-linearities, a key problem for Bayesian methods, is given by (Damianou, 2015).

Instead, the common way to proceed is to obtain a point estimate for θ by numerically optimizing the (log of the) numerator of Eq. (3), which is tractable. This approach is referred to as Maximum Likelihood Estimation (MLE) and represents a point estimate of Eq. (4). The results represents the most likely posterior predictive distribution for the latent function f (w.r.t. θ) which corresponds to a Gaussian distribution of approximating functions interpolating the observations (apart from a "tolerance" defined by σ_n^2). Hence the posterior predictive distribution can be

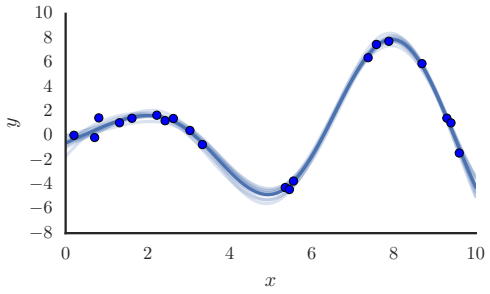


Fig. 1: Samples from the posterior on $\mathbf{f}$ for a small number of points. The hyperparameters θ were determined using MLE

regarded as an analytical weighting function for an infinite ensemble of approximating functions. This is illustrated in Fig. 1: here 10 samples for f are drawn from $p(f|\mathbf{x},\mathbf{X},\mathbf{y},\theta),\forall\mathbf{x}\in\mathscr{X}$. In practice, the mean of this posterior predictive distribution is typically used as surrogate model, whereas its variance may be used for different applications such as sampling as, for instance, is the case in the EGO algorithm.

2.2 Covariance functions

At the basis of the GP lies the covariance function, defined as $k:\mathscr{X}\times\mathscr{X}\rightarrow\mathbb{R}^{+}$. This function represents the underlying properties of the behavior of the response as it is the basis for the covariance matrix. Depending on the application several choices are possible, and some research has focused on searching over a space kernels (Duvenaud et al., 2013; Malkomes et al., 2016). In this chapter we focus on two popular stationary covariance functions: the Matérn $\frac{3}{2}$ correlation function (Stein, 1999)

$$k(\mathbf{x},\mathbf{x}')=\sigma^2\left(1+\sqrt{3}d\right)\exp\left(-\sqrt{3}d\right),$$

with $d=\sqrt{(\mathbf{x}-\mathbf{x}')^T\operatorname{diag}(\ell^{-1})(\mathbf{x}-\mathbf{x}')}$. In addition, and the popular Gaussian correlation function

$$k(\mathbf{x},\mathbf{x}')=\sigma^2\exp\left(-\frac{d^2}{2}\right).$$

Clearly, the choice of the *lengthscales* ℓ plays an important role in defining the covariance structure. Together with variance parameter σ^2 the lengthscales are included as hyperparameters θ_k. It is possible to consider a single lengthscale, or one per input dimension. The latter is referred to as Automated Relevance Determination (ARD) and permits identification of dimensions which contribute less to the response.

2.3 Kriging

Within the context of surrogate modeling GPs are typically used as part of a Kriging model to approximate deterministic noise-free data. This model has proven to be very useful for a variety of tasks and was used for the EGO algorithm (Jones et al., 1998).

A Kriging model is essentially a combination of a regression model $h(\mathbf{x}_\star) = \mathbf{B}_\star \alpha$, complemented with a unit variance Gaussian process with $\gamma(\mathbf{x}) = 0$ interpolating the residual.

$$f(\mathbf{x}) = h(\mathbf{x}) + \sigma^2 Z(\mathbf{x}).$$

The matrix $\mathbf{B}_\star$ is obtained by representing the test point $\mathbf{x}_\star$ in the basis spanned by the basis functions chosen for the regression model. The coefficients $\mathbf{c}$ can be determined by solving a generalized least-squares problem:

$$\mathbf{c} = \left(\mathbf{B}^T \mathbf{K}_{ff}^{-1} \mathbf{B}\right)^{-1} \mathbf{B}^T \mathbf{K}_{ff}^{-1} \mathbf{f}.$$

The GP Z is multiplied by the σ^2 signal variance parameter which under this setting can be computed analytically:

$$\sigma^2 = \frac{1}{n} (\mathbf{f} - \mathbf{B}\mathbf{c})^T \mathbf{K}_{ff}^{-1} (\mathbf{f} - \mathbf{B}\mathbf{c}).$$

This parameter is no longer a part of the covariance function. As such, in the context of Kriging the function k is now referred to as *correlation* function and the resulting matrix is $\mathbf{K}$ is referred to as *correlation* matrix. Predictions of a Kriging model are Gaussians, the first two moments are obtained by augmenting the formulas of Eqs. (5a) and (5b) with the regression model:

$$\begin{aligned}
\mu(\mathbf{x}_\star|\theta) &= \mathbf{B}_\star \mathbf{c} + \mathbf{K}_{\star f} \left(\mathbf{K}_{ff}\right)^{-1} (\mathbf{f} - \mathbf{B}\mathbf{c}), \\
s^2(\mathbf{x}_\star|\theta) &= \sigma^2 \left(1 - \mathbf{K}_{\star f} \mathbf{K}_{\star\star}^{-1} \mathbf{K}_{f\star} \frac{1 - \mathbf{B}^T \mathbf{K}_{ff}^{-1} \mathbf{K}_{f\star}}{\mathbf{B}^T \mathbf{K}_{ff}^{-1} \mathbf{B}}\right).
\end{aligned}$$

A thorough mathematically treatment of Kriging is given in (Santner et al., 2003; Forrester and Jones, 2008).

2.4 Marginalizing model hyperparameters

In case sufficient data is available, and an appropriate kernel was chosen which represents the covariance structure of f, the optimization of the numerator of Eq. (3) is usually successful as the global optimum is isolated and quite sharp. For some applications such as for instance Bayesian optimization, the point estimate can be insufficient as the likelihood surface becomes multi-modal. It is then interesting to

incorporate the uncertainty on θ somehow. One option is to approximate Eq. (4) by sampling θ from the numerator of Eq. (3) with MCMC. Under this setting each sample corresponds to a different posterior distribution on $f(\mathbf{x}_\star)$, hence the mean and variance of the posterior predictive distribution of the marginal GP can be approximated. Note that this distribution is not necessarily Gaussian: it was for instance shown that marginalizing the common σ^2 kernel parameter (which can still be tractable) the predictive distribution changes into a Student-t distribution (Gramacy and Apley, 2015). Given P hyperparameter θ_i sampled from the numerator of Eq. (3) and following the law of total cumulance (Brillinger, 1969):

$$\begin{aligned}
\mathbb{E}\left[f(\mathbf{x}_\star)|\mathbf{x}_\star, D\right] &= \mathbb{E}_{p(\theta|D)}\left[\mathbb{E}\left[f(\mathbf{x}_\star)|\mathbf{x}_\star, D, \theta\right]\right], \\
&= \mathbb{E}_{p(\theta|D)}\left[\mu(\mathbf{x}_\star|\theta)\right], \\
&\overset{\text{MCMC}}{\approx} \frac{1}{P}\sum_{i=1}^{P}\mu(\mathbf{x}_\star|\theta_i), \\
&= \tilde{\mathbb{E}}\left[p(f(\mathbf{x}_\star)|\mathbf{x}_\star, D)\right]. \\
\mathrm{Var}\left[f(\mathbf{x}_\star)|\mathbf{x}_\star, D\right] &= \mathbb{E}_{p(\theta|D)}\left[\mathrm{Var}\left[f(\mathbf{x}_\star)|\mathbf{x}_\star, D, \theta\right]\right] \\
&\quad + \mathrm{Var}_{p(\theta|D)}\left[\mathbb{E}\left[f(\mathbf{x}_\star)|\mathbf{x}_\star, D, \theta\right]\right], \\
&= \mathbb{E}_{p(\theta|D)}\left[s^2(\mathbf{x}_\star|\theta)\right] + \mathrm{Var}_{p(\theta|D)}\left[\mu(\mathbf{x}_\star|\theta)\right], \\
&\overset{\text{MCMC}}{\approx} \frac{1}{P}\sum_{i=1}^{P} s^2(\mathbf{x}_\star|\theta_i) + \left(\mu(\mathbf{x}_\star|\theta_i) - \tilde{\mathbb{E}}\left[p(f(\mathbf{x}_\star)|\mathbf{x}_\star, D)\right]\right)^2.
\end{aligned}$$

For both MCMC approximations, the samples θ_i are drawn from the numerator of Eq. (3). Finally, some analytical approximations of the marginal GP exist such as the method proposed by (Garnett et al., 2014). In the same work, an information-theoretic sampling method is proposed known as Bayesian Active Learning by Disagreement (BALD) which aims to select observations to reduce the uncertainty on θ, and hence enhances the quality of the point estimate.

2.5 Single-objective Bayesian optimization

We conclude this section with a brief review of some popular single-objective acquisition functions. These functions map the prediction of a model (typically a GP or Kriging model) to a score indicating how promising the sampling decision is expected to be in terms of our goal, in this case single-objective minimization. The approach known as Lower Confidence Bound (LCB) (Cox and John, 1997) is very basic, however it was shown to feature some strong theoretical guarantees (Freitas et al., 2012). It was shown that succesful Bayesian optimization comes to down finding the correct balance (in LCB explicitly present as a parameter) between minimizing the mean of the predictive distribution and incorporating variance into

the decision process. This corresponds to the well-known exploitation/exploration problem, present in a wide range of active learning problems.

Another criterion known as the Probability of Improvement was introduced, which corresponds to the part of the density of the predictive distribution below the current best observed value:

$$\begin{aligned}\alpha^{\text{PoI}}(\mathbf{x}_\star) &= \int_{-\infty}^{f_{\min}} p(f_\star|\mathbf{X},\mathbf{y},\mathbf{x}_\star)\,\mathrm{d}f_\star\,,\\ &= \Phi\left(\frac{\mu(\mathbf{x}_\star)-f_{\min}}{s(\mathbf{x}_\star)}\right).\end{aligned}$$

Incorporating the actual improvement into this integral yields the well-known and widely used Expected Improvement criterion (Močkus, 1975; Jones et al., 1998).

$$\begin{aligned}\alpha^{\text{EI}}(\mathbf{x}_\star) &= \int_{-\infty}^{\infty} \max(f_{\min}-f_\star,0)p(f_\star|\mathbf{X},\mathbf{y},\mathbf{x}_\star)\,\mathrm{d}f_\star\\ &= (f_{\min}-\mu(\mathbf{x}_\star))\,\Phi\left(\frac{\mu(\mathbf{x}_\star)-f_{\min}}{s(\mathbf{x}_\star)}\right)+s^2(\mathbf{x}_\star)\phi\left(\frac{\mu(\mathbf{x}_\star)-f_{\min}}{s(\mathbf{x}_\star)}\right)\end{aligned}$$

In these expressions, ϕ and Φ correspond to the probability and cumulative density functions of the standard normal distribution respectively.

More recently, some powerful results were obtained using information theoretic criteria, such as entropy search (Hennig and Schuler, 2012), predictive entropy search (Hernández-Lobato et al., 2014) and max-value entropy search (Wang and Jegelka, 2017). The latter work also proves a relation to LCB (Cox and John, 1997), linking the information theoretic work to the earlier regret-based approaches by providing an expression for the trade-off parameter.

3 Multi-objective acquisition functions

Given a multi-objective (or multi-task) deterministic optimization problem, each evaluated input $\mathbf{x}_i$ has p observed responses $\mathbf{f}_i = [f^{(1)}(\mathbf{x}_i),...,f^{(p)}(\mathbf{x}_i)]$. The observed responses are noiseless and together form a matrix $\mathbf{F} \in \mathbb{R}^{n\times p}$. The rows of this matrix correspond to points in the p-dimensional objective space. In terms of modeling, roughly three options are available

1. Train a single GP for multiple outputs. This however implies a single kernel and set of hyperparameters should be applicable to each objective which is often too restrictive.
2. Train a single-output GP for each objective: avoiding the problems of the first approach.
3. Train a single GP with a coregionalized kernel which enabled modeling all outputs and include correlation between the objectives (Shah and Ghahramani, 2016)

Of interest are the non-dominated solutions forming the Pareto set $P \subset \mathbf{F}$: acquisition functions for multi-objective optimization aim to improve the Pareto set by increasing the size of the dominated part of the objective space. One of the first methods introduced was ParEGO (Knowles, 2006) which corresponds to a weighted sum of EI scores for each objective individually. As the weights are unknown, a sampling scheme for the weights was developed to reweigh the objectives each iteration in order to cover the Pareto front. Recent algorithms include active learning of Pareto fronts (Campigotto et al., 2014), a multi-objective generalization of predictive entropy search (Hernández-Lobato et al., 2016), and minimum regret search (Metzen, 2016). A particular class of algorithms is based on the concept of improving the size of the hypervolume representing the dominated part of the objective space. We discuss these algorithms further in this section and conclude with four examples.

3.1 Hypervolume-based criteria

The hypervolume metric (or $\mathscr{S}$-metric) (Zitzler et al., 2003) is widely used in multi-objective optimization to assess the quality of a Pareto set or to drive multi-objective optimization algorithms (Beume et al., 2007). Ideally, we would like to identify the following point:

$$\tilde{\mathbf{x}} = \max_{\mathbf{x}_\star \in \mathscr{X}} I(\mathbf{f}_\star, P),$$

with $\mathbf{f}_\star = f(\mathbf{x}_\star)$ and $I(.)$ representing the improvement function which is defined using the hypervolume indicator as,

$$I(\mathbf{f}, P) = \begin{cases} \mathscr{H}(P \cup \mathbf{f}_\star) - \mathscr{H}(P) & \mathbf{f} \in \mathscr{D} \\ 0 & \text{otherwise.} \end{cases}$$

Here $\mathscr{D}$ represents the non-dominated section of the objective space and $\mathscr{H}(.)$ is defined as the hypervolume of the section of the objective space dominated by the Pareto set (bounded by a reference point $\mathbf{f}^{\max}$ dominated by all points of the Pareto set).

The situation is illustrated in Fig. 2: the exclusive (or contributing) hypervolume corresponds to $\mathscr{H}(P \cup \mathbf{f}) - \mathscr{H}(P)$. Because $\mathbf{f}$ is a (black-box) mapping of p objective functions of a candidate $\mathbf{x}$, and because of the assumption each evaluation is expensive, direct application of traditional numerical optimization methods is infeasible. Instead, we approximate each $f^{(i)}$ and optimize an acquisition function incorporating the information provided by the predictive distributions of the approximations of the objectives. The optimum of the acquisition yields a candidate $\tilde{\mathbf{x}}$ to be evaluated on all $f^{(i)}$.

Several hypervolume acquisition functions were introduced previously such as the Hypervolume-based EI (HvEI). Unfortunately, the EI integral is no longer tractable as the improvement is now expressed through the growth of a hypervolume, rather than a difference. It was proposed to compute this quantity using Monte Carlo

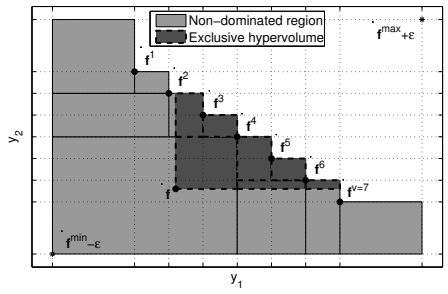


Fig. 2: Pareto set: Illustration (members illustrated by $\mathbf{f}^i$) with two objective functions. $\mathbf{f}^{\min}$ and $\mathbf{f}^{\max}$ denote the ideal and anti-ideal point respectively. The shaded areas (both light and dark) represent the non-dominated region and is decomposed into q cells by a binary partitioning procedure. These cells provide integration bounds to compute $I(\mathbf{f}, P)$. Courtesy of Couckuyt et al. (2014b)

techniques (Emmerich et al., 2006). More recently, a method was proposed for exact calculation for an arbitrary number of dimensions by decomposing the non-dominated region into a set of z cells spanned by upper and lower bounds $[\mathbf{l}^k, \mathbf{u}^k]$ (Emmerich et al., 2011). Unfortunately, the proposed mathematical expressions assume that the non-dominated region is decomposed into an uniform grid of cells based on the Pareto set, as represented by the dashed lines in Fig. 2. Hence, the number of cells required to evaluate the criterion grows exponentially in the number of Pareto points and objectives. Further developments of the HvEI resulted in faster methods to compute the acquisition function (Hupkens et al., 2014). In Couckuyt et al. (2014a) an approach was developed based on covering the non-dominated volume by a set of disjoint cells which permits computing the improvement in each cell independently. However, for HvEI a slower binary search needs to be used as HvEI requires a decomposition in disjoint cells.

A significant faster decomposition algorithm known as Walking Fish Group (WFG) (While et al., 2012), can be applied to most other acquisition functions. In this chapter we illustrate the Hypervolume Probability of Improvement (HvPoI) introduced by Couckuyt et al. (2014a) which permits decomposition of the non-dominated region with WFG. Formally, this acquisition function is defined as

$$\begin{aligned} \alpha^{\text{HvPoI}}(\mathbf{x}) &= I(\mu, P)p(\mathbf{x} \in \mathscr{D}), \\ \mu &= [\mu^{(1)}(\mathbf{x}), ..., \mu^{(p)}(\mathbf{x})]. \end{aligned}$$

The latter term of the multiplication represents the probability a new point is located in $\mathscr{D}$ and, hence, requires an integration over that region. Exact computation of this integral is performed by decomposing $\mathscr{D}$ into cells. We then make use of the predictive distribution of the GPs:

$$p(\mathbf{x} \in \mathscr{D}) = \sum_{k=1}^{z} \prod_{j=1}^{p} \left(\Phi\left(\frac{u_j^k - \mu^{(j)}(\mathbf{x})}{s^{2,(j)}(\mathbf{x})} \right) \Phi\left(\frac{l_j^k - \mu^{(j)}(\mathbf{x})}{s^{2,(j)}(\mathbf{x})} \right) \right).$$

In this context, Φ represents the cumulative density function of a standard normal distribution. In addition, we can simply compute the volume of the exclusive volume using the existing z cells with no extra computation as follows (assuming μ is non-dominated):

$$\mathscr{H}(P \cup \mu) - \mathscr{H}(P) = \sum_{k=1}^{z} \prod_{j=1}^{p} \left(u_j^k - \left(\max l_j^k, \mu_j(\mathbf{x}) \right) \right).$$

Similar to this method Keane (2006) defines the HvEI as the product of the PoI and an Euclidean distance-based improvement function.

3.2 Examples

A good set of configurable multi-objective benchmark problems has been proposed (Deb et al., 2001), of which four benchmark functions are chosen and adapted slightly to illustrate the hypervolume-based acquisition functions. A summary of the selected benchmark functions is found in Table 1. All benchmark functions are configured to have six input parameters. For a complete description of the benchmark functions the reader is referred to Deb et al. (2001).

Function	d	m	Reference point $\mathbf{f}^{\max}$
DTLZ1	6 inputs	3 objectives	$(400, 400, 400)$
DTLZ2	6 inputs	3 objectives	$(2.5, 2.5, 2.5)$
DTLZ7	6 inputs	4 objectives	$(1, 1, 1, 50)$
DTLZ5	6 inputs	6 objectives	$(2.5, 2.5, 2.5, 2.5, 2.5, 2.5)$

Table 1: Summary of the DTLZ benchmark functions

3.2.1 Experimental setup

An initial set of 65 samples is generated by a near-optimal maximin Latin Hypercube Design (LHD; (Van Dam et al., 2007)). Subsequently, a statistical criterion is optimized for each iteration to select the next point to evaluate. The criterion is optimized using a combination of Monte Carlo sampling and a local search. Specifically, $20 \times d$ Monte Carlo candidate points are generated and evaluated on the criterion. The best Monte Carlo candidate is further refined using Matlab's fmincon optimizer.

Various acquisition are applied on the benchmark functions for comparison: including the Euclidean distance-based criterion (Keane, 2006) referred to as EI-L2 and HvPoI using Kriging models with the Matérn $\frac{3}{2}$ correlation function (Rasmussen and Williams, 2006) and a constant regression function. The hyperparameters of the Kriging models are optimized using Sequential Quadratical Programming (SQP) implemented in SQPLab (Bonnans et al., 2006) utilizing likelihood derivative information.

Additionally, the runs of the EI-L2 criterion are repeated with Kriging models using the Gaussian correlation function. these runs are referred to as EI-L2-RBF in the results. Lastly, for the DTLZ1 and DTLZ2 functions the expensive hypervolume-based EI criterion (HvEI) (Emmerich et al., 2011) with Kriging models using the Matérn correlation function was also included in the comparison. Each of these configurations is repeated 10 times for statistical robustness and halts when the sample budget is met, namely, 250 samples.

These runs are compared against the NSGA-II (Deb et al., 2002), SPEA2 (Zitzler et al., 2001) and SMS-EMOA (Beume et al., 2007) MOEAs with a varying population size and maximum number of generations. The first run is configured with a population size of 25 and a maximum number of generations of 10 (total sample budget 250) and the second run is configured with a population size of 50 and a maximum number of generations of 50 (total sample budget 2500). The remaining parameters have been left to their default values. Similarly to the EMO runs, the evolutionary algorithm runs are repeated 10 times.

Besides assessing the performance of the algorithms using the hypervolume metric, the convergence measure is used too. The convergence measure is the mean distance of every point of the Pareto set to the closest Pareto point of the known Pareto front. In this work the known Pareto fronts are sampled with 100.000 Monte Carlo points.

3.2.2 Results

Results for the benchmark functions have been summarized in Table 2. Note that the differences on the hypervolume metric are more significant than they appear because of the conservative choice of the reference point $\mathbf{f}^{\max}$ (needed to accommodate the results of all test configurations).

In general, it is seen that the runs using multi-objective Bayesian optimization have better performance than the MOEAs in terms of hypervolume score for most functions except for DTLZ1. After a closer examination it is observed that the accuracy of the Kriging models of DTLZ1 for most statistical criteria is sub-optimal. In particular, the first objective function is difficult to approximate using the Kriging models, an issue further explored in Section 4.

A plot of the final Pareto sets generated of the DTLZ2 problem is shown in Fig. 4. It is seen that the hypervolume-based criteria emphasizes the edges of Pareto front more while leaving a small gap between the edge and the inner portion of the Pareto front. This is not unlike the DTLZ2 results as reported in (Beume et al., 2007) and

Problem	N	Algorithm	Convergence measure		Hypervolume	
			Mean	Std	Mean	Std
DTLZ1	250	EI-L2	93.2833	18.7840	6.3498e7	2.4970e5
		EI-L2-RBF	100.6741	14.2258	6.3650e7	1.2418e5
		HvEI	***37.6112***	2.9315	*6.3940e7*	6.0452e4
		HvPoI	66.9199	14.0029	6.3838e7	7.4330e4
		NSGA-II	75.8391	20.4219	6.3612e7	2.3441e5
		SPEA2	104.6259	0	6.3482e7	0
		SMS-EMOA	44.8818	7.9740	**6.3976e7**	8.0982e3
	2500	NSGA-II	16.6888	4.8071	6.3991e7	1.0227e4
		SPEA2	93.8381	0	6.3984e7	0
		SMS-EMOA	**9.5047**	2.8750	**6.4000e7**	324.0575
DTLZ2	250	EI-L2	0.0843	0.0205	14.9423	0.0181
		EI-L2-RBF	0.1481	0.0133	14.8994	0.0114
		HvEI	0.0411	0.0052	14.8834	0.0165
		HvPoI	***0.0106***	0.0021	***15.0326***	0.0054
		NSGA-II	0.2725	0.0460	13.6238	0.2725
		SPEA2	0.1643	0	14.4873	0
		SMS-EMOA	0.0388	0.0071	14.9021	0.0160
	2500	NSGA-II	0.1497	0.0185	14.6435	0.0460
		SPEA2	0.1544	0.0298	14.8503	0
		SMS-EMOA	**0.0030**	2.8954e-4	**15.0280**	3.4727e-4
DTLZ7	250	EI-L2	4.3888	2.8159	42.4629	0.4042
		EI-L2-RBF	1.7066	1.4069	42.6332	0.3295
		HvPoI	***0.0280***	0.0037	***43.5404***	0.0188
		NSGA-II	13.9371	2.3112	23.2392	5.4733
		SPEA2	10.1169	0	37.4830	0
		SMS-EMOA	3.4186	2.2457	41.2087	1.6529
	2500	NSGA-II	9.6799	2.3516	30.7966	4.2005
		SPEA2	5.4330	0	42.1191	0
		SMS-EMOA	**0.0236**	0.0015	**43.7127**	0.0953
DTLZ5	250	EI-L2	0.2259	0.0019	197.1390	0.1453
		EI-L2-RBF	0.2286	0.0013	196.8852	0.1777
		HvPoI	*0.0835*	0.0053	***198.6425***	0.1563
		NSGA-II	0.0656	0.0376	192.1285	2.0064
		SPEA2	0.1475	0	192.6617	0
		SMS-EMOA	**0.0467**	0.0268	196.0038	0.6004
	2500	NSGA-II	**0.0727**	0.0162	194.9017	0.3805
		SPEA2	0.2151	0	194.3750	0
		SMS-EMOA	0.1141	0.0070	**198.5351**	0.0343

Table 2: Results of the hypervolume-based acquisition functions, NSGA-II, SPEA2 and SMS-EMOA. The best results for each test function are highlighted in bold, for each performance metric and within the same sample budget. The best results are marked as italic

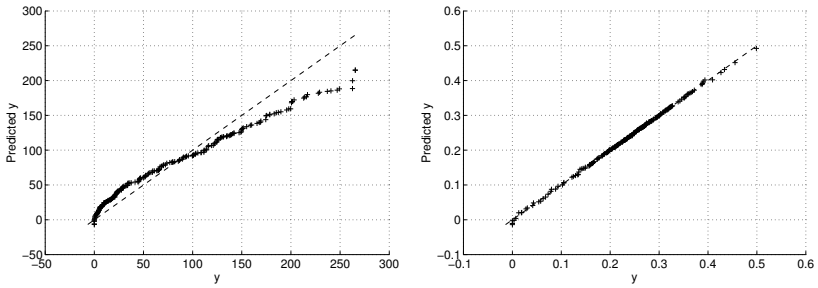


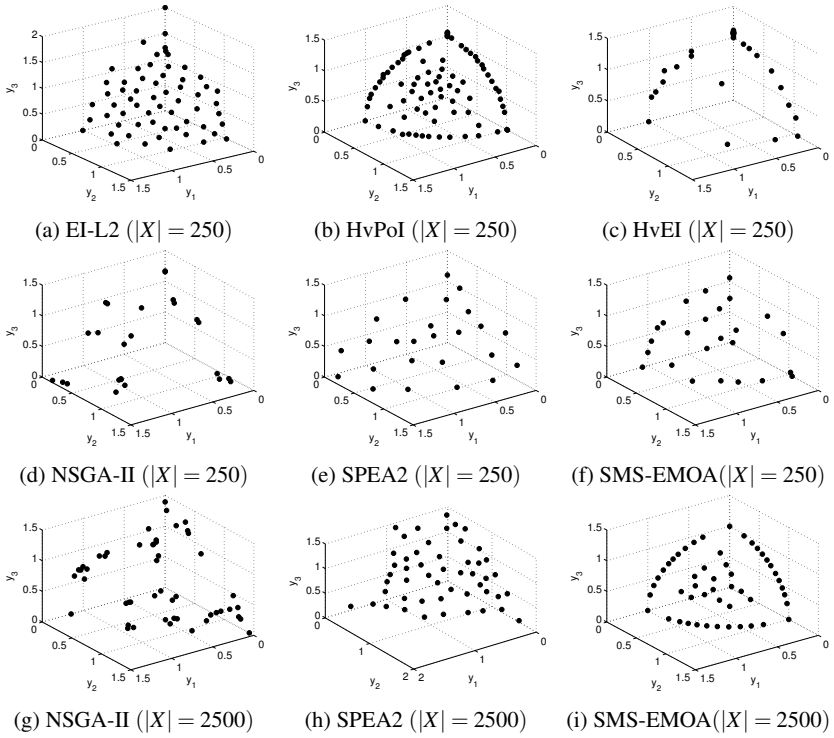
Fig. 3: 20-fold cross validation applied on the Kriging models based on 250 samples. The black dots denote the cross validated prediction values versus the real objective values. a) Final Kriging model of the first objective function of the DTLZ1 function. It is seen that Kriging has problems approximating the larger values of the objective function. b) Final Kriging model of the first objective function of the DTLZ5 function. Kriging is able to approximate the objective function quite well

is due to the nature of the hypervolume indicator. Logically, the farther away the reference point is located, the larger the exclusive hypervolume will be for points lying on the edge of the current Pareto set (as the exclusive hypervolume is then solely bounded by the reference point). Further research is needed to determine the influence of the choice of reference point $\mathbf{f}^{\max}$ on the statistical criteria (Auger et al., 2009).

While the Bayesian optimization algorithms outperforms the MOEAs on the hypervolume indicator on most problems, there are some limitations. These techniques, rely on the quality of the underlying surrogate model to guide the selection of new expensive data points. The Kriging models do not have to be accurate at the start of the algorithm when using the HvEI and HvPoI criteria, but they should be able to capture the behavior of the objective functions sufficiently well when enough samples become available, which might not always be the case (see Fig. 3 and the DTLZ1 results). Furthermore, the construction of the Kriging models and the evaluation of the statistical criteria comes at a computational cost, similar to the computational cost of MOEAs that rely on the hypervolume (i.e., SMS-EMOA), which might limit the practical usage of these criteria for some (less expensive) optimization problems.

4 Multi-objective Bayesian optimization with Student-*t* processes

As illustrated in Section 3.2, the quality of the model is crucial even though we do not require accuracy over the entire domain. If the model fit is problematic like in case of the DTLZ1 function, the optimization performance decreases significantly. In this section we consider the use of a different class of function priors known as



(a) EI-L2 ($|X| = 250$) (b) HvPoI ($|X| = 250$) (c) HvEI ($|X| = 250$)

(d) NSGA-II ($|X| = 250$) (e) SPEA2 ($|X| = 250$) (f) SMS-EMOA($|X| = 250$)

(g) NSGA-II ($|X| = 2500$) (h) SPEA2 ($|X| = 2500$) (i) SMS-EMOA($|X| = 2500$)

Fig. 4: Generated Pareto sets of the DTLZ2 function. The hypervolume-based metric focuses more on sampling the edge (extrema) of the Pareto front, while the Euclidean distance-based criterion performs a seemingly more uniform search over the Pareto front, though it performs slightly worse on the hypervolume metric

Student-*t* processes as surrogate model and show how it improves the performance for the DTLZ1 case.

4.1 Student-t processes

Given a d-dimensional input space $\mathscr{X} \subset \mathbb{R}^d$, f is a Student-*t* process with degrees of freedom $\nu > 2$, a continuous mean function γ and a parametrized kernel function k. For any set $\mathbf{X} \subset \mathscr{X}$ of n inputs $\mathbf{x}$, the (noisy) observations of the mapping of these inputs by f is distributed according to a Multivariate Student-t distribution (MVT): $\mathbf{y} \sim \mathrm{MVT}_n(\nu, \gamma, \mathbf{K} + \sigma_n^2 \mathbf{I})$ with $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. The likelihood corresponds to the probability density function of a MVT:

$$p(\mathbf{y}|\mathbf{X},\nu,\theta) = \frac{\Gamma\left(\frac{\nu+n}{2}\right)}{((\nu-2)\pi)^{\frac{n}{2}}\Gamma\left(\frac{\nu}{2}\right)}|\mathbf{K}|^{-1/2}\left(1+\frac{\beta}{\nu-2}\right)^{-\frac{\nu+n}{2}}, \tag{6}$$

with $\beta = (\mathbf{y}-\mu)^T K(\mathbf{y}-\mu)$. Shah et al. (2014) have shown that considering $\mathbf{y}|\sigma \sim GP(\gamma,(\nu-2)\sigma)$ and marginalizing σ out assuming an *inverse Wishart process* prior, recovers Eq. (6). For an arbitrary $\mathbf{x}_\star \in \mathscr{X}$ the predictive distribution is also a MVT:

$$\begin{aligned} f(\mathbf{x}_\star)|\mathbf{X},\mathbf{y},\theta,\nu &\sim \mathrm{MVT}_1\left(\nu+n,\mu(\mathbf{x}_\star),s^2_{\mathrm{tp}}(\mathbf{x}_\star)\right), \\ s^2_{\mathrm{tp}}(\mathbf{x}_\star) &= \frac{\nu+\beta-2}{\nu+n-2}s^2(\mathbf{x}_\star). \end{aligned} \tag{7}$$

The quantities μ and s^2 are identical to the predictive mean and variance of a GP (assuming the same kernel and parameters). Recent work also shows marginalizing the output scale also yields a related MVT predictive distribution (Gramacy and Apley, 2015; Montagna and Tokdar, 2016). This differs from non-analytical marginalization of the kernel lengthscales with Markov chain Monte Carlo methods as applied frequently in Bayesian optimization. See van der Herten et al. (2017) for a comparison of the latter with traditional maximum likelihood estimates.

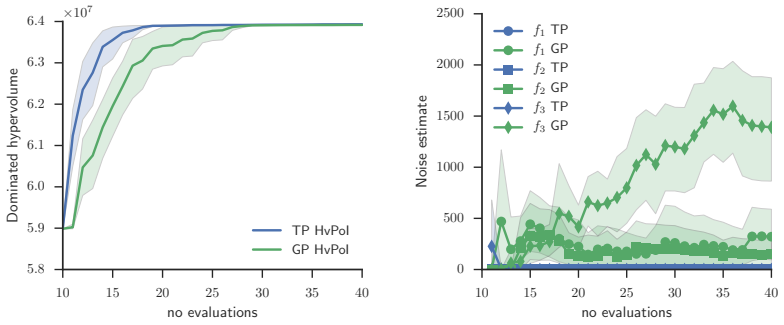
A fundamental difference is observed in Eq. (7): the variance prediction includes the observed responses, as opposed to GPs which only considers the space between inputs. This allows a TP to anticipate changes in covariance structure. Furthermore, it was proven that a GP is a special case of a TP, with $\nu \to \infty$. However, the approach applied for GPs to include noise as part of the likelihood can not be applied for TPs, as the sum of two independent MVT is not analytically tractable. Instead, a diagonal white noise kernel is added to allow approximation of noisy observations.

4.2 Hypervolume-based probability of improvement

We study the HvPoI as introduced earlier in Section 3 as it is tractable and scales to a higher number of objectives, however we assume each $f^{(i)} \sim TP$ instead of a GP. The algorithm only needs a single modification:

$$p(\mathbf{x}\in\mathscr{D}) = \sum_{k=1}^{z}\prod_{j=1}^{p}\left(\Phi_{\nu+n}\left(\frac{u_j^k-\mu^{(j)}(\mathbf{x})}{s^{2,(j)}_{\mathrm{tp}}(\mathbf{x})}\right)\Phi_{\nu+n}\left(\frac{l_j^k-\mu^{(j)}(\mathbf{x})}{s^{2,(j)}_{\mathrm{tp}}(\mathbf{x})}\right)\right).$$

Here, Φ_ν represents the cumulative density function of a $\mathrm{MVT}_1(\nu,0,1)$ instead of the standard normal distribution.



(a) (b)

Fig. 5: DTLZ1 function: (a) Comparison of the growth of the dominated hypervolume for the DTLZ1 function, for 10 experiments using both GP and TP priors for the objectives. The mean and 95% confidence intervals are shown. (b) The noise parameter for all three objectives approximated by GP and TP. For GP, the noise is part of the likelihood whereas for TP a diagonal matrix was added to the kernel matrix. Clearly, the TPs are more flexible and do not consider the evaluated data noisy.

4.3 Illustration

We illustrate the effectiveness of the TP prior by revisiting the DTLZ1 function. As illustrated, some difficulties approximating the first objective may occur, hence we try HvPoI in combination with GP priors, and compare it with the modified version as introduced in Section 4.2 with TP priors. The initial set of data points consists of an optimized Latin Hypercube of 10 points. The acquisition function is then permitted to select an additional 30 data points for evaluation. For both TP and GP, the RBF kernel was used, and the hyperparameters θ including ν were optimized with multi-start SQP. Note that the optimization can result in a very large value ν, causing the TP to become a GP. Hence, we expect better or equal performance, not worse. Both experiments were repeated 10 times.

As performance metric, the hypervolume indicator (size of the dominated hypervolume with respect to the reference point $\mathbf{f}^{\max}$) is recorded after every function evaluation. The average hypervolume and 95% confidence intervals were computed and plotted in Fig. 5a. Clearly, the runs using the TP approximations of the objectives obtain larger hypervolumes faster. The GP experiments lag behind although they also eventually manage to obtain the same hypervolume indicator performance after additional evaluations. In the end, TPs are able to find a decent hypervolume in about 30% of the function evaluations needed by the GPs for the same hypervolume indicator performance.

Closer investigation reveals the GP approximations for some of the objective functions have large noise levels, varying significantly as more evaluations are added,

whereas the TPs do not as illustrated in Fig. 5b. It seems the GP is not flexible enough to approximate the objective functions and has to increase the noise variance to avoid ill-conditioning of the kernel matrix. The TPs compensate for this by decreasing the degrees of freedom, which also affects the prediction variance resulting in better selection of evaluation candidates.

5 Conclusion

In this chapter, we reviewed the concept of multi-objective Bayesian optimization and discussed some key hypervolume-based algorithms, as well as recent developments in terms of modeling and acquisition functions. Some algorithms were illustrated on a set of benchmark functions. It was highlighted that model failures can severely affect the performance of the optimization as was clearly the case for the Kriging models for the first objective of the DTLZ1 function. By using Student-t processes instead, this objective can be approximated better, resulting in much better performance as the Pareto front is improved faster.

Several implementations of the methods discussed can be found online, in particular we highlight the inclusion of HvPoI in GPflowOpt[2], an opensource framework for implementation of Bayesian optimization methods based on GPflow (Matthews et al., 2017), a library for Gaussian Processes in TensorFlow. This framework also permits easy implementation of acquisition functions and supports multi-objective optimization.

Acknowledgements Ivo Couckuyt is a post-doctoral research fellow of FWO-Vlaanderen.

References

Auger, A., Bader, J., Brockhoff, D., Zitzler, E.: Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point. In: Workshop Foundation Genetic Algorithms (2009)

Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research **181**(3), 1653–1669 (2007)

Bonnans, J., Gilbert, J., Lemaréchal, C., Sagastizábal, C.: Numerical Optimization: Theoretical and Practical Aspects. Springer (2006)

Brillinger, D.R.: The calculation of cumulants via conditioning. Annals of the Institute of Statistical Mathematics **21**(1), 215–218 (1969)

[2] http://github.com/gpflow/GPflowOpt

Campigotto, P., Passerini, A., Battiti, R.: Active learning of Pareto fronts. Neural networks and learning systems, IEEE transactions on **25**(3), 506–519 (2014). DOI 10.1109/TNNLS.2013.2275918

Couckuyt, I., Deschrijver, D., Dhaene, T.: Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization. Journal of Global Optimization **60**(3), 575–594 (2014a). DOI 10.1007/s10898-013-0118-2

Couckuyt, I., Dhaene, T., Demeester, P.: ooDACE Toolbox: A Flexible Object-Oriented Kriging Implementation. Journal of Machine Learning Research **15**, 3183–3186 (2014b)

Cox, D.D., John, S.: SDO: A statistical method for global optimization. Multidisciplinary design optimization: state of the art pp. 315–329 (1997)

Damianou, A.: Deep Gaussian processes and variational propagation of uncertainty. Ph.D. thesis, University of Sheffield (2015)

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. Evolutionary Computation, IEEE Transactions on **6**(2), 182–197 (2002). DOI 10.1109/4235.996017

Deb, K., Thiele, L., Laummans, M., Zitzler, E.: Scalable test problems for evolutionary multi-objective optimization. Tech. Rep. 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (2001)

Duvenaud, D., Lloyd, J.R., Grosse, R., Tenenbaum, J.B., Ghahramani, Z.: Structure Discovery in Nonparametric Regression through Compositional Kernel Search. In: Proceedings of the 30th International Conference on Machine Learning, pp. 1166–1174 (2013)

Emmerich, M.T.M., Deutz, A.H., Klinkenberg, J.W.: Hypervolume-based expected improvement: Monotonicity properties and exact computation. In: Emmerich, M.T.M., Hingston, P. (eds.) Congress on Evolutionary Computation (CEC), pp. 2147–2154. IEEE, Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey, USA (2011). DOI 10.1109/CEC.2011.5949880

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

Forrester, A.I.J., Jones, D.R.: Global optimization of deceptive functions with sparse sampling. In: 12th AIAA/ISSMO multidisciplinary analysis and optimization conference, vol. 1012. Aerospace Research Central (2008). DOI 10.2514/6.2008-5996

Freitas, N.D., Zoghi, M., Smola, A.J.: Exponential Regret Bounds for Gaussian Process Bandits with Deterministic Observations. In: Langford, J., Pineau, J. (eds.) Proceedings of the 29th International Conference on Machine Learning (ICML-12), pp. 1743–1750. ACM, New York, NY, USA (2012)

Frohlich, H., Zell, A.: Efficient parameter selection for support vector machines in classification and regression via model-based global optimization. In: IEEE International Joint Conference on Neural Networks, IJCNN'05, vol. 3, pp. 1431–

1436. IEEE, Institute of Electrical and Electronics Engineers, Inc, Piscataway, New Jersey, USA (2005). DOI 10.1109/IJCNN.2005.1556085

Garnett, R., Osborne, M.A., Hennig, P.: Active learning of linear embeddings for Gaussian processes. In: M.L., Z., J., T. (eds.) Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence, pp. 230–239. AUAI Press (2014)

Goethals, K., Couckuyt, I., Dhaene, T., Janssens, A.: Sensitivity of night cooling performance to room/system design: Surrogate models based on CFD. Building and Environment **58**, 23–36 (2012). DOI 10.1016/j.buildenv.2012.06.015

Gramacy, R.B., Apley, D.W.: Local Gaussian process approximation for large computer experiments. Journal of Computational and Graphical Statistics **24**(2), 561–578 (2015). DOI 10.1080/10618600.2014.914442

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

Hernández-Lobato, D., Hernández-Lobato, J.M., Shah, A., Adams, R.P.: Predictive Entropy Search for Multi-objective Bayesian Optimization. In: Balcan, M.F., Weinberger, K.Q. (eds.) Proceedings of the 33rd International Conference on Machine Learning (ICML-16), *Proceedings of Machine Learning Research*, vol. 48, pp. 1492–1501. PMLR (2016)

Hernández-Lobato, J.M., Hoffman, M.W., Ghahramani, Z.: Predictive entropy search for efficient global optimization of black-box functions. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 918–926. Curran Associates, Inc. (2014)

van der Herten, J., Couckuyt, I., Deschrijver, D., Dhaene, T.: Fast Calculation of the Knowledge Gradient for Optimization of Deterministic Engineering Simulations. Submitted to the Journal of Machine Learning Research (JMLR) (2017)

Hupkens, I., Emmerich, M., Deutz, A.: Faster computation of expected hypervolume improvement. arXiv preprint arXiv:1408.7114 (2014)

Jones, D.R., Schonlau, M., Welch, W.J.: Efficient Global Optimization of Expensive Black-Box Functions. J. of Global Optimization **13**(4), 455–492 (1998). DOI 10.1023/A:1008306431147

Keane, A.J.: Statistical Improvement Criteria for Use in Multiobjective Design Optimization. AIAA Journal **44**(4), 879–891 (2006)

Knowles, J.: ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. Evolutionary Computation, IEEE Transactions on **10**(1), 50–66 (2006). DOI 10.1109/TEVC.2005.851274

Malkomes, G., Schaff, C., Garnett, R.: Bayesian optimization for automated model selection. In: Advances in Neural Information Processing Systems, pp. 2900–2908 (2016)

Matthews, A.G.d.G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., Hensman, J.: GPflow: A Gaussian process library using TensorFlow. Journal of Machine Learning Research **18**(40), 1–6 (2017). URL http://jmlr.org/papers/v18/16-537.html

Metzen, J.H.: Minimum Regret Search for Single- and Multi-Task Optimization. In: Balcan, M.F., Weinberger, K.Q. (eds.) Proceedings of the 33rd International

Conference on Machine Learning (ICML-16), *Proceedings of Machine Learning Research*, vol. 48, pp. 192–200. PMLR, New York, New York, USA (2016)

Montagna, S., Tokdar, S.T.: Computer Emulation with Nonstationary Gaussian Processes. SIAM/ASA Journal on Uncertainty Quantification **4**(1), 26–47 (2016). DOI 10.1137/141001512

Močkus, J.: On Bayesian methods for seeking the extremum. In: Marchuk, G. (ed.) Optimization Techniques IFIP Technical Conference, pp. 400–404. Springer, Springer Berlin Heidelberg, Berlin, Heidelberg (1975)

Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). MIT Press (2006)

Santner, T., Williams, B., Notz, W.: The design and analysis of computer experiments. Springer series in statistics. Springer-Verlag, New York (2003)

Shah, A., Ghahramani, Z.: Pareto Frontier Learning with Expensive Correlated Objectives. In: Balcan, M.F., Weinberger, K.Q. (eds.) Proceedings of The 33rd International Conference on Machine Learning, *Proceedings of Machine Learning Research*, vol. 48, pp. 1919–1927. PMLR, New York, New York, USA (2016)

Shah, A., Wilson, A.G., Ghahramani, Z.: Student-t Processes as Alternatives to Gaussian Processes. In: AISTATS, Proceedings of Machine Learning Research, pp. 877–885. PMLR (2014)

Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 2951–2959. Curran Associates, Inc. (2012)

Stein, M.: Interpolation of Spatial Data: Some Theory for Kriging. Springer-Verlag (1999)

Van Dam, E.R., Husslage, B., Den Hertog, D., Melissen, H.: Maximin Latin hypercube designs in two dimensions. Operations Research **55**(1), 158–169 (2007). DOI 10.1287/opre.1060.0317

Wang, Z., Jegelka, S.: Max-value Entropy Search for Efficient Bayesian Optimization. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, *Proceedings of Machine Learning Research*, vol. 70, pp. 3627–3635. PMLR, International Convention Centre, Sydney, Australia (2017)

While, L., Bradstreet, L., Barone, L.: A fast way of calculating exact hypervolumes. Evolutionary Computation, IEEE Transactions on **16**(1), 86–95 (2012). DOI 10.1109/TEVC.2010.2077298

Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Tech. rep. (2001)

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., da Fonseca, V.G.: Performance assesment of multiobjective optimizers: an analysis and review. Evolutionary Computation **7**(2), 117–132 (2003)