Optimization, Fast and Slow: Optimally Switching between Local and Bayesian Optimization

Mark McLeod [1] **Michael A. Osborne** [1 2] **Stephen J. Roberts** [1 2]

Abstract

We develop the first Bayesian Optimization algorithm, BLOSSOM, which selects between multiple alternative acquisition functions and traditional local optimization at each step. This is combined with a novel stopping condition based on expected regret. This pairing allows us to obtain the best characteristics of both local and Bayesian optimization, making efficient use of function evaluations while yielding superior convergence to the global minimum on a selection of optimization problems, and also halting optimization once a principled and intuitive stopping condition has been fulfilled.

1. Introduction

1.1. Bayesian Optimization with Gaussian Processes

In Bayesian Optimization we are concerned with the global optimization of a black box function. This function is considered to be expensive to evaluate, and we are therefore willing to undertake considerable additional computation in order to achieve efficient use of evaluations. Bayesian Optimization has been applied to may problems in machine learning, including hyperparameter tuning (Snoek et al., 2012; Hernndez-Lobato et al., 2014), sensor set selection (Garnett et al., 2010) and tuning robot gait parameters (Calandra et al., 2016; Lizotte et al., 2007; Tesch et al., 2011). A recent review of the field is Shahriari et al. (2016).

To achieve this aim at each iteration we first train a model of the objective conditioned on the data observed so far. This model is usually a Gaussian Process, a kernel-based model with particularly useful properties. A full introduction to the Gaussian Process is given by Rasmussen & Williams (2006). However, the relevant properties to this work are: that all posteriors produced by the GP are multivariate Gaussian, so provide both the estimated value and the uncertainty of that estimate; and that this joint Gaussian property also extends to derivatives of the function being modelled.

We next define an acquisition function over the optimization domain which states how useful we expect an evaluation at that location to be. This function is optimized to find the location predicted to be most useful. The true objective is then evaluated at this location. There are a large number of acquisition functions available. Two that are relevant to this work are Expected Improvement (Jones et al., 1998), in which we choose the point with the greatest improvement in expectation on the best value observed so far, and Predictive Entropy Search (PES) (Hernndez-Lobato et al., 2014), in which we choose the location expected to yield the greatest change in the information content of the distribution over our belief about the location of the global minimum.

We contribute a novel algorithm, Bayesian and Local Optimisation Sample-wise Switching Optimisation Method, BLOSSOM, which combines the desirable properties of both local and Bayesian optimization by selecting from multiple acquisition functions at each iteration. We retain the evaluation-efficient characteristics of Bayesian optimization, while also obtaining the superior convergence of local optimization. This is combined with a Bayesian stopping criterion allowing optimization to be terminated once a specified tolerance on expected regret has been achieved, removing the need to pre-specify a fixed budget.

1.2. Requirement for a Stopping Criterion

In the majority of work on Bayesian optimization the problems considered either fix the number of iterations or, less often, fix a computational budget. While this is clearly desirable for averaging over and comparing multiple repetitions of the same optimization, it is not desirable in practice: the number of steps to take (or budget to allow) is now an additional parameter that must be selected manually. This choice requires expert knowledge of Bayesian Optimization to select a number that hopefully will neither be too small, resulting in easy improvement being neglected, or too large, costing additional expensive evaluations for minimal gain. We are therefore motivated to seek an automatic stopping

[1]Department of Engineering Science, University of Oxford [2]Oxford-Man Institute of Quantitative Finance. Correspondence to: Mark McLeod <markm@robots.ox.ac.uk>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

criterion.

Early stopping has been considered by Lorenz et al. (2015) in an application of Bayesian Optimization to brain imaging experiments. They propose and test early stopping based on the Euclidean distance between consecutive evaluations of the objective, and based on the probability of improvement on the incumbent solution. Both of these criteria provide notable improvement on a fixed number of iterations. However, both these criteria are strictly *local* quantities with no consideration of the GP model at locations removed from the incumbent solution and proposed next location. The values must still be selected by the user so that optimization is not terminated undesirably early by an incremental exploitative step while regions of the optimization domain remain unexplored. We would prefer a stopping criterion which takes account of the full model of the objective, and which has a parameter more easily interpreted in terms of the expected difference between the proposed and true solutions.

1.3. Convergence Properties

Optimization has excellent empirical performance in identifying the minimizer of multi-modal functions with only a small number of evaluations. Bull (2011) has shown that $\mathcal{O}(n^{-\frac{v}{d}})$ convergence, where v is the smoothness of the kernel, can be achieved with a modified version of Expected Improvement. However the authors note that this is only applicable for fixed hyperparameters. We are not aware of any estimates on convergence for PES, which exhibits better performance empirically . Furthermore, even given a guarantee of convergence in theory, details of the implementation of Bayesian Optimization ensure that the final regret is unlikely to fall to less than a few orders of magnitude below the scale of the objective function. Firstly, we are not able to exactly maximize the acquisition function. Constraints placed on the number of evaluations available to the inner optimizer limit our ability to select evaluation points to a high degree of accuracy. This limit is also relevant to minimizing the posterior for our final recommendation. Secondly, even in a noiseless setting, we must add diagonal noise to our covariance matrix to resolve numerical errors in performing the Cholesky decomposition. This reduces the rate of convergence available to that of optimizing an objective with the noise level we have now implicitly imposed. As we cluster more evaluations closely around the minimum, the conditioning of the covariance matrix degrades further. The potential loss in performance due to diagonal noise is illustrated in Figure 1. We therefore also desire to create an optimization routine which does not suffer from this ineffective exploitation property.

This convergence issue has been addressed by Dhaenens et al. (2015) who switch from Bayesian Optimization to CMA-ES once a criteria on probability of improvement has been achieved. This provides excellent convergence on the objectives tested. However the switching relies on a heuristic based on a combination of previous values of the acquisition function and the number of steps without improvement of the incumbent. We would prefer to make use of a Bayesian criterion for any changes in behaviour, to avoid the need for additional parameters which must be manually chosen by an expert. By using a non-stationary kernel which replaces the squared exponential form with a quadratic kernel in regions around local minima, Wabersich & Toussaint (2016) also aim to achieve superior convergence. However, they do not show the final value achieved by their method in most experiments, and the use of fixed pre-trained hyperparameter samples makes their implementation unsuitable for an online setting.

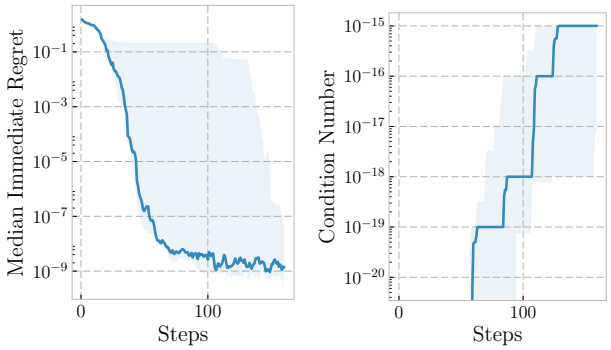

Figure 1. True immediate regret and conditioning number of the covariance matrix under optimization of the Hartman-4D function using the Predictive Entropy Search acquisition function. The median and quartiles of 16 repetitions of the optimization are shown. Rather than adding a fixed diagonal component to the GP kernel matrix to ensure stability we have added the minimum value which still allows the Cholesky decomposition to succeed, in decade increments starting from 10^{-20}. The optimization achieves a good result quickly, but then as jitter is added there is no further improvement beyond roughly $\sqrt{\text{jitter}}$.

We now develop our algorithm, which achieves superior convergence and terminates once a well-defined condition has been achieved. In §2 we outline the behaviour of the algorithm, which selects between multiple acquisition functions on each iteration. We detail in §3 how we approximate the numerical quantities required, then in §4 provide results demonstrating the effectiveness of our new method.

2. The Algorithm

2.1. Separating Global and Local Regret

In Bayesian Optimization we aim to minimize the difference between the function value at the final recommended point, $\hat{y} = f(\hat{x})$ and the value at the true global minimizer $y_* = f(x_*)$. This is the *regret* of selecting the current rec-

ommendation as the final solution. We shall now separate this concept into two distinct components which we treat separately. Let S be some region of note containing the incumbent solution. We define y_i as the minimum value of the objective within S and y_o as the minimum value outside S. We can then write

$$\begin{aligned} \text{Regret} &= \mathbb{E}(\hat{y} - y_*) \\ &= \mathbb{E}(\hat{y} - y_i) + \mathbb{E}(y_i - y_*) \\ &= \mathbb{E}(\hat{y} - y_i) + \mathbb{E}(\max(y_i - y_o, 0)) \\ &= R_{\text{local}} + R_{\text{global}} \,, \end{aligned} \tag{1}$$

where we have split the full regret into a local component, due to the difference between our candidate point and the associated local minimum, and a global component, due to the difference between the local and global minima. Both components are non-negative by definition, and we expect both to decrease as we learn about the objective. The local component represents the difference between our incumbent and the minimum within S. It is reduced by exploitation of the objective. The global component represents the difference between the local minimum and the global minimum. It will be reduced by exploration, and also by exploitation of other local minima. There is a finite probability that $R_{\text{global}} = 0$, corresponding to the probability that the global minimum is in fact the minimum of the basin containing our incumbent.

2.2. Multiple Acquisition Functions

To address the issues identified above, we split our optimization into four distinct modes, intending to use the most effective at each iteration. The modes are; Random Initialization, Bayesian Optimization, Global Regret Reduction and Local Optimization.

Random Initialization is, as usual, only required for the first few iterations. Bayesian Optimization using Predictive Entropy Search is our default acquisition function if no relevant conditions to change behaviour have been satisfied: PES provides the usual balance between exploration and exploitation.

In steps when a distinct candidate minimum can be identified, we switch to the predominantly explorative strategy of Global Regret Reduction, intended to reduce the global regret. By making this change, we avoid the inefficient convergence of exploitation due to poor conditioning in Gaussian Processes model when used for Bayesian Optimization. To identify a candidate minimum we require the existence of a region surrounding the minimum of the $\mathcal{GP}$ posterior with a high probability of being convex.

Once the predicted global regret has fallen below some target value we use a purely exploitative Local Optimization algorithm. In this work, we assume that we have access to noiseless evaluations of the objective functions so that we can employ a quasi-Newton local optimization routine, such as BFGS (Nocedal & Wright, 2006), which delivers super-linear convergence to the local minimum and is free of the numerical conditioning problems present in Gaussian Processes. We note that since we are starting our local optimization very close to the minimum (in fact we choose to start only when the $\mathcal{GP}$ model predicts a convex region), only a small number of steps should be needed to achieve any required local regret target. We are then able to stop optimization, having achieved a target total expected regret.

We now give further detail on the two new modes of optimization used by BLOSSOM. The methods used share many expensive computations with PES, so by reusing these results we do not incur too large an additional overhead.

2.2.1. Global Regret Reduction

Once a region around the posterior minimum has been identified within which local optimizations are likely to converge to the same location we do not wish to perform exploitation within this region with Gaussian Processes, as this leads to numerical conditioning issues and therefore does not use evaluations efficiently. Instead we wish to set this region aside for pure exploitation under a local search strategy. We therefore direct our efforts towards reducing the probability of any other local minima which might take lower values than our incumbent solution existing, reduction of the global regret. We use a modified form of expected improvement to achieve this, where instead of taking improvement with respect to the lowest observed objective value we compare to the estimated value of y_i that would be obtained by starting a local optimization from the current incumbent. The acquisition function used is therefore

$$\alpha_{\text{GRR}} = (\mathbb{E}(y_i) - \mu)\Phi\left(\frac{\mathbb{E}(y_i) - \mu}{\sigma}\right) + \sigma\phi\left(\frac{\mathbb{E}(y_i) - \mu}{\sigma}\right) \tag{2}$$

where μ and σ^2 are the $\mathcal{GP}$ posterior mean and variance, y_i is the minimum value within a defined region around the posterior minimum and Φ and ϕ are the unit Normal cdf and pdf respectively.

2.2.2. Local Optimization

Once we have a both sufficiently high certainty that the $\mathcal{GP}$ posterior minimum is close to a minimum of the objective, and that that minimum is in fact global, we wish to exploit fully.

We use the BFGS algorithm for local optimization. This is a second order algorithm which updates an estimate of the Hessian at each step. By using our estimate of the Hessian available from the GP model as the initial estimate in BFGS,

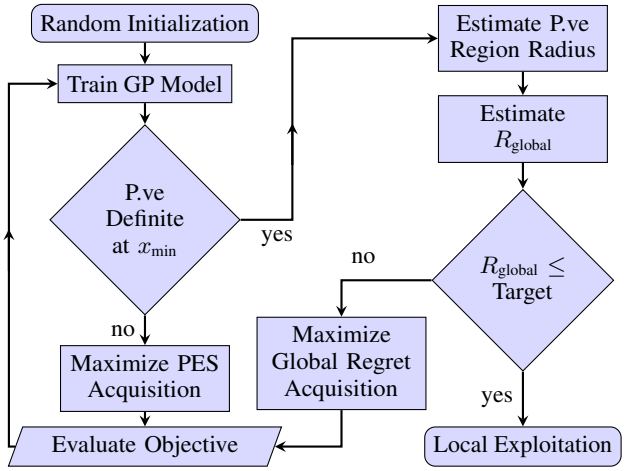

Figure 2. Flowchart for acquisition function switching behavior. Following initialization the acquisition function may switch between PES and Regret Reduction until R_{Global} achieves a sufficiently low value. The final steps are then under the local exploitation strategy.

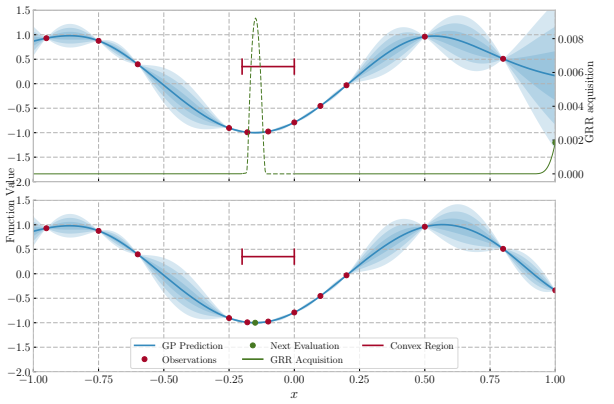

Figure 3. Two steps of BLOSSOM in an example problem. In the upper plot the GRR acquisition has a maximum at -0.15. However, since this falls within the convex region around the minimum of the posterior mean the next function evaluation is taken at the highest location outside this region. In the lower plot the estimated Global Regret is sufficiently low that no further Bayesian iterations are required. The next evaluation is the start of a local optimization from -0.15.

we hope to achieve convergence with fewer evaluations of the objective than otherwise. Rather than modifying the BFGS algorithm to use this estimate, we rescale the problem so that the expectation of the Hessian is the identity matrix. With a local function model

$$f(x) = \frac{1}{2}x^T Hx + x^T g + c \tag{3}$$

we define $z = Rx$ where $R^{-T} = C$, $H = CC^T$ the Cholesky decomposition of H. This gives us a modified function

$$g(z) = \frac{1}{2}z^T z + z^T R^T g + c \tag{4}$$

as required. Once we have started this process we are no longer performing Bayesian Optimization and can continue using BFGS until convergence. By selecting an appropriate stopping condition for local optimization we can ensure R_{local} falls below any desired target. We have selected a gradient estimate of less that 10^{-6} as our stopping condition, but any other method could be used.

2.3. Switching Between Acquisition Functions

We have specified that we wish to make decisions on the basis of the existence of a candidate minimum, and on the value of the global regret. In Figure 2 we show the decision making process used. However, we have not yet specified these criteria exactly. We choose to consider the existence of a sphere with non-zero radius, centred on the minimum of the GP posterior mean, x_{min}, within which the objective function has a high probability of being convex at all points. Local optimization routines have excellent performance on convex functions, so if our model predicts a convex region surrounding a local minimum with high confidence we no longer desire our Bayesian Optimization to recommend inefficient exploitative evaluations in this region. We therefore switch to the Global Regret Reduction acquisition function, which will place a high weight on exploration.

We have defined the global regret as the difference between the objective value at the local minimizer in some region and the true global minimum. We choose to use the positive definite sphere to define this region. We can then obtain an estimate of Global Regret. If this estimate falls below our target value we move to the final stage of optimization and use Local Exploitation, otherwise we continue with the Global Regret Reduction. This process is illustrated in Figure 3

3. Estimating Required Quantities

We now detail the procedures used to estimate the quantities required in our switching criteria. We first detail our method of determining a positive definite region, then provide a method for estimating the global regret and expected local minimum value.

3.1. Identifying a Convex Region

Convexity is characterized by the Hessian matrix of the objective being positive definite at all points. For a matrix H to be positive definite we require

$$x^T Hx > 0 \tag{5}$$

for all x. Given a Gaussian Process model of the objective we would like to construct:

1. A method to determine, using our GP model, the probability that the Hessian is positive definite at any point; and

2. A method to determine, using our GP model, the largest region centred on the current posterior minimum with a required probability of being convex at all points within that region.

3.1.1. Convexity at a Point

We make use of the Cholesky decomposition to determine if a matrix is positive definite. A unique real solution to the Cholesky decomposition of a matrix only exists if the matrix is positive definite. Implementations of the routine commonly return an error rather than computing the complex solution. We can make use of this behaviour to provide a test for positive definiteness returning a binary result: $D(X) : \mathbb{R}^{\frac{d(d+1)}{2}} \rightarrow \{0, 1\}$ where d is the dimensionality of the problem.

Since under a Gaussian Process model there will always be non-zero probability over all real values of inferred quantities we can never have certainty of positive definiteness. We therefore wish to determine the probability that the Hessian of our objective at some point x is positive definite (or if the point is on the boundary of the domain the Hessian for the remaining dimensions) under our GP model. All elements of the Hessian have a joint Normal distribution $H \mid x, M \sim \mathcal{N}\big(H_\mu(x), H_\sigma(x)\big)$ with mean and variance given by the GP posterior at x (only elements of the upper triangle need to be included in implementation since the Hessian is symmetric). The probability of positive definiteness at x is then

$$\begin{aligned} p\big(D(H) \mid x, M\big) &= \int p\big(D(H) \mid H\big) p\big(H \mid x, M\big) dH \\ &= \int D(H) p\big(H \mid x, M\big) dH \\ &= \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} D(h_i) \end{aligned} \tag{6}$$

where M is our GP model and the h_i have been drawn from the multivariate normal $p(H \mid x, M)$.

As our test of positive definiteness at a point we require all of some n samples of H to be positive definite. We treat the positive definiteness of samples from $p(H \mid x, M)$ as Bernoulli distributed with rate parameter θ, since it is a deterministic binary output function of H. Taking a uniform prior on θ the posterior expected value of θ is

$$\mathbb{E}[\theta] = \int_0^1 \theta p(\theta) d\theta = \frac{n+1}{n+2}\,. \tag{7}$$

Passing our test for positive definiteness at a point, as described in Algorithm 1, can therefore be interpreted as determining that $\mathbb{E}(\theta) = 1 - \epsilon$ where $\epsilon = \frac{1}{n+2}$ while failure implies $\mathbb{E}(\theta) < 1 - \epsilon$.

Algorithm 1 Positive Definite Test

Input: location x, tolerance ϵ
`G` $\leftarrow$ `GP_model`
`Hmean, Hvar` $\leftarrow$ `G.infer_Hessian`(x)
`PVEcount` $\leftarrow 0$
$n \leftarrow \frac{1}{\epsilon} - 2$
for $i = 1...n$ **do**
  $h \leftarrow$ `draw_Gaussian`$($`Hmean, Hvar`$)$
  $h^* \leftarrow$ `remove_boundary_elements`(h)
  if `Cholesky`$(h^*) \neq$ FAIL **then**
    `PVEcount` $\leftarrow$ `PVEcount` $+ 1$
  end if
end for
$p \leftarrow \frac{PVE\text{count}+1}{n+2}$
Return $p \geq 1 - \epsilon$

3.1.2. Radius of a Convex Region

The method above allows us to effectively test a point for convexity. We now wish to use this function to find a convex region centred around the posterior minimum (again we exclude any axes on the boundary of the search domain). We choose to find the hypersphere centred at x_{min} with the greatest possible radius R_{max}. As before we can not obtain a certain value. Instead we find an estimate, $\hat{R}_{\text{max}}$, which is the minimum distance to a non-positive definite over a finite set of test directions u.

We draw unit vectors, u, uniformly at random, by normalizing draws from the multivariate normal distribution $u = \frac{v}{|v|}$ where $v \sim \mathcal{N}(0, I_d)$. For each direction we obtain the positive definite radius

$$R_u(u) = \underset{PD(\hat{x}+ru)=1}{\arg\max} \; r \tag{8}$$

by performing a binary linesearch on r down to a resolution h_r. The first search is performed with the radius of the search domain as the upper limit, subsequent directions use the previous value of $R(u)$ as the upper limit and test the outer point first, moving on to the next direction if this point returns a convex result. We thus obtain

$$\hat{R}_{max} = \min_{u \in U} R_u(u) \tag{9}$$

which is in the minimum distance from $\hat{x}$ to the edge of the positive definite region out of $n_u = \|U\|$ random directions as our estimate of the radius of a convex spherical region centred on x_{min}.

To obtain an estimate of the global regret we must marginalize over the values of the local and global minima, y_o and y_i.

Algorithm 2 Positive Definite Sphere Radius

Input: center x_{min}, number of directions, n_u tolerance ϵ
$u \leftarrow$ `random_unit_vector`
$x_{\text{edge}} \leftarrow$ `dist_to_domain_boundary`
$\hat{R} \leftarrow \|x_{\text{min}} - x_{\text{edge}}\|$
for $i = 1...n_u$ **do**
  if $D(x + \hat{R}u) = 0$ **then**
    $\hat{R} \leftarrow$ `binarysearch`$(u, \hat{R})$
  end if
  $u \leftarrow$ `random_unit_vector`
end for
Return $\hat{R}$

We assume independence between these quantities, a reasonable assumption since alternative locations for the global minimizer are usually in separate basins to the incumbent. The expectation is therefore

$$R_{global} = \iint_{y_i \times y_o} \max(y_i - y_o, 0)\, p(y_i) p(y_o)\, dy_i\, dy_o \tag{10}$$

If we consider y_i to be well approximated by a Normal distribution $\mathcal{N}(\mu_i, \sigma_i^2)$ then we can perform the integral over y_i

$$\begin{aligned} R_{\text{global}} &= \int_{y_o} \int_{y_i=y_o}^{+\infty} \max(y_i - y_o, 0) p(y_i)\, dy_i\, p(y_o) dy_o \\ &= \int_{y_o} \left[(\mu_i - y_o)\Phi\left(\frac{\mu_i - y_o}{\sigma_i}\right) \right. \\ &\qquad \left. + \sigma_i \phi\left(\frac{\mu_i - y_o}{\sigma_i}\right) \right] p(y_o) dy_o \\ &\approx \sum_j^N (\mu_i - y_o^{(j)})\Phi\left(\frac{\mu_i - y_o^{(j)}}{\sigma_i}\right) + \sigma_i \phi\left(\frac{\mu_i - y_o^{(j)}}{\sigma_i}\right). \end{aligned} \tag{11}$$

Since we do not have an analytic form for $p(y_o)$ we are not able to perform the second integral. We instead approximate the marginalization over global regret as a summation.

To evaluate this expression we can draw N samples from our GP model and find the value of y_o in each case. This cannot be performed exactly, and instead we must take draws from the GP posterior over some set of support points X_s. Half of these are approximately drawn from the distribution of the global minimum using the method described by McLeod et al. (2017) (slice sampling over the Expected Improvement or LCB as suggested by Hennig & Schuler (2012) could equivalently be used), while half of them are drawn using rejection sampling with the GP posterior variance as an unnormalized distribution, to provide additional support in high variance regions outside the convex region. To evaluate μ_i and σ_i we can use the same set of draws, considering this time only points within the convex region, to obtain a sequence of samples of y_i which can be used for a maximum likelihood estimate of the mean and variance of a normal distribution.

4. Results

We compare BLOSSOM to Expected improvement with the PI stopping criteria of Lorenz et al. (2015), and to PES using the acquisition function value as the stopping criterion. For each algorithm we test multiple values of the stopping criteria, shown in the legend as appropriate.

4.1. In-Model Objectives

To demonstrate the effect of changing the target value of global regret we make use of objective drawn from a GP, since the effect may not be observable using any single fixed objective. For example, the Branin function has multiple equal-valued global minima. We will always achieve the global minimum, and the target regret only alters the number of steps required to terminate. We show in Figure 4 the mean regret over objectives drawn from the Matérn 5/2 kernel and note that we have achieved roughly the values we requested for expected regret.

4.2. Common Benchmark Functions

We now give results for several common test objectives for global optimization, illustrated in Figure 5. In these tests we have transformed the objectives by $y' = \log(y - y_* + 1)$. This is unrelated to our contributions, and is done as many of the functions used take the form of a flat plain surrounded by steep sides many orders of magnitude greater than the plain. This shape is extremely dissimilar to draws from the Matérn $\frac{5}{2}$ kernel used by our GP model, so yields very poor results. This is an ad-hoc transformation, and it would be preferable to either use a kernel more appropriate to the objective or learn a transform of the output space online as suggested by Snelson et al. (2004).

Neither the number of steps taken nor the regret achieved is alone a useful metric for the effectiveness of a stopping condition (few steps with high regret are obviously undesirable, but also a small decrease in regret may not be worth a much increased number of steps), so in Table 1 we have also shown the mean product of steps and regret, $\mathbb{E}[nR]$. Equal contours of this metric take the form of $y = \frac{a}{x}$, so low values indicate improved performance.

As is clear from the median curves in Figure 5, and mean final values in Table 1, we have been successful in achieving both superior local convergence and early stopping. BLOSSOM achieves the lowest mean terminal regret, and mean product of regret and iterations, for five of the six test objectives. There is considerable disparity between the plotted and tabulated results for the Hartman 3D and 4D functions.

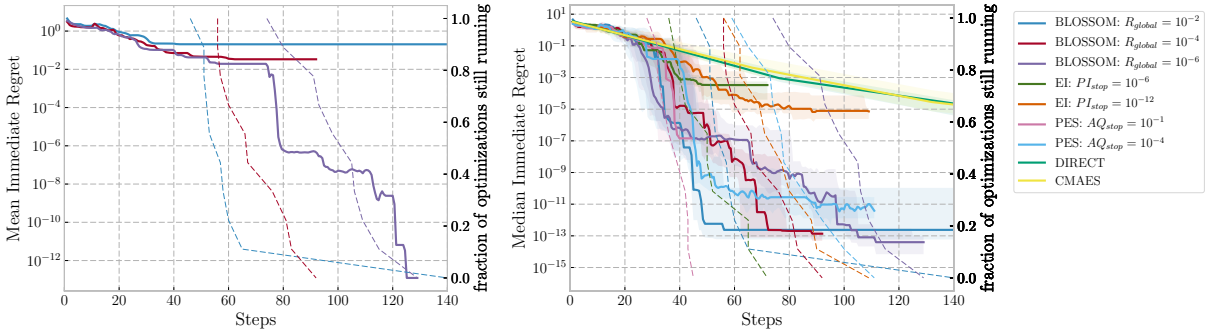

Figure 4. Comparison of methods on Draws from the Matérn 5/2 kernel in 2D (left), and illustration of the effect of changing our stopping parameter on the mean expected regret (right). Mean regret is heavily influenced by a few large values. Out of 35 runs there are those targeting 10^{-2} regret had 6 non-trivial results, targeting 10^{-4} yielded 4 non-trivial results, targeting 10^{-6} achieved the global minimum on every occasion while with no stopping condition only one failure occurred. Although with a limited number of the mean regret is not particularly close to the target values the decrease with more stringent stopping conditions is clear. We have also included DIRECT (Jones et al., 1993) and CMA-ES (**?**) to illustrate the performance on non-Bayesian methods.

However, we argue that this is in fact correct behaviour. These objectives are characterized by having multiple local minima of differing values. Usually Bayesian optimization will identify the correct basin as the global minimum and our local optimization converges to the correct value, as is evident in Figure 5. However, with some non-zero probability the GP will predict the global minimum and its surrounding positive definite region in the wrong location. If the estimated global regret is less than our target value when this occurs, the solution is accepted, leading to exploitation of a local minimum and a high final regret. This occurs on several runs of our algorithm when using a value of 10^{-2} as the target global regret. When the lower target value of 10^{-4} is used additional exploration steps are required to reduce the global regret estimate. These provide additional opportunities to correctly identify the basin of the global minimum. This leads to the much greater reliability observed in Table 1 at the cost of an increased number of objective evaluations.

4.3. GP Hyperparameter Optimization

Optimizing model hyperparameters is a common problem in machine learning. We use BLOSSSOM to optimize the input and output scale hyperparameters of a Gaussian Process using 6 months of half hourly measurements of UK electricity demand during 2015 [1]. As shown in Figure 6 we are able to obtain the best absolute result while terminating within a reasonable number of iterations, avoiding taking unnecessary further evaluations once the optimum has been achieved.

[1] www2.nationalgrid.com/UK/Industry-information/Electricity-transmission-operational-data/Data-explorer

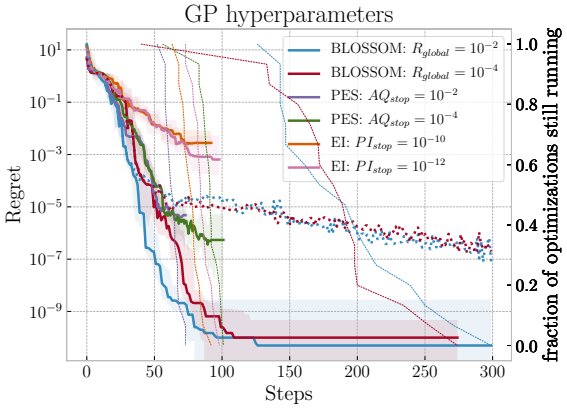

Figure 6. Comparison of stopping criteria optimizing the hyperparameter log-likelihood of a Gaussian Process. Regret is shown with respect to the best value achieved by any single optimization, the median and quartiles of 16 repetitions of each method are shown. Our method consistently obtains several orders of magnitude better convergence than PES or EI at termination.Also shown as dotted lines are the results if no final local optimization is used ($R_{\text{global}} = 0$).

5. Conclusion

We have developed BLOSSOM, a Bayesian Optimization algorithm making use of multiple acquisition functions in order to separately consider exploration and exploitation by actively selecting between Bayesian and local optimization. This separation allows us to avoid the poor local convergence of Gaussian Process methods. We are further able to halt optimization once a specified value of global regret has been achieved. This has the potential to save considerable computation in comparison to manual specification of the number of iterations to perform. We have shown that BLOSSOM is able to achieve an improvement in the final result of several orders of magnitude compared to existing methods.

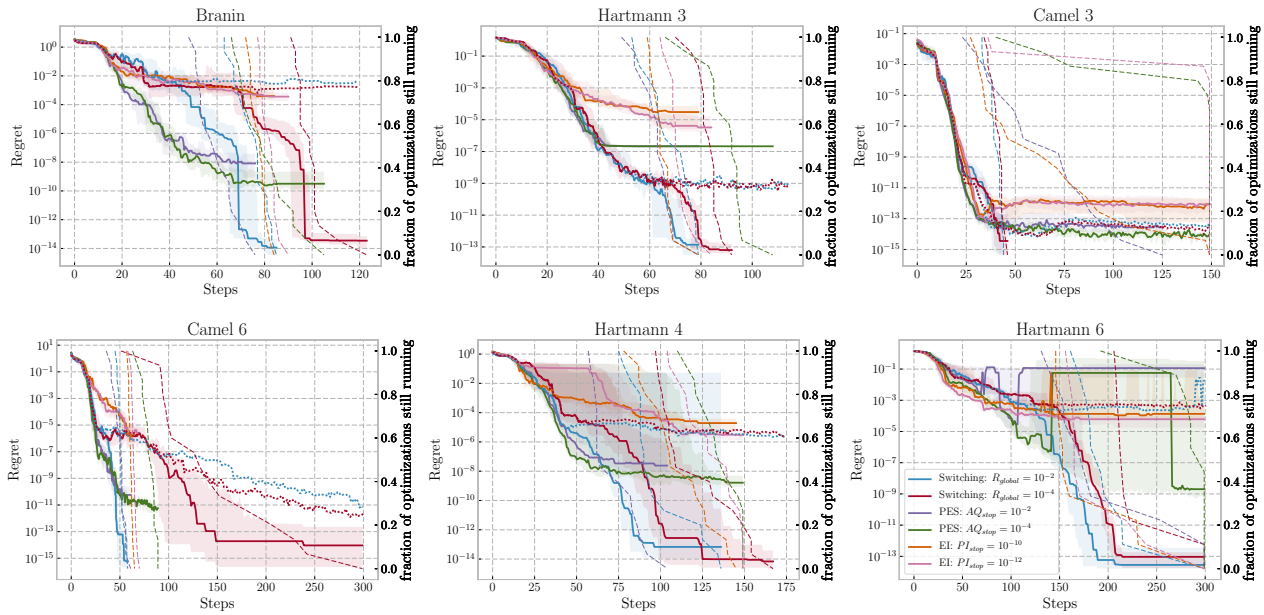

Figure 5. Comparison of various stopping methods. All stopping criteria allow an average saving compared to continuing to run optimization for many steps past convergence, but our method reliably achieves a low value before stopping. The median and quartiles of regret are shown. Fraction of optimizations still running after n steps are shown in thin dashed lines. Also shown as dotted lines are the results if no final local optimization is used ($R_{\text{global}} = 0$).

Objective	BLOSSOM 10^{-2}	BLOSSOM 10^{-4}	PES 10^{-8}	PES 10^{-10}	EI 10^{-10}	EI 10^{-12}
			Regret			
Branin	**3.32e-14**	5.2e-07	1.09e-07	2.9e-09	0.00221	0.00125
Camel 3hump	2.26e-13	1.79e-13	4.14e-13	**2.44e-14**	2.12e-12	1.57e-12
Camel 6hump	**2.28e-14**	7.95e-13	9.41e-11	1.62e-11	2.32e-05	2.35e-05
Hartmann 3D	0.107	**1.14e-13**	2.16e-07	2.16e-07	6.72e-05	0.000116
Hartmann 4D	0.0534	**5.21e-14**	0.0534	0.0133	6.06e-05	6.44e-06
Hartmann 6D	**0.00371**	0.0638	0.196	0.157	0.0229	0.0669
			Steps			
Branin	74.6	99.8	**59.6**	80.4	77.4	81.9
Camel 3hump	**39.6**	40.9	64.9	132	66.8	135
Camel 6hump	51.7	139	**49.8**	78.6	61.2	64.7
Hartmann 3D	67.8	82.6	**62.8**	89.4	65.1	71.1
Hartmann 4D	98.5	122	**72.3**	134	111	130
Hartmann 6D	199	230	196	281	**181**	190
			Steps × Regret			
Branin	**2.39e-12**	5.15e-05	5.69e-06	2.22e-07	0.167	0.103
Camel 3hump	8.56e-12	7.02e-12	1.18e-11	**2.73e-12**	1.07e-10	2.31e-10
Camel 6hump	**1.14e-12**	2.21e-10	4.84e-09	1.33e-09	0.00138	0.00157
Hartmann 3D	5.98	**9.41e-12**	1.35e-05	1.93e-05	0.00422	0.00746
Hartmann 4D	6.93	**5.89e-12**	4.36	1.95	0.00527	0.000853
Hartmann 6D	**1.11**	19.1	37.6	46.2	5.28	17.1

Table 1. Performance of selected stopping methods on various common objective functions. For two stopping criterion values for each algorithm we show the final regret, number of steps taken and step-regret product. Our methods have achieved the best regret and step-regret product on five of the six objectives used.

References

Bull, A. D. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(Oct):2879–2904, 2011. URL http://www.jmlr.org/papers/v12/bull11a.html.

Calandra, R., Seyfarth, A., Peters, J., and Deisenroth, M. P. Bayesian optimization for learning gaits under uncertainty: An experimental comparison on a dynamic bipedal walker. *Annals of Mathematics and Artificial Intelligence*, 76(1-2):5–23, February 2016. URL http://link.springer.com/10.1007/s10472-015-9463-9.

Dhaenens, C., Jourdan, L., and Marmion, M.-E. (eds.). *Learning and Intelligent Optimization*, volume 8994 of *Lecture Notes in Computer Science*. Springer International Publishing, Cham, 2015. ISBN 978-3-319-19083-9 978-3-319-19084-6. doi: 10.1007/978-3-319-19084-6. URL https://link.springer.com/chapter/10.1007/978-3-319-19084-6_29.

Garnett, R., Osborne, M. A., and Roberts, S. J. Bayesian optimization for sensor set selection. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pp. 209–219. ACM, 2010. URL http://www.robots.ox.ac.uk/~mosb/public/pdf/1242/ipsn673-garnett.pdf.

Hennig, P. and Schuler, C. J. Entropy Search for Information-Efficient Global Optimization. *Machine Learning Research*, 13(1999):1809–1837, 2012. URL http://jmlr.csail.mit.edu/papers/volume13/hennig12a/hennig12a.pdf.

Hernndez-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z. Predictive Entropy Search for Efficient Global Optimization of Black-box Functions. *Advances in Neural Information Processing Systems 28*, pp. 1–9, 2014. URL https://jmhldotorg.files.wordpress.com/2014/10/pes-final.pdf.

Jones, D. R., Law, C., and Law, C. Lipschitzian Optimization Without the Lipschitz Constant. 79(1), 1993.

Jones, D. R., Schonlau, M., and William, J. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13: 455–492, 1998. URL http://www.ressources-actuarielles.net/EXT/ISFA/1226.nsf/0/f84f7ac703bf5862c12576d8002f5259/$FILE/Jones98.pdf.

Lizotte, D. J., Wang, T., Bowling, M. H., and Schuurmans, D. Automatic Gait Optimization with Gaussian Process Regression. In *IJCAI*, volume 7, pp. 944–949, 2007. URL http://papersdb.cs.ualberta.ca/~papersdb/uploaded_files/352/additional_IJCAI07-152.pdf.

Lorenz, R., Monti, R. P., Violante, I. R., Faisal, A. A., Anagnostopoulos, C., Leech, R., and Montana, G. Stopping criteria for boosting automatic experimental design using real-time fMRI with Bayesian optimization. *arXiv preprint arXiv:1511.07827*, 2015. URL https://arxiv.org/abs/1511.07827.

McLeod, M., Osborne, M. A., and Roberts, S. J. Practical Bayesian Optimization for Variable Cost Objectives. *ArXiv e-prints*, March 2017. URL http://arxiv.org/abs/1703.04335.

Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006. URL http://www.springer.com/gb/book/9780387303031.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2006. ISBN 978-0-262-18253-9. URL http://www.gaussianprocess.org/gpml/chapters/RW.pdf. OCLC: ocm61285753.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and Freitas, N. d. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1):148–175, January 2016. URL https://www.cs.ox.ac.uk/people/nando.defreitas/publications/BayesOptLoop.pdf.

Snelson, E., Ghahramani, Z., and Rasmussen, C. E. Warped gaussian processes. In *Advances in neural information processing systems*, pp. 337–344, 2004. URL http://www.gatsby.ucl.ac.uk/~snelson/gpwarp.pdf.

Snoek, J., Larochelle, H., and Adams, R. P. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pp. 2951–2959, 2012. URL https://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf.

Tesch, M., Schneider, J., and Choset, H. Using response surfaces and expected improvement to optimize snake robot gait parameters. *IEEE International Conference on Intelligent Robots and Systems*, pp. 1069–1074, 2011. URL https://www.cs.cmu.edu/~schneide/IROS11snake.pdf.

Wabersich, K. P. and Toussaint, M. Advancing Bayesian Optimization: The Mixed-Global-Local (MGL) Kernel and Length-Scale Cool Down. *arXiv preprint arXiv:1612.03117*, 2016. URL https://arxiv.org/abs/1612.03117.