Parallelised Bayesian Optimisation via Thompson Sampling

Kirthevasan Kandasamy
Carnegie Mellon University
kandasamy@cs.cmu.edu

Akshay Krishnamurthy
University of Massachusetts, Amherst
akshay@cs.umass.edu

Jeff Schneider, Barnabás Póczos
Carnegie Mellon University
{schneide, bapoczos}@cs.cmu.edu

Abstract

We design and analyse variations of the classical Thompson sampling (TS) procedure for Bayesian optimisation (BO) in settings where function evaluations are expensive but can be performed in parallel. Our theoretical analysis shows that a direct application of the sequential Thompson sampling algorithm in either synchronous or asynchronous parallel settings yields a surprisingly powerful result: making n evaluations distributed among M workers is essentially equivalent to performing n evaluations in sequence. Further, by modelling the time taken to complete a function evaluation, we show that, under a time constraint, asynchronous parallel TS achieves asymptotically lower regret than both the synchronous and sequential versions. These results are complemented by an experimental analysis, showing that asynchronous TS outperforms a suite of existing parallel BO algorithms in simulations and in an application involving tuning hyper-parameters of a convolutional neural network. In addition to these, the proposed procedure is conceptually much simpler than existing work for parallel BO.

1 Introduction

Many real world problems require maximising an expensive unknown function f from noisy evaluations. As evaluations are typically expensive in such applications, we would like to optimise the function with a minimal number of evaluations. For example, consider the task of tuning the hyper-parameters of a machine learning model, which can be framed as a black-box optimisation problem where an evaluation to f at x trains the model on the hyper-parameters x, and returns the cross validation accuracy $f(x)$ on the validation set. The evaluation of $f(x)$ is noisy due to sources of randomness in the training procedure and is typically expensive, especially with large models. In drug discovery, each x characterises a candidate drug and $f(x)$ measures various qualities such as the potency, specificity, and solubility of the drug via an expensive *in vitro* or *in vivo* test. Bayesian optimisation (BO) refers to a suite of methods for black-box optimisation under Bayesian assumptions on f that has been successfully applied in hyper-parameter tuning, drug discovery and other applications in policy search, online advertising, and scientific experimentation [12, 17, 30, 32, 40].

In this paper, we develop new algorithms for *parallel Bayesian optimisation.* For example, in hyper-parameter tuning, with modern computing infrastructures, we have the ability to evaluate several hundred hyper-parameters in parallel. The training time for each hyper-parameter is influenced by a myriad of factors, including contention on shared compute resources and the actual hyper-parameter choices, so it typically exhibits significant variability. Our goal is to find a set of hyper-parameters that achieve low validation error, in a short amount of time. Similarly, in drug discovery, high throughput screening equipment can test several thousand candidate drugs at the same time.

Addressing this problem in the above and several other applications with parallel function evaluations, we design and analyse new algorithms for parallel Bayesian optimisation. Our algorithms are synchronous and asynchronous parallel versions of Thompson Sampling (TS), which we call synTS and asyTS, respectively. These algorithms are conceptually simple, easy to implement, and also scale to large number of parallel evaluations. In a departure from prior work on parallel BO, we explicitly model evaluation times and study the relationship between optimisation performance and time, in addition to the more standard relationship between optimisation and the number of function evaluations. Our main contributions are:

1. A theoretical analysis demonstrating that both synTS and asyTS making n evaluations distributed among M workers is almost as good as if the n evaluations were made in sequence.
2. We introduce and analyse simple regret with time as a resource in parallel settings. Under this definition, asyTS outperforms the synchronous and sequential versions up to constant factors.
3. Empirically, we demonstrate that TS significantly

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84. Copyright 2018 by the author(s).

outperforms existing methods for parallel BO in both the synchronous and asynchronous settings on several synthetic problems and a hyperparameter tuning task. A python implementation of our algorithm and experiments is available at github.com/kirthevasank/gp-parallel-ts.

Related Work

Bayesian optimisation methods start with a prior belief distribution for f and incorporate function evaluations into updated beliefs in the form of a posterior. Popular algorithms choose points to evaluate f via deterministic query rules such as expected improvement (EI) [21] or upper confidence bounds (UCB) [41]. We however, will focus on a randomised selection procedure known as Thompson sampling [42], which selects a point by maximising a random sample from the posterior. TS has been explored for sequential BO [4, 39] and some recent theoretical advances have characterised the performance of TS in sequential settings [3, 7, 27, 35, 36].

There has been a flurry of recent activity in parallelising BO [8, 9, 11, 13, 19, 26, 38, 43–46]. In comparison to this prior work, our approach enjoys one or more of the following advantages.

1. **Asynchronicity:** The majority of work on parallel BO are in the synchronous (batch) setting. To our knowledge, only [11, 19, 43] can handle asynchronous parallelisation.
2. **Theoretical underpinnings:** Most methods for parallel BO do not come with theoretical guarantees, with the exception of some work using UCB techniques [8, 9, 26]. Crucially, to the best of our knowledge, no theoretical guarantees are available for asynchronous methods.
3. **Conceptual simplicity:** *All* of the above methods either introduce additional hyper-parameters and/or ancillary computational subroutines. Some methods become computationally prohibitive when there are a large number of workers and must resort to approximations [19, 38, 43, 46]. In contrast, our approach is conceptually simple – a direct adaptation of the sequential TS algorithm to the parallel setting. Hence, it is robust in practice, especially with a large number of workers. Further, unlike existing methods, its computationally complexity does not increase with M and is exactly the same as the sequential version.

We mention that parallelised versions of TS have been explored to varying degrees in some applied domains of bandit and reinforcement learning research [15, 18, 31]. However, to our knowledge, we are the first to theoretically analyse parallel TS. More importantly, we are also the first to develop and study TS in an asynchronous parallel setting. Besides BO, there has been a line of work on online learning with delayed feedback (as we have in the parallel setting) [22, 33]. In addition, Jun et al. [23] study a best-arm identification problem when queries are issued in batches. But these papers only consider finite decision sets and do not model evaluation times to study trade-offs when time is viewed as the primary resource.

2 Preliminaries

We wish to maximise an unknown function $f : \mathcal{X} \to \mathbb{R}$ defined on a compact domain $\mathcal{X} \subset \mathbb{R}^d$, by repeatedly obtaining noisy evaluations of f; when we evaluate f at $x \in \mathcal{X}$, we observe $y = f(x)+\epsilon$ where the noise ϵ satisfies $\mathbb{E}[\epsilon] = 0$. We work in the Bayesian paradigm, modeling f itself as a random quantity. Following the plurality of Bayesian optimisation literature, we assume that f is a sample from a Gaussian process [34] and that the noise, $\epsilon \sim \mathcal{N}(0, \eta^2)$, is i.i.d normal. A Gaussian process (GP) is characterised by a mean function $\mu : \mathcal{X} \to \mathbb{R}$ and prior (covariance) kernel $\kappa : \mathcal{X}^2 \to \mathbb{R}$. If $f \sim \mathcal{GP}(\mu, \kappa)$, then $f(x)$ is distributed normally as $\mathcal{N}(\mu(x), \kappa(x, x))$ for all $x \in \mathcal{X}$. Additionally, given n observations $A = \{(x_i, y_i)\}_{i=1}^n$ from this GP, where $x_i \in \mathcal{X}$, $y_i = f(x_i) + \epsilon_i \in \mathbb{R}$, the posterior for f is also a GP with mean μ_A and covariance κ_A given by,

$$\begin{aligned} \mu_A(x) &= k^\top (K + \eta^2 I_n)^{-1} Y, \\ \kappa_A(x, \tilde{x}) &= \kappa(x, \tilde{x}) - k^\top (K + \eta^2 I_n)^{-1} \tilde{k}. \end{aligned} \qquad (1)$$

Here $Y \in \mathbb{R}^n$ such that $Y_i = y_i$, and $k, \tilde{k} \in \mathbb{R}^n$ are such that $k_i = \kappa(x, x_i), \tilde{k}_i = \kappa(\tilde{x}, x_i)$. The Gram matrix $K \in \mathbb{R}^{n\times n}$ is given by $K_{i,j} = \kappa(x_i, x_j)$, and $I_n \in \mathbb{R}^{n\times n}$ is the identity matrix. Some common choices for the kernel are the squared exponential (SE) kernel and the Matérn kernel. We refer the reader to Rasmussen and Williams [34] for more background on GPs.

Our goal is to find the maximiser $x_\star = \mathrm{argmax}_{x\in\mathcal{X}} f(x)$ of f through repeated evaluations. In the BO literature, this is typically framed as minimising the *simple regret*, which is the difference between the optimal value $f(x_\star)$ and the best evaluation of the algorithm. Since f is a random quantity, so is its optimal value and hence the simple regret. This motivates studying the *Bayes simple regret*, which is the expectation of the simple regret. Formally, we define the simple regret, SR(n), and Bayes simple regret, BSR(n), of an algorithm after n evaluations as,

$$\begin{aligned} \mathrm{SR}(n) &= f(x_\star) - \max_{j=1,\dots,n} f(x_j), \\ \mathrm{BSR}(n) &= \mathbb{E}[\mathrm{SR}(n)]. \end{aligned} \qquad (2)$$

The expectation in BSR(n) is with respect to the prior $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$, the noise in the observations $\epsilon_j \sim \mathcal{N}(0, \eta^2)$, and any randomness of the algorithm. We focus on simple regret here mostly to simplify exposition; our proof also applies for *cumulative regret*, which may be more familiar.

In many applications of BO, including hyperparameter tuning, the time required to evaluate the function is the dominant cost, and we are most interested in maximising f in

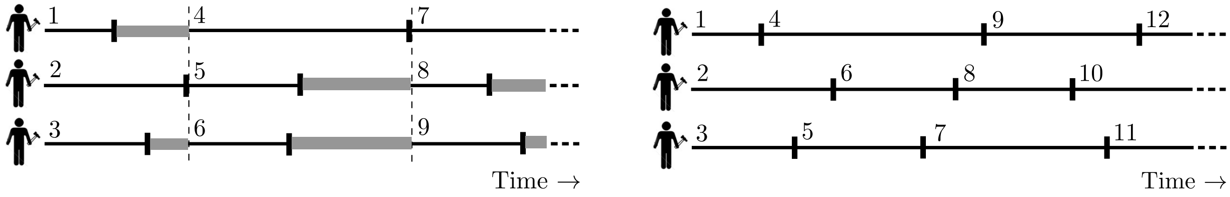


Figure 1: An illustration of the synchronous (left) and asynchronous (right) settings using $M = 3$ workers. The short vertical lines indicate when a worker finished its last evaluation. In the synchronous setting the grey shaded regions indicate idle time after a worker finishes its job. The horizontal location of a number indicates when the worker started its next evaluation while the number itself denotes the order in which the evaluation was dispatched by the algorithm.

a short period of time. Moreover, there is often considerable variability in the time required for different evaluations, caused by inherent differences between points in the domain, randomness of the environment, or other factors. For example, in the hyperparameter tuning application, unpredictable factors such as resource contention, initialisation, etc., may induce significant variability in evaluation times.

To adequately capture these settings, we model the time to complete an evaluation as a random variable, and measure performance in terms of the simple regret within a time budget, T. Specifically, letting $N = N(T)$ denote the (random) number of evaluations performed by an algorithm within time T, we define the simple regret $\mathrm{SR}'(T)$ and the Bayes simple regret $\mathrm{BSR}'(T)$ as

$$\mathrm{SR}'(T) = \begin{cases} f(x_\star) - \max_{j\leq N} f(x_j) & \text{if } N \geq 1 \\ \max_{x\in\mathcal{X}} |f(x_\star) - f(x)| & \text{otherwise} \end{cases},$$
$$\mathrm{BSR}'(T) = \mathbb{E}[\mathrm{SR}'(T)]. \tag{3}$$

This definition is similar to (2), except, when an algorithm has not completed an evaluation yet, its simple regret is the worst possible value. In $\mathrm{BSR}'(T)$, the expectation now also includes the randomness in the evaluation times in addition to the three sources of randomness in $\mathrm{BSR}(n)$. In this work, we will model evaluation times as random variables independent from f; specifically we consider uniform, half-normal, or exponential random variables. While the model does not precisely capture all aspects of evaluation times observed in practice, we prefer it because (a) it is fairly general, (b) it leads to a clean algorithm and analysis, and (c) the resulting algorithm has good performance on real applications, as we demonstrate in Section 4. Studying other models for the evaluation time is an intriguing question for future work and is discussed further in Section 5.

To our knowledge, all prior theoretical work for parallel BO [8, 9, 26], measures regret in terms of the total number of evaluations, i.e. $\mathrm{SR}(n), \mathrm{BSR}(n)$. However, explicitly modeling evaluation times and treating time as the main resource in the definition of regret is a better fit for applications and leads to new conclusions in the parallel setting.

Parallel BO: We are interested in parallel approaches for BO, where the algorithm has access to M workers that can evaluate f at different points in parallel. In this setup, we wish to differentiate between the synchronous and asynchronous settings, illustrated in Fig. 1. In the former, the algorithm issues a batch of M queries simultaneously, one per worker, and waits for all M evaluations to be completed before issuing the next batch. In contrast, in the asynchronous setting, a new evaluation may be issued as soon as a worker finishes its last job and becomes available. In the parallel setting, N in (3) will refer to the number of evaluations completed by *all* M workers.

One of our goals in the theoretical analysis will be to quantify the trade-offs between information accumulation and worker utilisation in the sequential, synchronous parallel and asynchronous parallel settings. When comparing the three settings purely in terms of the number of evaluations, i.e. $\mathrm{BSR}(n)$, the parallel settings are naturally at a disadvantage: the sequential algorithm makes use of feedback from all its previous evaluations when issuing a query, whereas a parallel algorithm could be missing up to $M-1$ of them. As we will see however, for our TS algorithms, this difference is fairly small - the bounds for the parallel algorithms are only slightly worse than for sequential variants. The advantage in the parallel setting however, is that we will be able to complete more evaluations than a sequential version within an allotted time. One can make a similar argument to compare the synchronous and asynchronous settings. When issuing queries, a synchronous algorithm has more information about f, since all previous evaluations complete before a batch is selected, whereas asynchronous algorithms always issue queries with $M-1$ missing evaluations. For example, in Fig. 1, when dispatching the fourth job, the synchronous version uses results from the first three evaluations whereas the asynchronous version uses just the result of the first evaluation. However, in the synchronous setting, workers may sit idle for some time waiting for the other workers to finish. Foreshadowing our results in Theorem 5, when there is significant variability in evaluation times, worker utilisation is more important than information accumulation, and hence the asynchronous setting will enable better bounds on $\mathrm{BSR}'(T)$. Next, we present our algorithms.

3 Thompson Sampling for Parallel BO

A review of sequential TS: Thompson sampling [42] is a randomised strategy for sequential decision making under uncertainty. At step j, TS samples x_j according to the posterior probability that it is the optimum. That is, x_j is drawn from the posterior density $p_{x_\star}(\cdot|\mathcal{D}_j)$ where $\mathcal{D}_j = \{(x_i, y_i)\}_{i=1}^{j-1}$ is the history of query-observation pairs up to step j. For GPs, this allows for a very simple and elegant algorithm. Observe that we can write $p_{x_\star}(x|\mathcal{D}_j) = \int p_{x_\star}(x|g)\, p(g|\mathcal{D}_j)\mathrm{d}g$, and that $p_{x_\star}(\cdot|g)$ puts all its mass at the maximiser $\mathrm{argmax}_x\, g(x)$ of g. Therefore, at step j, we draw a sample g from the posterior for f conditioned on $\mathcal{D}_j$ and set $x_j = \mathrm{argmax}_x\, g(x)$ to be the maximiser of g. We then evaluate f at x_j. The resulting procedure, called seqTS, is displayed in Algorithm 1.

```
Algorithm 1: seqTS                                   Thompson [42]
Require: Prior GP  $\mathcal{GP}(\mathbf{0}, \kappa)$.
1: $\mathcal{D}_1 \leftarrow \varnothing$,   $\mathcal{GP}_1 \leftarrow \mathcal{GP}(\mathbf{0}, \kappa)$.
2: for $j = 1, 2, \ldots$ do
3:     Sample $g \sim \mathcal{GP}_j$.
4:     $x_j \leftarrow \mathrm{argmax}_{x\in\mathcal{X}}\, g(x)$.
5:     $y_j \leftarrow$ Query $f$ at $x_j$.
6:     $\mathcal{D}_{j+1} \leftarrow \mathcal{D}_j \cup \{(x_j, y_j)\}$.
7:     Compute posterior $\mathcal{GP}_{j+1} = \mathcal{GP}(\mu_{\mathcal{D}_{j+1}}, \kappa_{\mathcal{D}_{j+1}})$
       conditioned on $\mathcal{D}_{j+1}$. See (1).
8: end for
```

Asynchronous Parallel TS: For the asynchronously parallel setting, we propose a natural adaptation of the above algorithm. Precisely, when a worker finishes an evaluation, we update the posterior with the query-feedback pair, sample g from the posterior, and re-deploy the worker with an evaluation at $x_j = \mathrm{argmax}_x\, g(x)$. The procedure, called asyTS, is displayed in Algorithm 2. In the first M steps, when at least one of the workers have not been assigned a job yet, the algorithm skips lines 3–5 and samples g from the prior GP, $\mathcal{GP}_1$, in line 6.

```
Algorithm 2: asyTS
Require: Prior GP  $\mathcal{GP}(\mathbf{0}, \kappa)$.
1: $\mathcal{D}_1 \leftarrow \varnothing$,   $\mathcal{GP}_1 \leftarrow \mathcal{GP}(\mathbf{0}, \kappa)$.
2: for $j = 1, 2, \ldots$ do
3:     Wait for a worker to finish.
4:     $\mathcal{D}_j \leftarrow \mathcal{D}_{j-1} \cup \{(x', y')\}$ where $(x', y')$ are the
       worker's previous query/observation.
5:     Compute posterior $\mathcal{GP}_j = \mathcal{GP}(\mu_{\mathcal{D}_j}, \kappa_{\mathcal{D}_j})$.
6:     Sample $g \sim \mathcal{GP}_j$,   $x_j \leftarrow \mathrm{argmax}\, g(x)$.
7:     Re-deploy worker to evaluate $f$ at $x_j$.
8: end for
```

Synchronous Parallel TS: To illustrate comparisons, we also introduce a synchronous parallel version, synTS, which makes the following changes to Algorithm 2. In line 3 we wait for all M workers to finish and compute the GP posterior with all M evaluations in lines 4–5. In line 6 we draw M samples and re-deploy all workers with evaluations at their maxima in line 7.

We wish to highlight the main methodological differences of our algorithms with prior work for parallel BO. Since existing methods select points using deterministic criteria such as UCB or EI, they need to *explicitly* enforce diversity of query points so as to prevent the algorithm from picking the same or similar points for all M workers. Consequently, such methods introduce additional hyperparameters and/or potentially expensive computational routines. In contrast, asyTS and synTS are essentially the same as their sequential counterpart and their computational complexity does not increase with M. In addition to this computational advantage, this conceptual simplicity results in robust empirical performance in practice. Our theoretical analysis shows that a straightforward application of TS works because its inherent randomness is sufficient to avoid redundant function evaluations when managing M workers in parallel. This phenomenon is confirmed by our experiments in Section 4, where we see that explicitly encouraging diversity does not improve the performance of asyTS. We demonstrate this empirically by constructing a variant asyHTS of asyTS which employs one such diversity scheme found in the literature. asyHTS performs either about the same as or slightly worse than asyTS in the many experiments we study in Section 4.

3.1 Theoretical Results

We now present our theoretical contributions. We analyse the performance of parallelised TS both with the number of evaluations n and the time budget T as the resource. In particular, we study how these rates change with the number of workers M and demonstrate that as M increases, while BSR(n) worsens slightly for the parallel settings when compared to the sequential setting, BSR$'(T)$ can improve dramatically. We provide theorem statements here to convey key intuitions, with all formal statements and proofs deferred to Appendices A and B. We use $\asymp, \lesssim$ to denote equality/inequality up to constant factors that are common across all theorem statements.

Maximum Information Gain (MIG): As in prior work, our regret bounds involve the MIG [41], which captures the statistical difficulty of the BO problem. It quantifies the maximum information a set of n observations provide about f. To define the MIG, and for subsequent convenience, we introduce one notational convention. For a finite subset $A \subset \mathcal{X}$, we use $y_A = \{(x, f(x) + \epsilon) \mid x \in A\}$ to denote the query-observation pairs corresponding to the set A. The MIG is then defined as $\Psi_n = \max_{A\subset\mathcal{X}, |A|=n} I(f; y_A)$ where I denotes the Shannon Mutual Information. Srinivas et al. [41] show that Ψ_n is sublinear in n for different classes of kernels; e.g. for the SE kernel, $\Psi_n \propto \log(n)^{d+1}$ and for the Matérn kernel with smoothness parameter ν, $\Psi_n \propto n^{1-\frac{\nu}{2\nu+d(d+1)}}$.

Our first goal is to compare the simple regret $\mathrm{BSR}(n)$ after n evaluations for synTS and asyTS with that of seqTS. To this end, we prove the following theorem for seqTS which is a straightforward extension of a result in [35].

Theorem 1 (Informal. $\mathrm{BSR}(n)$ for seqTS)**.** *Let $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$. Then, for* seqTS*,* $\mathrm{BSR}(n) \lesssim \sqrt{\Psi_n \log(n)/n}$*.*

The first theoretical result of this paper, presented below in Theorem 2, bounds $\mathrm{BSR}(n)$ for synTS.

Theorem 2 (Informal. $\mathrm{BSR}(n)$ for synTS)**.** *Let $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$. Then, for* synTS*,*

$$\mathrm{BSR}(n) \lesssim \frac{M\sqrt{\log(M)}}{n} + \sqrt{\frac{\Psi_n \log(n+M)}{n}}.$$

A comparison of the bounds in Theorems 1 and 2 reveals that, for reasons explained before in Section 2, seqTS outperforms synTS purely in terms of the number of evaluations n. However, for large n, the first term in the bound for synTS vanishes faster than the latter, and the dependence on M in the latter term is insignificant when $n \gg M$. Hence, the difference between the sequential and synchronous parallel algorithms is small and negligible for large n. We also note that the leading constant for the second term is the same as that in Theorem 1. This implies a powerful conclusion: synTS with M parallel workers is almost as good as the sequential version with as many evaluations.

To present the results for the asynchronous setting, we introduce the following quantity ξ_M, which bounds the information we can gain about f from the evaluations in progress. Assume that we have completed n evaluations to f at the points in $\mathcal{D}_n$ and that there are q evaluations in process at points in A_q. That is $\mathcal{D}_n, A_q \subset \mathcal{X}$, $|\mathcal{D}_n| = n$ and $|A_q| = q < M$. Then $\xi_M > 0$ satisfies the following for all $n \geq 1$,

$$\max_{A_q \subset \mathcal{X}, |A_q| < M} I(f; y_{A_q} | y_{\mathcal{D}_n}) \;\leq\; \frac{1}{2}\log(\xi_M). \qquad (4)$$

Our next result is a bound on $\mathrm{BSR}(n)$ for asyTS.

Theorem 3 (Informal. $\mathrm{BSR}(n)$ for asyTS)**.** *Let $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$. Then, for* asyTS*,*

$$\mathrm{BSR}(n) \lesssim \sqrt{\frac{\xi_M \Psi_n \log(n)}{n}}.$$

Unfortunately, this bound depends on ξ_M which can be quite large under general conditions. However, ξ_M is a well studied quantity in the GP literature; precisely, Desautels et al. [9], Krause et al. [28] show that ξ_M can be bounded by a kernel dependent constant C_κ by initially querying f using an uncertainty sampling procedure for γ_M samples. This sampling procedure, which iteratively samples the points with the largest variance in the GP, is asynchronously parallelisable. Desautels et al. [9] shows that for the Matérn kernel, with $\gamma_M \asymp \mathrm{poly}(M)$, this procedure guarantees $C_\kappa \leq e^e$, and for the SE kernel, with $\gamma_M \asymp M\mathrm{polylog}(M)$, we can achieve any constant $C_\kappa > 1$, depending on the order of the polylog term. These values for C_κ, γ_M are not absolute – by picking a larger γ_M we can achieve smaller C_κ. In all cases however, γ_M is at most polynomial in M and does not depend on n. We provide more details on the initialisation scheme in Appendix A.4. By initialising asyTS with this sampling scheme, we obtain the bound below.

Corollary 4 (Informal. $\mathrm{BSR}(n)$ for asyTS after initialisation)**.** *Let $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$. By first initialising* asyTS *with an uncertainty sampling scheme [9, 28], we have,*

$$\mathrm{BSR}(n) \lesssim \frac{\gamma_M}{n} + \sqrt{\frac{C_\kappa \Psi_n \log(n)}{n}}.$$

Despite the dependence on the initialisation scheme and the constant term C_κ, Corollary 4 is encouraging: since the first term in the bound vanishes faster than the latter, up to constant factors, asyTS with M parallel workers is almost as good as seqTS.

That said, we believe that bounds similar to Theorem 2 should be obtainable for asyTS without the additional constant C_κ and without the initialisation scheme. For instance, asyTS performs very well in our experiments even though we do not use this initialisation scheme. We leave it to future work to resolve this gap.

Now that we have bounds on the regret as a function of the number of evaluations, we can turn to bounding $\mathrm{BSR}'(T)$, the simple regret with time as the main resource. For this, we consider three different random distribution models for the time to complete a function evaluation: uniform, half-normal, and exponential. We choose these three distributions since they exhibit three different notions of tail decay, namely bounded, sub-Gaussian, and sub-exponential[1]. Table 1 describes these distributions and states the expected number of evaluations $n_{\mathrm{seq}}, n_{\mathrm{syn}}, n_{\mathrm{asy}}$ for seqTS, synTS, asyTS respectively with M workers in time T. The final theoretical result of this paper, presented below, bounds $\mathrm{BSR}'(T)$ for the Thompson sampling variants.

Theorem 5 (Informal, Simple regret with time for TS)**.** *Let $f \sim \mathcal{GP}(0, \kappa)$ and assume that for* asyTS*, ξ_M is bounded by C_κ after suitable initialisation. Assume that the times taken for an evaluation are i.i.d random variables with either uniform, half-normal or exponential distributions. Let $n_{seq}, n_{syn}, n_{asy}$ be as given in Table 1. Then $n_{seq} \leq n_{syn} \leq n_{asy}$ and $\mathrm{BSR}'(T)$ can be upper bounded by the following terms for* seqTS*,* synTS*, and* asyTS*.*

$$\textsf{seqTS:}\quad \sqrt{\frac{\Psi_{n_{\mathrm{seq}}} \log(n_{\mathrm{seq}})}{n_{\mathrm{seq}}}},$$

[1] While we study uniform, half-normal and exponential, analogous results for other distributions with similar tail behaviour are possible with the appropriate concentration inequalities. See Appendix B.

Distribution	pdf $p(x)$	seqTS	synTS	asyTS
$\text{Unif}(a,b)$	$\frac{1}{b-a}$ for $x \in (a,b)$	$n_{\text{seq}} = \frac{2T}{b+a}$	$n_{\text{syn}} = M\frac{T(M+1)}{a+bM}$	$n_{\text{asy}} = Mn_{\text{seq}}$ $(> n_{\text{syn}})$
$\mathcal{HN}(\zeta^2)$	$\frac{\sqrt{2}}{\zeta\sqrt{\pi}} e^{-\frac{x^2}{2\zeta^2}}$ for $x > 0$	$n_{\text{seq}} = \frac{T\sqrt{\pi}}{\zeta\sqrt{2}}$	$n_{\text{syn}} \asymp \frac{Mn_{\text{seq}}}{\sqrt{\log(M)}}$	$n_{\text{asy}} = Mn_{\text{seq}}$
$\text{Exp}(\lambda)$	$\lambda e^{-\lambda x}$ for $x > 0$	$n_{\text{seq}} = \lambda T$	$n_{\text{syn}} \asymp \frac{Mn_{\text{seq}}}{\log(M)}$	$n_{\text{asy}} = Mn_{\text{seq}}$

Table 1: The second column shows the probability density functions $p(x)$ for the uniform $\text{Unif}(a,b)$, half-normal $\mathcal{HN}(\zeta^2)$, and exponential $\text{Exp}(\lambda)$ distributions. The subsequent columns show the expected number of evaluations $n_{\text{seq}}, n_{\text{syn}}, n_{\text{asy}}$ for seqTS, synTS, and asyTS respectively with M workers. synTS always completes fewer evaluations than asyTS; e.g., in the exponential case, the difference could be a $\log(M)$ factor.

$$\text{synTS:}\quad \frac{M\sqrt{\log(M)}}{n_{\text{syn}}} + \sqrt{\frac{\Psi_{n_{\text{syn}}}\log(n_{\text{syn}}+M)}{n_{\text{syn}}}},$$

$$\text{asyTS:}\quad \frac{\text{poly}(M)}{n_{\text{asy}}} + \sqrt{\frac{C_\kappa\Psi_{n_{\text{asy}}}\log(n_{\text{asy}})}{n_{\text{asy}}}}.$$

As the above bounds are decreasing with the number of evaluations and since $n_{\text{seq}} < n_{\text{syn}} < n_{\text{asy}}$, the bound for $\text{BSR}'(T)$ shows the opposite trend to $\text{BSR}(n)$: asyTS is better than synTS which is better than seqTS. While the difference between synTS and asyTS is only a constant factor for the uniform distribution, it grows with the number of workers M for heavier tailed distributions; $\sqrt{\log(M)}$ for the half-normal and $\log(M)$ for the exponential. Hence, as the number of workers M increases, asyTS becomes increasingly attractive when compared to synTS. Intuitively, when there is more variability in evaluations, workers may sit idle for longer in the synchronous setting and hence synTS will complete fewer evaluations than asyTS.

Synopsis: The take-aways of our theoretical analysis can be summarised as follows. Theorems 2 and 3 show that since the synchronous setting has more information than the asynchronous setting, it achieves a better bound for $\text{BSR}(n)$. Therefore, if function evaluations deterministically take the same amount of time, the synchronous algorithm may be preferred. Further, in some applications, we are necessarily in the synchronous setting. For example, in pre-clinical drug discovery, high throughput screening equipment can test a few thousand compounds in parallel, but only in batches [16]. However, Theorem 5 contends that if there is significant variability in evaluation times, then it is prudent to be asynchronous despite the lack of information when compared to the synchronous setting.

A note on the proofs: To lift the sequential TS proof of Russo and Van Roy [35] to the synchronous case we exploit several properties of TS in this setting; for example, the distribution of $x_j|\mathcal{D}_j$ is the same as $x_\star|\mathcal{D}_j$, all jobs in one batch are completed before the next, and that conditioned on the randomness of the algorithm, the jobs in one batch are deterministic. These properties, along with a careful decomposition of terms in the instantaneous regret yields the bound. Unfortunately these properties do not hold in the asynchronous case, and we resort to techniques from Desautels et al. [9] to bound the MIG via uncertainty sampling. For Theorem 5, we establish concentration results for sums of random variables and sums of their maxima. The proofs of the uniform and half-normal cases uses standard sub-Gaussianity arguments whereas the proof for the exponential distribution uses a logarithmic Sobolev inequality and Herbst's argument [6].

4 Experiments

We compare parallelised TS with a comprehensive suite of parallel BO methods from the literature on a series of synthetic experiments and a hyper-parameter tuning task on the CIFAR-10 dataset.

Methods: *Synchronous Methods:* synRAND: synchronous random sampling, synTS: synchronous TS, synBUCB from [9], synUCBPE from [8]. *Aynchronous Methods:* asyRAND: asynchronous random sampling, asyHUCB: an asynchronous version of UCB with hallucinated observations [9, 11], asyUCB: asynchronous upper confidence bound [41], asyEI: asynchronous expected improvement [21], asyTS: asynchronous TS, asyHTS: asynchronous TS with hallucinated observations to explicitly encourage diversity. This last method is based on asyTS but bases the posterior on $\mathcal{D}_j \cup \{(x, \mu_{\mathcal{D}_j}(x))\}_{x\in F_j}$ in line 5 of Algorithm 2, where F_j are the points in evaluation by other workers at step j and $\mu_{\mathcal{D}_j}$ is the posterior mean conditioned on just $\mathcal{D}_j$; this preserves the mean of the GP, but shrinks the variance around the points in F_j. This method is inspired by [9, 11], who use such hallucinations for UCB/EI-type strategies so as to discourage picking points close to those that are already in evaluation. asyUCB and asyEI directly use the sequential UCB and EI criteria, since the asynchronous versions do not repeatedly pick the same point for all workers. asyHUCB adds hallucinated observations to encourage diversity and is similar to [11] (who use EI instead) and is also an asynchronous version of [9]. While there are other methods for parallel BO, many of them are either computationally quite expensive and/or require tuning several hyperparameters which might affect performance; They are not straightforward to implement and their implementations are not publicly available. Appendix C describes

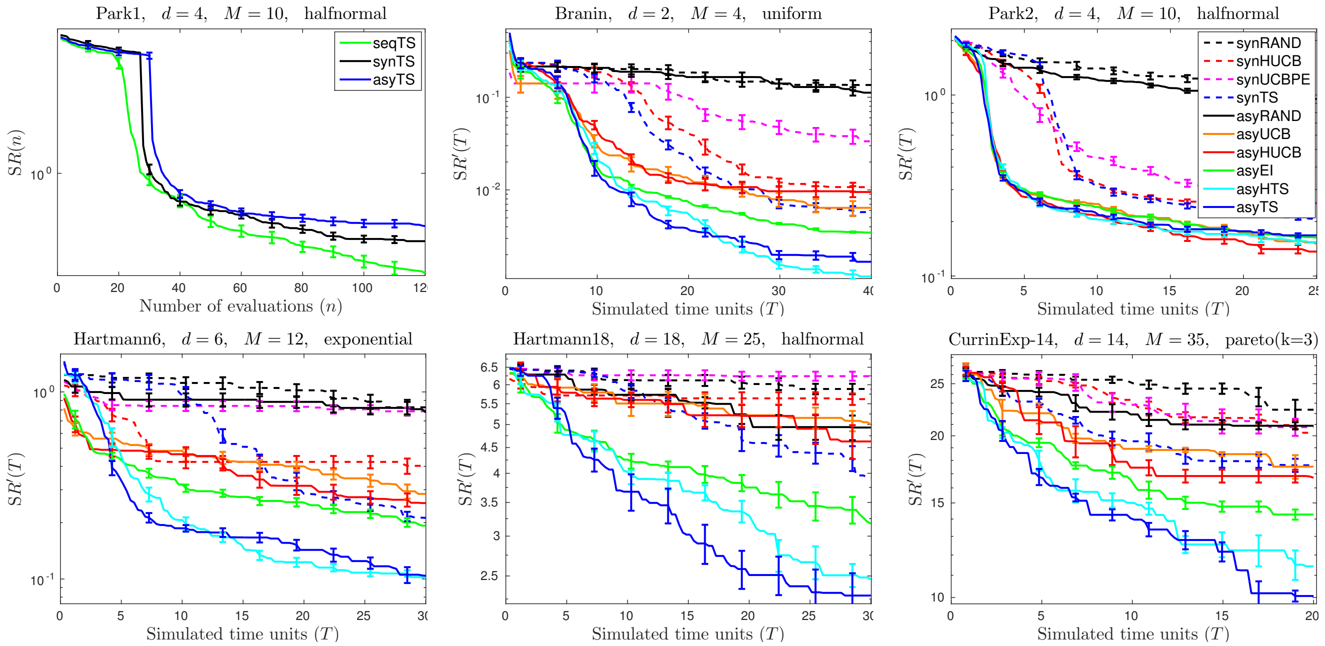


Figure 2: Results on the synthetic experiments. The title states the function used, its dimensionality d, the number of workers M and the distribution used for the time. All distributions were constructed so that the expected time for one evaluation was one time unit. The dotted lines depict synchronous methods while the solid lines are for asynchronous methods. The error bars indicate one standard error. All figures were averaged over at least 15 experiments.

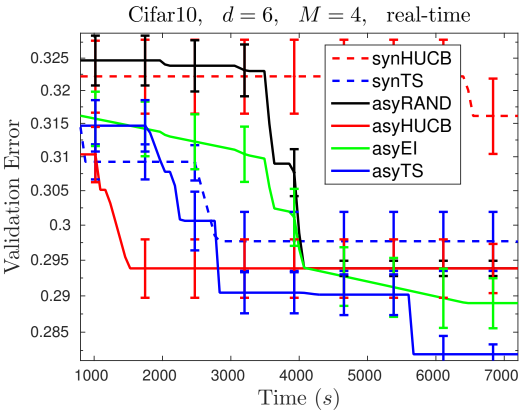
implementation details for all methods.

Synthetic Experiments: We first present some results on a suite of benchmarks for global optimisation. To better align with our theoretical analysis, we add Gaussian noise to the function value when querying. This makes the problem more challenging than standard global optimisation where evaluations are not noisy. In our first experiment, we corroborate the claims in Theorems 1, 2, and 3 by comparing the performance of seqTS, synTS, and asyTS in terms of the number of evaluations n on the Park1 function. The results, displayed in the first panel of Fig. 2, confirm that when comparing solely in terms of n, the sequential version outperforms the parallel versions while synchronous does marginally better than asynchronous.

Next, we present results on a series of global optimisation benchmarks with different values for the number of parallel workers M. We model the evaluation "time" as a random variable that is drawn from either a uniform, half-normal, exponential, or Pareto[2] distribution. Each time a worker makes an evaluation, we also draw a sample from this time distribution and maintain a queue to simulate the different start and finish times for each evaluation. The results are presented in Fig. 2 where we plot the simple regret $\mathrm{SR}'(T)$ against (simulated) time T. In the Park2 experiment, all asynchronous methods perform roughly the same and outperform the synchronous methods. On all other the other problems, asyTS performs best among the asynchronous methods and synTS among the synchronous methods. asyHTS, which also uses hallucinated observations, performs about the same or slightly worse than asyTS, demonstrating that there is no need to explicitly encourage diversity in TS. It is worth emphasizing that the improvement of TS over other methods become larger as M increases (e.g. $M > 20$). We believe that the ability to scale robustly with the number of workers is primarily due to the conceptual simplicity of our approach. Appendix C provides more details on these functions and additional synthetic experiments.

Image Classification on Cifar-10: We experiment with tuning hyperparameters of a 6 layer convolutional neural network on an image classification task on the Cifar-10 dataset [29]. We tune the number of filters/neurons at each layer in the range $(16, 256)$. Here, each function evaluation trains the model on 10K images for 20 epochs and computes the validation accuracy on a validation set of 10K images. Our implementation uses Tensorflow [1] and we use a parallel set up of $M = 4$ Titan X GPUs. The number of filters influences the training time which varied between ~ 4 to ~ 16 minutes depending on the size of the model. Note that this deviates from our theoretical analysis which treats function evaluation times as independent random variables, but it still introduces variability to evaluation times and demonstrates the robustness of our approach. Each method is given a budget of 2 hours to find the best model by optimising accuracy on a validation set. These evaluations are noisy since the result of each training procedure depends on the initial parameters of the network and other stochasticity in the training procedure. Since the true value of this func-

[2] A Pareto distribution with parameter k has a pdf which decays $p(x) \propto x^{-(k+1)}$.



synBUCB	synTS	asyRAND
25.63 ± 0.002	22.83 ± 1.01	23.93 ± 1.78

asyEI	asyHUCB	asyTS
$\mathbf{19.49 \pm 0.21}$	22.14 ± 1.12	$\mathbf{19.53 \pm 0.11}$

Figure 3: Results on the Cifar-10 experiment. Left: The best validation set error vs time for each method (lower is better). Top: Test set error after training the best model chosen by each method for 80 epochs. The results presented are averaged over 9 experiments.

tion is unknown, we simply report the best validation error achieved by each method. Due to the expensive nature of this experiment we only compare 6 of the above methods. The results are presented in Fig. 3. asyTS performs best on the validation error. The following are ranges for the number of evaluations for each method over 9 experiments; *synchronous:* synBUCB: 56 - 68, synTS: 56 - 68. *asynchronous:* asyRAND: 93 - 105, asyEI: 83 - 92, asyHUCB: 85 - 92, asyTS: 80 - 88.

While 20 epochs is insufficient to completely train a model, the validation error gives a good indication of how well the model would perform after sufficient training. In Fig. 3, we also give the error on a test set of 10K images after training the best model chosen by each algorithm to completion, i.e. for 80 epochs. asyTS and asyEI are able to recover the best models which achieve an accuracy of about 80%. While this falls short of state of the art results on Cifar-10 (for e.g. [14]), it is worth noting that we use only a small subset of the Cifar-10 dataset and a relatively small model. Nonetheless, it demonstrates the superiority of our approach over other baselines.

5 Conclusion

We study parallelised versions of TS for synchronous and asynchronous BO. Theoretically, we demonstrate that the algorithms synTS and asyTS perform as well as their purely sequential counterpart in terms of number of evaluations. However, when we factor in time, asyTS outperforms the other two versions. Practically, the main advantage of the proposed methods over prior approaches are conceptual simplicity and straightforward implementation, which enables us to scale robustly to a large number of workers.

We close with some avenues for future research. One challenge that we have already mentioned, is bounding the regret for asyTS without the initialisation procedure. Second, we are interested in other models for evaluation times, for example to capture correlations between the evaluation time and the query point $x_j \in \mathcal{X}$ that arise in practice, such as in our CNN experiment. We believe that in such problems, even sequential algorithms might be different from the conventional BO methods. One could also consider models where some workers are slower than the rest. Third, in the asynchronous setting, there might be instances where the algorithm might choose to kill an evaluation in progress based on the result of a completed job. The algorithm might also choose to wait for another evaluation to finish without immediately deploying a free worker, so as to incorporate additional information. Finally, an open challenge for TS in GPs is maximising the sample g (step 4, Algorithm 1) which can be computationally challenging as g is a random quantity (See Appendix C). We look forward to pursuing these directions.

Acknowledgements

This research is partly funded by DOE grant DESC0011114, NSF grant IIS1563887, the Darpa D3M program, and AFRL. KK is supported by a Facebook fellowship and a Siebel scholarship.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv:1603.04467*, 2016.

[2] Robert J Adler. *An Introduction to Continuity, Extrema, and Related Topics for General Gaussian Processes*. IMS, 1990.

[3] Shipra Agrawal and Navin Goyal. Analysis of thomp-

son sampling for the multi-armed bandit problem. In *Conference on Learning Theory (COLT)*, 2012.

[4] Hildo Bijl, Thomas B Schön, Jan-Willem van Wingerden, and Michel Verhaegen. A Sequential Monte Carlo approach to Thompson sampling for Bayesian optimization. *arXiv preprint arXiv:1604.00169*, 2016.

[5] Stéphane Boucheron and Maud Thomas. Concentration inequalities for order statistics. *Electronic Communications in Probability*, 2012.

[6] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press, 2013.

[7] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. *arXiv:1704.00445*, 2017.

[8] Emile Contal, David Buffoni, Alexandre Robicquet, and Nicolas Vayatis. Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *European Conference on Machine Learning (ECML/PKDD)*, 2013.

[9] Thomas Desautels, Andreas Krause, and Joel W Burdick. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *Journal of Machine Learning Research (JMLR)*, 2014.

[10] Subhashis Ghosal and Anindya Roy. Posterior consistency of Gaussian process prior for nonparametric binary regression". *Annals of Statistics*, 2006.

[11] David Ginsbourger, Janis Janusevskis, and Rodolphe Le Riche. Dealing with asynchronicity in parallel gaussian process based global optimization. In *Conference of the ERCIM WG on Computing and Statistics*, 2011.

[12] Javier Gonzalez, Joseph Longworth, David James, and Neil Lawrence. Bayesian Optimization for Synthetic Gene Design. In *NIPS Workshop on Bayesian Optimization in Academia and Industry*, 2014.

[13] Javier González, Zhenwen Dai, Philipp Hennig, and Neil D Lawrence. Batch Bayesian Optimization via Local Penalization. *arXiv:1505.08052*, 2015.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[15] José Miguel Hernández-Lobato, James Requeima, Edward O Pyzer-Knapp, and Alán Aspuru-Guzik. Parallel and distributed thompson sampling for large-scale accelerated exploration of chemical space. *arXiv preprint arXiv:1706.01825*, 2017.

[16] James P Hughes, Stephen Rees, S Barrett Kalindjian, and Karen L Philpott. Principles of early drug discovery. *British journal of pharmacology*, 162(6):1239–1249, 2011.

[17] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *LION*, 2011.

[18] Brett W Israelsen, Nisar R Ahmed, Kenneth Center, Roderick Green, and Winston Bennett. Towards adaptive training of agent-based sparring partners for fighter pilots. In *AIAA Information Systems-AIAA Infotech@ Aerospace*, page 0343, 2017.

[19] Janis Janusevskis, Rodolphe Le Riche, David Ginsbourger, and Ramunas Girdziusas. Expected Improvements for the Asynchronous Parallel Global Optimization of Expensive Functions: Potentials and Challenges. In *Learning and Intelligent Optimization*, 2012.

[20] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian Optimization Without the Lipschitz Constant. *J. Optim. Theory Appl.*, 1993.

[21] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *J. of Global Optimization*, 1998.

[22] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online learning under delayed feedback. In *International Conference on Machine Learning (ICML)*, 2013.

[23] Kwang-Sung Jun, Kevin Jamieson, Robert Nowak, and Xiaojin Zhu. Top arm identification in multi-armed bandits with batch arm pulls. In *Artificial Intelligence and Statistics (AISTATS)*, 2016.

[24] Kirthevasan Kandasamy, Jeff Schenider, and Barnabás Póczos. High Dimensional Bayesian Optimisation and Bandits via Additive Models. In *International Conference on Machine Learning*, 2015.

[25] Kirthevasan Kandasamy, Gautam Dasarathy, Junier Oliva, Jeff Schenider, and Barnabás Póczos. Gaussian Process Bandit Optimisation with Multi-fidelity Evaluations. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[26] Tarun Kathuria, Amit Deshpande, and Pushmeet Kohli. Batched gaussian process bandit optimization via determinantal point processes. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[27] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *ALT*, volume 12, pages 199–213. Springer, 2012.

[28] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb):235–284, 2008.

[29] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images, 2009.

[30] R. Martinez-Cantin, N. de Freitas, A. Doucet, and J. Castellanos. Active Policy Learning for Robot Planning and Exploration under Uncertainty. In *Proceedings of Robotics: Science and Systems*, 2007.

[31] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[32] David Parkinson, Pia Mukherjee, and Andrew R Liddle. A Bayesian model selection analysis of WMAP3. *Physical Review*, 2006.

[33] Kent Quanrud and Daniel Khashabi. Online learning with adversarial delays. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[34] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. Adaptative computation and machine learning series. University Press Group Limited, 2006.

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

[36] Daniel Russo and Benjamin Van Roy. An Information-theoretic analysis of Thompson sampling. *Journal of Machine Learning Research (JMLR)*, 2016.

[37] MW. Seeger, SM. Kakade, and DP. Foster. Information Consistency of Nonparametric Gaussian Process Methods. *IEEE Transactions on Information Theory*, 2008.

[38] Amar Shah and Zoubin Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[39] Bobak Shahriari, Ziyu Wang, Matthew W Hoffman, Alexandre Bouchard-Côté, and Nando de Freitas. An Entropy Search Portfolio for bayesian Optimization. *arXiv preprint arXiv:1406.4625*, 2014.

[40] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems*, 2012.

[41] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *ICML*, 2010.

[42] W. R. Thompson. On the Likelihood that one Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, 1933.

[43] Jialei Wang, Scott C Clark, Eric Liu, and Peter I Frazier. Parallel Bayesian Global Optimization of Expensive Functions. *arXiv:1602.05149*, 2016.

[44] Zi Wang, Chengtao Li, Stefanie Jegelka, and Pushmeet Kohli. Batched high-dimensional bayesian optimization via structural kernel learning. *arXiv:1703.01973*, 2017.

[45] Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In *AISTATS*, 2018.

[46] Jian Wu and Peter Frazier. The parallel knowledge gradient method for batch bayesian optimization. In *Advances In Neural Information Processing Systems*, 2016.

Appendix

A Theoretical Analysis for Parallelised Thompson Sampling in GPs

A.1 Some Relevant Results on GPs and GP Bandits

We first review some related results on GPs and GP bandits. We begin with the definition of the *Maximum Information Gain* (MIG) which characterises the statistical difficulty of GP bandits [41].

Definition 6 (Maximum Information Gain [41])**.** *Let $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$ where $\kappa : \mathcal{X}^2 \to \mathbb{R}$. Let $A = \{x_1, \dots, x_n\} \subset \mathcal{X}$ be a finite subset. Let $f_A, \epsilon_A \in \mathbb{R}^n$ such that $(f_A)_i = f(x_i)$ and $(\epsilon_A)_i \sim \mathcal{N}(0, \eta^2)$. Let $y_A = f_A + \epsilon_A \in \mathbb{R}^n$. Denote the Shannon Mutual Information by I. The MIG is the maximum information we can gain about f using n evaluations. That is,*

$$\Psi_n = \max_{A \subset \mathcal{X}, |A|=n} I(f; y_A).$$

Srinivas et al. [41] and Seeger et al. [37] provide bounds on the MIG for different classes of kernels. For example for the SE kernel, $\Psi_n \asymp \log(n)^{d+1}$ and for the Matérn kernel with smoothness parameter ν, $\Psi_n \asymp n^{\frac{d(d+1)}{2\nu+d(d+1)}} \log(n)$. The next theorem due to Srinivas et al. [41] bounds the sum of variances of a GP using the MIG.

Lemma 7 (Lemma 5.2 and 5.3 in [41])**.** *Let $f \sim \mathcal{GP}(0, \kappa)$, $f : \mathcal{X} \to \mathbb{R}$ and each time we query at any $x \in \mathcal{X}$ we observe $y = f(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \eta^2)$. Let $\{x_1, \dots, x_n\}$ be an arbitrary set of n evaluations to f where $x_j \in \mathcal{X}$ for all j. Let σ_{j-1}^2 denote the posterior variance conditioned on the first $j-1$ of these queries, $\{x_1, \dots, x_{j-1}\}$. Then, $\sum_{j=1}^n \sigma_{j-1}^2(x_j) \leq \frac{2}{\log(1+\eta^{-2})} \Psi_n$.*

Next we will need the following regularity condition on the derivatives of the GP sample paths. When $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$, it is satisfied when κ is four times differentiable, e.g. the SE kernel and Matérn kernel when $\nu > 2$ [10].

Assumption 8 (Gradients of GP Sample Paths [10])**.** *Let $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$, where $\kappa : \mathcal{X}^2 \to \mathbb{R}$ is a stationary kernel. The partial derivatives of f satisfies the following condition. There exist constants $a, b > 0$ such that,*

$$\textit{for all } J > 0, \;\; \textit{and for all } i \in \{1, \dots, d\}, \quad \mathbb{P}\left(\sup_x \left| \frac{\partial f(x)}{\partial x_i} \right| > J \right) \leq a e^{-(J/b)^2}.$$

Finally, we will need the following result on the supremum of a Gaussian process. It is satisfied when κ is twice differentiable.

Lemma 9 (Supremum of a GP [2])**.** *Let $f \sim \mathcal{GP}(0, \kappa)$ have continuous sample paths. Then, $\mathbb{E}\|f\|_\infty = \Xi < \infty$.*

This, in particular implies that in the definition of $\mathrm{BSR}'(T)$ in (3), $\max_{x \in \mathcal{X}} |f(x_\star) - f(x)| \leq 2\,\Xi$.

Finally, we will use the following result in our parallel analysis. Recall that the posterior variance of a GP does not depend on the observations.

Lemma 10 (Lemma 1 (modified) in [9])**.** *Let $f \sim \mathcal{GP}(0, \kappa)$. Let A, B be finite subsets of $\mathcal{X}$. Let $y_A \in \mathbb{R}^{|A|}$ and $y_B \in \mathbb{R}^{|B|}$ denote the observations when we evaluate f at A and B respectively. Further let σ_A, $\sigma_{A \cup B} : \mathcal{X} \to \mathbb{R}$ denote the posterior standard deviation of the GP when conditioned on A and $A \cup B$ respectively. Then,*

$$\textit{for all } x \in \mathcal{X}, \qquad \frac{\sigma_A(x)}{\sigma_{A \cup B}(x)} \leq \exp\big(I(f; y_B | y_A) \big)$$

The proof exactly mimics the proof in Desautels et al. [9]. Lemma 10 implies $\sigma_A(x) \leq \xi_M^{1/2} \sigma_{A \cup B}(x)$ where ξ_M is from (4).

A.2 Notation & Set up

We will require some set up in order to unify the analysis for the sequential, synchronously parallel and asynchronously parallel settings.

- The first is an indexing for the function evaluations. This is illustrated for the synchronous and asynchronous parallel settings in Figure 1. In our analysis, the index j or step j will refer to the j^{th} function evaluation dispatched by the algorithm. In the sequential setting this simply means that there were $j-1$ evaluations before the j^{th} . For synchronous strategies we index the first batch from $j = 1, \dots, M$ and then the next batch $j = M+1, \dots, 2M$ and so on as in Figure 1. For the asynchronous setting, this might differ as each evaluation takes different amounts of time. For example, in Figure 1, the first worker finishes the $j = 1^{\text{st}}$ job and then starts the $j = 4^{\text{th}}$, while the second worker finishes the $j = 2^{\text{nd}}$ job and starts the $j = 6^{\text{th}}$.

- Next, we define $\mathcal{D}_j$ at step j of the algorithm to be the query-observation pairs (x_k, y_k) for function evaluations completed by step j. In the sequential setting $\mathcal{D}_j = \{(x_k, y_k) : k \in \{1, \dots, j-1\}\}$ for all j. For the synchronous setting in Figure 1, $\mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_3 = \varnothing, \mathcal{D}_4 = \mathcal{D}_5 = \mathcal{D}_6 = \{(x_k, y_k) : k \in \{1, 2, 3\}\}, \mathcal{D}_7 = \mathcal{D}_8 = \mathcal{D}_9 = \{(x_k, y_k) : k \in \{1, 2, 3, 4, 5, 6\}\}$ etc. Similarly, for the asynchronous setting, $\mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_3 = \varnothing$, $\mathcal{D}_4 = \{(x_k, y_k) : k \in \{1\}\}$, $\mathcal{D}_5 = \{(x_k, y_k) : k \in \{1, 3\}\}$, $\mathcal{D}_6 = \{(x_k, y_k) : k \in \{1, 2, 3\}\}$, $\mathcal{D}_7 = \{(x_k, y_k) : k \in \{1, 2, 3, 5\}\}$ etc. Note that in the asynchronous setting $|\mathcal{D}_j| = j - M$ for all $j > M$. $\{\mathcal{D}_j\}_{j \geq 1}$ determines the filtration when constructing the posterior GP at every step j.

- Finally, in all three settings, $\mu_A : \mathcal{X} \to \mathbb{R}$ and $\sigma_A : \mathcal{X} \to \mathbb{R}_+$ will refer to the posterior mean and standard deviation of the GP conditioned on some evaluations A, i.e. $A \subset \mathcal{X} \times \mathbb{R}$ is a set of (x, y) values and $|A| < \infty$. They can be computed by plugging in the (x, y) values in A to (1). For example, $\mu_{\mathcal{D}_j}, \sigma_{\mathcal{D}_j}$ will denote the mean and standard deviation conditioned on the completed evaluations, $\mathcal{D}_j$. Finally, when using our indexing scheme above we will also overload notation so that σ_{j-1} will denote the posterior standard deviation conditioned on evaluations from steps 1 to $j-1$. That is $\sigma_{j-1} = \sigma_A$ where $A = \{(x_k, y_k)\}_{k=1}^{j-1}$.

A.3 Parallelised Thompson Sampling

In the remainder of this section, $\beta_n \in \mathbb{R}$ for all $n \geq 1$ will denote the following value.

$$\beta_n \;=\; 4(d+1)\log(n) + 2d\log(dab\sqrt{\pi}) \;\asymp\; d\log(n), \tag{5}$$

Here d is the dimension, a, b are from Assumption 8, and n will denote the number of evaluations. Our first theorem below is a bound on the simple regret for synTS after n completed evaluations.

Theorem 11. *Let $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$ where $\kappa : \mathcal{X}^2 \to \mathbb{R}$ satisfies Assumption 8. Further, without loss of generality $\kappa(x, x') \leq 1$. Then for synTS, the Bayes simple regret after n evaluations satisfies,*

$$\mathrm{BSR}(n) \;\leq\; \frac{C_1}{n} \;+\; \frac{(M-1)\beta_{M-1}^{1/2}}{n} \;+\; \sqrt{\frac{C_2 \beta_{n+M-1} \Psi_n}{n}},$$

where Ψ_n is the MIG in Definition 6, β_n is as defined in (5), ξ_M is from (4), and $C_1 = \pi^2/6 + \sqrt{2}\pi^{5/2}/12$, $C_2 = 2/\log(1 + \eta^{-2})$ are constants.

Proof. Our proof is based on techniques from Russo and Van Roy [35] and Srinivas et al. [41]. As part of analysis, we will discretise $\mathcal{X}$ at each step j of the algorithm. Our discretisation ν_j, is obtained via a grid of $\tau_j = j^2 dab\sqrt{\pi}$ equally spaced points along each coordinate and has size $|\nu_j| = \tau_j^d$. It is easy to verify that ν_j satisfies the following property: for all $x \in \mathcal{X}$, $\|x - [x]_j\|_1 \leq d/\tau_j$, where $[x]_j$ is the closest point to x in ν_j. This discretisation is deterministically constructed ahead of time and does not depend on any of the random quantities in the problem.

For the purposes of our analysis, we define the Bayes cumulative regret after n evaluations as,

$$\mathrm{BCR}(n) = \mathbb{E}\left[\sum_{j=1}^{n} f(x_\star) - f(x_j)\right].$$

Here, just as in (2), the expectation is with respect to the randomness in the prior, observations and algorithm. Since the average is larger than the minimum, we have $\frac{1}{n}\sum_j f(x_\star) - f(x_j) \geq \min_j (f(x_\star) - f(x_j)) = \mathrm{SR}(n)$; hence $\mathrm{BSR}(n) \leq \frac{1}{n}\mathrm{BCR}(n)$.

For the proof, we will use the following property of TS. The sampling distribution at time step j is $p(x_\star|\mathcal{D}_j)$. While TS is a randomised strategy, this distribution itself is constructed deterministically; the randomness comes from the algorithm.

We will denote this external source of randomness at step j by R_j. Now denote $U_j(\cdot) = \mu_{\mathcal{D}_j}(\cdot) + \beta_j^{1/2}\sigma_{\mathcal{D}_j}(\cdot)$ and $V_j(\cdot) = \mu_{\mathcal{D}_j}(\cdot) + \beta_{j+M-1}^{1/2}\sigma_{j-1}(\cdot)$. We begin by decomposing BCR(n) as follows.

$$\begin{aligned}
\mathrm{BCR}(n) &= \sum_{j=1}^{n} \mathbb{E}\left[f(x_\star) - f(x_j)\right] \\
&\overset{(a)}{=} \sum_{j=1}^{n} \mathbb{E}\big[f(x_\star) - f([x_\star]_j) + f([x_\star]_j) - U_j([x_\star]_j) + U_j([x_\star]_j) - V_j([x_\star]_j) + V_j([x_\star]_j) \\
&\qquad\qquad - V_j([x_j]_j) + V_j([x_j]_j) - f([x_j]_j) + f([x_j]_j) - f(x_j)\big] \\
&= \underbrace{\sum_{j=1}^{n} \mathbb{E}\big[f(x_\star) - f([x_\star]_j)\big]}_{A_1} + \underbrace{\sum_{j=1}^{n} \mathbb{E}\big[f([x_j]_j) - f(x_j)\big]}_{A_2} \\
&\qquad + \underbrace{\sum_{j=1}^{n} \mathbb{E}\big[f([x_\star]_j) - U_j([x_\star]_j)\big]}_{A_3} + \underbrace{\sum_{j=1}^{n} \mathbb{E}\big[V_j([x_j]_j) - f([x_j]_j)\big]}_{A_4} \\
&\qquad + \underbrace{\sum_{j=1}^{n} \mathbb{E}\big[U_j([x_\star]_j) - V_j([x_\star]_j)\big]}_{A_5} + \underbrace{\sum_{j=1}^{n} \mathbb{E}\big[V_j([x_\star]_j) - V_j([x_j]_j)\big]}_{A_6}.
\end{aligned}$$

In the first step we have added and subtracted $f([x_\star]_j), U_j([x_\star]_j), V_j([x_\star]_j), V_j([x_j]_j)$, and $f([x_j]_j)$ and in the second step we have separated the terms into the sums $A_1, \dots, A_6$. We will first control A_5 and A_6 using properties of TS.

By conditioning on $\mathcal{D}_j, \{R_k\}_{k=1}^{j-1}$ we argue that j^{th} term in A_6 vanishes, i.e. $\mathbb{E}\big[V_j([x_\star]_j) - V_j([x_j]_j)\big] = \mathbb{E}\big[\mathbb{E}[V_j([x_\star]_j) - V_j([x_j]_j)|\mathcal{D}_j, \{R_k\}_{k<j}]\big] = 0$. For this, first note that as x_j is sampled from the posterior distribution for $x_\star$ conditioned on $\mathcal{D}_j$, both $x_j|\mathcal{D}_j$ and $x_\star|\mathcal{D}_j$ have the same distribution. Since $R_j \notin \{R_k\}_{k<j}$ and this randomness is independent of everything else, $x_\star$ and x_j are equal in distribution conditioned on $\mathcal{D}_j, \{R_k\}_{k<j}$. Now observe that V_j is deterministic conditioned on $\mathcal{D}_j, \{R_k\}_{k<j}$. This is because at step j, $\mu_{\mathcal{D}_j}$ is a function of past query points and observations $\mathcal{D}_j$, $\sigma_{\mathcal{D}_j}$ depends only on past query points and σ_{j-1} depends only on past query points and those currently in evaluation; the latter is also deterministic since we are conditioning on $\{R_k\}_{k<j}$. Now, as the discretisation ν_j is fixed ahead of time, $V_j([x_j]_j)$ and $V_j([x_\star]_j)$ are also equal in distribution given $\mathcal{D}_j, \{R_k\}_{k<j}$. Therefore, both quantities are also equal in expectation.

Now, let us bound A_5. Noting that each term inside the summation A_5 is $\mathbb{E}[U_j([x_\star]_j) - V_j([x_\star]_j)] = \mathbb{E}[\beta_j^{1/2}\sigma_{\mathcal{D}_j}([x_\star]_j) - \beta_{j+M}^{1/2}\sigma_{j-1}([x_\star]_j)]$ we have,

$$\begin{aligned}
A_5 &= \sum_{j=1}^{M-1} \beta_j^{1/2}\mathbb{E}[\sigma_{\mathcal{D}_j}([x_\star]_j)] - \sum_{j=n-M+1}^{n} \mathbb{E}[\beta_{j+M-1}^{1/2}\sigma_{j-1}([x_\star]_j)] + \sum_{j=M}^{n-M} \beta_j^{1/2}\mathbb{E}\left[\sigma_{\mathcal{D}_j}([x_\star]_j) - \sigma_{j-M}([x_\star]_j)\right] \\
&\le (M-1)\beta_{M-1}^{1/2}\|\kappa\|_\infty^{1/2} + \sum_{j=M}^{n-M} \beta_j^{1/2}\mathbb{E}\left[\sigma_{\mathcal{D}_j}([x_\star]_j) - \sigma_{j-M}([x_\star]_j)\right] \qquad (6)
\end{aligned}$$

In the first step we have simply rearranged the terms and in the second step we have bounded the first sum by it largest possible values and dropped the second sum. For the synchronous case, we always have $\mathcal{D}_j \supseteq \{(x_k, y_k) : k \le j - M\}$. Hence, $\sigma_{\mathcal{D}_j} \le \sigma_{j-M}$ pointwise for all j and each term is bounded above by 0. Finally, since we have assumed $\|\kappa\|_\infty = 1$, $A_6 \le (M-1)\beta_{M-1}^{1/2}$.

To bound A_1, A_2 and A_3 we use the following Lemmas. The proofs are in Sections A.3.1 and A.3.2.

Lemma 12. *At step j, for all $x \in \mathcal{X}$, $\mathbb{E}[|f(x) - f([x]_j)|] \le \frac{1}{2j^2}$.*

Lemma 13. *At step j, for all $x \in \nu_j$, $\mathbb{E}[\mathbb{1}\{f(x) > U_j(x)\} \cdot (f(x) - U_j(x))] \le \frac{1}{j^2\sqrt{2\pi}|\nu_j|}$.*

Using Lemma 12 and the fact that $\sum_j j^{-2} = \pi^2/6$, we have $A_1 + A_2 \leq \pi^2/6$. We bound A_3 via,

$$\begin{aligned} A_3 \;&\leq\; \mathbb{E}\bigg[\sum_{j=1}^{n} \mathbb{1}\{f([x_\star]_j) > U_j([x_\star]_j)\} \cdot (f([x_\star]_j) - U_j([x_\star]_j))\bigg] \\ &\leq\; \sum_{j=1}^{n}\sum_{x\in\nu_j} \mathbb{E}\big[\,\mathbb{1}\{f(x) > U_j(x)\} \cdot (f(x) - U_j(x))\,\big] \leq \sum_{j=1}^{n}\sum_{x\in\nu_j} \frac{1}{j^2\sqrt{2\pi}|\nu_j|} = \frac{\sqrt{2\pi}}{12} \end{aligned}$$

In the first step we upper bounded A_3 by only considering the positive terms in the summation. The second step bounds the term for $[x_\star]_j$ by the sum of corresponding terms for all $x \in \nu_j$. We then apply Lemma 13.

Finally, we bound each term inside the summation of A_4 as follows,

$$\begin{aligned} \mathbb{E}[V_j([x_j]_j) - f([x_j]_j)] &= \mathbb{E}[\mu_{\mathcal{D}_j}([x_j]_j) + \beta_{j+M}^{1/2}\sigma_{j-1}([x_j]_j) - f([x_j]_j)] \\ &= \mathbb{E}[\mu_{\mathcal{D}_j}([x_j]_j) + \beta_{j+M}^{1/2}\sigma_{j-1}([x_j]_j) - \mathbb{E}[f([x_j]_j)|\mathcal{D}_j, \{R_k\}_{k<j}]] = \mathbb{E}[\beta_{j+M}^{1/2}\sigma_{j-1}([x_j]_j)] \end{aligned} \tag{7}$$

Once again, we have used the fact that $\mu_{\mathcal{D}_j}, \sigma_{j-1}$ are deterministic given $\mathcal{D}_j, \{R_k\}_{k<j}$. Therefore,

$$A_4 \overset{(a)}{\leq} \beta_{n+M}^{1/2}\mathbb{E}\left[\sum_{j=1}^{n}\sigma_{j-1}([x_j]_j)\right] \overset{(b)}{\leq} \beta_{n+M}^{1/2}\mathbb{E}\left[\left(n\sum_{j=1}^{n}\sigma_j^2([x_j]_j)\right)^{1/2}\right] \overset{(c)}{\leq} \sqrt{\frac{2n\beta_{n+M}\Psi_n}{\log(1+\eta^{-2})}} \tag{8}$$

Here, (a) uses (7) and that β_j is increasing in j (5). (b) uses the Cauchy-Schwarz inequality and (c) uses Lemma 7. Putting the bounds for $A_1, \dots, A_6$ together we get, $\mathrm{BCR}(n) \;\leq\; C_1 + (M-1)\beta_{M-1}^{1/2} + \sqrt{C_2 n\beta_n\Psi_n}$. The theorem follows from the relation $\mathrm{BSR}(n) \leq \frac{1}{n}\mathrm{BCR}(n)$. □

Next, we present our results for asyTS.

Theorem 14. *Assume the same setting and quantities as in Theorem 11. Then for* asyTS, *the Bayes simple regret after n evaluations satisfies,*

$$\mathrm{BSR}(n) \;\leq\; \frac{C_1}{n} \;+\; \sqrt{\frac{C_2\xi_M\beta_n\Psi_n}{n}}.$$

Here, all quantities are as defined in Theorem 11.

Proof. We will first assume that the n evaluations completed are the the evaluations indexed $j = 1, \dots, n$. Our proof will follow along the same lines as that for synTS, except we will use $U_j(\cdot) = V_j(\cdot) = \mu_{\mathcal{D}_j}(\cdot) + \beta_j^{1/2}\sigma_{\mathcal{D}_j}(\cdot)$. The terms A_1, A_2, A_3 are bound exactly the same way yielding $A_1 + A_2 + A_3 \leq C_1$. A_6 can be shown to be zero by conditioning on $\mathcal{D}_j$ and using a similar argument. $A_5 = 0$ since $U_j = V_j$. Hence, the only thing left to bound is A_4. Using a similar reasoning to (7), we have $\mathbb{E}[U_j([x_j]_j) - f([x_j]_j)] = \mathbb{E}[\beta_j^{1/2}\sigma_{\mathcal{D}_j}([x_j]_j)]$. Then,

$$\begin{aligned} A_4 &\overset{(a)}{\leq} \beta_n^{1/2}\sum_{j=1}^{n}\mathbb{E}[\sigma_{\mathcal{D}_j}([x_j]_j)] \overset{(b)}{\leq} \beta_n^{1/2}\xi_M^{1/2}\mathbb{E}\left[\sum_{j=1}^{n}\sigma_{j-1}([x_j]_j)\right] \\ &\overset{(c)}{\leq} \beta_n^{1/2}\xi_M^{1/2}\mathbb{E}\left[\left(n\sum_{j=1}^{n}\sigma_j^2([x_j]_j)\right)^{1/2}\right] \overset{(d)}{\leq} \sqrt{\frac{2\xi_M n\beta_n\Psi_n}{\log(1+\eta^{-2})}} \end{aligned} \tag{9}$$

Here, (a) uses that β_j is increasing in j (5). (c) uses the Cauchy-Schwarz inequality and (d) uses Lemma 7. For (b), first we note that $\mathcal{D}_j \subseteq \{(x^{(i)}, y^{(i)})\}_{i=1}^{j-1}$. In the asynchronous setting we will be missing up to $M-1$ evaluations during the first M steps and exactly $M-1$ evaluations thereafter. In either case, letting $A = \mathcal{D}_j$ and $B = \{(x^{(i)}, y^{(i)})\}_{i=1}^{j-1}\backslash\mathcal{D}_j$ in Lemma 10 we get,

$$\text{for all } x \in \mathcal{X}, \qquad \sigma_{\mathcal{D}_j}(x) \;\leq\; \exp\big(I(f; y_B|y_{\mathcal{D}_j})\big)\,\sigma_{j-1}(x) \;\leq\; \xi_M^{1/2}\sigma_{j-1}(x). \tag{10}$$

The last step uses (4) and that $|B| < M$.

Now consider the case where the n evaluations completed are not the first n dispatched. Since A_1, A_2, A_3 are bounded by constants summing over all n we only need to worry about A_4. In step (a) of (9), we have bounded A_4 by the sum of posterior variances $\sigma_{j-1}([x_j]_j)$. Since $\sigma_{j'-1}([x_j]_j) < \sigma_{j-1}([x_j]_j)$ for $j' > j$, the sum for any n completed evaluations can be bound by the same sum for the first n evaluations dispatched. The result follows accordingly. □

Finally, the bound for the sequential setting in Theorem 1 follows directly by setting $M = 1$ in Theorem 11. We state it formally below.

Corollary 15. *Assume the same setting and quantities as in Theorem 11. Then for* seqTS*, the Bayes' simple regret after* n *evaluations satisfies,*

$$\mathrm{BSR}(n) \;\leq\; \frac{C_1}{n} + \sqrt{\frac{C_2 \beta_n \Psi_n}{n}},$$

A.3.1 Proof of Lemma 12

Let $L = \sup_{i=1,\dots,d} \sup_{x\in\mathcal{X}} \left|\frac{\partial f(x)}{\partial x_i}\right|$. By Assumption 8 and the union bound we have $\mathbb{P}(L \geq t) \leq da \exp^{-t^2/b^2}$. Let $x \in \mathcal{X}$. We bound,

$$\mathbb{E}[|f(x) - f([x]_j)|] \leq \mathbb{E}[L\|x - [x]_j\|_1] \leq \frac{d}{\tau_j}\mathbb{E}[L] = \frac{d}{\tau_j}\int_0^\infty \mathbb{P}(L \geq t)\mathrm{d}t$$
$$\leq \frac{d}{\tau_j}\int_0^\infty a e^{t^2/b^2}\mathrm{d}t = \frac{dab\sqrt{\pi}}{2\tau_j} = \frac{1}{2j^2}.$$

The first step bounds the difference in the function values by the largest partial derivative and the L^1 distance between the points. The second step uses the properties of the discretisation ν_j and the third step uses the identity $\mathbb{E}X = \int \mathbb{P}(X > t)\mathrm{d}t$ for positive random variables X. The last step uses the value for τ_j specified in the main proof. □

A.3.2 Proof of Lemma 13

The proof is similar to Lemma 2 in [35], but we provide it here for completeness. We will use the fact that for $Z \sim \mathcal{N}(\mu, \sigma^2)$, we have $\mathbb{E}[Z\mathbb{1}(Z > 0)] = \frac{\sigma}{\sqrt{2\pi}}e^{-\mu^2/(2\sigma^2)}$. Noting that $f(x) - U_j(x)|\mathcal{D}_j \sim \mathcal{N}(-\beta_j^{1/2}\sigma_{\mathcal{D}_j}(x), \sigma^2_{\mathcal{D}_j}(x))$, we have,

$$\mathbb{E}[\mathbb{1}\{f(x) > U_j(x)\}\cdot(f(x) - U_j(x))|\mathcal{D}_j] = \frac{\sigma_{\mathcal{D}_j}(x)}{\sqrt{2\pi}}e^{\beta_j/2} \leq \frac{1}{\sqrt{2\pi}|\nu_j|j^2}.$$

Here, the last step uses that $\sigma_{\mathcal{D}_j}(x) \leq \kappa(x,x) \leq 1$ and that $\beta_j = 2\log(j^2|\nu_j|)$. □

A.4 On the Initialisation Scheme and Subsequent Results for asyTS – Theorem 4

Description of the initialisation scheme: The initialisation scheme [9, 28] is an uncertainty sampling procedure designed to reduce the posterior variance throughout the domain $\mathcal{X}$. Here, we first pick the point with the largest prior GP variance, $x_1^{\mathrm{init}} = \mathrm{argmax}_x\, \kappa(x,x)$. We then iterate $x_j^{\mathrm{init}} = \mathrm{argmax}_{x\in\mathcal{X}}\, \kappa_{j-1}(x,x)$ where κ_{j-1} denotes the posterior kernel with the previous $j-1$ evaluations. As the posterior variance of a GP does not depend on the observations, this scheme is asynchronously parallelisable: simply pre-compute the evaluation points and then deploy them in parallel.

Bounds on BSR(n) after initialisation: Desautels et al. [9] provide bounds for C_κ as a function of γ_M for different kernels (see Section 3.1). During this initialisation phase the best bound we can achieve on the instantaneous regret is $\mathbb{E}[f(x_\star) - f(x_j)] \leq 2\,\Xi$. Applying Theorem 14 after initialisation, we have asyTS:

$$\mathrm{BSR}(n) \;\leq\; \frac{C_1}{n} + \frac{2\,\Xi\,\gamma_M}{n} + \sqrt{\frac{C_2 C_\kappa \Psi_n \log(n)}{n}} \tag{11}$$

A more rigorous proof will simply replace the unconditional mutual information in the definition of the MIG with the mutual information conditioned on the first γ_M evaluations.We conjecture that asyTS will not need this initialisation and wish to resolve this in future work.

B Proofs for Parallelised Thompson Sampling with Random Evaluation Times

The goal of this section is to prove Theorem 5. In Section B.2 we derive some concentration results for uniform and half-normal distributes and their maxima. In Section B.3 we do the same for exponential random variables. We put everything together in Section B.4 to prove Theorem 5. We begin by reviewing some well known concepts in concentration of measure.

B.1 Some Relevant Results

We first introduce the notion of sub-Gaussianity, which characterises one of the stronger types of tail behaviour for random variables.

Definition 16 (Sub-Gaussian Random Variables). *A zero mean random variable is said to be τ* sub-Gaussian *if it satisfies, $\mathbb{E}[e^{\lambda X}] \;\leq\; e^{\frac{\tau^2\lambda^2}{2}}$ for all $\lambda \in \mathbb{R}$.*

It is well known that Normal $\mathcal{N}(0, \zeta^2)$ variables are ζ sub-Gaussian and bounded random variables with support in $[a, b]$ are $(b-a)/2$ sub-Gaussian. For sub-Gaussian random variables, we have the following important and well known result.

Lemma 17 (Sub-Gaussian Tail Bound). *Let $X_1, \ldots, X_n$ be zero mean independent random variables such that X_i is σ_i sub-Gaussian. Denote $S_n = \sum_{i=1}^n X_i$ and $\sigma^2 = \sum_{i=1}^n \sigma_i^2$. Then, for all $\epsilon > 0$,*

$$\mathbb{P}\left(S_n \geq \epsilon\right) \;\leq\; \exp\left(\frac{-\epsilon^2}{2\sigma^2}\right), \qquad \mathbb{P}\left(S_n \leq \epsilon\right) \;\leq\; \exp\left(\frac{-\epsilon^2}{2\sigma^2}\right).$$

We will need the following result for Lipschitz functions of Gaussian random variables in our analysis of the half-normal distribution for time, see Theorem 5.6 in Boucheron et al. [6].

Lemma 18 (Gaussian Lipschitz Concentration [6]). *Let $X \in \mathbb{R}^n$ such that $X_i \sim \mathcal{N}(0, \zeta^2)$ iid for $i = 1, \ldots, n$. Let $F : \mathbb{R}^n \to \mathbb{R}$ be an L-Lipschitz function, i.e. $|F(x) - F(y)| \leq L\|x - y\|_2$ for all $x, y \in \mathbb{R}^n$. Then, for all $\lambda > 0$, $\mathbb{E}[\exp^{\lambda F(X)}] \leq \exp\left(\frac{\pi^2 L^2 \zeta^2}{8}\lambda^2\right)$. That is, $F(X)$ is $\frac{\pi L \zeta}{2}$ sub-Gaussian.*

We also introduce Sub-Exponential random variables, which have a different tail behavior.

Definition 19 (Sub-Exponential Random Variables). *A zero mean random variable is said to be* sub-Exponential *with parameters (τ^2, b) if it satisfies, $\mathbb{E}[e^{\lambda X}] \;\leq\; e^{\frac{\tau^2\lambda^2}{2}}$ for all λ with $|\lambda| \leq 1/b$.*

Sub-Exponential random variables are a special case of Sub-Gamma random variables (See Chapter 2.4 in Boucheron et al. [6]) and allow for a Bernstein-type inequality.

Proposition 20 (Sub-Exponential tail bound [6]). *Let $X_1, \ldots, X_n$ be independent sub-exponential random variables with parameters (σ_i^2, b_i). Denote $S_n = \sum_{i=1}^n X_i$ and $\sigma^2 = \sum_{i=1}^n \sigma_i^2$ and $b = \max_i b_i$. Then, for all $\epsilon > 0$,*

$$\mathbb{P}\left(\left|S_n - \sum_{i=1}^n \mu_i\right| \geq \sqrt{2\sigma^2 t} + bt\right) \leq 2\exp(-t).$$

B.2 Results for Uniform and Half-normal Random Variables

In the next two lemmas, let $\{X_i\}_{i=1}^M$ denote a sequence of M i.i.d random variables and $Y = \max_i X_i$ be their maximum. We note that the results or techniques in Lemmas 21, 22 are not particularly new.

Lemma 21. *Let $X_i \sim \mathrm{Unif}(a, b)$. Then $\mathbb{E}X_i = \theta$ and $\mathbb{E}Y = \theta + \frac{M-1}{M+1}\frac{b-a}{2}$ where $\theta = (a+b)/2$.*

Proof. The proof for $\mathbb{E}X_i$ is straightforward. The cdf of Y is $\mathbb{P}(Y \leq t) = \prod_{i=1}^M \mathbb{P}(X_i \leq t) = (\frac{t-a}{b-a})^M$. Therefore its pdf is $p_Y(t) = M(t-a)^{M-1}/(b-a)^M$ and its expectation is

$$\mathbb{E}[Y] = \int_a^b tM(t-a)^{M-1}/(b-a)^M \mathrm{d}t = \frac{a + bM}{M+1} = \theta + \frac{M-1}{M+1}\frac{b-a}{2}.$$

□

Lemma 22. *Let $X_i \sim \mathcal{HN}(\zeta^2)$. Then $\mathbb{E}X_i = \zeta\sqrt{2/\pi}$ and $\mathbb{E}Y$ satisfies,*

$$\zeta K\sqrt{\log(M)} \leq \mathbb{E}Y \leq \zeta\sqrt{2\log(2M)}.$$

Here K is a universal constant. Therefore, $\mathbb{E}Y \in \Theta(\sqrt{\log(M)})\mathbb{E}X_i$.

Proof. The proof for $\mathbb{E}X_i$ just uses integration over the pdf $p_Y(t) = \frac{\sqrt{2}}{\sqrt{\pi\zeta^2}} e^{-\frac{t^2}{2\sigma^2}}$. For the second part, writing the pdf of $\mathcal{N}(0,\zeta^2)$ as $\phi(t)$ we have,

$$\mathbb{E}[e^{\lambda X_i}] = 2\int_0^\infty e^{\lambda t}\phi(t)\mathrm{d}t \leq 2\int_{-\infty}^\infty e^{\lambda t}\phi(t)\mathrm{d}t = 2\mathbb{E}_{Z\sim\mathcal{N}(0,\zeta^2)}[e^{\lambda Z}] = 2e^{\zeta^2\lambda^2/2}.$$

The inequality in the second step uses that the integrand is positive. Therefore, using Jensen's inequality and the fact that the maximum is smaller than the sum we get,

$$e^{\lambda\mathbb{E}[Y]} \leq \mathbb{E}[e^{\lambda Y}] \leq \sum_{i=1}^n \mathbb{E}[e^{\lambda X_i}] \leq 2Me^{\lambda^2\zeta^2/2} \quad\implies\quad \mathbb{E}[Y] \leq \frac{1}{\lambda}\log(2M) + \frac{\zeta^2\lambda}{2}.$$

Choosing $\lambda = \frac{\sqrt{2\log(2M)}}{\zeta}$ yields the upper bound. The lower bound follows from Lemma 4.10 of Adler [2] which establishes a $K\sqrt{\log(M)}$ lower bound for M i.i.d standard normals $Z_1,\dots,Z_M$. We can use the same lower bound since $|Z_i| \geq Z_i$. □

Lemma 23. *Suppose we complete a sequence of jobs indexed $j = 1, 2, \dots$. The time taken for the jobs $\{X_j\}_{j\geq 1}$ are i.i.d with mean θ and sub-Gaussian parameter τ. Let $\delta \in (0,1)$, and N denote the number of completed jobs after time T. That is, N is the random variable such that $N = \max\{n \geq 1; \sum_{j=1}^n X_j \leq T\}$. Then, with probability greater than $1-\delta$, for all $\alpha \in (0,1)$, there exists $T_{\alpha,\delta}$ such that for all $T > T_{\alpha,\delta}$, $N \in \left(\frac{T}{\theta(1+\alpha)} - 1, \frac{T}{\theta(1-\alpha)}\right)$.*

Proof. We will first consider the total time taken $S_n = \sum_{i=1}^n X_i$ after n evaluations. Let $\epsilon_n = \tau\sqrt{n\log(n^2\pi^2/(3\delta))}$ throughout this proof. Using Lemma 17, we have $\mathbb{P}(|S_n - n\theta| > \epsilon_n) = 6\delta/(n\pi^2)$. By a union bound over all $n \geq 1$, we have that with with probability greater than δ, the following event $\mathcal{E}$ holds.

$$\mathcal{E} = \{\forall n \geq 1,\ \ |S_n - n\theta| \leq \epsilon_n.\} \tag{12}$$

Since $\mathcal{E}$ is a statement about all time steps, it is also for true the random number of completed jobs N. Inverting the condition in (12) and using the definition of N, we have

$$N\theta - \epsilon_N \leq S_N \leq T \leq S_{N+1} \leq (N+1)\theta + \epsilon_{N+1}. \tag{13}$$

Now assume that there exists $T_{\alpha,\delta}$ such that for all $T \geq T_{\alpha,\delta}$ we have, $\epsilon_N \leq N\alpha\theta$. Since ϵ_n is sub-linear in n, it also follows that $\epsilon_{N+1} \leq (N+1)\alpha\theta$. Hence, $N\theta(1-\alpha) \leq T \leq (N+1)\theta(1+\alpha)$ and the result follows.

All that is left to do is to establish that such a $T_{\alpha,\delta}$ exists under event $\mathcal{E}$, for which we will once again appeal to (13). The main intuition is that as $\epsilon_N \asymp \sqrt{N\log(N)}$, the condition $\epsilon_N \leq N\alpha\theta$ is satisfied for N large enough. But N is growing with T, and hence it is satisfied for T large enough. More formally, since $\frac{N}{\epsilon_N} \asymp \frac{N+1}{\epsilon_{N+1}}$ using the upper bound for T it is sufficient to show $\frac{T}{\epsilon_{N+1}} \gtrsim \frac{1}{\alpha\theta}$ for all $T \geq T_{\alpha,\delta}$. But since $\epsilon_{N+1} \asymp \sqrt{N\log(N)}$ and the lower bound for T is $T \gtrsim N$, it is sufficient if $\frac{T}{\sqrt{T\log(T)}} \gtrsim \frac{1}{\alpha\theta}$ for all $T \geq T_{\alpha,\delta}$. This is achievable as the LHS is increasing with T and the RHS is constant. □

Our final result for the uniform and half-normal random variables follows as a consequence of Lemma 23.

Theorem 24. *Let the time taken X for completing an evaluation to f be a random variable.*

- *If $X \sim \mathrm{Unif}(a,b)$, denote $\theta = (a+b)/2$, $\theta_M = \theta + \frac{M-1}{M+1}\frac{b-a}{2}$, and $\tau = (b-a)/2$.*
- *If $X \sim \mathcal{HN}(\tau^2)$, denote $\theta = \zeta\sqrt{2/\pi}$, $\theta_M = \theta\cdot\Theta(\sqrt{\log(M)})$, and $\tau = \zeta\pi/2$.*

Denote the number of evaluations within time T by sequential, synchronous parallel and asynchronous parallel algorithms by $N_{\text{seq}}, N_{\text{syn}}, N_{\text{asy}}$ respectivey. Let $\delta \in (0,1)$. Then, with probability greater than $1-\delta$, for all $\alpha \in (0,1)$, there exists $T_{\alpha,\delta}$ such that for all $T \geq T_{\alpha,\delta}$, we have each of the following,

$$N_{\text{seq}} \in \left(\frac{T}{\theta(1+\alpha)} - 1, \frac{T}{\theta(1-\alpha)}\right), \quad N_{\text{syn}} \in \left(M\left[\frac{T}{\theta_M(1+\alpha)} - 1\right], \frac{MT}{\theta_M(1-\alpha)}\right),$$

$$N_{\text{asy}} \in \left(M\left[\frac{T}{\theta(1+\alpha)} - 1\right], \frac{MT}{\theta(1-\alpha)}\right).$$

Proof. We first show τ sub-Gaussianity of X and $Y = \max_{j=1,\ldots,M} X_j$ when $X, X_1, \ldots, X_M$ are either uniform or half-normal. For the former, both X and Y are $\tau = (b-a)/2$ sub-Gaussian since they are bounded in $[a,b]$. For the Half-normal case, we note that $X = |Z|$ and $Y = \max_{j=1,\ldots,M} |Z_i|$ for some i.i.d. $\mathcal{N}(0,\zeta^2)$ variables $Z, \{Z_i\}_{i=1}^M$. Both are 1-Lipschitz functions of Z_i and $(Z_{i1}, \ldots, Z_{iM})$ respectively and $\tau = \zeta\pi/2$ sub-Gaussianity follows from Lemma 18.

Now in synchronous settings, the algorithm dispatches the k^{th} batch with evaluation times $\{(X_{k1}, \ldots, X_{kM})\}$. It releases its $(k+1)^{\text{th}}$ batch when all evalutions finish after time $Y_k = \max_{i=1,\ldots,M} X_{ki}$. The result for N_{syn} follows by applying Lemma 23 on the sequence $\{Y_k\}_{k\geq 1}$. For the sequential setting, each worker receives its $(k+1)^{\text{th}}$ job after completing its k^{th} evaluation in time X_k. We apply Lemma 23 on the sequence $\{X_k\}_{k\geq 1}$ for one worker to obtain that the number of jobs completed by this worker is $N_{\text{seq}} \in \left(\frac{T}{\theta(1+\alpha)} - 1, \frac{T}{\theta(1-\alpha)}\right)$. In the asynchronous setting, a worker receives his new job immediately after finishing his last. Applying the same argument as the sequential version to all workers but with $\delta \leftarrow \delta/M$ in Lemma 23 and the union bound yields the result for N_{asy}. □

B.3 Results for the Exponential Random Variable

In this section we derive an analogous result to Theorem 24 for the case when the completion times are exponentially distributed. The main challenges stem from analysing the distribution of the maxima of a finite number of exponential random variables. Much of the analysis is based on results from Boucheron and Thomas [5] (See also chapter 6 of Boucheron et al. [6]).

In deviating from the notation used in Table 1, we will denote the parameter of the exponential distribution as θ, i.e. it has pdf $p(x) = \theta x^{-\theta x}$. The following fact about exponential random variables will be instrumental.

Fact 25. *Let $X_1, \ldots, X_n \sim \mathrm{Exp}(\theta)$ iid. Also let $E_1, \ldots, E_n \sim \mathrm{Exp}(\theta)$ iid and independent from X_1^n. If we define the order statistics $X_{(1)} \geq X_{(2)} \geq \ldots \geq X_{(n)}$ for $X_1, \ldots, X_n$, we have*

$$(X_{(n)}, \ldots, X_{(1)}) \sim \left(E_n/n, \ldots, \sum_{k=i}^{n} E_k/k, \ldots, \sum_{k=1}^{n} E_k/k\right).$$

Proof. This is Theorem 2.5 in [5] but we include a simple proof for completeness. We first must analyse the minimum of n exponentially distributed random variables. This is a simple calculation.

$$\mathbb{P}[\min_i X_i \geq t] = \prod_{i=1}^{n} \mathbb{P}[X_i \geq t] = \prod_{i=1}^{n} \exp(-\theta t) = \exp(-n\theta t)$$

This last expression is exactly the probability that an independent $\mathrm{Exp}(n\theta)$ random variables is at least t.

This actually proves the first part, since $E_n/n \sim \mathrm{Exp}(n\theta)$. Now, using the memoryless property, conditioning on $X_{(n)} = x$ and $X_{(n)} = X_i$ for some i, we know that for $j \neq i$

$$\mathbb{P}[X_j \geq x' + x | X_{(n)} = x, X_{(n)} = X_i] = \exp(-\theta x').$$

Removing the conditioning on the index achieving $X_{(n)}$, and using the same calculation for the minimum, we now get

$$\mathbb{P}[X_{(n-1)} \geq x' + x | X_{(n)} = x] = \exp(-(n-1)\theta x')$$

Thus we have that $X_{(n-1)} - X_{(n)} \sim \mathrm{Exp}((n-1)\theta)$. The claim now follows by induction. □
As before the first step of the argument is to understand the expectation of the maximum.

Lemma 26. *Let $X_i \sim \mathrm{Exp}(\theta)$. Then $\mathbb{E}X_i = 1/\theta$ and $\mathbb{E}Y = h_M/\theta$ where $Y = \max_{i=1,\ldots,M} X_i$ is the maximum of the X_i's and $h_M = \sum_{i=1}^{M} i^{-1}$ is the M^{th} harmonic number.*

Proof. Using the relationship between the order statistics and the spacings in Fact 25 we get

$$\mathbb{E}\max_i X_i = \mathbb{E}X_{(1)} = \mathbb{E}\sum_{k=1}^{M} E_k/k = \sum_{k=1}^{M} \frac{1}{k\theta} = \frac{h_M}{\theta}.$$ □

Recall that $h_M \asymp \log(M)$ accounting for the claims made in Table 1 and the subsequent discussion.

While obtaining polynomial concentration is straightforward via Chebyshev's inequality it is insufficient for our purposes, since we will require a union bound over many events. However, we can obtain exponential concentration, although the argument is more complex. Our analysis is based on Herbst's argument, and a modified logarithmic Sobolev inequality, stated below in Theorem 27. To state the inequality, we first define the entropy $\mathrm{Ent}[X]$ of a random variable X as follows (not to be confused with Shannon entropy),

$$\mathrm{Ent}[X] \triangleq \mathbb{E}[X\log(X)] - \mathbb{E}[X]\log(\mathbb{E}[X]).$$

Theorem 27 (Modified logarithmic Sobolev inequality (Theorem 6.6 in [6]))**.** *Let $X_1, \ldots, X_n$ be independent random variables taking values in some space $\mathcal{X}$, $f : \mathcal{X}^n \to \mathbb{R}$, and define the random variable $Z = f(X_1, \ldots, X_n)$. Further let $f_i : \mathcal{X}^{n-1} \to \mathbb{R}$ for $i \in \{1, \ldots, n\}$ be arbitrary functions and $Z_i = f_i(X^{(i)}) = f_i(X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n)$. Finally define $\tau(x) = e^x - x - 1$. Then for all $\lambda \in \mathbb{R}$*

$$\mathrm{Ent}[e^{\lambda Z}] \le \sum_{i=1}^{n} \mathbb{E}[e^{\lambda Z}\tau(-\lambda(Z - Z_i))].$$

Application of the logarithmic Sobolev inequality in our case gives:

Lemma 28. *Let $X_1, \ldots, X_M \sim \mathrm{Exp}(\theta)$ iid, $E \sim \mathrm{Exp}(\theta)$ also independently and define $Z = \max_{i\in\{1,\ldots,M\}} X_i$ and $\mu = \mathbb{E}Z$. Define $\tau(x) = e^x - x - 1$ and $\psi(x) = \exp(x)\tau(-x) = 1 + (x-1)e^x$. Then for any $\lambda \in \mathbb{R}$*

$$\begin{aligned}\mathrm{Ent}[\exp\{\lambda(Z-\mu)\}] &\le \mathbb{E}[\exp\{\lambda(Z-\mu)\}] \times \mathbb{E}\psi(\lambda E),\\ \mathrm{Ent}[\exp\{\lambda(\mu - Z)\}] &\le \mathbb{E}[\exp\{\lambda(\mu - Z)\}] \times \mathbb{E}\tau(\lambda E).\end{aligned}$$

Proof. We apply Theorem 27 with $Z = f(X_1, \ldots, X_M) = \max_i X_i$ and $Z_i = f_i(X^{(i)}) = \max_{j\neq i} X_j$. Notice that in this case, $Z_i = Z$ except when X_i is the maximiser, in which case $Z_i = X_{(2)}$ the second largest of the samples. This applies here since the maximiser is unique with probability 1. Thus, Theorem 27 gives

$$\begin{aligned}\mathrm{Ent}[\exp\{\lambda Z\}] &\le \sum_{i=1}^{M} \mathbb{E}[\exp\{\lambda Z\}\tau(-\lambda(Z - Z_i))] = \mathbb{E}\left[\exp\{\lambda Z\}\tau(-\lambda(X_{(1)} - X_{(2)}))\right]\\ &= \mathbb{E}\left[\exp\{\lambda X_{(2)}\}\exp\{\lambda(X_{(1)} - X_{(2)})\}\tau(-\lambda(X_{(1)} - X_{(2)}))\right]\\ &= \mathbb{E}[\exp\lambda X_{(2)}]\mathbb{E}[\psi(\lambda E)] \le \mathbb{E}[\exp\lambda X_{(1)}]\mathbb{E}[\psi(\lambda E)]\end{aligned}$$

The first inequality is Theorem 27, while the first equality uses the definitions of f_i and the fact that $Z_i \neq Z$ for exactly one index i. The second equality is straightforward and the third uses Fact 25 to write $X_{(1)} - X_{(2)}$ as an independent $\mathrm{Exp}(\theta)$ random variable, which also allows us to split the expectation. Finally since $X_{(2)} \le X_{(1)}$ almost surely, the final inequality follows. Multiplying both sides by $\exp(-\lambda\mu)$, which is non-random, proves the first inequality, since $\mathrm{Ent}(aX) = a\mathrm{Ent}(X)$.

The second inequality is similar. Set $Z = -\max_i X_i$ and $Z_i = -\max_{j\neq i} X_j$ and using the same argument, we get

$$\begin{aligned}\mathrm{Ent}[\exp\{-\lambda X_{(1)}\}] &\le \mathbb{E}[\exp\{-\lambda X_{(1)}\}\tau(\lambda(X_{(1)} - X_{(2)}))]\\ &= \mathbb{E}\left[\exp\left\{-\lambda(E_1 + \sum_{k=2}^{M} E_k/k)\right\}\tau(\lambda E_1)\right]\end{aligned}$$

Here the inequality follows from Theorem 27 and the identity uses Fact 25. We want to split the expectation, and to do so, we use Chebyshev's association principle. Observe that $\exp(-\lambda E_1)$ is clearly non-increasing in E_1 and that $\tau(\lambda E_1)$ is clearly non-decreasing in E_1 for $E_1 \ge 0$ ($E_1 > 0$ a.s.). Hence, we can split the expectation to get

$$\mathrm{Ent}[\exp\{-\lambda X_{(1)}\}] \le \mathbb{E}[\exp\{-\lambda X_{(1)}\}] \times \mathbb{E}[\tau(\lambda E)]$$

The second inequality follows now by multiplying both sides by $\exp(\lambda\mu)$. □

Theorem 29. *Let $X_1, \ldots, X_M \sim \mathrm{Exp}(\theta)$ iid and define $Z = \max_i X_i$ then $Z - \mathbb{E}Z$ is sub-exponential with parameters $(4/\theta^2, 2/\theta)$.*

Proof. We use the logarithmic Sobolev inequality, and proceed with Herbst's method. Unfortunately since our inequality is not in the standard form, we must reproduce most of the argument. However, we can unify the two tails by noticing that we currently have for centered Y (e.g., $Y = X_{(1)} - \mathbb{E}X_{(1)}$ or $Y = \mathbb{E}X_{(1)} - X_{(1)}$),

$$\mathrm{Ent}[\exp\{\lambda Y\}] \leq \mathbb{E}[\exp\{\lambda Y\}]f(\lambda) \tag{14}$$

for some differentiable function f, which involves either τ or ψ depending on the tail. We will use such an inequality to bound the moment generating function of Y.

For notational convenience, define $\phi(\lambda) = \log \mathbb{E}\exp\{\lambda Y\}$ and observe that

$$\phi'(\lambda) = \frac{1}{\lambda}\left(\frac{\mathrm{Ent}[\exp\{\lambda Y\}]}{\mathbb{E}\exp\{\lambda Y\}} + \log \mathbb{E}\exp\{\lambda Y\}\right)$$

Together with the inequality in Eq. (14), this gives

$$\begin{aligned}\lambda\phi'(\lambda) - \phi(\lambda) &= \frac{\mathrm{Ent}[\exp\{\lambda Y\}]}{\mathbb{E}\exp\{\lambda Y\}} \leq f(\lambda)\\ \Leftrightarrow \frac{\phi'(\lambda)}{\lambda} - \frac{\phi(\lambda)}{\lambda^2} &\leq f(\lambda)/\lambda^2, \qquad \forall\, \lambda > 0\end{aligned}$$

Observe now that the left hand side is precisely the derivative of the function $G(\lambda) = \phi(\lambda)/\lambda$. Hence, we can integrate both sides from 0 to λ, we get

$$\frac{\phi(\lambda)}{\lambda} \leq \int_0^\lambda f(t)/t^2 dt.$$

This last step is justified in part by the fact that $\lim_{t\to 0}\phi(t)/t = 0$ by L'Hopital's rule. Thus we have $\log \mathbb{E}\exp\{\lambda Y\} \leq \lambda\int_0^\lambda f(t)/t^2 dt$.

The upper tail: For the upper tail $Z - \mathbb{E}Z$, we have $f(t) = \mathbb{E}\psi(tE)$ where $E \sim \mathrm{Exp}(\theta)$ and $\psi(x) = 1 + (x-1)e^x$. By direct calculation, we have for $t < \theta$

$$\begin{aligned}\mathbb{E}\psi(tE) &= 1 + \mathbb{E}tE\exp(tE) - \mathbb{E}\exp(tE)\\ &= 1 - \frac{\theta}{\theta - t} + t\int_0^\infty x\exp(tx)\theta\exp(-\theta x)dx\\ &= 1 - \frac{\theta}{\theta - t} + \frac{t\theta}{(\theta - t)^2} = \frac{t^2}{(\theta - t)^2}.\end{aligned}$$

Thus, we get

$$\log \mathbb{E}\exp\{\lambda(Z - \mathbb{E}Z)\} \leq \lambda\int_0^\lambda \frac{1}{(\theta - t)^2}dt = \frac{\lambda^2}{\theta(\theta - \lambda)}.$$

If $\lambda \leq \theta/2$, this bound is $2\lambda^2/\theta^2$. Thus, according to definition 19, $Z - \mathbb{E}Z$ is sub-exponential with parameters $(4/\theta^2, 2/\theta)$.

The lower tail: For the lower tail $\mathbb{E}Z - Z$ we need to control $\mathbb{E}\tau(tE)$ where $E \sim \mathrm{Exp}(\theta)$, $\tau(x) = e^x - x - 1$. Direct calculation, using the moment generating function of exponential random variables gives

$$\mathbb{E}\tau(tE) = \frac{t^2}{\theta(\theta - t)}$$

So the integral bound is

$$\begin{aligned}\log \mathbb{E}\exp\{\lambda(\mathbb{E}Z - Z)\} \leq \lambda\int_0^\lambda \frac{1}{\lambda(\lambda - t)} &= \frac{\lambda}{\theta}\log\left(\frac{\theta}{\theta - \lambda}\right)\\ &= \frac{\lambda}{\theta}\left(\sum_{i=1}^\infty (\lambda/\theta)^i/i\right)\\ &= \frac{\lambda^2}{\theta^2}\left(\sum_{i=1}^\infty (\lambda/\theta)^{i-1}/i\right)\end{aligned}$$

If $\lambda/\theta \leq 1/2$ the series inside the paranthesis is clearly bounded by 2. Thus $\mathbb{E}Z - Z$ is sub-exponential with parameters $(4/\theta^2, 2/\theta)$ as before. □

Now that we have established that the maximum is sub-Exponential, we can bound the number of evaluations for the various methods. This is the main result for this section.

Theorem 30. *Let the time taken X for completing an evaluation to f be a random variable that is $\mathrm{Exp}(\theta)$ distributed. Let $\delta \in (0,1)$ and denote N_{syn} and N_{asy} denote the number of evaluations by synchronous and asynchronous algorithms with time T. Then with probability at least $1-\delta$, for any $\alpha \in (0,1)$ there exists $T_{\alpha,\theta}$ such that*

$$N_{\text{seq}} \in \left(\frac{T\theta}{(1+\alpha)} - 1, \frac{MT\theta}{(1-\alpha)}\right), \qquad N_{\text{syn}} \in \left(M\left(\frac{T\theta}{h_M(1+\alpha)} - 1\right), \frac{MT\theta}{h_M(1-\alpha)}\right)$$
$$N_{\text{asy}} \in \left(M\left(\frac{T\theta}{(1+\alpha)} - 1\right), \frac{MT\theta}{(1-\alpha)}\right)$$

Proof. In the synchronous setting, the k^{th} batch issues M jobs with lengths $(X_{k1}, \dots, X_{kM})$ and the batch ends after $Y_k = \max_i X_{ki}$. Since the sequence of random variables $\{Y_k\}_{k\geq 1}$ are all iid and sub-exponential with parameters $(4/\theta^2, 2/\theta)$, in a similar way to the proof of Lemma 23, with $S_n = \sum_{k=1}^n Y_k$ we get that

$$\mathbb{P}\left(\exists n;\ |S_n - \mathbb{E}S_n| \geq \underbrace{\sqrt{8n\theta^{-2}\log(n^2\pi^2/(3\delta))} + \frac{2}{\theta}\log(n^2\pi^2/(3\delta))}_{\triangleq \epsilon_n}\right) \leq \delta$$

This follows from Bernstein's inequality (Proposition 20) and the union bound. As in Lemma 23 this means that:

$$\frac{Nh_M}{\theta} - \epsilon_N \leq S_N \leq T \leq S_{N+1} \leq \frac{(N+1)h_M}{\theta} + \epsilon_{N+1}.$$

Here we also used the fact that $\mathbb{E}Y_k = h_M/\theta$ from Lemma 26. Now assuming there exists $T_{\alpha,\delta}$ such that for all $T \geq T_{\alpha,\delta}$, we have $\epsilon_N \leq Nh_M\alpha/\theta$, we get

$$\frac{Nh_M}{\theta}(1-\alpha) \leq T \leq \frac{(N+1)h_m}{\theta}(1+\alpha).$$

The existence of $T_{\alpha,\delta}$ is based on the same argument as in Lemma 23. Re-arranging these inequalities, which gives a bound on the number of batches completed, leads to the bounds on the number of evaluations for the synchronous case.

Applying the same argument to a single worker on the sequence $\{X_k\}_{k\geq 1}$, we get

$$\frac{N_{\text{seq}}}{\theta}(1-\alpha) \leq T \leq \frac{N_{\text{seq}}+1}{\theta}(1+\alpha).$$

Repeating this argument for all M workers with $\delta \leftarrow \delta/M$ and then taking a union bound yields the result for N_{asy}. □

B.4 Putting it altogether

Finally, we put the results in Theorems 11, 14, 24 and 30 together to obtain the following result. This is a formal version of Theorem 5 in the main text.

Theorem 31. *Let $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$ where $\kappa : \mathcal{X}^2 \to \mathbb{R}$ satisfies Assumption 8 and $\kappa(x,x') \leq 1$. Then for all $\alpha > 0$, the Bayes simple regret for* seqTS, synTS *and* asyTS, *satisfies the following for sufficiently large T.*

$$\text{seqTS:} \qquad \mathrm{BSR}'(T) \leq \frac{C_1'}{n_{\text{seq}}} + \sqrt{\frac{C_2\beta_{n_{\text{seq}}}\Psi_{n_{\text{seq}}}}{n_{\text{seq}}}}, \qquad n_{\text{seq}} = \frac{T}{\theta(1+\alpha)} - 1$$
$$\text{synTS:} \qquad \mathrm{BSR}'(T) \leq \frac{C_1'}{n_{\text{syn}}} + \frac{(M-1)\beta_{M-1}^{1/2}}{n_{\text{syn}}} + \sqrt{\frac{C_2\beta_{n_{\text{syn}}+M}\Psi_{n_{\text{syn}}}}{n_{\text{syn}}}}, \qquad n_{\text{syn}} = M\left[\frac{T}{\theta_M(1+\alpha)} - 1\right]$$
$$\text{asyTS:} \qquad \mathrm{BSR}'(T) \leq \frac{C_1'}{n_{\text{asy}}} + \frac{2\,\Xi\,\gamma_M}{n_{\text{asy}}} + \sqrt{\frac{C_2C_\kappa\beta_{n_{\text{asy}}}\Psi_{n_{\text{asy}}}}{n_{\text{asy}}}}, \qquad n_{\text{asy}} = M\left[\frac{T}{\theta(1+\alpha)} - 1\right]$$

Here, θ, θ_M are defined as follows for the uniform, half-normal and exponential cases.

$$\begin{aligned}
\text{Unif}(a,b): \qquad & \theta = \frac{a+b}{2}, \qquad && \theta_M = \frac{a+bM}{M+1} \\
\mathcal{HN}(\zeta^2): \qquad & \theta = \frac{\zeta\sqrt{2}}{\sqrt{\pi}}, \qquad && \theta_M \in \zeta \cdot \Theta(\log(M)) \\
\text{Exp}(\lambda): \qquad & \theta = \frac{1}{\lambda}, \qquad && \theta_M = \frac{h_M}{\lambda}
\end{aligned}$$

Further, Ψ_n is the MIG in Definition 6, β_n is as defined in (5), ξ_M is from (4), and $C_1 = \pi^2/6 + \sqrt{2\pi}/12 + 1$, $C_2 = 2/\log(1+\eta^{-2})$ are constants.

Proof. We will prove the result for asyTS as the others are obtained by an identical argument. Let $\mathcal{V}$ denote the event that $N \geq M\big[\frac{T}{\theta(1+\alpha)} - 1\big]$. Theorems 24 and 30 give us control on this event with probability at least $1-\delta$. We will choose $\delta = \frac{1}{2\,\Xi\, n_{\text{asy}}}$ where Ξ is the expected maximum of the GP in Lemma 9. Since the randomness in the evaluation times are independent of the prior, noise and the algorithm, we can decompose $\text{BSR}'(T)$ as follows and use the result in (11) for $\text{BSR}(n)$.

$$\begin{aligned}
\text{BSR}'(T) &\leq \mathbb{E}[\text{BSR}(N)|\mathcal{V}]\mathbb{P}(\mathcal{V}) + \mathbb{E}[\text{BSR}(N)|\mathcal{V}^c]\mathbb{P}(\mathcal{V}^c) \\
&\leq \text{BSR}(n_{\text{asy}}) \cdot 1 \;+\; 2\,\Xi\,\delta
\end{aligned}$$

Here we have used the definition of the Bayes simple regret with time in (3) which guarantees that it is never worse than $\sup_x |f(x_\star) - f(x)| \leq 2\,\Xi$. The theorem follows by plugging in values for $\text{BSR}(n)$ and δ. The "sufficiently large T" requirement is because Theorems 24 and 30 hold only for $T > T_{\alpha,\delta} = T_{\alpha, \frac{1}{2\,\Xi\, n_{\text{asy}}}}$. Since the dependencies of δ on $T_{\alpha,\delta}$ is polylogarithmic, and as n_{asy} is growing linearly with T, the above condition is equivalent to $T \gtrsim \text{polylog}(T)$ which is achievable for large enough T. □

C Addendum to Experiments

C.1 Implementation Details for BO methods

We describe some implementation details for all BO methods below.

1. **Domain:** Given a problem with an arbitrary d dimensional domain, we map it to $[0,1]^d$ by linearly transforming each coordinate.

2. **Initialisation:** All BO methods were initialised by uniformly randomly picking n_{init} points in the domain. To facilitate a fair comparison with the random strategies, we also afford them with the same initialisation, and begin our comparisons in the figures after the initialisation.

3. **GP kernel and other hyperarameters:** The GP hyper-parameters are first learned by maximising the marginal likelihood [34] after the initialisation phase and then updated every 25 iterations. For all BO methods, we use a SE kernel and tune the bandwidth for each dimension, the scale parameter of the kernel and the GP noise variance (η^2). The mean of the GP is set to be the median of all observations.

4. **UCB methods:** Depending on the methods used, the UCB criterion typically takes a form $\mu + \beta_j^{1/2}\sigma$ where μ, σ are the posterior mean and standard deviation of the GP. $\beta_j^{1/2}$ is a parameter that controls the exploration exploitation trade-off in UCB methods. Following recommendations in [24], we set it $\beta_j = 0.2d\log(2j+1)$.

5. **Selection of x_j:** In all BO methods, the selection of x_j typically takes the form $x_j = \text{argmax}_x\, \varphi_j(x)$ where φ_j is a function of the GP posterior at step j. φ_j is usually called the acquisition in the BO literature and its maximisation is, in general, an intractable computational problem. For TS, the acquisition is simply the random sample drawn in line 4 of Algorithm 1. For all BO methods, at each iteration, we randomly sample $10d^2j$ points, where d is the dimension and $j-1$ is the number of completed evaluations, and pick the point with the largest $\varphi_j(x)$ value. We found this strategy to be quite robust to model misspecification, particularly given the GP kernel learning procedure in 3. See [4, 20, 40] for more methods for selecting x_j.

On maximising the acquisition:
While selecting maximiser of φ_j from a random or pre-fixed set of points is common in the BO literature, some deterministic criteria also use adaptive branch and bound methods or gradient based methods. However, using such methods can be problematic for TS since an adaptive procedure would require conditioning on the $j-1$ points where we have already evaluated f and the s points where we have already evaluated the random sample – resulting in a complexity that grows $O((j+s)^3)$. In practice, we found that picking the maximum from a random set of discrete points as mentioned above in 5 worked consistently well in practice for *all* BO methods, as it was robust to model misspecification, particularly given the GP kernel learning procedure in 3.

It is worth reiterating that the cruicial advantage to the parallelised TS algorithms we propose and study is that they do not scale with the number of workers M, making it now feasible to scale to a large number of workers, unlike previous methods. By making progress on maximising the sample in TS, such as some recent work by Bijl et al. [4], we believe one could achieve a computationally attractive algorithm for large scale parallel BO.

C.2 Synthetic Experiments

Additional Experiments: In Figures 4 and 5 we present results on additional synthetic experiments and also repeat those in the main text in larger figures. The last panel of Figure 5 compares seqTS, synTS, and asyTS on the Park1 function.

We describe the construction of the synthetic experiments below. All the design choices were made arbitrarily.

Construction of benchmarks: To construct our test functions, we start with the following benchmarks for global optimisation commonly used in the literature: Branin $(d=2)$, Currin-exponential $(d=2)$, Hartmann3 $(d=3)$, Park1 $(d=4)$, Park2 $(d=4)$, and Hartmann6 $(d=6)$. The descriptions of these functions are available in, for e.g. [25]. To construct the high dimensional variants, we repeat the same function by cycling through different groups of coordinates and add them up. For e.g. the Hartmann12 function was constructed as $f(x_{1:12}) = g(x_{1:6}) + g(x_{7:12})$ where g is the Hartmann6 function. Similarly, for the Park2-16 function we used the Park2-function 4 times, for Hartmann18, we used Hartmann6 thrice, and for CurrinExp-14 we used the Currin-exponential function 7 times.

Noise: To reflect the bandit setting, we added Gaussian noise with standard deviation η in our experiments. We used $\eta = 0.2$ for CurrinExp, Branin, Park1, Park2, Hartmann3, and Hartmann6; $\eta = 1$ for Park2, Park2-16, Hartmann12, CurrinExp-14, and Hartmann18. The two choices were to reflect the “scale” of variability of the function values themselves on each test problem.

Time distributions: The time distributions are indicated on the top of each figure. In all cases, the time distributions were constructed so that the expected time to complete one evaluation is 1 time unit. Therefore, for e.g. in the Hartmann6 problem, an asynchronous version would use roughly $12 \times 30 = 360$ evaluations while a synchronous version would use roughly $\frac{12\times 30}{\log(8)} \approx 173$ evaluations.

C.3 Cifar-10 Experiment

In the Cifar-10 experiment we use a 6 layer convolutional neural network. The first 5 layers use convolutional filters while the last layer is a fully connected layer. We use skip connections [14] between the first and third layers and then the third and fifth layers; when doing so, instead of just using an indentity transformation $\phi(x) = x$, we use a linear transformation $\phi(x) = Wx$ as the number of filters could be different at the beginning and end of a skip connection. The weights of W are also learned via back propagation as part of the training procedure. This modification to the Resnet was necessary in our set up as we are tuning the number of filters at each layer.

The following are ranges for the number of evaluations for each method over 9 experiments:
synchronous: synBUCB: 56 - 68, synTS: 56 - 68.
asynchronous: asyRAND: 93 - 105, asyEI: 83 - 92, asyHUCB: 85 - 92, asyTS: 80 - 88.

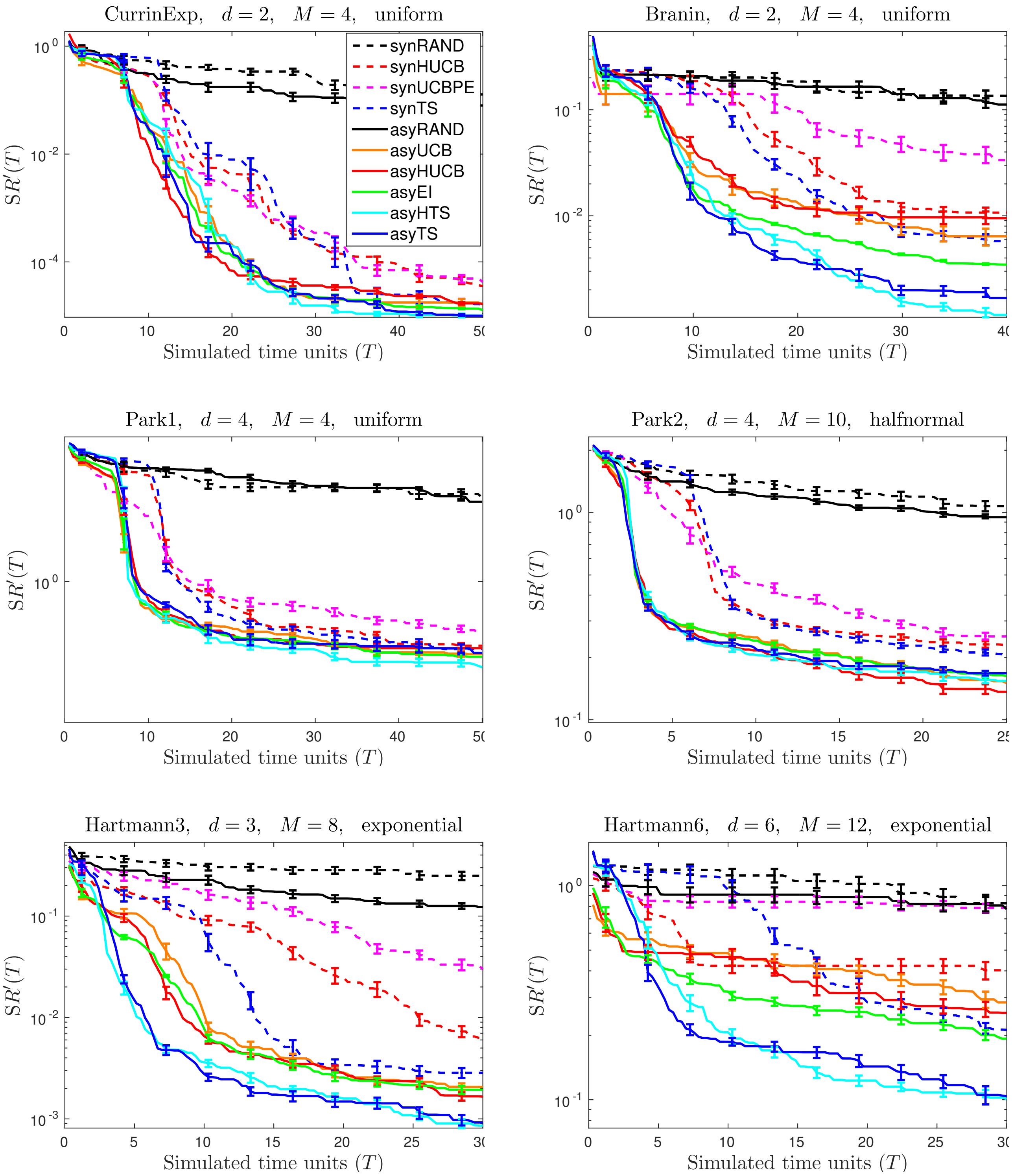


Figure 4: Results on the synthetic experiments. The title states the function used, its dimensionality d, the number of workers M and the distribution used for the time. All distributions were constructed so that the expected time for one evaluation was one time unit (for e.g., in the half normal $\mathcal{HN}(\zeta^2)$ in Table 1, we used $\zeta = \sqrt{\pi/2}$). The dotted lines depict synchronous methods while the solid lines are for asynchronous methods. The error bars indicate one standard error. All figures were averaged over at least 15 experiments.

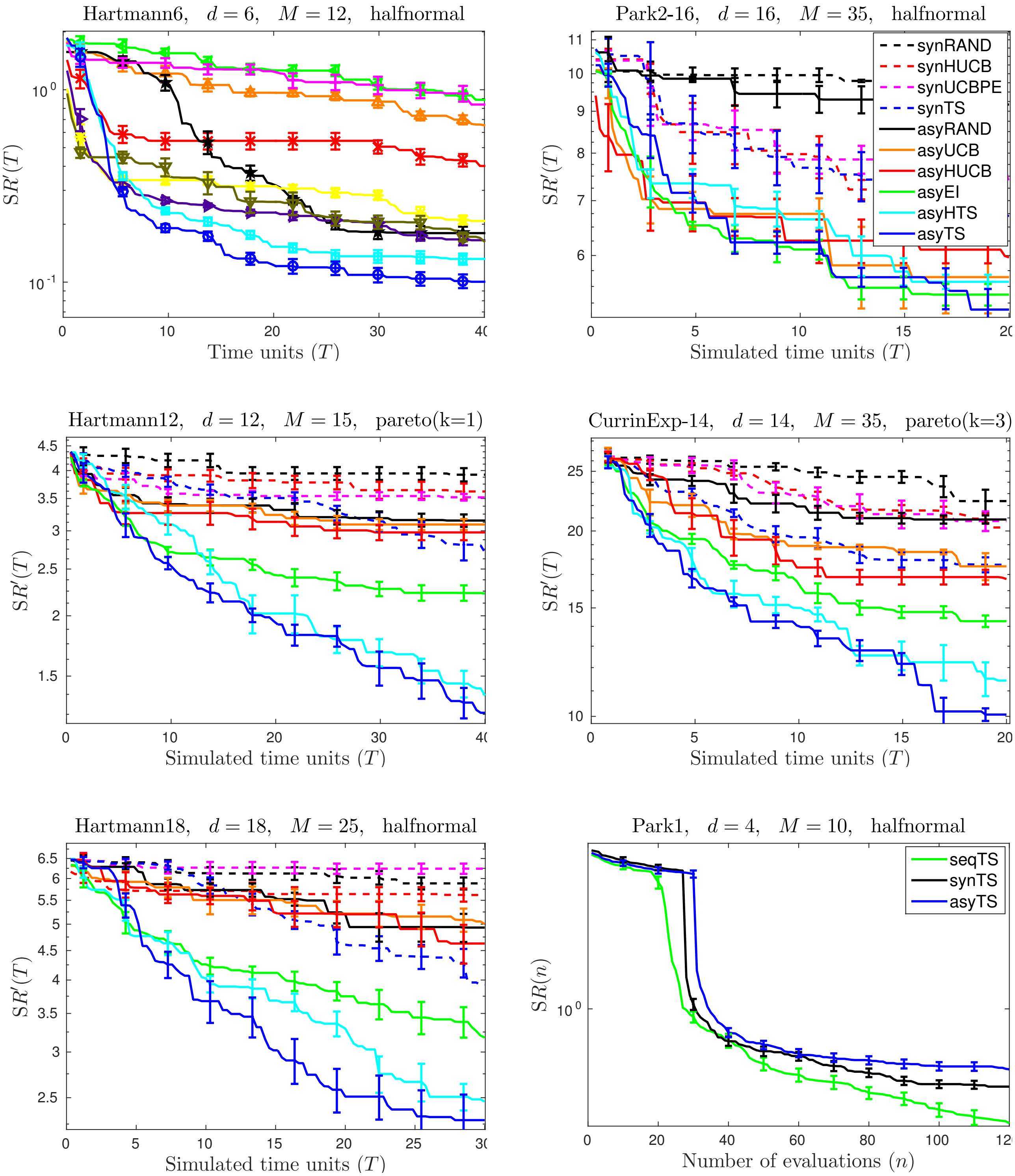


Figure 5: The first five panels are results on synthetic experiments. See caption under Figure 4 for more details. The last panel compares seqTS, synTS, and asyTS against the number of evaluations n.