ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-THPG85

UPDATES TO Xopt FOR ONLINE ACCELERATOR OPTIMIZATION AND CONTROL

R. Roussel*, D. Kennedy, T. Boltz, C. Mayes, A. Edelen
SLAC National Accelerator Laboratory, Menlo Park, CA, USA
K. Baker, Science and Technology Facilities Council, Swindon, United Kingdom

Abstract

The recent development of advanced black box optimization algorithms has promised order of magnitude improvements in optimization speed when solving accelerator physics problems. These algorithms have been implemented in the Python package Xopt, which has been used to solve online and offline accelerator optimization problems at a variety of facilities, including SLAC, Argonne, BNL, DESY, and ESRF. This work describes updates to the Xopt framework that expand its capabilities and improve optimization performance in solving online optimization problems. We also discuss how Xopt has been incorporated into the Badger graphical user interface that allows easy access to these advanced control algorithms in the accelerator control room. Finally, we explain how to integrate machine learning-based surrogate models provided by the LUME-model package into online optimization via Xopt.

INTRODUCTION

Advanced optimization algorithms, such as Bayesian optimization (BO) [1], RCDS [2], CNSGA [3], and extremum seeking [4], have been developed to solve complex optimization problems in accelerator physics, such as online accelerator tuning or simulated accelerator design. Despite their wide applicability, connecting these algorithms to optimization problems can be challenging due to the diverse algorithmic and measurement interfaces. To address this issue, we developed the Xopt framework in Python [5, 6], which implements a wide range of advanced optimization algorithms for easy use by non-experts. On the other hand, Xopt modules are flexible and extendable, allowing experienced users to customize and develop optimization algorithms for novel applications.

Xopt connects optimization algorithms to arbitrary problems using a modular approach which defines separate objects for input and output parameter spaces, algorithm options, and objective/constraint function evaluation. Evaluating objectives and constraints is defined using a simple Python callable that takes a dictionary of arguments and returns a dictionary of results. This flexibility allows Xopt (and by extension Xopt algorithms) to be used in the same way at multiple different accelerator facilities (SLAC, Argonne, DESY, BNL, ERSF, etc.) or in simulations. Additionally, there are multiple ways to interact with the Xopt library depending on the application, as shown in Fig. 1. This includes a text file based interface for dispatching parallel job evaluation on high performance computing clusters, a Python-based API for complex algorithm testing and development, and a GUI interface known as Badger [7] for online use inside accelerator control rooms.

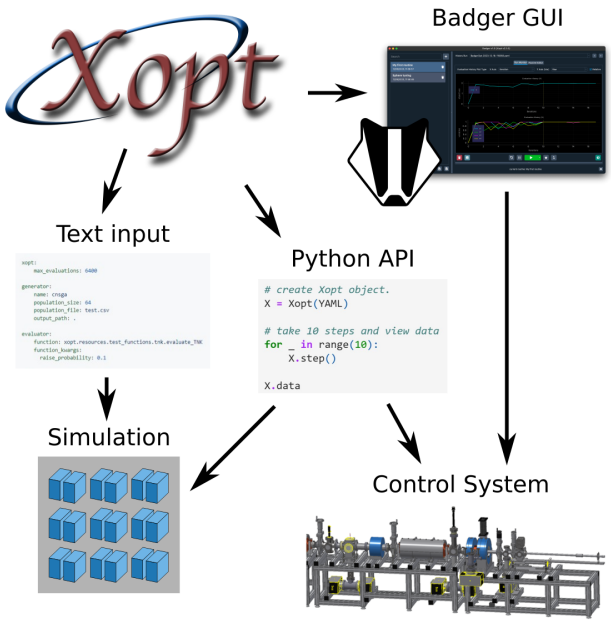

Figure 1: Xopt ecosystem for connecting advanced optimization algorithms with arbitrary optimization problems in accelerator physics.

In this work, we describe recent enhancements of Bayesian optimization algorithms in Xopt, which are applicable to improving the performance and capabilities for online and offline accelerator optimization.

INTERPOLATED MEASUREMENTS

When using Xopt to perform online accelerator control, it is important to consider the computational costs of making control decisions relative to the cost of evaluating objectives and constraints. For example, evaluating the objectives and/or constraints takes significantly less time than the algorithm to make a control decision, it is optimal to make multiple measurements in-between calls to the algorithm to make control decisions. Furthermore, when using algorithms to make accelerator control decisions, it is necessary to reduce the size of parameter changes to maintain the stability of accelerator components and internal feedback mechanisms.

To address both of these issues, we added the capability of performing interpolated measurements to BO

* rroussel@slac.stanford.edu

ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-THPG85

algorithms implemented in Xopt, through the keyword `n_interpolate_points`. Instead of simply changing the parameters of the accelerator to the ones proposed by the BO algorithm, we generate a sequence of parameters in small steps along the line from the current parameter set point to the point proposed by BO. We then sequentially evaluate these points by altering the accelerator parameters and then measuring the objectives/constraints at each point. This reduces the instantaneous size of parameter changes from one measurement to the next, and adds additional data to the Gaussian process (GP) model used in BO to make decisions, without incurring additional computational costs, as discussed in [1]. While this technique is likely to reduce overall optimization costs due to additional measurement data per optimization step, it may have some drawbacks. First, it is primarily useful when optimizing objectives that are inexpensive to evaluate, as more evaluations are done over the course of optimization. Second, since more data is gathered over the course of optimization, decision-making costs of the BO algorithm will increase, since the computational costs of evaluating Gaussian process models scales as $O(N^3)$. Effective use of this technique will need to balance the trade-offs between gathering more information about the objective function and additional computational expense.

TRUST REGION BAYESIAN OPTIMIZATION

One drawback of BO algorithms is that they tends to over-prioritize exploration over exploitation in high-dimensional parameter spaces. This is due to the relatively large posterior uncertainties of GP models that result from the exponential growth of parameter space volume with dimensionality (models in high dimensional space need more data to update prior function distributions). As a result, BO tends to pick points at the extremes of the domain in high dimensional parameter spaces even if optimal points are found in a local region. Trust region BO (TurBO) [8] addresses these issues by restricting optimization of the acquisition function to a so-called “trust region" around previous measurements where the model is expected to be the most accurate. The trust region is a local region centered at the best previously observed measurement so far during optimization, with side lengths equal to a base length L multiplied by the relative length scale of the GP model along each axis in parameter space. As optimization progresses, the location and size of the trust region is continuously updated to be centered at the best measured point in parameter space and scaled to match length scales of the GP model.

We implemented TurBO in Xopt for single objective Bayesian algorithms via the `turbo_controller` property of Bayesian generators. Using the keyword argument `turbo_controller="optimize"` implements TurBO for single objective problems, with a default side length of 25% of the input domain specified by VOCS and a threshold for expanding or contracting the trust region according to the recommendations given in [8]. TurBO in Xopt is used regularly at SLAC as part of beam and FEL optimization. It was also used at ESRF to tune a large number of sextupole magnets in order to improve synchrotron lifetime, as shown in Fig. 2 and reported in [9].

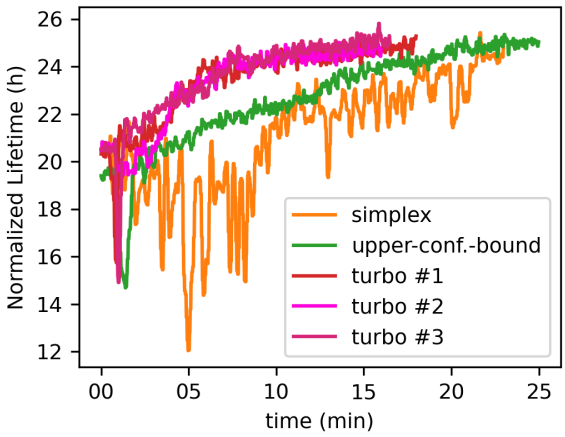

Figure 2: Demonstration of using TurBO to optimize sextupoles at ESRF compared to basic online tuning algorithms and raster scans performed by operators. In this case, TurBO is used through the Badger GUI. Reproduced from [1].

BAYESIAN ALGORITHM EXECUTION FOR VIRTUAL OBJECTIVES

In some cases, beam properties are difficult to optimize since they require multiple measurements of the beam distribution to determine the property of interest. For example, measurements of the beam centroid as a function of quadrupole strengths can be used to determine the centroid offset of the beam with respect to the magnetic center of the quadrupole. Additionally, characterizing the beam emittance requires measurements of the RMS beam size at a number of different quadrupole strengths. Multiple measurements increase the cost of optimizing these objectives and does not efficiently use information gathered during objective calculation.

To address these optimization challenges we have implemented the Bayesian Algorithm Execution (BAX) optimization algorithm that uses so-called “virtual objectives” [10]. Virtual objectives, such as the rate of centroid deflection as a function of quadrupole strength or transverse beam emittance [11], are characterized not by direct measurement, but are calculated from GP model predictions of underlying observables, the beam centroid and size respectively. An example of this is shown in Fig. 3, where BAX is used to optimize the steering magnet parameter in order to minimize the centroid deflection as a function of the quadrupole strength (which corresponds to good alignment with the quadrupole magnetic center). Additionally, in the case shown here, we explicitly specify a Polynomial kernel function for the GP model based on physics knowledge that the centroid posi-

ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-THPG85

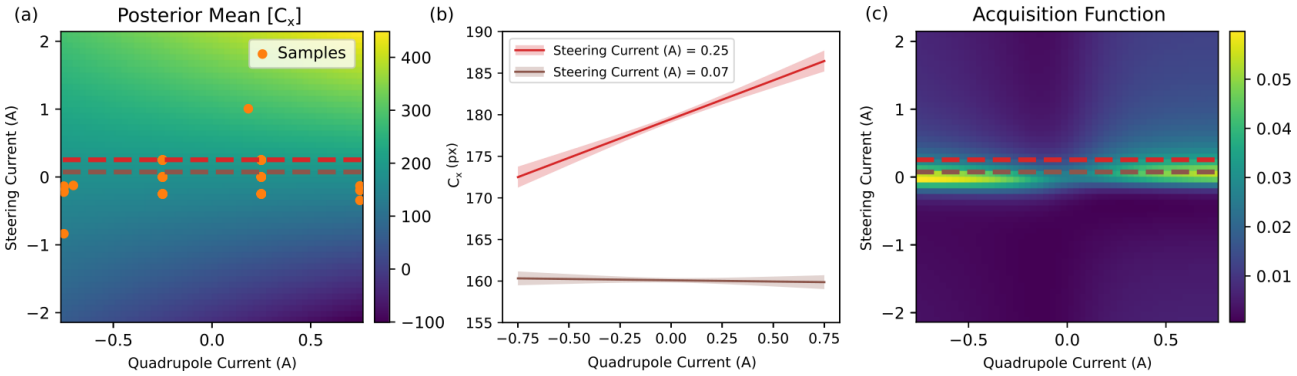

Figure 3: Visualization of the BAX process for beam steering through quadrupole magnets. (a) Experimental measurements are used to build a GP model of the horizontal beam centroid position at a downstream screen C_x as a function of the quadrupole strength and steering parameter. Note that the GP model is built with a 1st order polynomial kernel, constraining predictions to planar surfaces. Dashed lines denote cross sections of the GP model shown in (b). (c) The BAX acquisition function which predicts the information gained about the ideal steering current by making future measurements.

tion is linear with respect to quadrupole and steering magnet strengths. This functionality in Xopt is now regularly used in AWA operations to align the beam through two quadrupole magnets in approximately than 10 minutes, about 3 times faster than human operators.

NON-CONSTANT PRIOR MEANS

We can speed up optimization by incorporating prior notions of the objective function into the GP model [12, 13]. To support this technique in practice, Xopt now allows incorporating any PyTorch Module into the GP as a prior mean function through the `StandardModelConstructor` class. This can be combined with LUME-Model [14] which provides a wrapper around neural network modules which allows for dictionary based inputs/outputs and unit conversion in and out of the neural network. By incorporating a prior model from simulated data, we were able to significantly improve convergence to objectives on the real machine using this feature of Xopt, as shown in Fig. 4 and reported in [13].

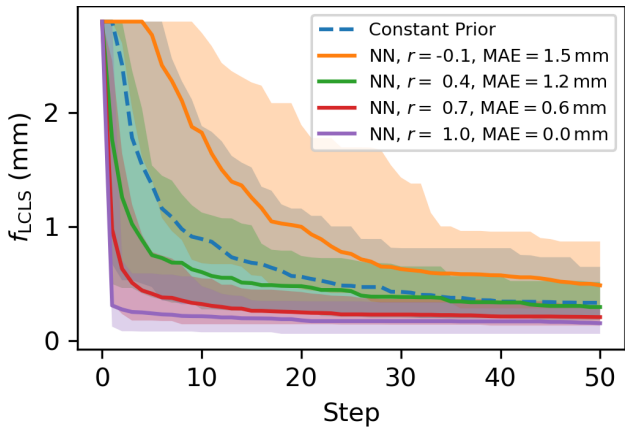

Figure 4: Demonstration of using neural network prior means in Xopt to increase convergence speed of optimization on the LCLS injector problem. Reproduced from [13].

CONCLUSION

This work reports enhancements to the Xopt library, which implements state-of-the-art algorithms to tackle simulated and experimental optimization challenges. These features are used during daily accelerator operations at a number of accelerator facilities, accessible from text, programmatic, and graphical user interfaces.

ACKNOWLEDGEMENTS

This work was funded by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Contract No. DE-AC02-76SF00515.

REFERENCES

[1] R. Roussel *et al.*, "Bayesian optimization algorithms for accelerator physics", 2023. doi:10.48550/arXiv.2312.05667

[2] X. Huang, J. Corbett, J. Safranek, and J. Wu, "An algorithm for online optimization of accelerators", *Nucl. Instrum. Methods Phys. Res., Sect. A*, vol. 726, pp. 77–83, 2013. doi:10.1016/j.nima.2013.05.046

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii", *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002. doi:10.1109/4235.996017

[4] A. Scheinker, E.-C. Huang, and C. Taylor, "Extremum seeking-based control system for particle accelerator beam loss minimization", *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 5, pp. 2261–2268, 2022. doi:10.1109/TCST.2021.3136133

[5] R. Roussel, A. Edelen, A. Bartnik, and C. Mayes, "Xopt: A simplified framework for optimization of accelerator problems using advanced algorithms", in *Proc. IPAC'23*, Venezia, 2023, pp. 4796–4799. doi:doi:10.18429/jacow-ipac2023-thpl164

[6] R. Roussel *et al.*, *Christophermayes/xopt: Xopt v2.2.3*, version v2.2.3, 2024. doi:10.5281/zenodo.11130509

ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-THPG85

[7] Z. Zhang *et al.*, "Badger: The missing optimizer in acr", 2022. doi:10.18429/JACoW-IPAC2022-TUPOST058

[8] D. Eriksson, M. Pearce, J. Gardner, R. D. Turner, and M. Poloczek, "Scalable global optimization via local bayesian optimization", in *Proc. 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 5496–5507. https://proceedings.neurips.cc/paper_files/paper/2019/file/6c990b7aca7bc7058f5e98ea909e924b-Paper.pdf

[9] S. Liuzzo *et al.*, "Optimisation of the touschek lifetime in synchrotron light sources using badger", in *Proc. 19th Int. Conf. Accel. Large Exp. Phys. Control Syst. (ICALEPCS'23)*, Cape Town, South Africa, 2023, pp. 108–115. doi:10.18429/JACoW-ICALEPCS2023-MO3AO01

[10] W. Neiswanger, K. A. Wang, and S. Ermon, "Bayesian algorithm execution: Estimating computable properties of black-box functions using mutual information", in *Proc. International Conference on Machine Learning (ICML 2021)*, PMLR, 2021, pp. 8005–8015.

[11] S. A. Miskovich *et al.*, "Multipoint-bax: A new approach for efficiently tuning particle accelerator emittance via virtual objectives", *Mach. Learn.: Sci. Technol.*, vol. 5, no. 1, p. 015 004, 2024.

[12] K. Hwang *et al.*, "Prior-mean-assisted Bayesian optimization application on FRIB Front-End tunning", 2022. doi:10.48550/arXiv.2211.06400

[13] T. Boltz *et al.*, "More sample-efficient tuning of particle accelerators with bayesian optimization and prior mean models", 2024. doi:10.48550/arXiv.2403.03225

[14] C. Mayes *et al.*, "Lightsource Unified Modeling Environment (LUME), a Start-to-End Simulation Ecosystem", in *Proc. IPAC'21*, Campinas, SP, Brazil, 2021, paper THPAB217, pp. 4212–4215. doi:10.18429/JACoW-IPAC2021-THPAB217