Randomised Gaussian Process Upper Confidence Bound for Bayesian Optimisation

Julian Berk*, Sunil Gupta, Santu Rana and Svetha Venkatesh
Applied Artificial Intelligence Institute
{jmberk, sunil.gupta, santu.rana, svetha.venkatesh}@deakin.edu.au

Abstract

In order to improve the performance of Bayesian optimisation, we develop a modified Gaussian process upper confidence bound (GP-UCB) acquisition function. This is done by sampling the exploration-exploitation trade-off parameter from a distribution. We prove that this allows the expected trade-off parameter to be altered to better suit the problem without compromising a bound on the function’s Bayesian regret. We also provide results showing that our method achieves better performance than GP-UCB in a range of real-world and synthetic problems.

1 Introduction

Global optimisation is a cornerstone of modern scientific innovation. Optimisation of alloys and other materials has allowed us to create massive vehicles that are both strong and light enough to fly. Optimisation in medical science has seen us live longer and healthier lives than previously thought possible. This optimisation usually involves a trial-and-error approach of repeated experiments with different inputs to determine which input produces the most desirable output. Unfortunately, many systems are expensive to sample, and the heuristic methods commonly used to select inputs are not sample-efficient. This can lead to these optimisation experiments being prohibitively costly. As such, methods that can select inputs in a sample-efficient manner can lead to faster and cheaper innovation in a wide range of fields.

Bayesian optimisation is one of the most sample efficient methods for optimising expensive, noisy systems [Brochu et al., 2010; Shahriari et al., 2015]. It has shown excellent performance on a range of practical problems, including problems in biomedical science [Turgeon et al., 2016; Gonzalez et al., 2015], materials science [Li et al., 2017; Kikuchi et al., 2018; Ju et al., 2017], and machine learning [Snoek et al., 2012; Klein et al., 2017; Xia et al., 2017]. It does so by using the data from previous samples to generate a statistical model of the system. This model is then used to suggest the next input through an acquisition function. The design of the acquisition function is non-trivial; and is critically important to the algorithms performance. It must balance selecting points with the goal of improving the statistical model (exploration), and selecting points with the goal of utilising the improved statistical model to find the global optima (exploitation). In addition to this, the costly nature of these problems means that it is desirable to have a theoretical guarantee of the algorithm’s performance.

There are a large range of acquisition functions, all with different balances of exploration and exploitation, but we focus on Gaussian process upper confidence bound (GP-UCB) [Srinivas et al., 2010] in this work. This controls its exploration-exploitation trade-off with a single hyperparameter, \( \beta_t \). It has strong theoretical guarantees on the overall convergence rate, but the bound they give is fairly loose. This causes the value of \( \beta_t \) to be too large, causing significant over-exploration and hence poor practical performance. In practice, the theoretical guarantees need to be weakened by selecting a far smaller \( \beta_t \).

We introduce a novel modification to the GP-UCB acquisition function that significantly improves its exploration-exploitation balance while still having a strong convergence guarantee. This is done by sampling its trade-off parameter with a distribution that allows for a range of exploration factors. However, the distribution is chosen such that convergence is guaranteed to be sub-linear while the sampled \( \beta_t \) is generally smaller than the traditional GP-UCB. This reduction leads to a direct improvement on the convergence efficiency in practice. We demonstrate this improved performance over the standard GP-UCB implementation in a range of both synthetic benchmark functions and real-world applications. In summary, our contributions are:

- The development of a modified acquisition function: RGP-UCB.
- A convergence analysis of Bayesian optimisation using RGP-UCB.
- The demonstration of the performance of our method on a range of synthetic and real-world problems.

2 Background

In this section, we provide a brief overview of Bayesian optimisation, with an emphasis on acquisition functions and regret bounds. For a more in-depth overview of Bayesian
optimisation, we refer readers to [Rasmussen and Williams, 2006], [Shahriari et al., 2015], and [Brochu et al., 2010].

### 2.1 Bayesian Optimisation

Bayesian optimisation is a method for optimising expensive, noisy black-box functions. It represents the system being optimised as an unknown function, $f$. As this is black-box, it is impossible to directly observe. However, we can sample it with an input variable, $x$, to obtain a noisy output, $y = f(x) + \epsilon_{\text{noise}}$, where $\epsilon_{\text{noise}} \sim \mathcal{N}(0, \sigma_{\text{noise}})$ is the random noise corrupting the measurement. Bayesian optimisation seeks to efficiently find the optimal input for such systems over a bounded search space, $\mathcal{X}$:

$$x^* = \arg \max_{x \in \mathcal{X}} f(x) \quad (1)$$

To do this, it creates a statistical model of $f$ using all previously sampled input-output pairs, $D_{t-1} = \{x_i, y_i\}_{i=1}^{t-1}$. This statistical model is usually a Gaussian process, but other models can be used. The statistical model is then used to create an acquisition function, $\alpha(x)$. This is essentially a map of our belief of how useful a given input will be for optimising the system. As such, it can be used to suggest $x_t$; the next input with which to sample the system. This input and its corresponding output can then be added to the previous data, $D_t = D_{t-1} \cup \{x_t, y_t\}$. This process can be iterated, with the Gaussian process improving with each iteration, until a stopping condition has been met. As Bayesian optimisation is generally used on expensive systems, this stopping condition is often a maximum number of experiments, $T$.

### 2.2 Gaussian Process

The Gaussian process is the most common statistical model used in Bayesian optimisation. It models each point in $x \in \mathcal{X}$ as a Gaussian random variable. As such, it is completely characterised by a mean function, $\mu(x)$, and a variance function, $\sigma^2(x)$. However, in order to model sensibly behaved functions there must be correlation between neighbouring points. This is done by conditioning the distribution on the data via a kernel function. There are many viable kernel functions, but one of the simplest and most popular is the squared exponential kernel as it only depends on a single hyperparameter, the lengthscale $l$. This is given by

$$k_{SE}(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2l^2}\right).$$

Using a kernel such as this, a predictive distribution, $f(x) \sim \mathcal{N}(\mu(x), \sigma^2(x))$, can be obtained by conditioning the Gaussian process on the data, $D_t$:

$$\mu_t(x) = k_s(K_t + \sigma^2_{\text{noise}}I)^{-1}y,$$

$$\sigma^2_t(x) = k_{ss} - k_s(K_t + \sigma^2_{\text{noise}}I)^{-1}k_s^T \quad (2)$$

where $K_{t(i,j)} = k(x_i, x_j)$ is the kernel matrix, $k_{ss} = k_t(x_t, x_t)$, and $k_s = [k(x_1, x), k(x_2, x), \ldots, k(x_t, x)]$. Here $I$ is the identity matrix with the same dimensions as $K_t$, and $\sigma_{\text{noise}}$ is the output noise standard deviation.

### 2.3 Acquisition Functions

Once the Gaussian process has been generated, it must then be used to create an acquisition function. This is chosen such that its global maxima in $\mathcal{X}$ will be the best next point to sample:

$$x_t = \arg \max_{x \in \mathcal{X}} \alpha_{t-1}(x) \quad (3)$$

However, the design of such a function is non-trivial. It must first suggest points spread over the search space to improve the Gaussian process. This is called exploration. Once the Gaussian process has been improved enough, it must then transition to suggesting points in regions that have a high probability of containing the global optima. This is called exploitation. If an acquisition function does not explore enough, it may get stuck exploiting sub-optimal regions and never find the global optima. However, if it explores too much, it may waste costly evaluations improving an already adequate Gaussian process. This makes balancing exploration and exploitation vital. There is a wide range of common acquisition functions, all of which have different balances of exploration and exploitation. These include entropy search (ES) by [Hennig and Schuler, 2012], predictive entropy search (PES) by [Hernández-Lobato et al., 2014], knowledge gradient (KG) by [Scott et al., 2011], and others. However, we will only consider the Gaussian process upper confidence bound (GP-UCB) by [Srinivas et al., 2010], Thompson sampling by [Russo and Van Roy, 2014], max-value entropy search (MES) by [Wang and Jegelka, 2017], and expected improvement (EI) by [Jones et al., 1998], with the latter three only being used as baselines.

#### GP-UCB

GP-UCB [Srinivas et al., 2010] is one of the most intuitive acquisition functions. It balances exploration and exploitation through a single hyperparameter, $\beta_t$:

$$\alpha_{t-1}^{GP-UCB}(x) = \mu_t(x) + \sqrt{\beta_t \sigma_t(x)} \quad (4)$$

Increasing $\beta_t$ makes the acquisition function favour points with high variance, causing more exploration. Decreasing $\beta_t$ will make the acquisition function favour points with high mean, causing more exploitation. However, the selection of $\beta_t$ is not done to optimally balance exploitation and exploration, but is done such that the cumulative regret is bounded. It has been proved that, assuming the chosen kernel satisfies

$$P \left\{ \sup_{x \in \mathcal{X}} |\partial f/\partial x_i| > L \right\} \leq ae^{-(L/b)^2}, \quad i = 1 \ldots t \quad (5)$$

for some constants $a, b > 0$, then with probability $1 - \delta$, the algorithm will have sub-linear regret if

$$\beta_t = 2 \log(t^2 \pi^2/(3\delta)) + 2d \log \left( t^2 \log \log(4da/\delta) \right) \quad (6)$$

While the regret bound provided by this choice of $\beta_t$ is desirable, it unfortunately is far larger than needed. This leads to sub-optimal real world performance due to over-exploration. In their own paper, the authors divided the suggested $\beta_t$ by a factor of 5 to achieve better performance [Srinivas et al., 2010].

### 3 Proposed Method

We describe our improved GP-UCB acquisition function, randomised Gaussian process upper confidence bound (RGP-UCB), and prove that it has a sub-linear regret bound.
Bayesian Optimisation with RGP-UCB

\textbf{Algorithm 1} Bayesian Optimisation with RGP-UCB

\textbf{Input:} \( D_{t_0} = \{x_i, y_i\}_{i=1}^{t_0}, \) \( \Gamma \) scale parameter, \( \theta, \) \# Iterations \( T, \) Kernel lengthscale \( l \)

1: for \( t = t_0 \) to \( T \) do
2: \hspace{1em} Build a Gaussian Process (GP) with \( D_t. \)
3: \hspace{1em} Set \( \kappa_t = \frac{\log(\frac{1}{\sigma_t(\theta)\sqrt{2\pi}})}{\log(1+\theta/2)} \) and draw \( \beta_t \) from \( \Gamma(\kappa_t, \theta) \)
4: \hspace{1em} Optimise the acquisition function, \( \alpha_t(x) = \mu_t(x) + \sqrt{2\theta}(x) \) to obtain \( x_{t+1} \) and use it to sample \( y_{t+1} = f(x_{t+1}) + \epsilon_t \) from the system.
5: \hspace{1em} Augment the data, \( D_{t+1} = D_t \cup \{x_{t+1}, y_{t+1}\}. \)
6: end for
7: \hspace{1em} return \( x^* = x_t \) such that \( y_t = \max_{t \in [1,T]} y_t. \)

3.1 RGP-UCB

While the standard GP-UCB method has a desirable regret bound, it has relatively poor performance. This is due to the \( \beta_t \) used to satisfy this bound being far too large, forcing significant over-exploration. As such, a method for selecting a smaller \( \beta_t \) while maintaining a regret bound is desirable. We show that this can be done by sampling \( \beta_t \) from a distribution, and as such, we call our method \textit{randomised Gaussian process upper confidence bound} (RGP-UCB). Doing so means that we bound the Bayesian regret instead of the regret, but it allows for far greater freedom in selecting \( \beta_t, \) letting it be set far smaller while still maintaining convergence guarantees. However, we require \( \beta_t > 0 \) since a negative \( \beta_t \) will punish exploration. We also do not want our distribution to suggest a very large \( \beta_t \) as that will cause over-exploration. As such, we draw \( \beta_t \) from a \( \Gamma(\kappa_t, \theta) \) distribution. Examples of this distribution can be seen in \textbf{Figure 1} and the complete algorithm shown in \textbf{Algorithm 1}. Much like with standard GP-UCB, the parameters of this distribution are chosen to satisfy a regret bound, as per \textbf{Theorem 3}. However, we show that we only need to set one of the two distribution parameters for the bound to hold. Unlike the standard GP-UCB, this allows us to tune \( \beta_t \) to substantially improve our methods performance without compromising its theoretical guarantees.

3.2 Theoretical Analysis

Bayesian optimisation is commonly used in high value problems. As such, theoretical measures of its convergence are desirable. The cumulative regret is one such measure, and is the cornerstone of GP-UCB \cite{Srinivas2010}. Regret is simply the difference between the current sampled value and the global optima. The cumulative regret is the sum of the regret over all iterations:

\[ R_T = \sum_{t=1}^{T} [f(x^*) - f(x_t)] \] (7)

As RGP-UCB is probabilistic, we instead need to use the \textit{Bayesian regret} by Russo et al \cite{Russo2014}:

\[ BR_T = \sum_{t=1}^{T} \mathbb{E}[f(x^*) - f(x_t)] \] (8)

However, their proof was for Gaussian processes with a finite search space, i.e. \( |X| < \infty. \) As such, we follow a method similar to \cite{Srinivas2010} and \cite{Kandasamy2017} and introduce a discretisation of our search space into a \( \delta \) grid of equally spaced points, \( \lambda_{\text{dis}}. \) We denote \( [x]_\tau \) as the closest point to \( x \) in \( \lambda_{\text{dis}}. \)

With this, we can begin bounding the Bayesian regret of our algorithm by decomposing it into components that are easier to bound.

\textbf{Lemma 1.} The Bayesian regret of a probabilistic RGP-UCB algorithm, \( \alpha_t \) over \( T \) iterations can be decomposed as

\[
BR_T \leq \sum_{t=1}^{T} \mathbb{E}[\alpha_{t-1}([x_t]_\tau) - f([x_t]_\tau)] + \sum_{t=1}^{T} \mathbb{E}[f([x^*]_\tau) - \alpha_{t-1}([x^*]_\tau)] + \sum_{t=1}^{T} \mathbb{E}[f([x_t]_\tau) - f([x^*]_\tau)] + \sum_{t=1}^{T} \mathbb{E}[f([x_t]_\tau) - f(x_t)]
\]

(9)

\textit{Proof.} This simply follows from the fact that, as \( x_t = \arg\max_{x \in \lambda_{\text{dis}}} \alpha_t(x), \) we have that \( \alpha(x_t) \geq \alpha(x^*). \)

With this decomposition, we simply need to find a bound for each term to bound the Bayesian regret. We will start with the second term.

\textbf{Theorem 1.} Assuming that \( \beta_t \) is drawn from a \( \Gamma(\kappa_t, \theta) \) distribution with \( \kappa_t = \frac{\log(\frac{1}{\sigma_t(\theta)\sqrt{2\pi}})}{\log(1+\theta/2)} \), the following bound holds

\[
R_2 = \sum_{t=1}^{T} \mathbb{E}[f([x^*]_\tau) - \alpha_{t-1}([x^*]_\tau)] \leq \sum_{t=1}^{T} \frac{1}{t^2 + 1}
\]

(10)
As we want this to decay at a sub-linear rate, we need the following bound holds

\[ E_{t-1} \left( \frac{1}{(1 + \theta_{t-1}/2)^{\kappa_{t-1}}} \right) \leq \frac{\kappa_{t-1}}{\log (1 + \theta_{t-1}/2)} \]  

(13)

Setting \( \kappa_{t-1} = \frac{\log (t^2 + 1)}{\log (1 + \theta_{t-1}/2)} \) to satisfy this, equation \( R_2 \) is bounded by the following:

\[
R_2 = \sum_{t=1}^{T} E \left[ f(x^*) - \alpha_{t-1}(x^*) \right] \\
\leq \sum_{t = 1}^{T} \sum_{x \in \mathcal{X}} \mathbb{E}_{\beta_t} \left[ \mathbb{I} \left\{ f(x) - \alpha_{t-1}(x) \geq 0 \right\} \right] \\
\leq \sum_{t = 1}^{T} \frac{1}{t^2 + 1} 
\]

(14)

Next, we attempt to bound the first component.

**Theorem 2.** Assuming that \( \beta_t \) is drawn from a \( \Gamma(k, \theta) \) distribution, the following bound holds:

\[
R_1 = \sum_{t=1}^{T} E \left[ \alpha_{t-1}(x_t^*) - f(x_t) \right] \\
\leq \sqrt{\frac{k - 1}{F-1 (1 - \frac{1}{T})}} \gamma + F^{-1} \left( 1 - \frac{1}{T} \right) \\
\times \sqrt{T \sum_{t=1}^{T} \sigma_{t-1}(x_t)} 
\]

(15)

\[
\text{Proof.} \quad \text{Using Jensen’s Inequality we have}
\]

\[
R_1 = \sum_{t=1}^{T} E \left[ \alpha_{t-1}(x_t^*) - f(x_t) \right] \\
= \sum_{t=1}^{T} \mathbb{E}_{\beta_t} \left[ \mathbb{I} \left\{ f(x) - \alpha_{t-1}(x) \geq 0 \right\} \right] \\
= \sum_{t=1}^{T} \mathbb{E}_{\beta_t} \left[ \sqrt{\beta_t} \sigma_{t-1}(x_t) \right] \\
\leq \sum_{t=1}^{T} \alpha_{t-1}(x_t) \sqrt{\mathbb{E}_{\beta_t} [\beta_t]} 
\]

(16)

We can then use the Cauchy-Schwartz inequality to get

\[
R_1 \leq \sqrt{\frac{T \mathbb{E}_{\beta_t} \left[ \max_{1 \leq t \leq T} \beta_t \right]}{\sum_{t=1}^{T} \sigma_{t-1}(x_t)}} 
\]

(17)

As \( \beta_t \) is a gamma distribution with shape parameter \( \kappa_t \) and scale parameter \( \theta_t \), its maximum is given by

\[
E \left[ \max_{1 \leq t \leq T} \beta_t \right] \approx \left( 1 + \frac{\kappa_t - 1}{F^{-1} (1 - \frac{1}{T})} \right) \gamma + F^{-1} \left( 1 - \frac{1}{T} \right) 
\]

(18)

where \( \gamma \) is the Euler-Mascheroni constant and \( F^{-1}(x) \) is the inverse CDF of \( \beta_t \). This finally gives us the following bound:

\[
R_i \leq \sqrt{\frac{1 + \frac{k - 1}{F^{-1} (1 - \frac{1}{T})}}{\sum_{t=1}^{T} \sigma_{t-1}(x_t)}} \gamma + F^{-1} \left( 1 - \frac{1}{T} \right) 
\]

(19)

Finally, we need to bound components \( R_3 \) and \( R_4 \). For these, we can use lemma 10 from [Kandasamy et al., 2017].

**Lemma 2.** At step \( t \), \( \forall x \in \mathcal{X} \), \( E[|f(x) - f(x_t)|] \leq \frac{1}{2\pi^2} \).

This means that we have that

\[
R_3 + R_4 \leq 2 \sum_{t=1}^{T} \frac{1}{2\pi^2} \leq \frac{\pi^2}{6} 
\]

(20)

With this we can finally find our Bayesian regret bound.

**Theorem 3.** If \( \beta_t \) is sampled from a \( \Gamma(\kappa_t, \theta) \) distribution with

\[
\kappa_t = \frac{\log (\frac{1}{\sqrt{2\pi}} (t^2 + 1))}{\log (1 + \theta/2)} 
\]

(21)
then the RGP-UCB acquisition function has its Bayesian regret bounded by

\[
BR_T \leq \sqrt{\left[ 1 + \frac{\kappa_t - 1}{F^{-1}(1 - \frac{1}{T})} \right] \gamma + F^{-1}\left( 1 - \frac{1}{T} \right)} 
\times \sqrt{T \sum_{t=1}^{T} \sigma_{t-1}^2(x_t) + \sum_{t=1}^{T} \frac{1}{t^2 + 1} + \frac{\pi^2}{6}} \quad (22)
\]

where \( \gamma \) is the Euler-Mascheroni constant and \( F^{-1}(x) \) is the inverse CDF of \( \beta_t \).

Proof. The result follows simply by combining the bounds for the various components. \( \square \)

4 Results

In this section we present results that demonstrate the performance of RGP-UCB in comparison to other common acquisition functions. We also demonstrate the impact of varying the \( \theta \) parameter of the gamma distribution used to sample \( \beta_t \). The Python code used for this paper can be found at https://github.com/jmaberk/RGPUCB.

4.1 Experimental Setup

We test our method against a selection of common acquisition functions on a range of Bayesian optimisation problems. These include a range of synthetic benchmark functions and real-world optimisation problems. These are all transformed into continuous maximisation problems for consistency. In each case, the experiment was run for 40d iterations and repeated 10 times with 3d + 1 different initial points. The initial points are chosen randomly with a Latin hypercube sample scheme [Jones, 2001]. The methods being tested are:

- Our randomised Gaussian process upper confidence bound with \( \theta = 8 \) (RGP-UCB \( \theta = 8 \)).
- Our randomised Gaussian process upper confidence bound with \( \theta = 1 \) (RGP-UCB \( \theta = 1 \)).
- Our randomised Gaussian process upper confidence bound with \( \theta = 0.5 \) (RGP-UCB \( \theta = 0.5 \)).
- Standard Gaussian process upper confidence bound (GP-UCB) [Srinivas et al., 2010].
- Expected improvement (EI) [Jones et al., 1998].
- Thompson sampling [Russo and Van Roy, 2014].
- Max-value entropy search (MES) [Wang and Jegelka, 2017].

Note that we turn all functions that are traditionally minimised into maximisation problems by taking their negative for consistency. As such, higher results are always better.

4.2 Selection of the Trade-off Parameter

An advantage of our method is that it can change its exploration-exploitation balance without compromising its convergence guarantee. This is done by changing the \( \theta \) parameter in the \( \beta_t \sim \Gamma(\kappa_t, \theta) \) distribution. Increasing \( \theta \) will increase the expected \( \beta_t \), increasing exploration.

As different problems favour different exploration-exploitation balances, we tested a range of \( \theta \) values on a range of different problems. In Figure 2, we show the performance of a range of \( \theta \) values on an exploitation favouring problem, the Alpine 2 (5D) function, and an exploration favouring problem, the Dropwave (2D) function.

It was found that \( \theta = 8 \) gives good performance on both the above exploration-favouring problem and other similar problems tested. Likewise, \( \theta = 0.5 \) is a good choice for exploitation favouring problems. We also note that \( \theta = 1 \) has decent performance on both problems, making it a good choice for problems where the required exploration-exploitation balance is completely unknown.

4.3 Synthetic Benchmark Functions

The first demonstration of our methods performance is the optimisation of several common synthetic benchmark functions. These are the Dropwave (2D), Sphere (4D), Alpine 2 (5D), and Ackley (5D) functions. Results for these are shown in Figure 3.

Here we can see that RGP-UCB has competitive performance in all of the above cases. In general, it does significantly better than the standard GP-UCB and the Thompson sampling acquisition functions. EI has better early performance in many cases, as it starts exploiting earlier. However, RGP-UCB tends to have better exploration and therefore often able to beat it in the long-term. MES, on the other hand, has poor early performance in general but catches up to other methods later. However, explorative RGP-UCB (\( \theta = 8 \)) beat it in the exploration favouring dropwave function. In the other synthetic functions, both the balanced (\( \theta = 1 \)) and exploitative (\( \theta = 0.5 \)) RGP-UCB achieved superior results.

The leftmost functions were chosen to be pathological cases which disproportionately favours exploration (Dropwave 2D) and exploitation (Sphere 4D). These can be seen as best-case examples for GP-UCB and EI respectively. However, RGP-UCB is able to out-perform them even on these if its \( \theta \) parameter is chosen properly, and does so while maintaining its convergence guarantee. This formulation of EI does not have a known regret bound as it follows the standard implementation and hence doesn’t satisfy the assumptions required by the current bounds [Bull, 2011; Ryzhov, 2016; Wang and de Freitas, 2014; Nguyen et al., 2017].

1 All benchmark functions use the recommended parameters from https://www.sfu.ca/~ssurjano/optimization.html
In the middle two plots, RGP-UCB is superior even with the conservative parametrisation of $\theta = 1$.

### 4.4 Machine Learning Hyperparameter Tuning

Our first demonstration of the real-world performance of RGP-UCB is the hyperparameter tuning of a support vector regression (SVR) [Drucker et al., 1997] algorithm. This is the support vector machine classification algorithm extended to work on regression problems, with performance measured in root mean squared error (RMSE). It has three hyperparameters, the threshold, $\epsilon$, the kernel parameter, $\gamma$, and a soft margin parameter, $C$. All experiments are done with the public Space GA scale dataset\textsuperscript{2}. The results are shown in Figure 3.

We can see that, while MES is superior here, the final performance of all three variants of our method exceeds that of standard GP-UCB. The high exploitation and balanced variants are competitive with EI, with the former achieving higher final performance. As with many real-world problems, SVR is known to favour higher exploitation, and is therefore an example of when the user would know to try a smaller $\theta$.

### 4.5 Materials Science Application: Alloy Heat Treatment

The second demonstration of RGP-UCB’s performance is the optimisation of an Aluminium-Scandium alloy heat treatment simulation [Robson et al., 2003]. The goal of the simulation is to optimise the resulting alloys hardness, measured in MPa. The hardening process is controlled through multiple cooking stages, each with two hyperparameters, the duration and a temperature. As we use a two-stage cooking simulation, there is a total of four hyperparameters to optimise through Bayesian optimisation. The results are shown in Figure 3.

Outside of MES, the results are very similar to the previous SVR example, with the high-exploitation method having the best performance and the balance method being competitive with EI.

### 4.6 Conclusion

We have developed a modified UCB based acquisition function that has substantially improved performance while maintaining a sub-linear regret bound. We have proved that this bound holds in terms of Bayesian regret while allowing for some flexibility in the selection of its parameters. We have also demonstrated the impact of said parameters on the performance. Moreover, we have shown that its performance is competitive or greater than existing methods in a range of synthetic and real-world applications. In particular, we show that it out-performs the standard GP-UCB implementation on most problems, even when its hyperparameter is chosen conservatively.

### Acknowledgements

This research was supported by an Australian Government Research Training Program (RTP) Scholarship awarded to JMA Berk, and was partially funded by the Australian Government through the Australian Research Council (ARC). Prof Venkatesh is the recipient of an ARC Australian Laureate Fellowship (FL170100006).

### References

[Brochu et al., 2010] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expen-
sive cost functions, with application to active user model-
ning and hierarchical reinforcement learning. arXiv preprint

[Bull, 2011] Adam D Bull. Convergence rates of effi-
cient global optimization algorithms. Journal of Machine

[Drucker et al., 1997] Harris Drucker, Christopher JC
Burges, Linda Kaufman, Alex J Smola, and Vladimir
Vapnik. Support vector regression machines. In Advances
in neural information processing systems, pages 155–161,
1997.

[Gonzalez et al., 2015] Javier Gonzalez, Joseph Longworth,
David C James, and Neil D Lawrence. Bayesian op-
timization for synthetic gene design. arXiv preprint

[Hennig and Schuler, 2012] Philipp Hennig and Christian J
Schuler. Entropy search for information-efficient global
optimization. Journal of Machine Learning Research,

[Hernández-Lobato et al., 2014] José Miguel Hernández-
Lobato, Matthew W Hoffman, and Zoubin Ghahramani.
Predictive entropy search for efficient global optimization
of black-box functions. In Advances in neural information
processing systems, pages 918–926, 2014.

[Jones et al., 1998] Donald R Jones, Matthias Schonlau,
and William J Welch. Efficient global optimization of expen-
sive black-box functions. Journal of Global optimization,

methods based on response surfaces. Journal of global

[Ju et al., 2017] Shenghong Ju, Takuma Shiga, Lei Feng,
Zhufeng Hou, Koji Tsuda, and Junichiro Shiomi. Designing
nanostuctures for phonon transport via bayesian optimi-

[Kandasamy et al., 2017] Kirthivasan Kandasamy, Akshay
Krishnamurthy, Jeff Schneider, and Barnabás Póczos.
Asynchronous parallel bayesian optimisation via thomp-

[Kikuchi et al., 2018] Shun Kikuchi, Hiromi Oda, Shin Kiy-
ohara, and Teruyasu Mizoguchi. Bayesian optimization for
efficient determination of metal oxide grain boundary
structures. Physica B: Condensed Matter, 532:24–28,
2018.

[Klein et al., 2017] Aaron Klein, Stefan Falkner, Simon Bar-
tels, Philipp Hennig, and Frank Hutter. Fast bayesian op-
timization of machine learning hyperparameters on large
datasets. In International Conference on Artificial Intel-
ligence and Statistics (AISTATS 2017), pages 528–536.
PMLR, 2017.

[Li et al., 2017] Cheng Li, David Rubín de Celis Leal, Santu
Rana, Sunil Gupta, Alessandra Sutti, Stewart Greenhill,
Teo Slezak, Murray Height, and Svetla Venkatesh. Rapid
bayesian optimisation for synthesis of short polymer fiber

[Nguyen et al., 2017] Vu Nguyen, Sunil Gupta, Santu Rana,
Cheng Li, and Svetla Venkatesh. Regret for expected im-
provement over the best-observed value and stopping con-
dition. In Asian Conference on Machine Learning, pages

[Rasmussen and Williams, 2006] Carl Edward Rasmussen
and Christopher KI Williams. Gaussian Processes for Ma-

[Robson et al., 2003] JD Robson, MJ Jones, and
PB Prangnell. Extension of the n-model to predict competing homogeneous and heterogeneous precipitation

[Russo and Van Roy, 2014] Daniel Russo and Benjamin
Van Roy. Learning to optimize via posterior sampling.
Mathematics of Operations Research, 39(4):1221–1243,
2014.

[Ryzhov, 2016] Ilya O Ryzhov. On the convergence rates
of expected improvement methods. Operations Research,

[Scott et al., 2011] Warren Scott, Peter Frazier, and Warren
Powell. The correlated knowledge gradient for simul-
tation optimization of continuous parameters using gaus-
sian process regression. SIAM Journal on Optimization,

[Shahriari et al., 2015] Bobak Shahriari, Kevin Swersky,
Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking
the human out of the loop: A review of bayesian optimiza-

[Snoek et al., 2012] Jasper Snoek, Hugo Larochelle, and
Ryan P Adams. Practical bayesian optimization of ma-
chine learning algorithms. In Advances in neural informa-

[Srinivas et al., 2010] Niranjan Srinivas, Andreas Krause,
Sham M Kakade, and Matthias Seeger. Gaussian pro-
cess optimization in the bandit setting: no regret and
experimental design. In Proceedings of the 27th Inter-
national Conference on Machine Learning, pages 1015–
1022, 2010.

[Turgeon et al., 2016] Martine Turgeon, Cindy Lustig, and
Warren H Meck. Cognitive aging and time perception:
roles of bayesian optimization and degeneracy. Frontiers
in aging neuroscience, 8:102, 2016.

[Wang and de Freitas, 2014] Ziyu Wang and Nando de Fre-
itas. Theoretical analysis of bayesian optimisation with
unknown gaussian process hyper-parameters. NIPS Work-
shop on Bayesian Optimization, 2014.

Max-value entropy search for efficient bayesian optimiza-
tion. In Proceedings of the 34th International Confer-
mence on Machine Learning-Volume 70, pages 3627–3635.
JMLR. org, 2017.

[Xia et al., 2017] Yufei Xia, Chuanzhe Liu, YuYing Li, and
Nana Liu. A boosted decision tree approach using
bayesian hyper-parameter optimization for credit scoring.