# Practical Path-based Bayesian Optimization

**Jose Pablo Folch**                                                JOSE.FOLCH16@IMPERIAL.AC.UK
*Department of Mathematics, Imperial College London*

**James Odgers**                                                JAMES.ODGERS16@IMPERIAL.AC.UK
*Department of Computing, Imperial College London*

**Shiqiang Zhang**                                                S.ZHANG21@IMPERIAL.AC.UK
*Department of Computing, Imperial College London*

**Robert M Lee**                                                ROBERT-MATTHEW.LEE@BASF.COM
*BASF SE*

**Behrang Shafei**                                                BEHRANG.SHAFEI@BASF.COM
*BASF SE*

**David Walz**                                                DAVID-SIMON.WALZ@BASF.COM
*BASF SE*

**Calvin Tsay**                                                C.TSAY@IMPERIAL.AC.UK
*Department of Computing, Imperial College London*

**Mark van der Wilk**                                                M.VDWILK@IMPERIAL.AC.UK
*Department of Computing, Imperial College London*

**Ruth Misener**                                                R.MISENER@IMPERIAL.AC.UK
*Department of Computing, Imperial College London*

## Abstract

There has been a surge in interest in data-driven experimental design with applications to chemical engineering and drug manufacturing. Bayesian optimization (BO) has proven to be adaptable to such cases, since we can model the reactions of interest as expensive black-box functions. Sometimes, the cost of this black-box functions can be separated into two parts: (a) the cost of the experiment itself, and (b) the cost of changing the input parameters. In this short paper, we extend the SnAKe algorithm to deal with both types of costs simultaneously. We further propose extensions to the case of a maximum allowable input change, as well as to the multi-objective setting.

**Keywords:** Machine Learning, Bayesian Optimization

## 1. Introduction and related work

Efficiently optimizing the desired qualities of materials or chemicals in experimental settings is important, given the costliness of experiments. Bayesian Optimization (BO) (Jones et al., 1998; Shahriari et al., 2016) has emerged as a potent tool for this purpose, e.g., see Griffiths and Hernández-Lobato (2020). However, the diversity of experimental setups necessitates extending current methods to accommodate common constraints and objectives.

BO seeks to optimize an expensive black-box function through the use of probabilistic models, usually Gaussian Processes (GPs) (Rasmussen and Williams, 2005), which are then

used to select inputs in sequential fashion. This usually involves optimizing a tractable acquisition function, $\alpha(x)$, so that the selected input is given by:

$$x_{t+1} = \underset{x \in \mathcal{X}}{\arg\max}\, \alpha(x|D_t) \tag{1}$$

where $D_t$ is the data-set at iteration $t$, and $x_i$ is the $i$-th experiment.

Recent work has focused on path-based BO, where the cost of changing inputs between subsequent experiments must be taken into account. This surge has been inspired by scientific applications to environmental monitoring (Bogunovic et al., 2016; Mutny et al., 2023; Folch et al., 2022; Yang et al., 2023; Samaniego et al., 2021), chemistry (Folch et al., 2022; Vellanki et al., 2017), and energy systems (Ramesh et al., 2022).

Approaches for path-based BO have been considered through meta-learning (Yang et al., 2023), and in the contextual BO setting (Ramesh et al., 2022). Mutny et al. (2023) approach a similar problem by considering the optimization of trajectories in active learning. Our recent work (Folch et al., 2022) proposes SnAKe, a path-based variation of BO that is able to minimize the cumulative cost of input changes, by iteratively: (i) creating large batches of possible experiments using Thompson Sampling (Kandasamy et al., 2018), (ii) solving a Travelling Salesman Problem to order the experiments in a way that minimizes input costs, and (iii) updating the path as observations arrive, using a 'point-deletion' heuristic that helps avoid getting stuck in local optima.

SnAKe is limited in some practical aspects. This work presents three extensions:

I. SnAKe requires the total number of experiments to be selected *a priori*, but, in practice we might not know how many experiments are needed, or we might want to minimize the number of experiments, e.g., to save on materials.

II. Even though SnAKe minimizes cumulative input changes, individual input changes between iterations are unbounded. This could pose a practical problem in some applications, where large changes may be unsafe or infeasible. For example, there may be limitations on maximum temperature or the rate of temperature change.

III. The algorithm assumes a single objective function. In many applications, experimentalists are interested in problems with many objectives. In multi-objective settings, we seek to find a Pareto front of solutions as opposed to optimizing a single function.

Other limitations of SnAKe include poor performance in high-dimensional settings, or when the experimental budget is low ($< 100$). Yang et al. (2023) introduce MONGOOSE, addressing both problems using meta-learning.

Further related work includes Nguyen et al. (2017), who explore using the Expected Improvement (EI) acquisition function as a stopping criteria, similar to the use of Expected Improvement per unit cost (EIpu) (Snoek et al., 2012) which we introduce here. Paulson et al. (2022) and Paulson et al. (2023) enable incorporation of local movement constraints in BO using approximate dynamic programming, and by combining local and global steps depending on the local expected improvement. Samaniego et al. (2021) explore the use of truncation for minimizing movement costs under the context of environmental monitoring. Multi-objective optimization has seen recent research interest, in particular at the intersection of chemistry and machine learning (Thebelt et al., 2022; Tu et al., 2022).

## 2. Practical Path-based Bayesian optimization

### 2.1 Self-stopping SnAKe

---
**Algorithm 1** Expected Improvement Point Deletion

---
INPUT: Proposed batch, $\mathcal{P}_t$, of size $(T - t)$. Model-misspecification safety parameter $\nu$, termination parameter $\delta$.

**for** $x_p \in \mathcal{P}_t$ **do**

    **if** ( $EI(x_p)/C^{(0)}(x_p) < \delta$ ) and ( $\text{Var}[\ x_p \mid D_t\ ] < \nu$ ) **then**

        $\mathcal{P}_t \leftarrow \mathcal{P}_t \backslash \{x_p\}$

    **end if**

**end for**

**if** $|\mathcal{P}_t| = 0$ **then**

    Terminate optimization

**end if**

OUTPUT: Reduced Batch, $\tilde{\mathcal{P}}$, of unknown size. Optimization termination criteria.

---

We begin by investigating the first problem. We assume that we want to optimize an expensive black-box function $f$ and that the cost of experiment $t + 1$ can be described by:

$$C(x_t, x_{t+1}) = C^{(0)}(x_{t+1}) + C^{(\Delta)}(x_t, x_{t+1}) \tag{2}$$

where $x_{t+1}$ is the next experimental input we want to query, $x_t$ is our current input, $C^{\Delta}(x_t, x_{t+1})$ is an input-change cost function equal to zero when $x_{t+1} = x_t$, and $C^{(0)}(x_{t+1})$ represents the fixed cost of each experiment. The goal is to optimize the function for the smallest possible total cost. Finding an exact optimum of a black-box function is practically impossible; therefore the BO loop should terminate once we have a good approximation.

To combine BO stopping conditions and SnAKe, we modify the Point Deletion criteria (Folch et al., 2022, Section 3.7) using EIpu (Snoek et al., 2012) as a stopping condition. For a certain point $x$, if the EIpu, $EI(x)/C^{(0)}(x)$, is less than a given threshold $\delta > 0$, then we delete it. We also add an extra criterion based on the prediction variance of the point to avoid early termination in the case where the model is misspecified: we will not delete points that still have significant uncertainty. Using this same criteria (to avoid model misspecification), we were able to significantly boost EIpu's performance, achieving comparable results to SnAKe. We detail the method in Algorithm 1.

### 2.2 Truncated SnAKe

We now investigate the problem of maximum allowable input changes, motivated by, e.g., safety or physical limits. In particular, we assume the optimization now involves the *rate-of-change* constraint:

$$|x_{t+1} - x_t| < \delta_{MAX}, \quad \forall\ t \in \{1, ..., T\} \tag{3}$$

We modify the criteria that SnAKe uses when selecting which experiment to query next: truncate SnAKe's input change whenever it exceeds $\delta_{MAX}$. The modification is shown in Algorithm 2, where $[v]^{(u)}$ denotes the unit-length vector in direction $v$.

---
**Algorithm 2** Truncated SnAKe (TrSnAKe)
---
Run SnAKe as normal with the following extra criterion for each new experiment, $x_{t+1}$:
**if** $|x_{t+1} - x_t| > \delta_{MAX}$ **then**
$\quad x_{t+1} \leftarrow x_t + [x_{t+1} - x_t]^{(u)}\delta_{MAX}$
**end if**
---

## 2.3 Multi-objective SnAKe

We consider a simple multi-objective extension of SnAKe by using random scalarizations of the objective functions (Paria et al., 2020). We define the problem as trying to find the Pareto front of a set of objective functions $\{f^{(k)}(x)\}_{k=1}^{K}$. The Pareto front (in the maximization setting) is defined as the set of points where we cannot increase the value of an objective without decreasing the value of another.

SnAKe creates a large batch of points using Thompson Sampling, which is then considered for evaluation and ordered in such a way that input costs are minimized. Paria et al. (2020) give a multi-objective extension of Thompson Sampling, where we consider a random parametric scalarization of the objectives as a single objective function.

Assume we have $K$ objectives, each modelled by an independent GP. We choose a scalarization function $\mathcal{S}(\{f^{(k)}\}_{k=1}^{K}, \lambda)$ where $\lambda$ is a random parameter with distribution $p(\lambda)$. Specifically, we test a linear scalarization function $\mathcal{S}(\{f^{(k)}\}_{k=1}^{K}, \lambda)(x) = \sum_{k=1}^{K} \lambda_k f^{(k)}(x)$ and Tchebyshev scalarization $\mathcal{S}(\{f^{(k)}\}_{k=1}^{K}, \lambda)(x) = \min_{k=1,...,K} \lambda_k (f^{(k)}(x) - z^{(k)})$, where $z^{(k)}$ are some reference points (Nakayama et al., 2009). The latter can handle non-convex Pareto fronts. To create the batch, we simply sample both the independent GPs and $\lambda \sim p(\lambda)$ and then optimize the single objective (one for each GP sample). We choose this particular multi-objective method for its computational simplicity, which is required when creating large batches. We outline the procedure in Algorithm 3.

---
**Algorithm 3** Multi-objective SnAKe (MO-SnAKe)
---
Run SnAKe as normal, but replace the Thompson Sampling in the batch creation step with Random Scalarization Thompson Sampling. That is, create the batch with the following algorithm:
**for** $i = 1, ..., T$ **do**
$\quad f^{(1)}, ..., f^{(K)} \sim GP(\mu_t, \sigma_t | D_t)$
$\quad \lambda \sim p(\lambda)$
$\quad x_{t+1}^{(i)} = \arg\max_{x \in \mathcal{X}} \mathcal{S}(\{f^{(k)}\}_{k=1}^{K}, \lambda)(x)$
**end for**
---

## 3. Experimental Results

All implementations use GPyTorch (Gardner et al., 2018) and BoTorch (Balandat et al., 2020). To implement SnAKe, we use the non-parametric version of point-deletion when applicable (denoted by an $\ell$ prefix). In all plots, we zoom into the area of interest (low-cost region) and plot $\pm 0.5$ the standard deviations of 25 runs with random initial conditions. Dashed vertical lines in plots show the average cost used. As in Folch et al. (2022), the

hyper-parameters of the GP are optimized by maximizing the marginal-likelihood every 25 iterations. Random refers to uniform samples ordered by solving the Travelling Salesman Problem. We highlight a few benchmarks here and include further studies in the Appendix.

We tested self-stopping SnAKe (ssSnAKe) in three synthetic benchmarks: Branin2D, Hartmann3D and Hartmann6D. We compare against SnAKe, vanilla EIpu, and a new implementation of EIpu that also considers model certainty as termination criteria.

SnAKe achieves the best regret; however, it expends the entire budget and therefore incurs high costs for very small gains. ssSnAKe achieves similar performance to SnAKe up to the point where it automatically terminates. Our modification to EIpu significantly improves the performance, achieving the best results in Hartmann3D and comparable results in Branin2D benchmarks. It is good to note that the ssSnAKe seems robust to the maximum budget, using approximately the same number of experiments, even when the maximum budget is increased. We showcase the Hartmann6D benchmark in Figure 1.
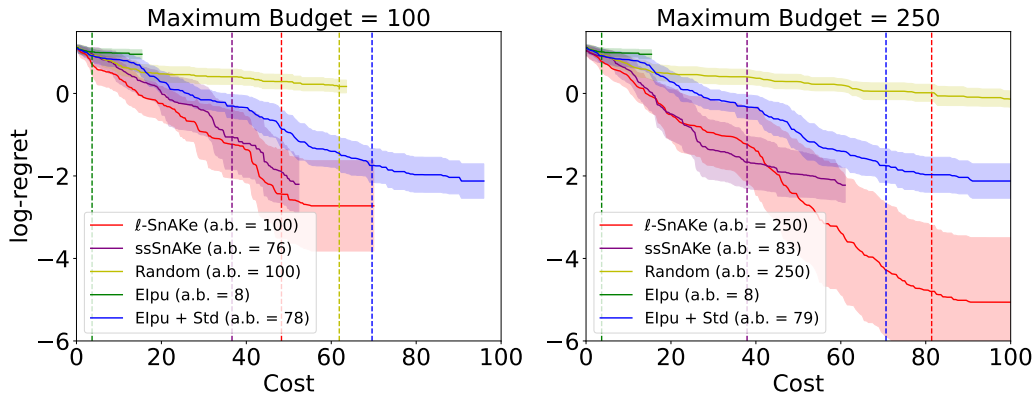


Figure 1: Self-stopping optimization of Hartmann 6D with different maximum budgets. Vanilla EIpu stops too early due to model misspecification; our modification that includes uncertainty criteria greatly improves performance. The average budget used (a.b.) is robust to maximum budget choice.

For rate-of-change constraint problems, we test on two synthetic benchmarks: Branin2D and Hartmann3D. We compare Truncated SnAKe (TrSnAKe), Truncated Expected Improvement (TrEI) (Samaniego et al., 2021), SnAKe, and EIpu. We emulate movement constraints using a cost function with a jump when the movement is larger than $\delta_{MAX}$:

$$\mathcal{C}(x_t, x_{t+1}) = 0.2||x_{t+1} - x_t||_2 + \mathbb{1}(||x_{t+1} - x_t||_2 > \delta_{MAX})$$

For each benchmark, we test with $\delta_{MAX} \in \{0.1, 0.025\}$. TrSnAKe performs best in three out of four cases. Methods that cannot consider movement constraints are heavily penalized and achieve poor performance. Figure 2 shows the results for the Branin2D benchmark.

Finally, we test multi-objective SnAKe (MO-SnAKe) on two benchmarks: the Schekel function, which was used to model pollution in Ypacarai Lake (Samaniego et al., 2021), and the SnAr chemistry benchmark, implemented in Summit (Felton et al., 2021). We compare MO-SnAKe and Scalarized Thompson Sampling (Paria et al., 2020). We note that Folch et al. (2022) used SnAr as a real-world application; however, they restricted the problem to single objective. This work is able to consider the original multi-objective problem.
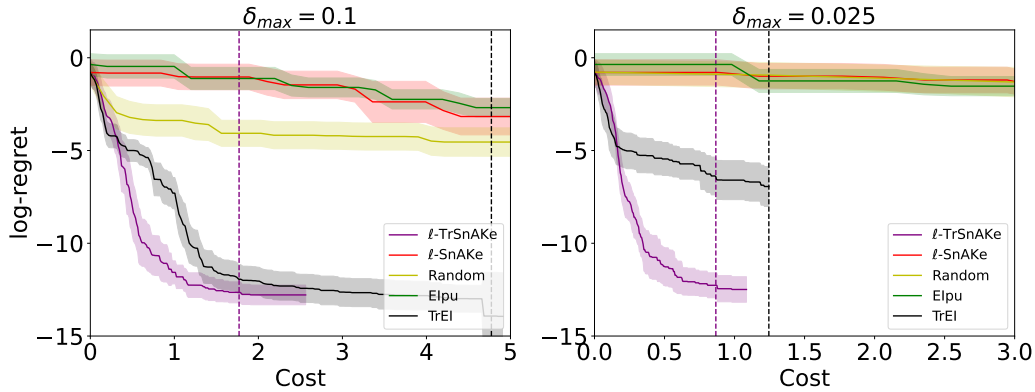
Figure 2: Results for truncated Branin2D with a budget of 250 queries. The cost of non-truncated methods is very high due to penalization for breaking movement constraints. TrSnAKe achieves the best performance for low cost.

Table 1: Results of multi-objective optimization experiments on four dimensional benchmarks. Red and blue indicate linear and Tchebyshev scalarazations respectively.

| | Method | GD | | IGD | MPFE | Cost | |
|---|---|---|---|---|---|---|---|
| Schekel | $\ell$-MO-SnAKe | **0.05** $\pm$ | 0.06 | **0.09** $\pm$ 0.07 | **0.23** $\pm$ 0.08 | **9.60** $\pm$ | 6.12 |
| | $\ell$-MO-SnAKe | 0.07 $\pm$ | 0.05 | 0.15 $\pm$ 0.05 | 0.29 $\pm$ 0.07 | 11.24 $\pm$ | 9.94 |
| | TS | **0.05** $\pm$ | 0.04 | 0.11 $\pm$ 0.07 | 0.24 $\pm$ 0.08 | 57.21 $\pm$ | 12.97 |
| | TS | 0.06 $\pm$ | 0.04 | 0.16 $\pm$ 0.05 | 0.30 $\pm$ 0.07 | 63.18 $\pm$ | 12.30 |
| | Random | 0.22 $\pm$ | 0.05 | 0.22 $\pm$ 0.04 | 0.31 $\pm$ 0.05 | 59.62 $\pm$ | 1.08 |
| SnAr | $\ell$-MO-SnAKe | 0.00040 $\pm$ 0.00015 | | 0.19 $\pm$ 0.06 | 0.67 $\pm$ 0.22 | 614.94 $\pm$ 229.98 | |
| | $\ell$-MO-SnAKe | 0.00044 $\pm$ 0.00020 | | 0.18 $\pm$ 0.07 | 0.65 $\pm$ 0.26 | **478.43** $\pm$ 219.16 | |
| | TS | 0.00035 $\pm$ 0.00008 | | 0.18 $\pm$ 0.06 | 0.64 $\pm$ 0.22 | 1967.56 $\pm$ 447.31 | |
| | TS | **0.00031** $\pm$ 0.00008 | | 0.18 $\pm$ 0.07 | 0.63 $\pm$ 0.25 | 1602.68 $\pm$ 401.3 | |
| | Random | 0.02498 $\pm$ 0.01157 | | **0.16** $\pm$ 0.04 | **0.35** $\pm$ 0.15 | 1195.68 $\pm$ 42.07 | |

For performance metrics, we use Euclidean generational distance (GD) which measures the average distance from the the approximated Pareto front to the true one, inverted generational distance (IGD) which measure the average distance from the true Pareto front to the approximated one, and the maximum Pareto frontier error (MPFE) which measures the largest error from the approximated Pareto frontier to the true one. We obtain a 'true' set of Pareto points by running `NSGA-II` (Deb et al., 2002). Table 1 shows MO-SnAKe performs comparable to Scalarized Thompson Sampling (Paria et al., 2020) at much lower cost.

## 4. Conclusion

In this paper, we explore extensions of SnAKe to path-based BO, to make it feasible to more practical situations. For each situation, we mention areas where the particular modification might be useful, and empirically show the advantages of using our modifications.

# References

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020.

Ilija Bogunovic, Jonathan Scarlett, Andreas Krause, and Volkan Cevher. Truncated variance reduction: A unified approach to Bayesian optimization and level-set estimation. *Advances in Neural Information Processing Systems*, 29, 2016.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

Kobi Felton, Jan Rittig, and Alexei Lapkin. Summit: Benchmarking Machine Learning Methods for Reaction Optimisation. *Chemistry Methods*, February 2021.

Jose Pablo Folch, Shiqiang Zhang, Robert Lee, Behrang Shafei, David Walz, Calvin Tsay, Mark van der Wilk, and Ruth Misener. SnAKe: Bayesian Optimization with Pathwise Exploration. In *Advances in Neural Information Processing Systems*, volume 35, pages 35226–35239, 2022.

Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix Gaussian Process inference with GPU acceleration. *Advances in Neural Information Processing Systems*, pages 7576–7586, 2018.

Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science*, 11(2): 577–586, 2020.

Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using NetworkX. Technical report, Los Alamos National Lab. (LANL), 2008.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabas Poczos. Parallelised Bayesian Optimisation via Thompson Sampling. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pages 133–142, 2018.

Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, 12 2014.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

Mojmir Mutny, Tadeusz Janik, and Andreas Krause. Active Exploration via Experiment Design in Markov Chains. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206, pages 7349–7374, 2023.

Hirotaka Nakayama, Yeboon Yun, and Min Yoon. *Sequential approximate multiobjective optimization using computational intelligence*. Springer Science & Business Media, 2009.

Vu Nguyen, Sunil Gupta, Santu Rana, Cheng Li, and Svetha Venkatesh. Regret for Expected Improvement over the Best-Observed Value and Stopping Condition. In *Asian conference on machine learning*, pages 279–294, 2017.

Biswajit Paria, Kirthevasan Kandasamy, and Barnabás Póczos. A flexible framework for multi-objective bayesian optimization using random scalarizations. In *Uncertainty in Artificial Intelligence*, pages 766–776. PMLR, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, pages 8024–8035. Curran Associates, Inc., 2019.

Joel A. Paulson, Farshud Sorouifar, and Ankush Chakrabarty. Efficient Multi-Step Lookahead Bayesian Optimization with Local Search Constraints. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 123–129, 2022.

Joel A Paulson, Farshud Sorouifar, Christopher R Laughman, and Ankush Chakrabarty. LSR-BO: Local search region constrained Bayesian optimization for performance optimization of vapor compression systems. In *2023 American Control Conference (ACC)*, pages 576–582. IEEE, 2023.

Shyam Sundhar Ramesh, Pier Giuseppe Sessa, Andreas Krause, and Ilija Bogunovic. Movement Penalized Bayesian Optimization with Application to Wind Energy Systems. In *Advances in Neural Information Processing Systems*, volume 35, pages 27036–27048, 2022.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

Federico Peralta Samaniego, Daniel Gutiérrez Reina, Sergio L Toral Marín, Mario Arzamendia, and Derlis O Gregor. A Bayesian Optimization Approach for Water Resources Monitoring Through An Autonomous Surface Vehicle: The Ypacarai Lake Case Study. *IEEE Access*, 9:9163–9179, 2021.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25, 2012.

Alexander Thebelt, Calvin Tsay, Robert M Lee, Nathan Sudermann-Merx, David Walz, Tom Tranter, and Ruth Misener. Multi-objective constrained optimization for energy applications via tree ensembles. *Applied Energy*, 306:118061, 2022.

Ben Tu, Axel Gandy, Nikolas Kantas, and Behrang Shafei. Joint entropy search for multi-objective Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 35, pages 9922–9938, 2022.

Pratibha Vellanki, Santu Rana, Sunil Gupta, David Rubin, Alessandra Sutti, Thomas Dorin, Murray Height, Paul Sanders, and Svetha Venkatesh. Process-constrained Batch Bayesian Optimisation. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017.

James Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Deisenroth. Efficiently Sampling Functions from Gaussian Process Posteriors. In *International Conference on Machine Learning*, pages 10292–10302, 13-18 Jul 2020.

Adam X Yang, Laurence Aitchison, and Henry B Moss. MONGOOSE: Path-wise Smooth Bayesian Optimisation via Meta-learning. *arXiv preprint arXiv:2302.11533*, 2023.

## Appendix A. Acknowledgements

## Appendix B. Complete numerical results

In this section we include further experimental results and more implementation details. As per Folch et al. (2022), we implement Thompson Sampling using the efficient sampling procedure introduced by Wilson et al. (2020), solve the Travelling Salesman using Simulated Annealing (Kirkpatrick et al., 1983) and the NetworkX package (Hagberg et al., 2008). Optimizations of the Thompson Samples are done in PyTorch (Paszke et al., 2019) using Adam (Kingma and Ba, 2014). The code used to produce the results will be available after peer-review in the original SnAKe repository (`https://github.com/cog-imperial/SnAKe`).

### B.1 Early stopping experiments

We tested self-terminating SnAKe on three total benchmarks. Hartmann 6D can be seen in Figure 1. The cost function used in the problem is given by:

$$\mathcal{C}(x_t, x_{t+1}) = \alpha + ||x_{t+1} - x_t||_2 \qquad (4)$$

Where $\alpha = 0.05, 0.1, 0.25$ in each of Branin2D, Hartmann3D, and Hartmann6D respectively. Figure 3 shows the results of the remaining two benchmarks. ssSnAKe gives the strongest performance in Hartmann6D, EIpu with the additional variance stopping condition is strongest in Hartmann3D, and the methods are comparable in Branin2D.

### B.2 Truncated Experiments

We further tested on Hartmann3D for the experiments with rate-of-change constraints. In this case Truncated SnAKe (TrSnAKe) achieves stronger results when the rate of change is larger, giving it more freedom to explore the space. Truncated Expected Improvement (TrEI) gives better results when restricted to a smaller constraint.

### B.3 Multi-objective experiments

Finally, we consider a relaxed Pareto front to explain why Random appears to outperform BO in IGD and MPFE in the SnAr chemistry benchmark. This is due to the Pareto front being almost a horizontal line, and therefore the algorithms prefer to explore the part of the line that maximizes the second objective. We consider a relaxed Pareto front by taking into account all points where we cannot improve on our current objective without decreasing the second objective by more than 0.01 (before this limit was strictly zero).

Using this definition for our approximated Pareto front, Table 2 shows improved performance in IGD and MPFE, at the expense of a decrese in performance in GD (as now point
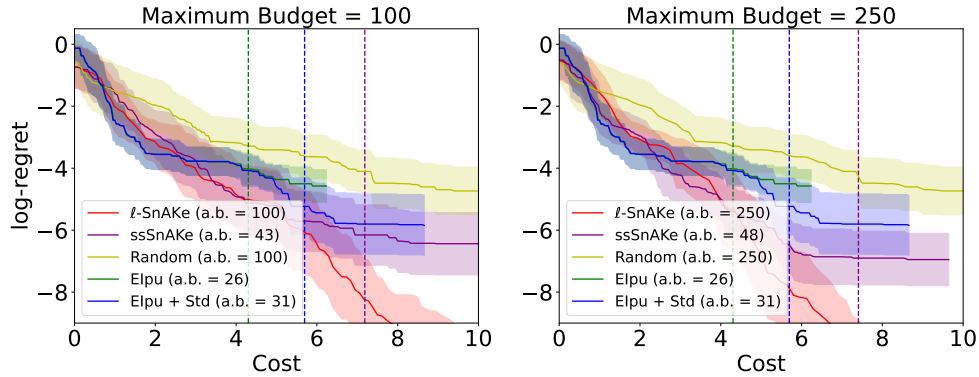
(a) Branin2D
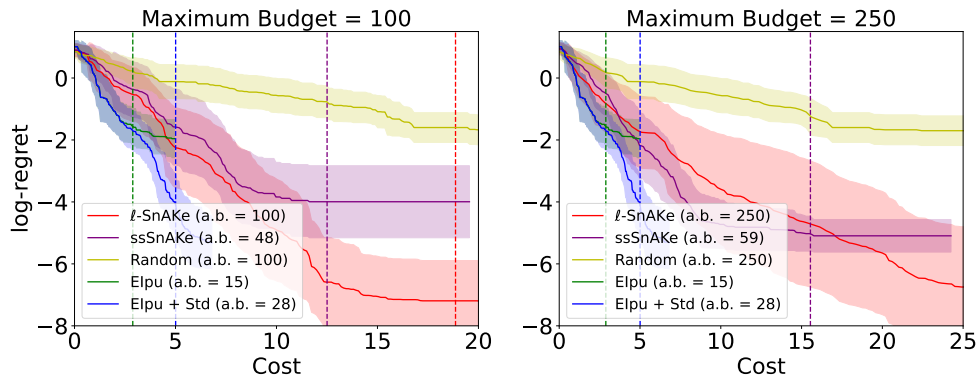


(b) Hartmann3D

Figure 3: Results of self-stopping experiment on remaining synthetic benchmarks. We show average budget (a.b.) used in the captions, and the average cost as vertical lines. In all cases vanilla EIpu seems to terminate too early. Once we have included the uncertainty reduction criteria (EIpu+Std), the performance improves significantly. EIpu+Std achieves the best performance in Hartmann3D, and comparable performance to self-stopping SnAKe (ssSnAKe) in Branin2D. In all cases ssSnAKe follows the performance of SnAKe closely, up to the point of termination. In all cases ssSnAKe and EIpu+Std appear to be robust to the maximum budget, terminating the optimizaiton at approximately the same time even when the maximum budget is increased.

in the approximated Pareto front may not be Pareto optimal) showcasing the difficulty in measuring good performance in multi-objective optimization.
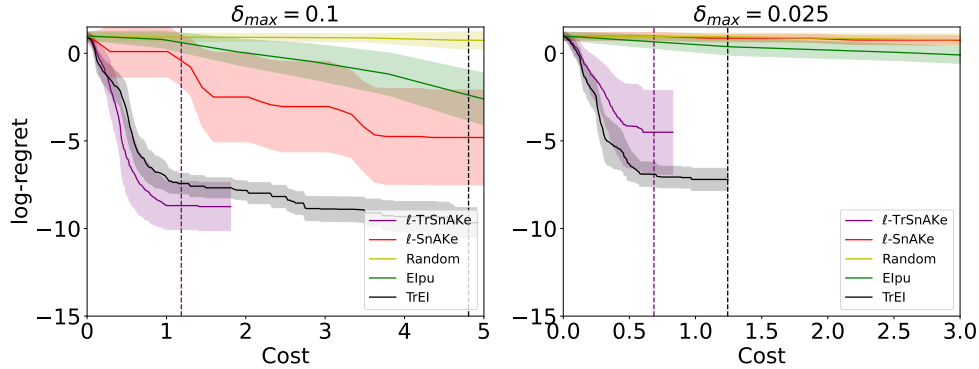
Figure 4: Results for truncated Hartmann3D. The cost of non-truncated methods is very high due to constantly being penalized for breaking movement constraints. We show results for a budget of 250 queries, with two different constraint sizes. TrSnAKe achieves the best performance when the movement constraint is larger, and TrEI the best when the movement constraint is lower.

Table 2: Results of multi-objective optimization experiments on four dimensional benchmarks with relaxed Pareto front. Even though the approximated Pareto front now contains observations known to *not* be Pareto optimal, the performance in the IGD and MPFE metrics improves. Red and blue indicate linear and Tchebyshev scalarazations respectively.

| Problem | Method | GD | | IGD | MPFE | Cost | |
|---|---|---|---|---|---|---|---|
| Schekel | $\ell$-MO-SnAKe | **0.05** $\pm$ | 0.05 | **0.09** $\pm$ 0.07 | **0.22** $\pm$ 0.09 | **9.60** $\pm$ | 6.12 |
| | $\ell$-MO-SnAKe | 0.06 $\pm$ | 0.04 | 0.15 $\pm$ 0.05 | 0.28 $\pm$ 0.07 | 11.24 $\pm$ | 9.94 |
| | TS | **0.05** $\pm$ | 0.04 | 0.11 $\pm$ 0.07 | 0.24 $\pm$ 0.08 | 57.21 $\pm$ | 12.97 |
| | TS | 0.06 $\pm$ | 0.04 | 0.16 $\pm$ 0.05 | 0.30 $\pm$ 0.08 | 63.18 $\pm$ | 12.30 |
| | Random | 0.22 $\pm$ | 0.05 | 0.22 $\pm$ 0.04 | 0.31 $\pm$ 0.05 | 59.62 $\pm$ | 1.08 |
| SnAr | $\ell$-MO-SnAKe | 0.00248 $\pm$ 0.00118 | | **0.06** $\pm$ 0.02 | **0.24** $\pm$ 0.08 | 614.94 $\pm$ 229.98 | |
| | $\ell$-MO-SnAKe | 0.00262 $\pm$ 0.00106 | | **0.06** $\pm$ 0.04 | 0.27 $\pm$ 0.15 | **478.43** $\pm$ 219.16 | |
| | TS | 0.00214 $\pm$ 0.00091 | | 0.07 $\pm$ 0.03 | 0.29 $\pm$ 0.08 | 1967.56 $\pm$ 447.31 | |
| | TS | **0.00171** $\pm$ 0.00047 | | **0.06** $\pm$ 0.02 | **0.24** $\pm$ 0.08 | 1602.68 $\pm$ 401.3 | |
| | Random | 0.02725 $\pm$ 0.01345 | | 0.14 $\pm$ 0.04 | 0.25 $\pm$ 0.09 | 1195.68 $\pm$ 42.07 | |