Adapting Multi-Objective Bayesian Optimization for Online Particle Accelerator Tuning

Ryan Roussel
Department of Physics
University of Chicago
Chicago, IL 60637
rroussel@uchicago.edu

Adi Hanuka
SLAC National Laboratory
Menlo Park, CA, 94025
adiha@slac.stanford.edu

Auralee Edelen
SLAC National Laboratory
Menlo Park, CA, 94025
edelen@slac.stanford.edu

Abstract

Particle accelerators require constant tuning during operation to meet goals for beam quality, total charge and particle energy for use in a wide variety of physics, chemistry and biology experiments. Maximizing the performance of an accelerator facility often necessitates multi-objective optimization, where operators must balance trade-offs between objectives, often using limited, real-time, temporally expensive beam observations. Unfortunately parallelized methods typically used to solve multi-objective problems don't have sufficient sample efficiency to be used practically during accelerator operation. This is due, in part, because fitness evaluation of a given input must be done serially. Here, we introduce modifications to a multi-objective Bayesian optimization scheme for use in practical particle accelerator control algorithms, by including optimization constraints, objective preferences and localized parameter tuning.

1 Introduction

Optimizing particle accelerator parameters during operation (i.e. "online tuning") is a tedious but often necessary part of any accelerator facility's daily operation. Due to their large number of components and variability of external factors, such as vibrations or temperature changes, accelerators must be continuously re-tuned and optimized to meet various beam quality objectives. This often requires hours of tuning by experienced operators to maintain accelerator performance, which in turn, reduces the overall scientific output as experimenters using the facility do not have access to the beam during these periods.

Accelerator optimization can be framed as a multi-objective optimization problem, as several aspects of the beam must be optimized simultaneously. However, parallelized methods previously used to solve multi-objective problems such as genetic [1] or swarm [2] optimization are not suitable for practical online optimization where the fitness of each individual can only be evaluated in serial. Since parallelized multiple objective algorithms require thousands of observations to effectively find the Pareto front, take too much time to converge for use in online tuning. On top of this, accelerator measurements often take significant time and resources to preform, further increasing time required for optimization.

We use the recent development of Multi-Objective Bayesian Optimization (MOBO) [3] to enable serialized optimization of accelerators when solving multi-objective problems. Optimization of multi-objective problems entails finding the Pareto front $\mathcal{P}$, which is a set of points in objective space that optimally balances the trade-offs between each objective, as optimizing one objective often comes at the expense of another. In the case described here we wish to minimize each objective relative to a reference point $\mathbf{r}$ which represents the largest expected values of each objective. The figure of merit for a Pareto front is its hypervolume indicator $\mathcal{H}$, which is the volume in N-dimensional

Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020), Vancouver, Canada.

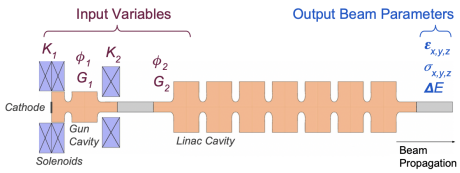

Figure 1: Cartoon of the AWA photoinjector and the first linac cavity. Input variables and output beam parameters used in optimization are labeled. Reproduced from [6].

sub-domain defined by the Pareto front set and the reference point. As detailed in [3], Gaussian processes [4] corresponding to each objective can be used to predict which points in input space are likely to optimally increase the hypervolume until convergence to the true hypervolume is reached. This serialized optimization strategy efficiently samples the input space to reduce the number of observations needed to find the Pareto front, enabling it to be used in online accelerator optimization. In this paper, we demonstrate how making slight modifications the hypervolume improvement acquisition function enables us to solve unique issues that make accelerator optimization difficult, including specifying optimization preferences, objective constraints and smooth exploration of input parameter space.

2 Adapting MOBO to Particle Accelerator Problems

A diagram showing the problem we wish to solve is shown in Figure 1. The Argonne Wakefield Accelerator (AWA) at Argonne National Lab generates electron bunches by shining a laser pulse onto a metallic cathode inside a photo-injector [5] and then using radio frequency fields to accelerate the electron bunch to the right. We wish to minimize 7 output beam parameters of the electron bunch at the end of this portion of the accelerator. These parameters (colored blue in Figure 1) include the bunch size in three dimensions $\sigma_x, \sigma_y, \sigma_x$, the phase space area of electron trajectories in three dimensions $\varepsilon_x, \varepsilon_y, \varepsilon_z$ and the energy spread of the electrons δE. These beam attributes are influenced by several input variables (colored red in Figure 1). These include the magnetic field strength of a pair of solenoids (K_1, K_2) and the amplitude (G_1, G_2) and phase (ϕ_1, ϕ_2) of accelerating electric fields.

Before using this algorithm to control the physical accelerator, we developed and tested it on a simulated model. Normally, a full 3D physics simulation is used to predict beam attributes from a given set of parameters, however recent progress in using neural network based surrogate models of particle accelerators [6] allowed us to speed up algorithm development significantly by using a neural network surrogate model that executed roughly $\mathcal{O}(10^6)$ faster than conventional 3D physics simulations.

We base our multi-objective optimization strategy on the hypervolume based acquisition function developed in [3]. An independent Gaussian process regressor is used to model each beam attribute (7 in total). Since the dimensionality of objective space is very large, we used the upper confidence bound hypervolume improvement acquisition function (UCB-HVI) [7], which allows the use of efficient exact [8] or approximate [9] hypervolume calculation algorithms to quickly predict the hypervolume improvement.

2.1 Adding optimization preferences and constraints

One advantage of the MOBO approach is the ability to specify a preference towards optimizing certain objectives over others during optimization by explicitly constraining objective space. Instead of a single reference point, this preferential algorithm specifies both a maximum and minimum reference point in objective space and calculates the truncated hypervolume improvement [10]. If we specify the truncated domain $\mathcal{T} = [\mathbf{A}, \mathbf{B}]$ defined by the minimum objective point $\mathbf{A}$ and the maximum objective point $\mathbf{B}$ the truncated version of the UCB-HVI is given by

$$\alpha_{TUHVI}(\mu(\mathbf{x}), \sigma(\mathbf{x}), \mathcal{P}, \beta, \mathbf{A}, \mathbf{B}) := \begin{cases} \text{HVI}(\mathcal{P}, \mathbf{y}(\mathbf{x}), \mathbf{B}) & \mathbf{y} \in \mathcal{T} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\mathbf{y}(\mathbf{x}) = \mu(\mathbf{x}) - \sqrt{\beta}\sigma(\mathbf{x})$ where $\mu(\mathbf{x}), \sigma(\mathbf{x})$ are the predicted mean and standard deviation from each objective Gaussian process at an input point $\mathbf{x}$ and β specifies the trade-off between exploration ($\beta \gg 1$) and exploitation ($\beta \ll 1$). The hypervolume improvement $\mathrm{HVI}(\mathcal{P}, \mathbf{y}, \mathbf{B})$ measures the hypervolume of adding a point $\mathbf{y}$ to the current Pareto front $\mathcal{P}$, subtracted by the hypervolume of the current Pareto front, both with the reference point $\mathbf{B}$.

Alternatively, we can specify an inequality constraint that needs to be satisfied during optimization. We create another GP surrogate model that predicts the probability of a constraint being satisfied and simply multiply the hypervolume improvement acquisition function by this probability [11]. The probability of a point $\mathbf{x}$ satisfying the constraint condition $g(\mathbf{x}) \leq h$, modeled by a Gaussian process trained on a dataset $\mathcal{D}_g$ is given by

$$P_g(\mathbf{x}) := Pr[g(\mathbf{x}) \leq h] = \int_{-\infty}^{h} p(g(\mathbf{x})|\mathcal{D}_g) dg(\mathbf{x}) \tag{2}$$

which is simply the univariate Gaussian cumulative distribution function. Now we can define a new constrained version of the acquisition function $\hat{\alpha}(\mathbf{x})$ as $\hat{\alpha}(\mathbf{x}) = \alpha(\mathbf{x}) P_g(\mathbf{x})$. As a result, the acquisition function will be negatively biased anywhere the model predicts that the constraint is likely to be violated.

2.2 Smooth input space exploration

One unique aspect of accelerator optimization is the cost associated with exploring the input parameter space. Changes to input parameters (magnetic field settings, RF phase settings, etc.) often takes a nontrivial amount of time to fully execute in practice, often scaling proportionally to the magnitude of the change. Thus, it is desirable to bias the acquisition function to smoothly explore input space, while still increasing the Pareto front hypervolume.

Achieving this is done by biasing the acquisition function towards prioritizing nearby points relative to the most recent observation location in input space. We multiply our original acquisition function $\alpha(\mathbf{x})$ by a multivariate Gaussian distribution, centered at the most recently observed point in input space $\mathbf{x}_0$, and a precision matrix $\mathbf{\Sigma}$

$$\tilde{\alpha}(\mathbf{x}, \mathbf{x}_0) = \alpha(\mathbf{x}) \exp\Big[-\frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \mathbf{\Sigma} (\mathbf{x} - \mathbf{x}_0) \Big]. \tag{3}$$

We name the modified acquisition function $\tilde{\alpha}(\mathbf{x}, \mathbf{x}_0)$, smooth UCB-HVI (S-UCB-HVI). The precision matrix specifies the cost associated with changing each input variable, where larger elements correspond to a harsher penalty. This allows travel in each parameter direction to cost a variable amount, which is often the case in accelerator operations when different classes of beamline parameters are modified. With an appropriately chosen cost matrix $\mathbf{\Sigma}$, the acquisition function still allows the optimizer to make large jumps in input space if the unmodified acquisition function $\alpha(\mathbf{x})$ is large enough. This maintains the optimizers' ability to escape localized extrema and explore regions of unobserved input space, while significantly reducing the frequency and amplitude of large jumps.

3 Photo-injector Optimization Results

3.1 Pareto front convergence

We assume that the functional form of each objective is smooth, thus we choose the standard radial basis function kernel with an anisotropic precision matrix $\Sigma = \mathrm{diag}(\mathbf{l})^{-1}$. Initially, a randomly generated Latin-Hypercube distribution of 20 input points with corresponding objective observations is used to train each corresponding GP. Hyperparameter training is done by maximizing the log marginal likelihood [4]. We use the UCB-HVI acquisition function with $\beta = 0.01$ to do multi-objective Bayesian optimization with 300 sequential observations. The UCB-HVI is maximized using a particle swarm optimization algorithm implemented in the PyGMO package [12] with 64 individuals and 10 generations. In order to account for new information gained from the optimization, we retrain the hyperparameters with the accumulated dataset every 10 observations. The resulting hypervolume as a function of iteration number is shown in Figure 2f. We observe that this algorithm converges to a Pareto front, that matches results from previous experiments [6], in about 300 iterations.

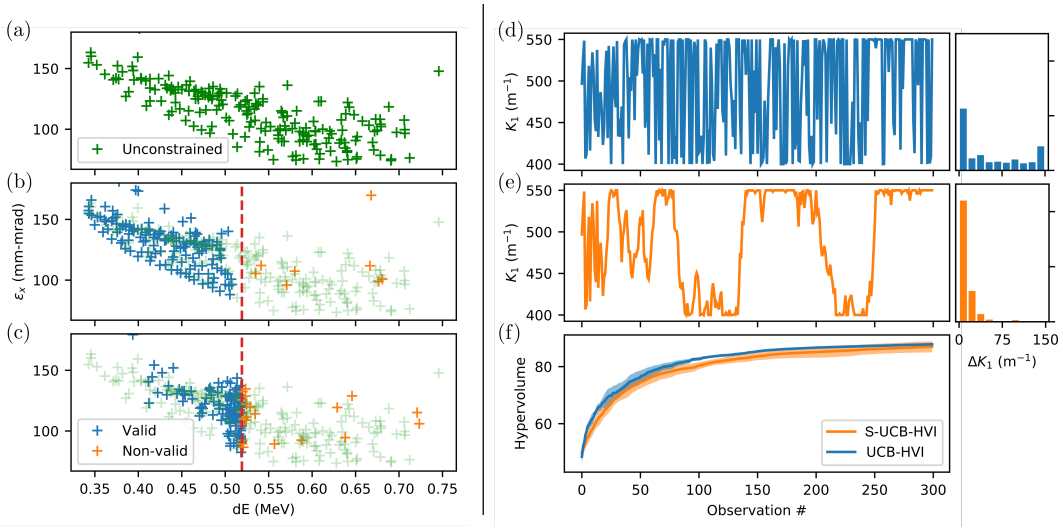

Figure 2: Left: Energy spread (dE) and horizontal beam emittance (ε_x) objective space after 200 observations with MOBO. (a) MOBO with no constraints. (b) MOBO with an optimization preference of $dE < 0.52$ MeV. (c) MOBO with an inequality constraint of $dE < 0.52$ MeV. The dotted line in (b) and (c) denotes the preference/constraint limit (0.52 MeV). Right: Comparison between normal UCB-HVI and localized UCB-HVI acquisition functions when used to perform optimization of the AWA photoinjector. Solenoid 1 strength parameter over 300 observations when normal (d) and localized UCB-HVI (e) is used. (f) Average Pareto front hypervolume of 10 optimization runs with random initialization sets. Shading denotes one sigma variance. Smoother traversal through parameter space, as seen in (e) is particularly important for practical online use in accelerators.

3.2 Constrained optimization

We now investigate the effect of preferential or constrained treatment of an objective on the optimization result. First, we consider a case where want to optimize the same objectives as the previous problem but wish to only find solutions where the energy spread satisfies $dE < 0.52$ mega-electron-volts (MeV). To judge how this modification effects the optimization we compare the observed points in the projected 2D dE vs. ε_x objective space after 200 iterations in Figure 2. When preferential treatment is added to the algorithm (Figure 2(b)), the algorithm observes almost no points that violate this preference. Furthermore, since the volume of the objective space is significantly reduced, the optimizer finds a higher-quality Pareto front in the same number of steps as the unconstrained case.

Second, we consider a case where we wish to relax this constraint, removing the energy spread minimization objective and replacing it with the inequality constraint $dE < 0.52$ MeV. The resulting observation distribution appears significantly different in this case (Figure 2(c)) as the optimizer allows the energy spread to increase up to the constraint value in order to optimize the six remaining objectives. In this case, more observations are made that violate the constraint than in the previous experiment, which is necessary to accurately model the constraining function near the boundary.

3.3 Smooth optimization

Finally, we demonstrate the use of S-UCB-HVI on optimizing the AWA problem. We start with the same set of 10 initial sets of observations as in Section 3.1 with the same hyperparameter training schedule. However, this time we run MOBO optimization using the S-UCB-HVI acquisition function, with a isotropic covariance matrix (see Eq. 3) $\mathbf{\Sigma} = 0.25\mathbf{I}$ in normalized input space.

Results from these optimization runs are presented in Figure 2. We observe that during optimization, when the UCB-HVI acquisition function is used, the solenoid strength parameter is wildly varied to increase the hypervolume as much as possible. However, when a smoothing term is added to the acquisition function, the frequency and amplitude of large jumps in parameter space are both decreased. While not shown here, this change in behavior is mirrored in each of the other 5 variables. Furthermore, the use of S-UCB-HVI acquisition function over the generic UCB-HVI function

only minimally reduces the overall speed at which the method finds maximizes the Pareto front hypervolume (Figure 2f).

4 Conclusion

In this paper we have demonstrated that the MOBO framework can be used to solve multi-objective optimization problems in accelerator physics. This method efficiently finds the Pareto front in a serialized manner, which makes multi-objective online optimization of accelerators viable for the first time. In the simple photo-injector optimization case shown here our algorithm reached reasonable convergence to the Pareto front in approximately 300 iterations, corresponding to about 50 minutes of online optimization time (assuming 10 seconds for each measurement), which is similar to tuning times needed today to optimize a similar accelerator towards a single objective. The framework also allows the operator to easily specify objective preferences and constraints. Finally, we demonstrated that adding a smoothing term to the acquisition function effectively reduces the number and frequency of large jumps in input space. These modifications to the MOBO optimization framework are especially important for practical use in accelerator facilities, where operational time is at a premium.

5 Broader Impact

The authors do not believe that this work has any ethical or future societal impacts.

6 Acknowledgements

This work was supported by the U.S. National Science Foundation under Award No. PHY-1549132, the Center for Bright Beams.

References

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. 6(2):182–197. Conference Name: IEEE Transactions on Evolutionary Computation.

[2] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4.

[3] Michael Emmerich, Kaifeng Yang, André Deutz, Hao Wang, and Carlos M. Fonseca. A multicriteria generalization of bayesian global optimization. In Panos M. Pardalos, Anatoly Zhigljavsky, and Julius Žilinskas, editors, *Advances in Stochastic and Deterministic Global Optimization*, Springer Optimization and Its Applications, pages 229–242. Springer International Publishing.

[4] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press. OCLC: ocm61285753.

[5] M E Conde, S P Antipov, D S Doran, W Gai, Q Gao, and G Ha. Research program and recent results at the argonne wakefield accelerator facility (AWA). page 3.

[6] Auralee Edelen, Nicole Neveu, Yannick Huber, Mattias Frey, Christopher Mayes, and Andreas Adelmann. Machine learning for orders of magnitude speedup in multi-objective optimization of particle accelerator systems.

[7] M.T.M. Emmerich, K.C. Giannakoglou, and B. Naujoks. Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. 10(4):421–439. Conference Name: IEEE Transactions on Evolutionary Computation.

[8] Lyndon While, Lucas Bradstreet, and Luigi Barone. A fast way of calculating exact hypervolumes. 16(1):86–95. Conference Name: IEEE Transactions on Evolutionary Computation.

[9] Weisen Tang, Hai-Lin Liu, Lei Chen, Kay Chen Tan, and Yiu-ming Cheung. Fast hypervolume approximation scheme based on a segmentation strategy. 509:320–342.

[10] Kaifeng Yang, Andre Deutz, Zhiwei Yang, Thomas Back, and Michael Emmerich. Truncated expected hypervolume improvement: Exact computation and application. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 4350–4357.

[11] Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian optimization with inequality constraints. In *ICML*, volume 2014, pages 937–945.

[12] Francesco Biscani and Dario Izzo. A parallel global multiobjective framework for optimization: pagmo. 5(53):2338.