A penalisation method for batch multi-objective Bayesian optimisation with application in heat exchanger design

Andrei Paleyes   AP2169@CAM.AC.UK
Department of Computer Science and Technology, University of Cambridge

Henry B. Moss & Victor Picheny
Secondmind.ai

Piotr Zulawski & Felix Newman
Reaction Engines

Abstract

We present HIghly Parallelisable Pareto Optimisation (HIPPO) — a batch acquisition function that enables multi-objective Bayesian optimisation methods to efficiently exploit parallel processing resources. Multi-Objective Bayesian Optimisation (MOBO) is a very efficient tool for tackling expensive black-box problems. However, most MOBO algorithms are designed as purely sequential strategies, and existing batch approaches are prohibitively expensive for all but the smallest of batch sizes. We show that by encouraging batch diversity through penalising evaluations with similar predicted objective values, HIPPO is able to cheaply build large batches of informative points. Our extensive experimental validation demonstrates that HIPPO is at least as efficient as existing alternatives whilst incurring an order of magnitude lower computational overhead and scaling easily to batch sizes considerably higher than currently supported in the literature. Additionally, we demonstrate the application of HIPPO to a challenging heat exchanger design problem, stressing the real-world utility of our highly parallelisable approach to MOBO.

Keywords: Multi-objective optimisation, Bayesian optimisation, Batch optimisation

1. Introduction

Bayesian Optimisation (BO, Shahriari et al., 2015) is a well-established strategy to tackle expensive-to-evaluate black-box optimisation problems, and in particular the subclass of problems for which one seeks a trade-off between several equally important but competing objectives. In this so-called *Multi-Objective Optimisation* (MOO) context an improvement in one metric may harm another, and the objective is to find a set of points corresponding to optimal trade-offs, called the *Pareto set*[1]. Numerous problems in science and industry are framed in that way (Park et al., 2018; Nahvi et al., 2019; Avent et al., 2020).

An increasingly popular approach for speeding up MOO is to exploit parallel processing resources, i.e. allowing multiple evaluations to be queried in batches. Access to such parallel resources is common in life science and industry applications, where large batches of experiments can be ran concurrently, or in machine learning, where cloud computing resources can be used to parallelise computation. Although the classical formulation of BO is primarily sequential, extending BO ideas to this batch settings has been an active area of research and there exists many efficient methods for single-objective BO that allow arbitrary degrees of parallelisation. However, for multi-objective BO, most currently available

---

1. Its image in the objective space being referred to as the *Pareto front*.

approaches for parallel optimisation allow only modest batch sizes, and their performance drops significantly as the optimisation loop progresses or as the batch size increases.

In this work, we introduce HIPPO (HIghly Parallelisable Pareto Optimisation), a computationally efficient method for collecting large batches of points within multi-objective BO. Inspired by the penalisation method of González et al. (2016), we propose an iterative batch acquisition strategy with multiplicative distance-based penalisation. Compared to existing alternatives, HIPPO incurs a significantly smaller computational overhead when collecting each new batch point and is agnostic to the surrogate model of the objectives.

2. Related Work

For non-batch BO, Expected Hypervolume Improvement (EHVI, Emmerich, 2005) is a popular acquisition function for MOO. It aims to find the point which induces a Pareto front with maximimal expected improvement in hypervolume. Under the efficient implementation suggested by Yang et al. (2019), EHVI is currently considered a gold standard, and forms the basis of our HIPPO acquisition function.

There are three primary ways to achieve batch BO. Firstly, acquisition functions can be defined over batches of points, for instance the (single objective) multi-point expected improvement (Chevalier and Ginsbourger, 2013) with recent extensions to multi-objective optimisation (Daulton et al., 2020, 2021). These approaches measure a Monte-Carlo approximation of the likely improvement in the Pareto front provided by candidate batches and is henceforth referred to as Batch MC. However, such approaches scale very poorly, as they usually incur a high computational cost and acquisition optimisation over entire batches is challenging.

An alternative approach for batch BO is to define a set of diverse acquisition functions and optimise each one independently, with Thompson sampling being the canonical example. Here, each acquisition function is a sample from the GP model (Vakili et al., 2021). Similar methods can also be applied to multi-objective BO, for example Diversity-Guided Efficient Multi-objective Optimisation (DGEMO, Konakovic-Lukovic et al., 2020) which divides the currently discovered front into sectors and independently picks points from each sector, or random scalarisation-based Thompson sampling (Paria et al., 2020). While these approaches have the advantage of trivial parallelisation, diversity of batches is only controlled indirectly, i.e. there is no guarantee that batches are diverse.

The third approach for batch BO, often referred to as "greedy", constructs a batch by sequentially adding the maximiser of an acquisition function to the batch, then updating the acquisition function based on the location of the newly added point. As examples for single objectives: Kriging Believer (KB, Ginsbourger et al., 2010) conditions the GP on fake observations at the points already added to the batch (effectively modifying the posterior distribution and hence also the acquisition function); local penalisation (LP, González et al., 2016; Alvi et al., 2019) applies multiplicative penalisation terms based on assumed Lipschitz continuity of the objective function; and GIBBON (Moss et al., 2021) sequentially accumulates points expected to provide large reductions in our current uncertainty of the function's maximum value. Among greedy approaches, KB can be used in the multi-objective setting, but only with GP regression models and for very small batch sizes. Recent work has also applied entropy-based method to multi-objective optimization, however, these do not scale to provide batches containing more that a handful of points and so are not considered in

this paper (Qing et al., 2022). Despite convincing performance in the single objective case, no penalisation methods have been proposed yet for MOO: this is the purpose of HIPPO.

3. HIPPO: Penalisation for Multi-objective Optimisation

We now present the primary contribution of this work — a penalisation method for effective batch multi-objective optimisation of a black-box multi-output function. Mathematically, we wish to discover the Pareto front between the k outputs of a function

$$f(x) = (f_1(x), f_2(x), ..., f_k(x)), x \in \mathcal{X} \subset \mathcal{R}^n, f_i(x) \in \mathcal{R}, \tag{1}$$

where n is the dimensionality of the input space, and k — of the output space.

We built HIPPO based on the following principles, which are also shared with the single-objective LP approaches: 1) simplicity to ensure scalability with batch size, 2) model-agnosticity, 3) acquisition function agnosticity. Existing LP approaches directly encourage batch diversity through a repulsion mechanism in the input space, where the strength of this repulsion reduces in the vicinity of the current optimum (points in the batch can be close to each other *only* if we believe they are close to optimal). However, such approaches are not applicable to multi-objective problems where the goal is to discover a set rather than a single point. Ensuring diversity in the input space may not achieve diversity along the Pareto front, as distant parts of the input domain might correspond to the same area of the Pareto front.

Instead, we propose a penalty based on distances in the objective space, i.e. a candidate point will have its acquisition function penalised if its predictive objectives are close to the predicted objectives of a point already added to the batch. This may ensure the generation of batches of points that have high diversity along the Pareto front. Specifically, we consider a warped Mahalanobis distance defined between the posterior distributions of our surrogate models at the candidates points. Note that if we follow a common assumption of independence between Gaussian process models of each objective (Svenson and Santner, 2016), the Mahalanobis distance $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}$ becomes:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} \frac{(\mu(x[i]) - \mu(y[i])^2}{(\sigma(y[i]))^2}}, \tag{2}$$

where $\mu(x) \in \mathcal{R}^k$ and $\sigma(x)^2 \in \mathcal{R}$ are the mean and variance of our surrogate models at location x, respectively. Intuitively, this distance will induce a penalty that increases if both predictive means are similar; and this penalty will increase further if the prediction variance is small, i.e. if the relative position of the points is known with certainty.

Following the construction of González et al. (2016) (where each new point in the batch incurs a multiplicative penalty term to the acquisition), we propose the HIPPO acquisition function for the $(t+1)$st element of the batch as

$$\alpha_{t+1}(x) = \alpha(x) \times \prod_{j=1}^{t} \phi(x, x_j^*) = \alpha(x) \times \prod_{j=1}^{t} w(d(x, x_j)), \tag{3}$$

where $1 \leq t \leq b-1$ is the number of points in the batch collected so far, b is the total batch size, $\phi(.,.)$ is the penalisation function, $w : \mathcal{R}-> [0,1)$ is a smooth warping function and $\alpha : \mathcal{X} \rightarrow \mathcal{R}$ is the well-known (but non-batch) EHVI acquisition function.
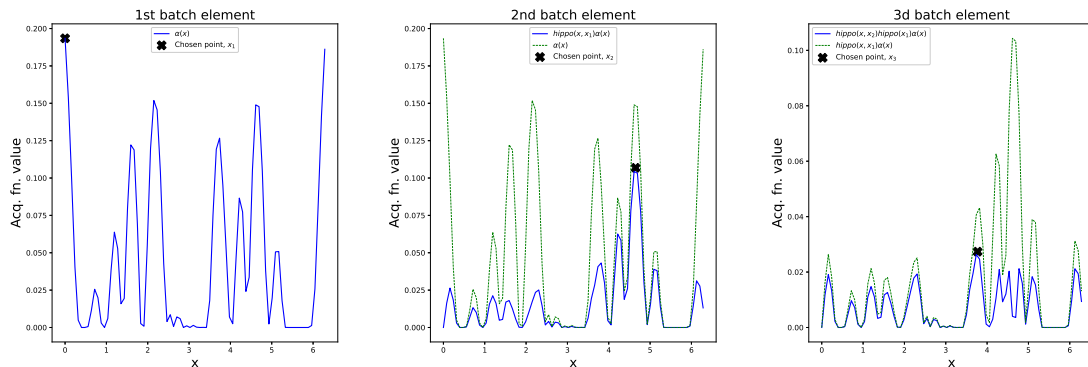
3

Figure 1: The process of collecting three consecutive batch points with HIPPO. The first point x_1 is collected by maximising the acquisition function $\alpha(x)$ (left). The second point x_2 is obtained by multiplying the acquisition function $\alpha(x)$ with the HIPPO penaliser $hippo(x, x_1)$ and maximising the resulting penalised acquisition (center). Third point x_3 is obtained by applying HIPPO penalisation to $\alpha(x)$ at both x_1 and x_2 (right). Dashed green lines on steps 2 and 3 show the acquisition function from the previous step.

In practice we found HIPPO to be resilient to a particular choice of warping function w as long as it scaled the distance value to $[0, 1)$. For our experiments we used $w(y) = \frac{2}{\pi} \times arctan(y)$. Figure 1 illustrates the first three steps of the batch collection process with HIPPO for a simple function with 1D input.

4. Experimental evaluation

4.1 Benchmark results

This section compares HIPPO with several state-of-the-art methods for multi-objective batch BO on a set of commonly used benchmark functions, considering first modest and then large batch sizes. As baselines for comparison we use Batch MC, KB and DGEMO. To measure the performance of each method we use hypervolume (HV) regret, the difference between the hypervolumes defined by the discovered Pareto fronts and the exact ones. If $HV(\mathcal{P})$ is a function that calculates hypervolume of a given Pareto front $\mathcal{P}$ (Wagner et al., 2010), then HV regret is understood as $\Delta H = HV(\mathcal{P}_0) - HV(\mathcal{P}_1)$, where $\mathcal{P}_0$ is the exact Pareto front of a given benchmark function (which we calculated with pymoo (Blank and Deb, 2020)), and $\mathcal{P}_1$ is the currently discovered Pareto front.

Additionally we measured the time taken for a single optimisation step to show the gain in computational efficiency HIPPO offers compared to other methods. Each optimisation was repeated 10 times. Batches of size four were used for all experiments, while the number of optimisation steps depended on the problem. We used the `trieste` library (Berkeley et al., 2022) to implement all the experiments[2]. We used following benchmark functions: *VLMOP2* (van Veldhuizen and Lamont, 1999), *DTLZ2* (Deb et al., 2002), and combination of *Hartmann* and *Ackley* functions (Picheny et al., 2013; Adorio and Diliman, 2005).

Results of the experiments are shown on Figure 2. As the HV regret plots show, HIPPO produces Pareto fronts that are on par or better with those produced by other methods.

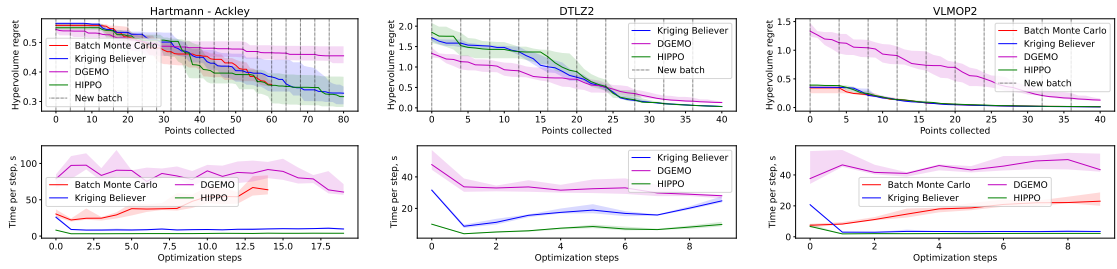2. Experiments code can be found at `https://github.com/apaleyes/hippo`

Figure 2: HV regret (top) and time per optimisation step (bottom) on optimisation benchmark functions. The solid lines show the mean and the shaded areas the 25th to 75th percentiles envelope. Batch MC performance is not shown for DTLZ2 and last five iterations on Hartmann - Ackley because it was running out of memory in our tests.
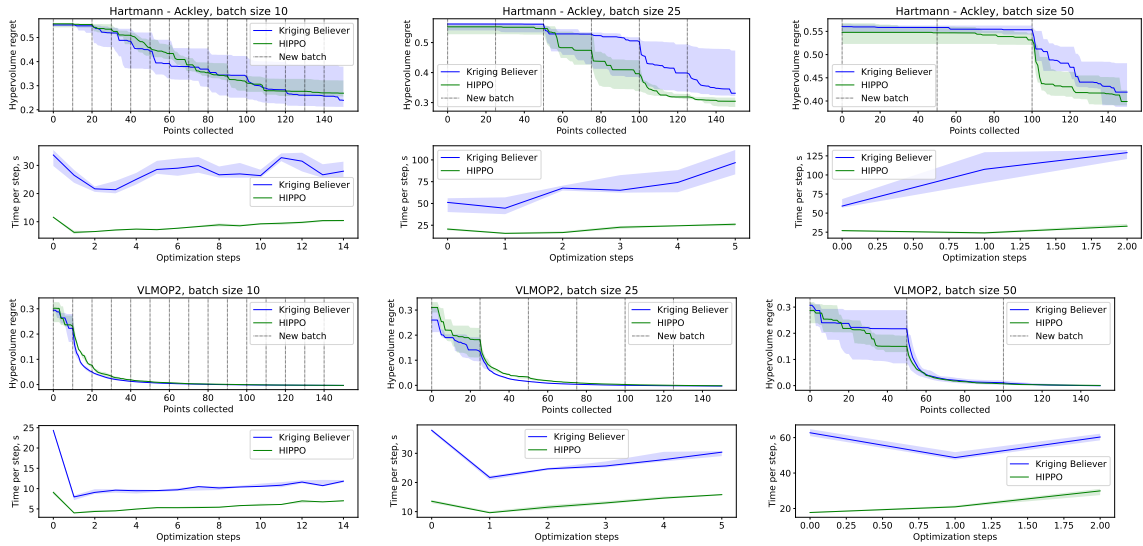

Figure 3: HV regret and time per optimisation step with variable batch sizes. The solid lines show the mean and the shaded areas the 25th to 75th percentiles envelope.

However, as can be seen on the bottom row, even for a modest batch size of 4 HIPPO introduces significantly smaller computational overhead than Batch MC and DGEMO.

To study scalability of HIPPO, we conducted further experiments with variable batch size. Here we are using *Hartmann - Ackley* and *VLMOP2* as test problems, and *KB* as a baseline, since this method showed the best computational performance of all baselines used in the previous experiment. We conducted the experiment for batch sizes 10, 25 and 50. All experiments were run with the total budget of 150 points, thus yielding 15, 6 and 3 optimisation steps respectively. Experiment results can be found on Figure 3. Again, we can observe very close results in terms of HV regret, however HIPPO has clearly lower computational overhead that grows slowly with the increase in batch size.

4.2 Optimisation of a heat exchanger

We now consider the real-world MOO task of designing an effective yet light-weight automotive heat exchanger (radiator). The heat exchanger is parameterised with respect to its
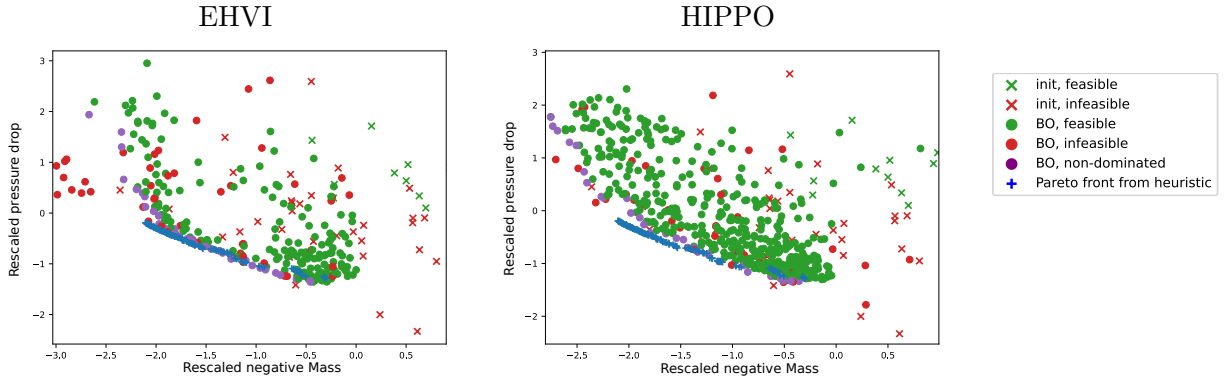
Figure 4: Observation sets in the objective space obtained by EHVI and HIPPO on the heat exchanger problem. Both objectives need to be minimised and the utopia point is set as the bottom left corner.

dimensions and internal geometry. The performance of a particular configuration is given by an expensive-to-evaluate and highly non-linear digital twin based on aero-thermal analysis (a coupling of heat transfer and fluid dynamics).

The input space consists of 9 parameters, with 6 varying continuously between a lower and an upper bound (e.g. radius of component, etc.), as well as 3 taking a finite number of values (e.g. number of modules, etc.). The two objectives are mass and air-side pressure drop. Additionally, as a minimal performance is requested on two other outputs (heat load and tube-side pressure drop), we also have two inequality constraints. Finally, some configurations are physically impossible or create very unstable conditions which lead to evaluation crashes. We encoded this as an additional binary constraint.

The two objectives and two quantitative constraints are modelled with standard GPR models (with a Matern 5/2 kernel and constant mean function), while the binary constraint is modelled with a variational GP model with Bernoulli likelihood. As a base acquisition function, we use here EHVI, multiplied by the probability that the three constraints are satisfied (as recommended in Emmerich et al., 2006; Shimoyama et al., 2013).

We compare a purely sequential EHVI run with HIPPO using a batch size of 20. Both start with 100 observations (randomly sampled across the input space) and 500 additions. To assess the overall performance of both, a bespoke evolutionary strategy is also ran for a very large number of iterations, which represents the current solutions used in the design of heat exchangers.

Both Pareto fronts are shown in Figure 4. One can see that the fronts are broadly similar, matching or improving upon the evolutionary baseline on the right of the front and providing a wider range of solutions on the left-most part. Both EHVI and HIPPO returned a balanced set of feasible and non-feasible solutions and a relatively uniform coverage of the Pareto front. However, while EHVI concentrated most of its search on the rightmost part of the front, HIPPO shows a much more balanced search, as expected by its definition. Overall, despite using large batches (20), HIPPO provides comparable performance to EHVI, hence allowing consirable speed-ups of the optimisation pipeline.

References

Ernesto P Adorio and U Diliman. MVF-multivariate test functions library in C for unconstrained global optimization. *Quezon City, Metro Manila, Philippines*, 44, 2005.

Ahsan S. Alvi, Binxin Ru, Jan-P. Calliess, Stephen J. Roberts, and Michael A. Osborne. Asynchronous batch Bayesian optimisation with improved local penalisation. In *ICML*, 2019.

Brendan Avent, Javier Gonzalez, Tom Diethe, Andrei Paleyes, and Borja Balle. Automatic discovery of privacy–utility Pareto fronts. *Proceedings on Privacy Enhancing Technologies*, 2020(4):5–23, 2020.

Joel Berkeley, Henry B. Moss, Artem Artemev, Sergio Pascual-Diaz, Uri Granta, Hrvoje Stojic, Ivo Couckuyt, Jixiang Qing, Nasrulloh Loka, Andrei Paleyes, Sebastian W. Ober, and Victor Picheny. Trieste, 2022. URL https://github.com/secondmind-labs/trieste.

J. Blank and K. Deb. pymoo: Multi-objective optimization in Python. *IEEE Access*, 8: 89497–89509, 2020.

Clément Chevalier and David Ginsbourger. Fast computation of the multi-points expected improvement with applications in batch selection. In *International Conference on Learning and Intelligent Optimization*, pages 59–69. Springer, 2013.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. *Advances in Neural Information Processing Systems*, 33:9851–9864, 2020.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *Advances in Neural Information Processing Systems*, 34, 2021.

Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 1, pages 825–830. IEEE, 2002.

Michael Emmerich. *Single- and multi-objective evolutionary design optimization assisted by Gaussian random field metamodels.* PhD thesis, Dortmund, Univ., Diss., 2005, 2005.

Michael TM Emmerich, Kyriakos C Giannakoglou, and Boris Naujoks. Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. Kriging is well-suited to parallelize optimization. 2010.

Javier González, Zhenwen Dai, Philipp Hennig, and Neil D. Lawrence. Batch Bayesian optimization via local penalization. In *AISTATS*, 2016.

Mina Konakovic-Lukovic, Yunsheng Tian, and Wojciech Matusik. Diversity-guided multi-objective Bayesian optimization with batch evaluations. In *NeurIPS*, 2020.

Henry B. Moss, David S. Leslie, Javier González, and Paul Rayson. GIBBON: General-purpose Information-Based Bayesian OptimisatioN. *J. Mach. Learn. Res.*, 22:235:1–235:49, 2021.

Ali Nahvi, Mohammadkazem Sadoughi, Ali Arabzadeh, Alireza Sassani, Chao Hu, Halil İbrahim Ceylan, and Sunghwan Kim. Multi-objective Bayesian optimization of super hydrophobic coatings on asphalt concrete surfaces. *J. Comput. Des. Eng.*, 6:693–704, 2019.

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.

Seongeon Park, Jonggeol Na, Minjun Kim, and Jong Min Lee. Multi-objective Bayesian optimization of chemical reactor design using computational fluid dynamics. *Computers & Chemical Engineering*, 119:25–37, 2018.

Victor Picheny, Tobias Wagner, and David Ginsbourger. A benchmark of kriging-based infill criteria for noisy optimization. *Structural and multidisciplinary optimization*, 48(3): 607–626, 2013.

Jixiang Qing, Henry B. Moss, Tom Dhaene, and Ivo Couckuyt. PF2ES: Parallel feasible Pareto frontier entropy search for multi-objective Bayesian optimization under unknown constraints. *ArXiv*, abs/2204.05411, 2022.

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, 2015.

Koji Shimoyama, Koma Sato, Shinkyu Jeong, and Shigeru Obayashi. Updating kriging surrogate models based on the hypervolume indicator in multi-objective optimization. *Journal of Mechanical Design*, 135(9):094503, 2013.

Joshua Svenson and Thomas J. Santner. Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models. *Comput. Stat. Data Anal.*, 94:250–264, 2016.

Sattar Vakili, Henry Moss, Artem Artemev, Vincent Dutordoir, and Victor Picheny. Scalable Thompson sampling using sparse Gaussian process models. *Advances in Neural Information Processing Systems*, 34, 2021.

David A. van Veldhuizen and Gary B. Lamont. Multiobjective evolutionary algorithm test suites. In *Proceedings of the 1999 ACM Symposium on Applied Computing*, SAC '99, page 351–357, New York, NY, USA, 1999. Association for Computing Machinery. ISBN 1581130864. doi: 10.1145/298151.298382. URL https://doi.org/10.1145/298151.298382.

Tobias Wagner, Michael Emmerich, André Deutz, and Wolfgang Ponweiser. On expected-improvement criteria for model-based multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 718–727. Springer, 2010.

Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. Efficient computation of expected hypervolume improvement using box decomposition algorithms. *Journal of Global Optimization*, 75(1):3–34, 2019.