
Distributed Batch Gaussian Process Optimization

Erik A. Daxberger¹ Bryan Kian Hsiang Low²

Abstract

This paper presents a novel *distributed batch Gaussian process upper confidence bound* (DB-GP-UCB) algorithm for performing batch *Bayesian optimization* (BO) of highly complex, costly-to-evaluate black-box objective functions. In contrast to existing batch BO algorithms, DB-GP-UCB can jointly optimize a batch of inputs (as opposed to selecting the inputs of a batch one at a time) while still preserving scalability in the batch size. To realize this, we generalize GP-UCB to a new batch variant amenable to a Markov approximation, which can then be naturally formulated as a multi-agent distributed constraint optimization problem in order to fully exploit the efficiency of its state-of-the-art solvers for achieving linear time in the batch size. Our DB-GP-UCB algorithm offers practitioners the flexibility to trade off between the approximation quality and time efficiency by varying the Markov order. We provide a theoretical guarantee for the convergence rate of DB-GP-UCB via bounds on its cumulative regret. Empirical evaluation on synthetic benchmark objective functions and a real-world optimization problem shows that DB-GP-UCB outperforms the state-of-the-art batch BO algorithms.

1. Introduction

Bayesian optimization (BO) has recently gained considerable traction due to its capability of finding the global maximum of a highly complex (e.g., non-convex, no closed-form expression nor derivative), noisy black-box objective

function with a limited budget of (often costly) function evaluations, consequently witnessing its use in an increasing diversity of application domains such as robotics, environmental sensing/monitoring, automatic machine learning, among others (Brochu et al., 2010; Shahriari et al., 2016). A number of acquisition functions (e.g., probability of improvement or *expected improvement* (EI) over the currently found maximum (Brochu et al., 2010), entropy-based (Villemonteix et al., 2009; Hennig & Schuler, 2012; Hernández-Lobato et al., 2014), and *upper confidence bound* (UCB) (Srinivas et al., 2010)) have been devised to perform BO: They repeatedly select an input for evaluating/querying the black-box function (i.e., until the budget is depleted) that intuitively trades off between sampling where the maximum is likely to be given the current, possibly imprecise belief of the function modeled by a *Gaussian process* (GP) (i.e., exploitation) vs. improving the GP belief of the function over the entire input domain (i.e., exploration) to guarantee finding the global maximum.

The rapidly growing affordability and availability of hardware resources (e.g., computer clusters, sensor networks, robot teams/swarms) have motivated the recent development of BO algorithms that can repeatedly select a *batch* of inputs for querying the black-box function in *parallel* instead. Such batch/parallel BO algorithms can be classified into two types: On one extreme, batch BO algorithms like *multi-points EI* (*q*-EI) (Chevalier & Ginsbourger, 2013), *parallel predictive entropy search* (PPES) (Shah & Ghahramani, 2015), and the *parallel knowledge gradient method* (*q*-KG) (Wu & Frazier, 2016) jointly optimize the batch of inputs and hence scale poorly in the batch size. On the other extreme, greedy batch BO algorithms (Azimi et al., 2010; Contal et al., 2013; Desautels et al., 2014; González et al., 2016) boost the scalability by selecting the inputs of the batch one at a time. We argue that such a highly sub-optimal approach to gain scalability is an overkill: In practice, each function evaluation is often much more computationally and/or economically costly (e.g., hyperparameter tuning for deep learning, drug testing on human subjects), which justifies dedicating more time to obtain better BO performance. In this paper, we show that it is in fact possible to jointly optimize the batch of inputs and still preserve scalability in the batch size by giving practitioners the flexibility to trade off BO performance for time efficiency.

¹Ludwig-Maximilians-Universität, Munich, Germany. A substantial part of this research was performed during his student exchange program at the National University of Singapore under the supervision of Bryan Kian Hsiang Low and culminated in his Bachelor's thesis. ²Department of Computer Science, National University of Singapore, Republic of Singapore. Correspondence to: Bryan Kian Hsiang Low <lowkh@comp.nus.edu.sg>.

To achieve this, we first observe that, interestingly, batch BO can be perceived as a cooperative multi-agent decision making problem whereby each agent optimizes a separate input of the batch while coordinating with the other agents doing likewise. To the best of our knowledge, this has not been considered in the BO literature. In particular, if batch BO can be framed as some known class of multi-agent decision making problems, then it can be solved efficiently and scalably by the latter’s state-of-the-art solvers. The key technical challenge would therefore be to investigate how batch BO can be cast as one of such to exploit its advantage of scalability in the number of agents (hence, batch size) while at the same time theoretically guaranteeing the resulting BO performance.

To tackle the above challenge, this paper presents a novel distributed batch BO algorithm (Section 3) that, in contrast to greedy batch BO algorithms (Azimi et al., 2010; Contal et al., 2013; Desautels et al., 2014; González et al., 2016), can jointly optimize a batch of inputs and, unlike the batch BO algorithms (Chevalier & Ginsbourger, 2013; Shah & Ghahramani, 2015; Wu & Frazier, 2016), still preserve scalability in the batch size. To realize this, we generalize GP-UCB (Srinivas et al., 2010) to a new batch variant amenable to a Markov approximation, which can then be naturally formulated as a multi-agent *distributed constraint optimization problem* (DCOP) in order to fully exploit the efficiency of its state-of-the-art solvers for achieving linear time in the batch size. Our proposed *distributed batch GP-UCB* (DB-GP-UCB) algorithm offers practitioners the flexibility to trade off between the approximation quality and time efficiency by varying the Markov order. We provide a theoretical guarantee for the convergence rate of our DB-GP-UCB algorithm via bounds on its cumulative regret. We empirically evaluate the cumulative regret incurred by our DB-GP-UCB algorithm and its scalability in the batch size on synthetic benchmark objective functions and a real-world optimization problem (Section 4).

2. Problem Statement, Background, and Notations

Consider the problem of sequentially optimizing an unknown objective function $f : \mathcal{D} \rightarrow \mathbb{R}$ where $\mathcal{D} \subset \mathbb{R}^d$ denotes a domain of d -dimensional input feature vectors. We consider the domain to be discrete as it is known how to generalize results to a continuous, compact domain via suitable discretizations (Srinivas et al., 2010). In each iteration $t = 1, \dots, T$, a batch $\mathcal{D}_t \subset \mathcal{D}$ of inputs is selected for evaluating/querying f to yield a corresponding column vector $\mathbf{y}_{\mathcal{D}_t} \triangleq (y_{\mathbf{x}})_{\mathbf{x} \in \mathcal{D}_t}^\top$ of noisy observed outputs $y_{\mathbf{x}} \triangleq f(\mathbf{x}) + \epsilon$ with i.i.d. Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ and noise variance σ_n^2 .

Regret. Supposing our goal is to get close to the global maximum $f(\mathbf{x}^*)$ as rapidly as possible where $\mathbf{x}^* \triangleq \arg \max_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x})$, this can be achieved by minimizing a standard batch BO objective such as the *batch* or *full cumulative regret* (Contal et al., 2013; Desautels et al., 2014): The notion of regret intuitively refers to a loss in reward from not knowing \mathbf{x}^* beforehand. Formally, the instantaneous regret incurred by selecting a single input \mathbf{x} to evaluate its corresponding f is defined as $r_{\mathbf{x}} \triangleq f(\mathbf{x}^*) - f(\mathbf{x})$. Assuming a fixed cost of evaluating f for every possible batch \mathcal{D}_t of the same size, the batch and full cumulative regrets are, respectively, defined as sums (over iteration $t = 1, \dots, T$) of the smallest instantaneous regret incurred by any input within every batch \mathcal{D}_t , i.e., $R_T \triangleq \sum_{t=1}^T \min_{\mathbf{x} \in \mathcal{D}_t} r_{\mathbf{x}}$, and of the instantaneous regrets incurred by all inputs of every batch \mathcal{D}_t , i.e., $R'_T \triangleq \sum_{t=1}^T \sum_{\mathbf{x} \in \mathcal{D}_t} r_{\mathbf{x}}$. The convergence rate of a batch BO algorithm can then be assessed based on some upper bound on the average regret R_T/T or R'_T/T (Section 3) since the currently found maximum after T iterations is no further away from $f(\mathbf{x}^*)$ than R_T/T or R'_T/T . It is desirable for a batch BO algorithm to asymptotically achieve *no regret*, i.e., $\lim_{T \rightarrow \infty} R_T/T = 0$ or $\lim_{T \rightarrow \infty} R'_T/T = 0$, implying that it will eventually converge to the global maximum.

Gaussian Processes (GPs). To guarantee no regret (Section 3), the unknown objective function f is modeled as a sample of a GP. Let $\{f(\mathbf{x})\}_{\mathbf{x} \in \mathcal{D}}$ denote a GP, that is, every finite subset of $\{f(\mathbf{x})\}_{\mathbf{x} \in \mathcal{D}}$ follows a multivariate Gaussian distribution (Rasmussen & Williams, 2006). Then, the GP is fully specified by its *prior* mean $m_{\mathbf{x}} \triangleq \mathbb{E}[f(\mathbf{x})]$ and covariance $k_{\mathbf{x}\mathbf{x}'}$ $\triangleq \text{cov}[f(\mathbf{x}), f(\mathbf{x}')] for all $\mathbf{x}, \mathbf{x}' \in \mathcal{D}$, which, for notational simplicity (and w.l.o.g.), are assumed to be zero, i.e., $m_{\mathbf{x}} = 0$, and bounded, i.e., $k_{\mathbf{x}\mathbf{x}'} \leq 1$, respectively. Given a column vector $\mathbf{y}_{\mathcal{D}_{1:t-1}} \triangleq (y_{\mathbf{x}})_{\mathbf{x} \in \mathcal{D}_{1:t-1}}^\top$ of noisy observed outputs for some set $\mathcal{D}_{1:t-1} \triangleq \mathcal{D}_1 \cup \dots \cup \mathcal{D}_{t-1}$ of inputs after $t-1$ iterations, a GP model can perform probabilistic regression by providing a predictive distribution $p(\mathbf{f}_{\mathcal{D}_t} | \mathbf{y}_{\mathcal{D}_{1:t-1}}) = \mathcal{N}(\mu_{\mathcal{D}_t}, \Sigma_{\mathcal{D}_t \mathcal{D}_t})$ of the latent outputs $\mathbf{f}_{\mathcal{D}_t} \triangleq (f(\mathbf{x}))_{\mathbf{x} \in \mathcal{D}_t}^\top$ for any set $\mathcal{D}_t \subseteq \mathcal{D}$ of inputs selected in iteration t with the following *posterior* mean vector and covariance matrix:$

$$\begin{aligned} \mu_{\mathcal{D}_t} &\triangleq K_{\mathcal{D}_t \mathcal{D}_{1:t-1}} (K_{\mathcal{D}_{1:t-1} \mathcal{D}_{1:t-1}} + \sigma_n^2 I)^{-1} \mathbf{y}_{\mathcal{D}_{1:t-1}}, \\ \Sigma_{\mathcal{D}_t \mathcal{D}_t} &\triangleq K_{\mathcal{D}_t \mathcal{D}_t} - K_{\mathcal{D}_t \mathcal{D}_{1:t-1}} (K_{\mathcal{D}_{1:t-1} \mathcal{D}_{1:t-1}} + \sigma_n^2 I)^{-1} K_{\mathcal{D}_{1:t-1} \mathcal{D}_t} \end{aligned} \quad (1)$$

where $K_{\mathcal{B}\mathcal{B}'} \triangleq (k_{\mathbf{x}\mathbf{x}'})_{\mathbf{x} \in \mathcal{B}, \mathbf{x}' \in \mathcal{B}'}$ for all $\mathcal{B}, \mathcal{B}' \subset \mathcal{D}$.

GP-UCB and its Greedy Batch Variants. Inspired by the UCB algorithm for the multi-armed bandit problem, the GP-UCB algorithm (Srinivas et al., 2010) selects, in each iteration, an input $\mathbf{x} \in \mathcal{D}$ for evaluating/querying f that trades off between sampling close to an expected maximum (i.e., with large posterior mean $\mu_{\{\mathbf{x}\}}$) given the current GP belief of f (i.e., exploitation) vs. that of high predictive un-

certainty (i.e., with large posterior variance $\Sigma_{\{\mathbf{x}\}|\{\mathbf{x}\}}$) to improve the GP belief of f over \mathcal{D} (i.e., exploration), that is, $\max_{\mathbf{x} \in \mathcal{D}} \mu_{\{\mathbf{x}\}} + \beta_t^{1/2} \Sigma_{\{\mathbf{x}\}|\{\mathbf{x}\}}^{1/2}$ where the parameter $\beta_t > 0$ is set to trade off between exploitation vs. exploration for bounding its cumulative regret.

Existing generalizations of GP-UCB such as *GP batch UCB* (GP-BUCB) (Desautels et al., 2014) and *GP-UCB with pure exploration* (GP-UCB-PE) (Contal et al., 2013) are greedy batch BO algorithms that select the inputs of the batch one at a time (Section 1). Specifically, to avoid selecting the same input multiple times within a batch (hence reducing to GP-UCB), they update the posterior variance (but not the posterior mean) after adding each input to the batch, which can be performed prior to evaluating its corresponding f since the posterior variance is independent of the observed outputs (1). They differ in that GP-BUCB greedily adds each input to the batch using GP-UCB (without updating the posterior mean) while GP-UCB-PE selects the first input using GP-UCB and each remaining input of the batch by maximizing only the posterior variance (i.e., pure exploration). Similarly, a recently proposed UCB-DPP-SAMPLE algorithm (Kathuria et al., 2016) selects the first input using GP-UCB and the remaining inputs by sampling from a *determinantal point process* (DPP). Like GP-BUCB, GP-UCB-PE, and UCB-DPP-SAMPLE, we can theoretically guarantee the convergence rate of our DB-GP-UCB algorithm, which, from a theoretical point of view, signifies an advantage of GP-UCB-based batch BO algorithms over those (e.g., q -EI and PPES) inspired by other acquisition functions such as EI and PES. Unlike these greedy batch BO algorithms (Contal et al., 2013; Desautels et al., 2014), our DB-GP-UCB algorithm can jointly optimize the batch of inputs while still preserving scalability in batch size by casting as a DCOP to be described next.

Distributed Constraint Optimization Problem (DCOP).

A DCOP can be defined as a tuple $(\mathcal{X}, \mathcal{V}, \mathcal{A}, h, \mathcal{W})$ that comprises a set \mathcal{X} of input random vectors, a set \mathcal{V} of $|\mathcal{X}|$ corresponding finite domains (i.e., a separate domain for each random vector), a set \mathcal{A} of agents, a function $h : \mathcal{X} \rightarrow \mathcal{A}$ assigning each input random vector to an agent responsible for optimizing it, and a set $\mathcal{W} \triangleq \{w_n\}_{n=1, \dots, N}$ of non-negative payoff functions such that each function w_n defines a constraint over only a subset $\mathcal{X}_n \subseteq \mathcal{X}$ of input random vectors and represents the joint payoff that the corresponding agents $\mathcal{A}_n \triangleq \{h(\mathbf{x}) | \mathbf{x} \in \mathcal{X}_n\} \subseteq \mathcal{A}$ achieve. Solving a DCOP involves finding the input values of \mathcal{X} that maximize the sum of all functions w_1, \dots, w_n (i.e., social welfare maximization), that is, $\max_{\mathcal{X}} \sum_{n=1}^N w_n(\mathcal{X}_n)$. To achieve a truly decentralized solution, each agent can only optimize its local input random vector(s) based on the assignment function h but communicate with its neighboring agents: Two agents are considered neighbors if there is a function/constraint involving input random vectors that

the agents have been assigned to optimize. Complete and approximation algorithms exist for solving a DCOP; see (Chapman et al., 2011; Leite et al., 2014) for reviews of such algorithms.

3. Distributed Batch GP-UCB (DB-GP-UCB)

A straightforward generalization of GP-UCB (Srinivas et al., 2010) to jointly optimize a batch of inputs is to simply consider summing the GP-UCB acquisition function over all inputs of the batch. This, however, results in selecting the same input $|\mathcal{D}_t|$ times within a batch, hence reducing to GP-UCB, as explained earlier in Section 2. To resolve this issue but not suffer from the suboptimal behavior of greedy batch BO algorithms such as GP-BUCB (Desautels et al., 2014) and GP-UCB-PE (Contal et al., 2013), we propose a batch variant of GP-UCB that jointly optimizes a batch of inputs in each iteration $t = 1, \dots, T$ according to

$$\max_{\mathcal{D}_t \subset \mathcal{D}} \mathbf{1}^\top \mu_{\mathcal{D}_t} + \alpha_t^{1/2} \mathbb{I}[\mathbf{f}_{\mathcal{D}}; \mathbf{y}_{\mathcal{D}_t} | \mathbf{y}_{\mathcal{D}_{1:t-1}}]^{1/2} \quad (2)$$

where the parameter $\alpha_t > 0$, which performs a similar role to that of β_t in GP-UCB, is set to trade off between exploitation vs. exploration for bounding its cumulative regret (Theorem 1) and the conditional mutual information¹ $\mathbb{I}[\mathbf{f}_{\mathcal{D}}; \mathbf{y}_{\mathcal{D}_t} | \mathbf{y}_{\mathcal{D}_{1:t-1}}]$ can be interpreted as the information gain on f over \mathcal{D} (i.e., equivalent to $\mathbf{f}_{\mathcal{D}} \triangleq (f(\mathbf{x}))_{\mathbf{x} \in \mathcal{D}}^\top$) by selecting the batch \mathcal{D}_t of inputs for evaluating/querying f given the noisy observed outputs $\mathbf{y}_{\mathcal{D}_{1:t-1}}$ from the previous $t - 1$ iterations. So, in each iteration t , our proposed batch GP-UCB algorithm (2) selects a batch $\mathcal{D}_t \subset \mathcal{D}$ of inputs for evaluating/querying f that trades off between sampling close to expected maxima (i.e., with a large sum of posterior means $\mathbf{1}^\top \mu_{\mathcal{D}_t} = \sum_{\mathbf{x} \in \mathcal{D}_t} \mu_{\{\mathbf{x}\}}$) given the current GP belief of f (i.e., exploitation) vs. that yielding a large information gain $\mathbb{I}[\mathbf{f}_{\mathcal{D}}; \mathbf{y}_{\mathcal{D}_t} | \mathbf{y}_{\mathcal{D}_{1:t-1}}]$ on f over \mathcal{D} to improve its GP belief (i.e., exploration). It can be derived that $\mathbb{I}[\mathbf{f}_{\mathcal{D}}; \mathbf{y}_{\mathcal{D}_t} | \mathbf{y}_{\mathcal{D}_{1:t-1}}] = 0.5 \log |I + \sigma_n^{-2} \Sigma_{\mathcal{D}_t \mathcal{D}_t}|$ (Appendix A), which implies that the exploration term in (2) can be maximized by spreading the batch \mathcal{D}_t of inputs far apart to achieve large *posterior* variance individually and small magnitude of *posterior* covariance between them to encourage diversity.

Unfortunately, our proposed batch variant of GP-UCB (2) involves evaluating prohibitively many batches of inputs (i.e., exponential in the batch size), hence scaling poorly in the batch size. However, we will show in this section that our batch variant of GP-UCB is, interestingly, amenable to a Markov approximation, which can then be naturally formulated as a multi-agent DCOP in order to fully exploit the

¹In contrast to the BO algorithm of Contal et al. (2014) that also uses mutual information, our work here considers *batch* BO by exploiting the correlation information between inputs of a batch in our acquisition function in (2) to encourage diversity.

efficiency of its state-of-the-art solvers for achieving linear time in the batch size.

Markov Approximation. The key idea is to design the structure of a matrix $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}$ whose log-determinant can closely approximate that of $\Psi_{\mathcal{D}_t \mathcal{D}_t} \triangleq I + \sigma_n^{-2} \Sigma_{\mathcal{D}_t \mathcal{D}_t}$ residing in the $\mathbb{I}[\mathbf{f}_{\mathcal{D}}; \mathbf{y}_{\mathcal{D}_t} | \mathbf{y}_{\mathcal{D}_{1:t-1}}]$ term in (2) and at the same time be decomposed into a sum of log-determinant terms, each of which is defined by submatrices of $\Psi_{\mathcal{D}_t \mathcal{D}_t}$ that all depend on only a subset of the batch. Such a decomposition enables our resulting approximation of (2) to be formulated as a DCOP (Section 2).

At first glance, our proposed idea may be naively implemented by constructing a *sparse* block-diagonal matrix $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}$ using, say, the $N > 1$ diagonal blocks of $\Psi_{\mathcal{D}_t \mathcal{D}_t}$. Then, $\log |\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}|$ can be decomposed into a sum of log-determinants of its diagonal blocks², each of which depends on only a disjoint subset of the batch. This, however, entails an issue similar to that discussed at the beginning of this section of selecting the same $|\mathcal{D}_t|/N$ inputs N times within a batch due to the assumption of independence of outputs between different diagonal blocks of $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}$. To address this issue, we significantly relax this assumption and show that it is in fact possible to construct a more refined, *dense* matrix approximation $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}$ by exploiting a Markov assumption, which consequently correlates the outputs between all its constituent blocks and is, perhaps surprisingly, still amenable to the decomposition to achieve scalability in the batch size.

Specifically, evenly partition the batch \mathcal{D}_t of inputs into $N \in \{1, \dots, |\mathcal{D}_t|\}$ disjoint subsets $\mathcal{D}_{t1}, \dots, \mathcal{D}_{tN}$ and $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}$ ($\Psi_{\mathcal{D}_t \mathcal{D}_t}$) into $N \times N$ square blocks, i.e., $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t} \triangleq [\bar{\Psi}_{\mathcal{D}_{tn} \mathcal{D}_{tn'}}]_{n, n'=1, \dots, N}$ ($\Psi_{\mathcal{D}_t \mathcal{D}_t} \triangleq [\Psi_{\mathcal{D}_{tn} \mathcal{D}_{tn'}}]_{n, n'=1, \dots, N}$). Our first result below derives a decomposition of the log-determinant of any symmetric positive definite block matrix $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}$ into a sum of log-determinant terms, each of which is defined by a separate diagonal block of the Cholesky factor of $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}^{-1}$:

Proposition 1. Consider the Cholesky factorization of a symmetric positive definite $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}^{-1} \triangleq U^\top U$ where Cholesky factor $U \triangleq [U_{nn'}]_{n, n'=1, \dots, N}$ (i.e., partitioned into $N \times N$ square blocks) is an upper triangular block matrix (i.e., $U_{nn'} = \mathbf{0}$ for $n > n'$). Then, $\log |\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}| = \sum_{n=1}^N \log |(U_{nn}^\top U_{nn})^{-1}|$.

Its proof (Appendix B) utilizes properties of the determinant and that the determinant of an upper triangular block matrix is a product of determinants of its diagonal blocks (i.e., $|U| = \prod_{n=1}^N |U_{nn}|$). Proposition 1 reveals a subtle possibility of imposing some structure on the *inverse* of

²The determinant of a block-diagonal matrix is a product of determinants of its diagonal blocks.

$\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}$ such that each diagonal block U_{nn} of its Cholesky factor (and hence each $\log |(U_{nn}^\top U_{nn})^{-1}|$ term) will depend on only a subset of the batch. The following result presents one such possibility:

Proposition 2. Let $B \in \{1, \dots, N-1\}$ be given. If $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}^{-1}$ is B -block-banded³, then

$$(U_{nn}^\top U_{nn})^{-1} = \bar{\Psi}_{\mathcal{D}_{tn} \mathcal{D}_{tn}} - \bar{\Psi}_{\mathcal{D}_{tn} \mathcal{D}_{tn}^B} \bar{\Psi}_{\mathcal{D}_{tn}^B \mathcal{D}_{tn}^B}^{-1} \bar{\Psi}_{\mathcal{D}_{tn}^B \mathcal{D}_{tn}} \quad (3)$$

for $n = 1, \dots, N$ where $\eta \triangleq \min(n + B, N)$, $\mathcal{D}_{tn}^B \triangleq \bigcup_{n'=n+1}^{\eta} \mathcal{D}_{tn'}$, $\bar{\Psi}_{\mathcal{D}_{tn} \mathcal{D}_{tn}^B} \triangleq [\bar{\Psi}_{\mathcal{D}_{tn} \mathcal{D}_{tn'}}]_{n'=n+1, \dots, \eta}$, $\bar{\Psi}_{\mathcal{D}_{tn}^B \mathcal{D}_{tn}^B} \triangleq [\bar{\Psi}_{\mathcal{D}_{tn'} \mathcal{D}_{tn''}}]_{n', n''=n+1, \dots, \eta}$, and $\bar{\Psi}_{\mathcal{D}_{tn}^B \mathcal{D}_{tn}} \triangleq \bar{\Psi}_{\mathcal{D}_{tn} \mathcal{D}_{tn}^B}^\top$.

Its proof follows directly from a block-banded matrix result of (Asif & Moura, 2005) (i.e., Theorem 1). Proposition 2 indicates that if $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}^{-1}$ is B -block-banded (Fig. 1b), then each $\log |(U_{nn}^\top U_{nn})^{-1}|$ term depends on only the subset $\mathcal{D}_{tn} \cup \mathcal{D}_{tn}^B = \bigcup_{n'=n}^{\eta} \mathcal{D}_{tn'}$ of the batch \mathcal{D}_t (Fig. 1c).

Our next result defines a structure of $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}$ in terms of the blocks *within* the B -block band of $\Psi_{\mathcal{D}_t \mathcal{D}_t}$ to induce a B -block-banded inverse of $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}$:

Proposition 3. Let

$$\bar{\Psi}_{\mathcal{D}_{tn} \mathcal{D}_{tn'}} \triangleq \begin{cases} \Psi_{\mathcal{D}_{tn} \mathcal{D}_{tn'}} & \text{if } |\Delta| \leq B, \\ \Psi_{\mathcal{D}_{tn} \mathcal{D}_{tn}^B} \Psi_{\mathcal{D}_{tn}^B \mathcal{D}_{tn}^B}^{-1} \bar{\Psi}_{\mathcal{D}_{tn}^B \mathcal{D}_{tn'}} & \text{if } \Delta < -B, \\ \bar{\Psi}_{\mathcal{D}_{tn} \mathcal{D}_{tn}^B} \Psi_{\mathcal{D}_{tn}^B \mathcal{D}_{tn'}}^{-1} \Psi_{\mathcal{D}_{tn}^B \mathcal{D}_{tn'}} & \text{if } \Delta > B; \end{cases} \quad (4)$$

where $\Delta \triangleq n - n'$ for $n, n' = 1, \dots, N$ (see Fig. 1a). Then, $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}^{-1}$ is B -block-banded (see Fig. 1b).

Its proof follows directly from a block-banded matrix result of (Asif & Moura, 2005) (i.e., Theorem 3). It can be observed from (4) and Fig. 1 that (a) though $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}^{-1}$ is a sparse B -block-banded matrix, $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}$ is a dense matrix approximation for $B = 1, \dots, N-1$; (b) when $B = N-1$ or $N = 1$, $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t} = \Psi_{\mathcal{D}_t \mathcal{D}_t}$; and (c) the blocks *within* the B -block band of $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}$ (i.e., $|n - n'| \leq B$) coincide with that of $\Psi_{\mathcal{D}_t \mathcal{D}_t}$ while each block *outside* the B -block band of $\bar{\Psi}_{\mathcal{D}_t \mathcal{D}_t}$ (i.e., $|n - n'| > B$) is fully specified by the blocks within the B -block band of $\Psi_{\mathcal{D}_t \mathcal{D}_t}$ (i.e., $|n - n'| \leq B$) due to its recursive series of $|n - n'| - B$ reduced-rank approximations (Fig. 1a). Note, however, that the $\log |(U_{nn}^\top U_{nn})^{-1}|$ terms (3) for $n = 1, \dots, N$ depend on only the blocks within (and not outside) the B -block band of $\Psi_{\mathcal{D}_t \mathcal{D}_t}$ (Fig. 1c).

Remark 1. Proposition 3 provides an attractive principled interpretation: Let $\varepsilon_{\mathbf{x}} \triangleq \sigma_n^{-1} (y_{\mathbf{x}} - \mu_{\{\mathbf{x}\}})$ denote a scaled

³A block matrix $P \triangleq [P_{nn'}]_{n, n'=1, \dots, N}$ (i.e., partitioned into $N \times N$ square blocks) is B -block-banded if any block $P_{nn'}$ outside its B -block band (i.e., $|n - n'| > B$) is $\mathbf{0}$.

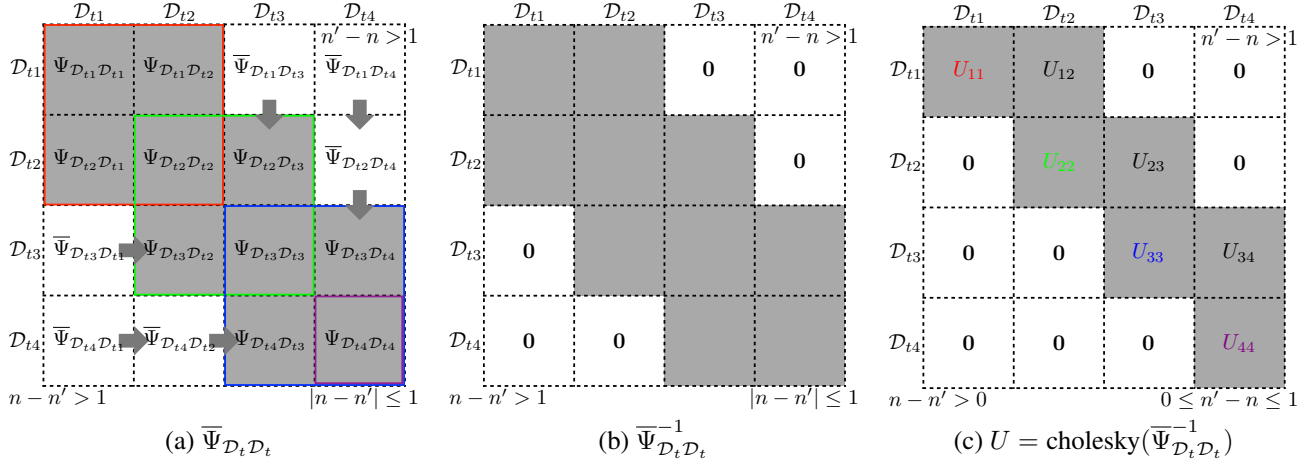


Figure 1. $\bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t}$, $\bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t}^{-1}$, and U with $B = 1$ and $N = 4$. (a) Shaded blocks (i.e., $|n - n'| \leq B$) form the B -block band while unshaded blocks (i.e., $|n - n'| > B$) fall outside the band. Each arrow denotes a recursive call. (b) Unshaded blocks outside the B -block band of $\bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t}^{-1}$ (i.e., $|n - n'| > B$) are $\mathbf{0}$, which result in the (c) unshaded blocks of its Cholesky factor U being $\mathbf{0}$ (i.e., $n - n' > 0$ or $n' - n > B$). Using (3) and (4), U_{11}, U_{22}, U_{33} , and U_{44} depend on only the shaded blocks of $\bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t}$ enclosed in red, green, blue, and purple, respectively.

residual incurred by the GP predictive mean (1). Its covariance is then $\text{cov}[\varepsilon_{\mathbf{x}}, \varepsilon_{\mathbf{x}'}] = \Psi_{\{\mathbf{x}\}\{\mathbf{x}'\}}$. In the same spirit as a Gaussian Markov random process, imposing a B -th order Markov property on the residual process $\{\varepsilon_{\mathbf{x}}\}_{\mathbf{x} \in \mathcal{D}_t}$ is equivalent to approximating $\Psi_{\mathcal{D}_t\mathcal{D}_t}$ with $\bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t}$ (4) whose inverse is B -block-banded. In other words, if $|n - n'| > B$, then $\{\varepsilon_{\mathbf{x}}\}_{\mathbf{x} \in \mathcal{D}_{t_n}}$ and $\{\varepsilon_{\mathbf{x}}\}_{\mathbf{x} \in \mathcal{D}_{t_{n'}}$ are conditionally independent given $\{\varepsilon_{\mathbf{x}}\}_{\mathbf{x} \in \mathcal{D}_t \setminus (\mathcal{D}_{t_n} \cup \mathcal{D}_{t_{n'}})}$. This conditional independence assumption therefore becomes more relaxed with a larger batch \mathcal{D}_t . Proposition 2 demonstrates the importance of such a B -th order Markov assumption (or, equivalently, the sparsity of B -block-banded $\bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t}^{-1}$) to achieving scalability in the batch size.

Remark 2. Regarding the approximation quality of $\bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t}$ (4), the following result (see Appendix C for its proof) shows that the *Kullback-Leibler* (KL) distance of $\bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t}$ from $\Psi_{\mathcal{D}_t\mathcal{D}_t}$ measures an intuitive notion of the *approximation error* of $\bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t}$ being the difference in information gain when relying on our Markov approximation, which can be bounded by some quantity ν_t :

Proposition 4. Let the KL distance $D_{\text{KL}}(\Psi, \tilde{\Psi}) \triangleq 0.5(\text{tr}(\Psi\tilde{\Psi}^{-1}) - \log|\Psi\tilde{\Psi}^{-1}| - |\mathcal{D}_t|)$ between two symmetric positive definite $|\mathcal{D}_t| \times |\mathcal{D}_t|$ matrices Ψ and $\tilde{\Psi}$ measure the error of approximating Ψ with $\tilde{\Psi}$. Also, let $\tilde{\mathbb{I}}[\mathbf{f}_{\mathcal{D}}; \mathbf{y}_{\mathcal{D}_t} | \mathbf{y}_{\mathcal{D}_{1:t-1}}] \triangleq 0.5 \log |\bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t}|$ denote the approximated information gain, and $C \geq \mathbb{I}[\mathbf{f}_{\{\mathbf{x}\}}; \mathbf{y}_{\mathcal{D}_t} | \mathbf{y}_{\mathcal{D}_{1:t-1}}]$ for all $\mathbf{x} \in \mathcal{D}$ and $t \in \mathbb{N}$. Then, for all $t \in \mathbb{N}$,

$$\begin{aligned} & D_{\text{KL}}(\Psi_{\mathcal{D}_t\mathcal{D}_t}, \bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t}) \\ &= \tilde{\mathbb{I}}[\mathbf{f}_{\mathcal{D}}; \mathbf{y}_{\mathcal{D}_t} | \mathbf{y}_{\mathcal{D}_{1:t-1}}] - \mathbb{I}[\mathbf{f}_{\mathcal{D}}; \mathbf{y}_{\mathcal{D}_t} | \mathbf{y}_{\mathcal{D}_{1:t-1}}] \\ &\leq (\exp(2C) - 1) \mathbb{I}[\mathbf{f}_{\mathcal{D}}; \mathbf{y}_{\mathcal{D}_t} | \mathbf{y}_{\mathcal{D}_{1:t-1}}] \triangleq \nu_t. \end{aligned}$$

Proposition 4 implies that the approximated information gain $\tilde{\mathbb{I}}[\mathbf{f}_{\mathcal{D}}; \mathbf{y}_{\mathcal{D}_t} | \mathbf{y}_{\mathcal{D}_{1:t-1}}]$ is never smaller than the exact information gain $\mathbb{I}[\mathbf{f}_{\mathcal{D}}; \mathbf{y}_{\mathcal{D}_t} | \mathbf{y}_{\mathcal{D}_{1:t-1}}]$ since $D_{\text{KL}}(\Psi_{\mathcal{D}_t\mathcal{D}_t}, \bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t}) \geq 0$ with equality when $N = 1$, in which case $\bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t} = \Psi_{\mathcal{D}_t\mathcal{D}_t}$ (4). Thus, intuitively, our proposed Markov approximation *hallucinates information* into $\Psi_{\mathcal{D}_t\mathcal{D}_t}$ to yield an *optimistic estimate* of the information gain (by selecting a particular batch), ultimately making our resulting algorithm *overconfident* in selecting a batch. This overconfidence is information-theoretically quantified by the *approximation error* $D_{\text{KL}}(\Psi_{\mathcal{D}_t\mathcal{D}_t}, \bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t}) \leq \nu_t$.

Remark 3. The KL distance $D_{\text{KL}}(\Psi_{\mathcal{D}_t\mathcal{D}_t}, \bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t})$ of $\bar{\Psi}_{\mathcal{D}_t\mathcal{D}_t}$ from $\Psi_{\mathcal{D}_t\mathcal{D}_t}$ is also the least among all $|\mathcal{D}_t| \times |\mathcal{D}_t|$ matrices with a B -block-banded inverse, as proven in Appendix D.

DCOP Formulation. By exploiting the approximated information gain $\tilde{\mathbb{I}}[\mathbf{f}_{\mathcal{D}}; \mathbf{y}_{\mathcal{D}_t} | \mathbf{y}_{\mathcal{D}_{1:t-1}}]$ (Proposition 4), Proposition 1, (3), and (4), our batch variant of GP-UCB (2) can be reformulated in an approximate sense⁴ to a *distributed batch GP-UCB* (DB-GP-UCB) algorithm⁵ that jointly optimizes a batch of inputs in each iteration $t = 1, \dots, T$ according to

$$\begin{aligned} \bar{\mathcal{D}}_t &\triangleq \arg \max_{\mathcal{D}_t \subset \mathcal{D}} \sum_{n=1}^N w_n(\mathcal{D}_{t_n} \cup \mathcal{D}_{t_n}^B) \\ w_n(\mathcal{D}_{t_n} \cup \mathcal{D}_{t_n}^B) &\triangleq \mathbf{1}^\top \mu_{\mathcal{D}_{t_n}} + (0.5\alpha_t \log |\Psi_{\mathcal{D}_{t_n}\mathcal{D}_{t_n} | \mathcal{D}_{t_n}^B}|)^{1/2} \\ &\text{with } \Psi_{\mathcal{D}_{t_n}\mathcal{D}_{t_n} | \mathcal{D}_{t_n}^B} \triangleq \Psi_{\mathcal{D}_{t_n}\mathcal{D}_{t_n}} - \Psi_{\mathcal{D}_{t_n}\mathcal{D}_{t_n}^B} \Psi_{\mathcal{D}_{t_n}^B\mathcal{D}_{t_n}^B}^{-1} \Psi_{\mathcal{D}_{t_n}^B\mathcal{D}_{t_n}}. \end{aligned} \quad (5)$$

⁴Note that our acquisition function (5) uses $\sum_{n=1}^N (\log |\cdot|)^{1/2}$ instead of $(\sum_{n=1}^N \log |\cdot|)^{1/2}$ to enable the decomposition.

⁵Pseudocode for DB-GP-UCB is provided in Appendix E.

Note that (5) is equivalent to our batch variant of GP-UCB (2) when $N = 1$. It can also be observed that (5) is naturally formulated as a multi-agent DCOP (Section 2) whereby every agent $a_n \in \mathcal{A}$ is responsible for optimizing a disjoint subset \mathcal{D}_{tn} of the batch \mathcal{D}_t for $n = 1, \dots, N$ and each function w_n defines a constraint over only the subset $\mathcal{D}_{tn} \cup \mathcal{D}_{tn}^B = \bigcup_{n'=n}^N \mathcal{D}_{tn'}$ of the batch \mathcal{D}_t and represents the joint payoff that the corresponding agents $\mathcal{A}_n \triangleq \{a_{n'}\}_{n'=n}^N \subseteq \mathcal{A}$ achieve. As a result, (5) can be efficiently and scalably solved by the state-of-the-art DCOP algorithms (Chapman et al., 2011; Leite et al., 2014). For example, the time complexity of an iterative message-passing algorithm called max-sum (Farinelli et al., 2008) scales exponentially in only the largest arity $\max_{n \in \{1, \dots, N\}} |\mathcal{D}_{tn} \cup \mathcal{D}_{tn}^B| = (B+1)|\mathcal{D}_t|/N$ of the functions w_1, \dots, w_N . Given a limited time budget, a practitioner can set a maximum arity of ω for any function w_n , after which the number N of functions is adjusted to $\lceil (B+1)|\mathcal{D}_t|/\omega \rceil$ so that the time incurred by max-sum to solve the DCOP in (5) is $\mathcal{O}(|\mathcal{D}_t|^\omega \omega^3 B |\mathcal{D}_t|)^6$ per iteration (i.e., linear in the batch size $|\mathcal{D}_t|$ by assuming ω and the Markov order B to be constants). In contrast, our batch variant of GP-UCB (2) incurs exponential time in the batch size $|\mathcal{D}_t|$. The max-sum algorithm is also amenable to a distributed implementation on a cluster of parallel machines to boost scalability further. If a solution quality guarantee is desired, then a variant of max-sum called bounded max-sum (Rogers et al., 2011) can be used⁷. Finally, the Markov order B can be varied to trade off between the approximation quality of $\bar{\Psi}_{\mathcal{D}_t, \mathcal{D}_t}$ (4) and the time efficiency of max-sum in solving the DCOP in (5).

Regret Bounds. Our main result to follow derives probabilistic bounds on the cumulative regret of DB-GP-UCB:

Theorem 1. *Let $\delta \in (0, 1)$ be given, $C_1 \triangleq 4/\log(1 + \sigma_n^{-2})$, $\gamma_T \triangleq \max_{\mathcal{D}_1, T \subset \mathcal{D}} \mathbb{I}[\mathbf{f}_{\mathcal{D}}; \mathbf{y}_{\mathcal{D}_1, T}]$, $\alpha_t \triangleq C_1 |\mathcal{D}_t| \exp(2C) \log(|\mathcal{D}_t| t^2 \pi^2 / (6\delta))$, and $\bar{\nu}_T \triangleq \sum_{t=1}^T \nu_t$. Then, for the batch and full cumulative regrets incurred by our DB-GP-UCB algorithm (5),*

$$\begin{aligned} R_T &\leq 2 (T |\bar{\mathcal{D}}_T|^{-2} \alpha_T N (\gamma_T + \bar{\nu}_T))^{1/2} \text{ and} \\ R'_T &\leq 2 (T \alpha_T N (\gamma_T + \bar{\nu}_T))^{1/2} \end{aligned}$$

hold with probability of at least $1 - \delta$.

⁶We assume the use of online sparse GP models (Csató & Opper, 2002; Hensman et al., 2013; Hoang et al., 2015; 2017; Low et al., 2014b; Xu et al., 2014) that can update the GP predictive/posterior distribution (1) in constant time in each iteration.

⁷Bounded max-sum is previously used in (Rogers et al., 2011) to solve a related maximum entropy sampling problem (Shewry & Wynn, 1987) formulated as a DCOP. But, the largest arity of any function w_n in this DCOP is still the batch size $|\mathcal{D}_t|$ and, unlike the focus of our work here, no attempt is made in (Rogers et al., 2011) to reduce it, thus causing max-sum and bounded max-sum to incur exponential time in $|\mathcal{D}_t|$. In fact, our proposed Markov approximation can be applied to this problem to reduce the largest arity of any function w_n in this DCOP to again $(B+1)|\mathcal{D}_t|/N$.

Its proof (Appendix F), when compared to that of GP-UCB (Srinivas et al., 2010) and its greedy batch variants (Contal et al., 2013; Desautels et al., 2014), requires tackling the additional technical challenges associated with jointly optimizing a batch \mathcal{D}_t of inputs in each iteration t . Note that the uncertainty sampling based initialization strategy proposed by Desautels et al. (2014) can be employed to replace the $\sqrt{\exp(2C)}$ term (i.e., growing linearly in $|\mathcal{D}_t|$) appearing in our regret bounds by a kernel-dependent constant factor of C' that is independent of $|\mathcal{D}_t|$; values of C' for the most commonly-used kernels are replicated in Table 2 in Appendix G (see section 4.5 in (Desautels et al., 2014) for a more detailed discussion on this issue).

Table 1 in Appendix G compares the bounds on R_T of DB-GP-UCB (5), GP-UCB-PE, GP-BUCB, GP-UCB, and UCB-DPP-SAMPLE. Compared to the bounds on R_T of GP-UCB-PE and UCB-DPP-SAMPLE, our bound includes the additional kernel-dependent factor of C' , which is similar to GP-BUCB. In fact, our regret bound is of the same form as that of GP-BUCB except that our bound incorporates a parameter N of our Markov approximation and an upper bound $\bar{\nu}_T$ on the cumulative approximation error, both of which vanish for our batch variant of GP-UCB (2):

Corollary 1. *For our batch variant of GP-UCB (2), the cumulative regrets reduce to $R_T \leq 2 (T |\bar{\mathcal{D}}_T|^{-2} \alpha_T \gamma_T)^{1/2}$ and $R'_T \leq 2 (T \alpha_T \gamma_T)^{1/2}$.*

Corollary 1 follows directly from Theorem 1 and by noting that for our batch variant (2), $N = 1$ (since $\bar{\Psi}_{\mathcal{D}_t, \mathcal{D}_t}$ then trivially reduces to $\Psi_{\mathcal{D}_t, \mathcal{D}_t}$) and $\nu_t = 0$ for $t = 1, \dots, T$.

Finally, the convergence rate of our DB-GP-UCB algorithm is dominated by the growth behavior of $\gamma_T + \bar{\nu}_T$. While it is well-known that the bounds on the maximum mutual information γ_T established for the commonly-used linear, squared exponential, and Matérn kernels in (Srinivas et al., 2010; Kathuria et al., 2016) (i.e., replicated in Table 2 in Appendix G) only grow sublinearly in T , it is not immediately clear how the upper bound $\bar{\nu}_T$ on the cumulative approximation error behaves. Our next result reveals that $\bar{\nu}_T$ in fact only grows sublinearly in T as well:

Corollary 2. $\bar{\nu}_T \leq (\exp(2C) - 1) \gamma_T$.

Corollary 2 follows directly from the definitions of ν_t in Proposition 4 and $\bar{\nu}_T$ and γ_T in Theorem 1 and applying the chain rule for mutual information. Since γ_T grows sublinearly in T for the above-mentioned kernels (Srinivas et al., 2010) and C can be chosen to be independent of T (e.g., $C \triangleq \gamma_{|\bar{\mathcal{D}}_t| - 1}$) (Desautels et al., 2014), it follows from Corollary 2 that $\bar{\nu}_T$ grows sublinearly in T . As a result, Theorem 1 guarantees sublinear cumulative regrets for the above-mentioned kernels, which implies that our DB-GP-UCB algorithm (5) asymptotically achieves no regret, regardless of the degree of our proposed Markov approxi-

mation (i.e., configuration of $[N, B]$). Thus, our batch variant of GP-UCB (2) achieves no regret as well.

4. Experiments and Discussion

This section evaluates the cumulative regret incurred by our DB-GP-UCB algorithm (5) and its scalability in the batch size empirically on two synthetic benchmark objective functions such as Branin-Hoo (Lizotte, 2008) and gSobol (González et al., 2016) (Table 3 in Appendix H) and a real-world pH field of Broom’s Barn farm (Webster & Oliver, 2007) (Fig. 3 in Appendix H) spatially distributed over a 1200 m by 680 m region discretized into a 31×18 grid of sampling locations. These objective functions and the real-world pH field are each modeled as a sample of a GP whose prior covariance is defined by the widely-used squared exponential kernel $k_{\mathbf{x}\mathbf{x}'} \triangleq \sigma_s^2 \exp(-0.5(\mathbf{x} - \mathbf{x}')^\top \Lambda^{-2}(\mathbf{x} - \mathbf{x}'))$ where $\Lambda \triangleq \text{diag}[\ell_1, \dots, \ell_d]$ and σ_s^2 are its length-scale and signal variance hyperparameters, respectively. These hyperparameters together with the noise variance σ_n^2 are learned using maximum likelihood estimation (Rasmussen & Williams, 2006).

The performance of our DB-GP-UCB algorithm (5) is compared with the state-of-the-art batch BO algorithms such as GP-BUCB (Desautels et al., 2014), GP-UCB-PE (Contal et al., 2013), SM-UCB (Azimi et al., 2010), q -EI (Chevalier & Ginsbourger, 2013), and BBO-LP by plugging in GP-UCB (González et al., 2016), whose implementations⁸ are publicly available. These batch BO algorithms are evaluated using a performance metric that measures the cumulative regret incurred by a tested algorithm: $\sum_{t=1}^T f(\mathbf{x}^*) - f(\tilde{\mathbf{x}}_t)$ where $\tilde{\mathbf{x}}_t \triangleq \arg \max_{\mathbf{x}_t \in \mathcal{D}} \mu\{\mathbf{x}_t\}$ (1) is the recommendation of the tested algorithm after t batch evaluations. For each experiment, 5 noisy observations are randomly selected and used for initialization. This is independently repeated 64 times and we report the resulting mean cumulative regret incurred by a tested algorithm. All experiments are run on a Linux system with Intel® Xeon® E5-2670 at 2.6GHz with 96 GB memory.

For our experiments, we use a *fixed* budget of $T|\mathcal{D}_T| = 64$ function evaluations and analyze the trade-off between batch size $|\mathcal{D}_T|$ (i.e., 2, 4, 8, 16) vs. time horizon T (respectively, 32, 16, 8, 4) on the performance of the tested algorithms. This experimental setup represents a practical scenario of costly function evaluations: On one hand, when a function evaluation is computationally costly (i.e., time-consuming), it is more desirable to evaluate f for a larger batch (e.g., $|\mathcal{D}_T| = 16$) of inputs in parallel in each iteration t (i.e., if hardware resources permit) to reduce the total time needed (hence smaller T). On the other hand,

when a function evaluation is economically costly, one may be willing to instead invest more time (hence larger T) to evaluate f for a smaller batch (e.g., $|\mathcal{D}_T| = 2$) of inputs in each iteration t in return for a higher frequency of information and consequently a more adaptive BO to achieve potentially better performance. In some settings, both factors may be equally important, that is, moderate values of $|\mathcal{D}_T|$ and T are desired. To the best of our knowledge, such a form of empirical analysis does not seem to be available in the batch BO literature.

Fig. 2 shows results⁹ of the cumulative regret incurred by the tested algorithms to analyze their trade-off between batch size $|\mathcal{D}_T|$ (i.e., 2, 4, 8, 16) vs. time horizon T (respectively, 32, 16, 8, 4) using a fixed budget of $T|\mathcal{D}_T| = 64$ function evaluations for the Branin-Hoo function (left column), gSobol function (middle column), and real-world pH field (right column). Our DB-GP-UCB algorithm uses the configurations of $[N, B] = [4, 2], [8, 5], [16, 10]$ in the experiments with batch size $|\mathcal{D}_T| = 4, 8, 16$, respectively; in the case of $|\mathcal{D}_T| = 2$, we use our batch variant of GP-UCB (2) which is equivalent to DB-GP-UCB when $N = 1$. It can be observed that DB-GP-UCB achieves lower cumulative regret than GP-BUCB, GP-UCB-PE, SM-UCB, and BBO-LP in all experiments (with the only exception being the gSobol function for the smallest batch size of $|\mathcal{D}_T| = 2$ where BBO-LP performs slightly better) since DB-GP-UCB can jointly optimize a batch of inputs while GP-BUCB, GP-UCB-PE, SM-UCB, and BBO-LP are greedy batch algorithms that select the inputs of a batch one at time. Note that as the real-world pH field is not as well-behaved as the synthetic benchmark functions (see Fig. 3 in Appendix H), the estimate of the Lipschitz constant by BBO-LP is potentially worse, hence likely degrading its performance. Furthermore, DB-GP-UCB can scale to a much larger batch size of 16 than the other batch BO algorithms that also jointly optimize the batch of inputs, which include q -EI, PPES (Shah & Ghahramani, 2015) and q -KG (Wu & Frazier, 2016): Results of q -EI are not available for $|\mathcal{D}_T| \geq 4$ as they require a prohibitively huge computational effort to be obtained¹⁰ while PPES can only operate with a small batch size of up to 3 for the Branin-Hoo function and up to 4 for other functions, as reported in (Shah & Ghahramani, 2015), and q -KG can only operate with a small batch size of 4 for all tested functions (including the Branin-Hoo function and four others), as reported in (Wu & Frazier, 2016). The scalability of DB-GP-UCB is attributed to our proposed Markov approximation of our

⁹Error bars are omitted in Fig. 2 to preserve the readability of the graphs. A replication of the graphs in Fig. 2 including standard error bars is provided in Appendix K.

¹⁰In the experiments of González et al. (2016), q -EI can reach a batch size of up to 10 but performs much worse than GP-BUCB, which is likely due to a considerable downsampling of possible batches available for selection in each iteration.

⁸Details on the used implementations are given in Table 4 in Appendix I. We implemented DB-GP-UCB in MATLAB to exploit the GPML toolbox (Rasmussen & Williams, 2006).

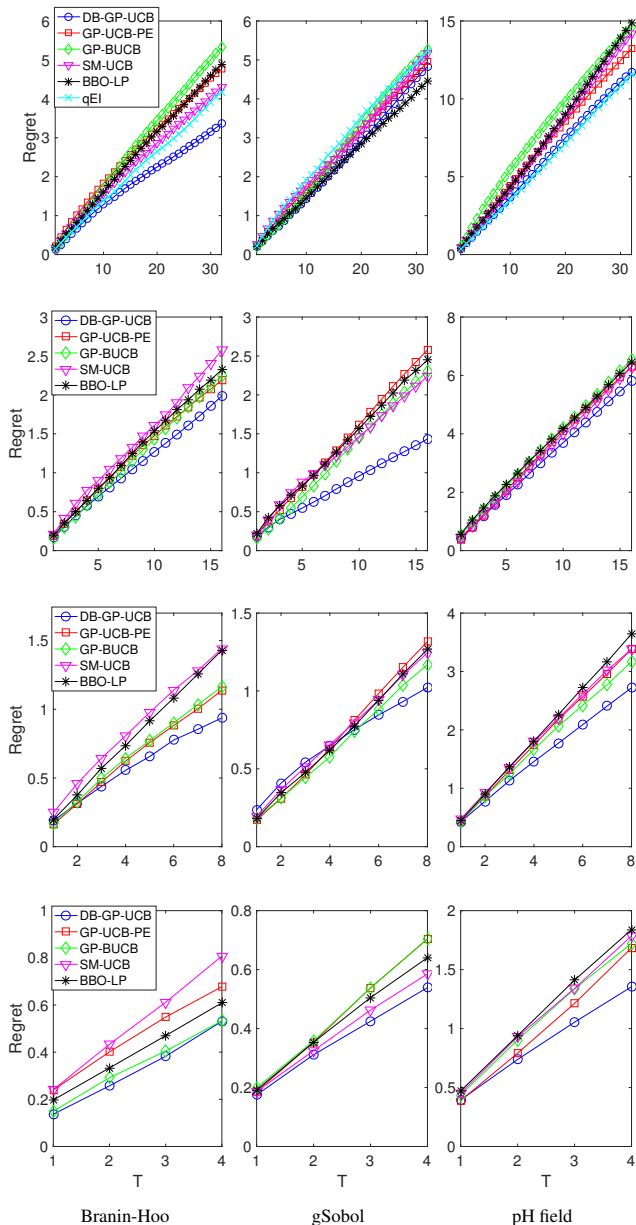


Figure 2. Cumulative regret incurred by tested algorithms with varying batch sizes $|\mathcal{D}_T| = 2, 4, 8, 16$ (rows from top to bottom) using a fixed budget of $T|\mathcal{D}_T| = 64$ function evaluations for the Brainin-Hoo function, gSobol function, and real-world pH field.

batch variant of GP-UCB (2) (Section 3), which can then be naturally formulated as a multi-agent DCOP (5) in order to fully exploit the efficiency of one of its state-of-the-art solvers called max-sum (Farinelli et al., 2008). In the experiments with the largest batch size of $|\mathcal{D}_T| = 16$, we have reduced the number of iterations in max-sum to less than 5 without waiting for convergence to preserve the efficiency of DB-GP-UCB, thus sacrificing its BO performance. Nevertheless, DB-GP-UCB can still outperform the other tested

batch BO algorithms.

We have also investigated and analyzed the trade-off between approximation quality and time efficiency of our DP-GP-UCB algorithm and reported the results in Appendix J due to lack of space. To summarize, it can be observed from our results that the approximation quality improves near-linearly with an increasing Markov order B at the expense of higher computational cost (i.e., exponential in B).

5. Conclusion

This paper develops a novel *distributed batch GP-UCB* (DB-GP-UCB) algorithm for performing batch BO of highly complex, costly-to-evaluate, noisy black-box objective functions. In contrast to greedy batch BO algorithms (Azimi et al., 2010; Contal et al., 2013; Desautels et al., 2014; González et al., 2016), our DB-GP-UCB algorithm can jointly optimize a batch of inputs and, unlike (Chevalier & Ginsbourger, 2013; Shah & Ghahramani, 2015; Wu & Frazier, 2016), still preserve scalability in the batch size. To realize this, we generalize GP-UCB (Srinivas et al., 2010) to a new batch variant amenable to a Markov approximation, which can then be naturally formulated as a multi-agent DCOP in order to fully exploit the efficiency of its state-of-the-art solvers such as max-sum (Farinelli et al., 2008; Rogers et al., 2011) for achieving linear time in the batch size. Our proposed DB-GP-UCB algorithm offers practitioners the flexibility to trade off between the approximation quality and time efficiency by varying the Markov order. We provide a theoretical guarantee for the convergence rate of our DB-GP-UCB algorithm via bounds on its cumulative regret. Empirical evaluation on synthetic benchmark objective functions and a real-world pH field shows that our DB-GP-UCB algorithm can achieve lower cumulative regret than the greedy batch BO algorithms such as GP-BUCB, GP-UCB-PE, SM-UCB, and BBO-LP, and scale to larger batch sizes than the other batch BO algorithms that also jointly optimize the batch of inputs, which include q -EI, PPES, and q -KG. For future work, we plan to generalize DB-GP-UCB (a) to the nonmyopic context by appealing to existing literature on nonmyopic BO (Ling et al., 2016) and active learning (Cao et al., 2013; Hoang et al., 2014a;b; Low et al., 2008; 2009; 2011; 2014a) as well as (b) to be performed by a multi-robot team to find hotspots in environmental sensing/monitoring by seeking inspiration from existing literature on multi-robot active sensing/learning (Chen et al., 2012; 2013b; 2015; Low et al., 2012; Ouyang et al., 2014). For applications with a huge budget of function evaluations, we like to couple DB-GP-UCB with the use of parallel/distributed sparse GP models (Chen et al., 2013a; Hoang et al., 2016; Low et al., 2015) to represent the belief of the unknown objective function efficiently.

Acknowledgements

This research is supported by Singapore Ministry of Education Academic Research Fund Tier 2, MOE2016-T2-2-156. Erik A. Daxberger would like to thank Volker Tresp for his advice throughout this research project.

References

- Anderson, B. S., Moore, A. W., and Cohn, D. A non-parametric approach to noisy and costly optimization. In *Proc. ICML*, 2000.
- Asif, A. and Moura, J. M. F. Block matrices with L -block-banded inverse: Inversion algorithms. *IEEE Trans. Signal Processing*, 53(2):630–642, 2005.
- Azimi, J., Fern, A., and Fern, X. Z. Batch Bayesian optimization via simulation matching. In *Proc. NIPS*, pp. 109–117, 2010.
- Brochu, E., Cora, V. M., and de Freitas, N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv:1012.2599, 2010.
- Cao, N., Low, K. H., and Dolan, J. M. Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms. In *Proc. AAMAS*, pp. 7–14, 2013.
- Chapman, A., Rogers, A., Jennings, N. R., and Leslie, D. A unifying framework for iterative approximate best-response algorithms for distributed constraint optimisation problems. *The Knowledge Engineering Review*, 26(4):411–444, 2011.
- Chen, J., Low, K. H., Tan, C. K.-Y., Oran, A., Jaillet, P., Dolan, J. M., and Sukhatme, G. S. Decentralized data fusion and active sensing with mobile sensors for modeling and predicting spatiotemporal traffic phenomena. In *Proc. UAI*, pp. 163–173, 2012.
- Chen, J., Cao, N., Low, K. H., Ouyang, R., Tan, C. K.-Y., and Jaillet, P. Parallel Gaussian process regression with low-rank covariance matrix approximations. In *Proc. UAI*, pp. 152–161, 2013a.
- Chen, J., Low, K. H., and Tan, C. K.-Y. Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system. In *Proc. RSS*, 2013b.
- Chen, J., Low, K. H., Jaillet, P., and Yao, Y. Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems. *IEEE Trans. Autom. Sci. Eng.*, 12: 901–921, 2015.
- Chevalier, C. and Ginsbourger, D. Fast computation of the multi-points expected improvement with applications in batch selection. In *Proc. 7th International Conference on Learning and Intelligent Optimization*, pp. 59–69, 2013.
- Contal, E., Buffoni, D., Robicquet, A., and Vayatis, N. Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *Proc. ECML/PKDD*, pp. 225–240, 2013.
- Contal, E., Perchet, V., and Vayatis, N. Gaussian process optimization with mutual information. In *Proc. ICML*, pp. 253–261, 2014.
- Csató, L. and Opper, M. Sparse online Gaussian processes. *Neural Computation*, 14(3):641–669, 2002.
- Desautels, T., Krause, A., and Burdick, J. W. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *JMLR*, 15:4053–4103, 2014.
- Farinelli, A., Rogers, A., Petcu, A., and Jennings, N. R. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *Proc. AAMAS*, pp. 639–646, 2008.
- González, J., Dai, Z., Hennig, P., and Lawrence, N. D. Batch Bayesian optimization via local penalization. In *Proc. AISTATS*, 2016.
- Hennig, P. and Schuler, C. J. Entropy search for information-efficient global optimization. *JMLR*, 13: 1809–1837, 2012.
- Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian processes for big data. In *Proc. UAI*, 2013.
- Hernández-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z. Predictive entropy search for efficient global optimization of black-box functions. In *Proc. NIPS*, pp. 918–926, 2014.
- Hoang, Q. M., Hoang, T. N., and Low, K. H. A generalized stochastic variational Bayesian hyperparameter learning framework for sparse spectrum Gaussian process regression. In *Proc. AAAI*, pp. 2007–2014, 2017.
- Hoang, T. N., Low, K. H., Jaillet, P., and Kankanhalli, M. Active learning is planning: Nonmyopic ϵ -Bayes-optimal active learning of Gaussian processes. In *Proc. ECML/PKDD Nectar Track*, pp. 494–498, 2014a.
- Hoang, T. N., Low, K. H., Jaillet, P., and Kankanhalli, M. Nonmyopic ϵ -Bayes-optimal active learning of Gaussian processes. In *Proc. ICML*, pp. 739–747, 2014b.
- Hoang, T. N., Hoang, Q. M., and Low, K. H. A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data. In *Proc. ICML*, pp. 569–578, 2015.

- Hoang, T. N., Hoang, Q. M., and Low, K. H. A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In *Proc. ICML*, pp. 382–391, 2016.
- Kathuria, T., Deshpande, A., and Kohli, P. Batched Gaussian process bandit optimization via determinantal point processes. In *Proc. NIPS*, pp. 4206–4214, 2016.
- Leite, A. R., Enembreck, F., and Barthès, J.-P. A. Distributed constraint optimization problems: Review and perspectives. *Expert Systems with Applications*, 41(11): 5139–5157, 2014.
- Ling, C. K., Low, K. H., and Jaillet, P. Gaussian process planning with Lipschitz continuous reward functions: Towards unifying Bayesian optimization, active learning, and beyond. In *Proc. AAAI*, pp. 1860–1866, 2016.
- Lizotte, D. J. *Practical Bayesian optimization*. Ph.D. Thesis, University of Alberta, 2008.
- Low, K. H., Dolan, J. M., and Khosla, P. Adaptive multi-robot wide-area exploration and mapping. In *Proc. AAMAS*, pp. 23–30, 2008.
- Low, K. H., Dolan, J. M., and Khosla, P. Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. In *Proc. ICAPS*, pp. 233–240, 2009.
- Low, K. H., Dolan, J. M., and Khosla, P. Active Markov information-theoretic path planning for robotic environmental sensing. In *Proc. AAMAS*, pp. 753–760, 2011.
- Low, K. H., Chen, J., Dolan, J. M., Chien, S., and Thompson, D. R. Decentralized active robotic exploration and mapping for probabilistic field classification in environmental sensing. In *Proc. AAMAS*, pp. 105–112, 2012.
- Low, K. H., Chen, J., Hoang, T. N., Xu, N., and Jaillet, P. Recent advances in scaling up Gaussian process predictive models for large spatiotemporal data. In Ravela, S. and Sandu, A. (eds.), *Dynamic Data-Driven Environmental Systems Science: First International Conference, DyDESS 2014*, pp. 167–181. LNCS 8964, Springer International Publishing, 2014a.
- Low, K. H., Xu, N., Chen, J., Lim, K. K., and Özgül, E. B. Generalized online sparse Gaussian processes with application to persistent mobile robot localization. In *Proc. ECML/PKDD Nectar Track*, pp. 499–503, 2014b.
- Low, K. H., Yu, J., Chen, J., and Jaillet, P. Parallel Gaussian process regression for big data: Low-rank representation meets Markov approximation. In *Proc. AAAI*, pp. 2821–2827, 2015.
- Ouyang, R., Low, K. H., Chen, J., and Jaillet, P. Multi-robot active sensing of non-stationary Gaussian process-based environmental phenomena. In *Proc. AAMAS*, pp. 573–580, 2014.
- Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- Rogers, A., Farinelli, A., Stranders, R., and Jennings, N. R. Bounded approximate decentralised coordination via the max-sum algorithm. *AIJ*, 175(2):730–759, 2011.
- Shah, A. and Ghahramani, Z. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *Proc. NIPS*, pp. 3312–3320, 2015.
- Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.
- Shewry, M. C. and Wynn, H. P. Maximum entropy sampling. *J. Applied Statistics*, 14(2):165–170, 1987.
- Srinivas, N., Krause, A., Kakade, S., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. ICML*, pp. 1015–1022, 2010.
- Villemonteix, J., Vazquez, E., and Walter, E. An informational approach to the global optimization of expensive-to-evaluate functions. *J. Glob. Optim.*, 44(4):509–534, 2009.
- Webster, R. and Oliver, M. *Geostatistics for Environmental Scientists*. John Wiley & Sons, Inc., 2007.
- Wu, J. and Frazier, P. The parallel knowledge gradient method for batch Bayesian optimization. In *Proc. NIPS*, pp. 3126–3134, 2016.
- Xu, N., Low, K. H., Chen, J., Lim, K. K., and Özgül, E. B. GP-Localize: Persistent mobile robot localization using online sparse Gaussian process observation model. In *Proc. AAAI*, pp. 2585–2592, 2014.