

BEAM LINE OPTIMIZATION USING DERIVATIVE-FREE ALGORITHMS

S. Appel*, S. Reimann, GSI, Darmstadt, Germany

Abstract

The present study focuses on the beam line optimization from the heavy-ion synchrotron SIS18 to the HADES experiment. BOBYQA (Bound Optimization BY Quadratic Approximation) solves bound constrained optimization problems without using derivatives of the objective function. The Bayesian optimization is another strategy for global optimization of costly, noisy functions without using derivatives. A python programming interface to MADX allow the use of the python implementation of BOBYQA and Bayesian method. This gave the possibility to use tracking simulation with MADX to determine the loss budget for each lattice setting during the optimization and compare both optimization methods.

INTRODUCTION

FAIR—the Facility for Antiproton and Ion Research will provide antiproton and ion beams of unprecedented intensities as well as qualities to drive forefront heavy ion and antimatter research [1]. The complexity of the FAIR facility demands a high level of automation to keep anticipated manpower requirements within acceptable levels, as shown in [2]. An example of complexity is the High Energy Beam Transport System (HEBT) of FAIR which forms a complex system connecting among other things seven storage rings and experiment caves and has a total length of 2350 metres [3]. A similar and current optimization problem is the beam line from the heavy-ion synchrotron SIS18 to the HADES experiment. This beam line is part of the high-energy beam transport system of the existing GSI facility called HEST. The HADES beam line is over 150-meter-long and contains 22 focusing quadrupoles as well as 11 steers magnets. The aim of the optimization has been to minimize the beam loss along the transport and reach simultaneous a small focused beam at the experiment target.

An automatized machine based optimization with derivative-free algorithms may improve the time for optimization and control of beam lines, if a model-driven optimization is not sufficient. The HADES experiment requires a slow extracted beam from the SIS18, therefore an online evaluation of beam line must be low as possible and additionally not exceed boundary conditions by machine and radiation protection. BOBYQA and Bayesian optimization are for engineering problems know to be rapid convergence derivative-free algorithms. Therefore, a comparison study between has been carried out to determine the fastest algorithms of both.

The optimization of the parameters for the SIS18 multi-turn-injection (MTI) using a genetic algorithm has already been simulated and has been successfully demonstrated ex-

perimentally at the CRYRING@ESR [4–6]. The nature-inspired optimization has potential to reduce the manpower requirements and variations of quality performance due to the manual procedure. These algorithms are stable, but typically require many evaluation.

In many real-life problems, multi-quantities - as well as for the HADES beam line - have to be optimized. In addition, these quantities can be contradicting and there is more than one equally valid solution. These solutions form a so-called Pareto front (PA front) in the solution space. A solution is Pareto optimal if it is not dominated by any other solution. By using a non-dominated selection algorithm one tries to find solutions near the optimal Pareto set [7]. Genetic algorithms allow multi-objective optimization, whereby usually derivative-free algorithms do not enable this. Therefore, a weighting factor approach is used for multi-objective reduction and the composite objective function is than optimized. The composite objective function is constructed from a set of individual objective functions using a user-specified set of weighting factors. The higher the weighting factor, the more dominant a particular objective function will be in the optimization process.

DERIVATIVE-FREE ALGORITHMS

Bound Optimization By Quadratic Approximation (BOBYQA) is a deterministic method and widely used for engineering problems. The Bayesian method follow a stochastic (probabilistic) approach and is therefore a non-deterministic method. Deterministic methods typically require fewer executions of an experiment than is the case with non-deterministic procedures. In particular, they work demonstrably efficiently when good initial values are available, such as empirical values of previous operating parameters [8]. Nevertheless, for online accelerator optimization the probabilistic Bayesian method has been demonstrated as powerful [9].

BOBYQA Optimization

BOBYQA is a bound constrained optimization for finding a minimum of a function without using derivatives of the objective function proposed by Michael J. D. Powell in 2009. Each iteration employs a quadratic approximation of optimization function by using trust regions [10]. Quadratic models of objective functions are highly useful in many optimization algorithms. They are updated regularly to include new information about the objective function, such as the difference between two gradient vectors. An online algorithm needs to be aware of the noise and take action more cautiously [11]. *Py-BOBYQA* is an open-source and user-friendly python implementation of BOBYQA which implements robustness to noise strategies by introducing a new adaptive measure of accuracy for the data profiles of noisy

* s.appel@gsi.de

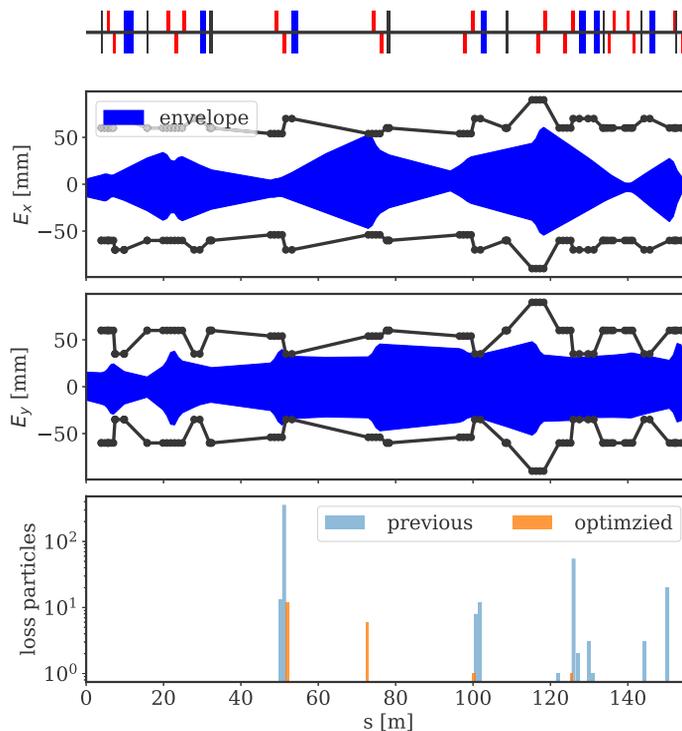


Figure 1: The beam line from SIS18 to the HADES experiment is 156 meter long and contains several quadrupoles (red), dipole magnets (blue) and steer magnets (black). Possible loss areas are visible for the horizontal (top) and vertical (middle) beam envelope. The loss along the beam line is shown in logarithmic scale (below) for previous and an optimized setting.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

functions that strikes a balance between measuring the true and the noisy objective improvement. *Py-BOBYQA* is particularly useful when evaluations of the objective function are expensive and/or noisy. *Py-BOBYQA* is a site-package of the *DFO-LS* software for derivative-free optimization for nonlinear Least-Squares problems with optional bound constraints [12].

Bayesian Optimization

The Bayesian optimization is another strategy for global optimization of costly, noisy functions without using derivatives. Bayesian optimization has been developed for the drilling industry and has recently become popular for training expensive machine-learning models. During a Bayesian optimization, a probabilistic model of optimization function is construed and then exploits this model to make decisions, where to evaluate the function next. This results in a procedure that can find the minimum of difficult non-convex functions with relatively few evaluations, at the cost of performing more computation to determine the next point to try [13]. Two major assumptions have to be made for performing a Bayesian optimization. The Gaussian process prior, the first assumption, will express estimation about the optimization function. The second assumption is the chosen acquisition function. The acquisition function defined from the Gaussian surrogate model, where to evaluate the function

next [14]. The *BayesianOptimization* is a Python implementation of global optimization with Gaussian processes [15].

OPTIMIZATION OF HADES BEAM LINE

A comparison study between BOBYQA and Bayesian optimization with tracking simulation has been carried out, to allow a decision which of the two methods is better suited for an online optimization of beam lines. Another comparison study between genetic algorithms, BOBYQA and the Powell's method showed that BOBYQA is the better studied algorithms for beam line optimization [16]. A python programming interface to MADX allow the use of the python implementation of BOBYQA and Bayesian optimization. This gave the possibility to use tracking simulation with MADX [17] to determine the loss budget for each lattice setting during the optimization. The HADES experiment requires a small focused beam and high beam intensity on target. Due to radiation protection reason the loss along the HADES beam line must be low as possible. For the simultaneous optimization of beam loss and focused beam, a high weighing of 0.9 for beam loss and an equal weighing factor of 0.05 for the vertical and horizontal beta-function has been applied. Figure 1 show an optimized horizontal and vertical envelope obtain with the BOBYQA algorithms. The optimized loss patterns along the beam line is shown together with the previous one. BOBYQA found a solution

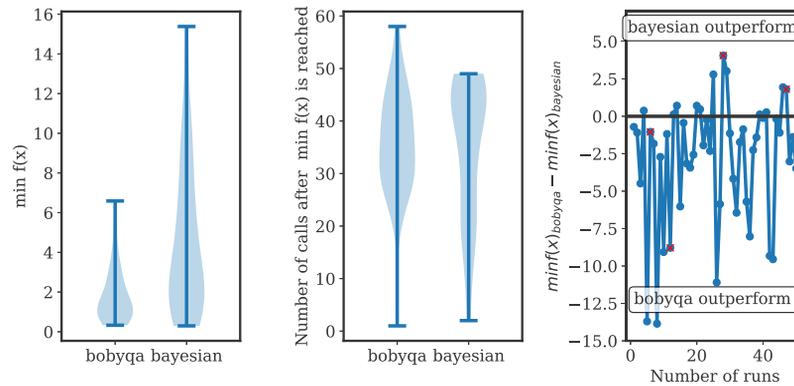


Figure 2: Comparison between BOBYQA and Bayesian optimization. Right: A violin plot of the achieved minimum; Middle: A violin plot of needed evaluation steps; Left: Difference of the achieved minimum.

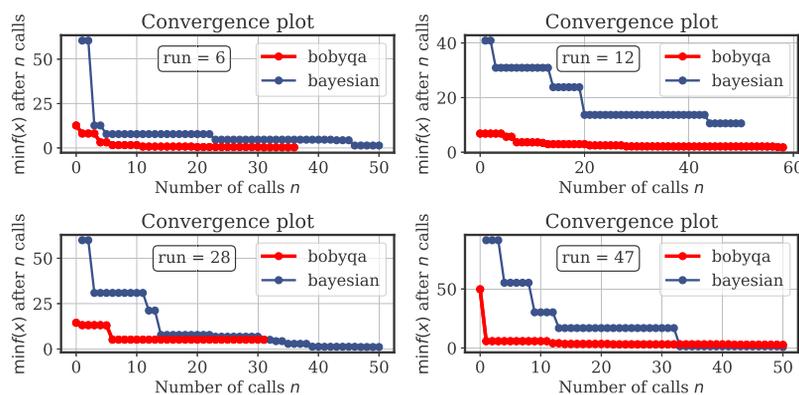


Figure 3: Convergence plot for four selected optimization cases, which are indicated in red on the left plot of Figure 2.

where low loss after $s=100$ m is reached, whereby a large horizontal envelope at $s=75$ m has to be accepted. The optimized setting reach in addition a slightly smaller focused beam at the target.

As BOBYQA and Bayesian optimization are constrained optimization techniques, the upper and under bounds has been specified from the know adequate quadruples strength with $\pm 20\%$. The initial values for both optimization algorithms have been generated randomly in this bound. For the comparison studies, only a part of the beam line elements (up to second dipol magnets, in total 7 elements) have been varied by the optimization algorithms for fast convergence. This has been sufficient to find a much better optimum, as an optimization with all beam line elements has not found better solutions. Figure 2 show the comparison between BOBYQA and Bayesian optimization. Figure 3 show convergence plots for four selected optimization case, which are indicated in red on the left plot of Figure 2. The python *BayesianOptimization* implementation is often outperform from the *Py-BOBYQA* for the HADES beam line optimization. *Py-BOBYQA* found more often a lower minimum of the composite objective function, whereby few more evaluation steps were necessary. The BOBYQA algorithms is able in round of 40 - 50 evaluation steps to minimize of the beam

loss and reach simultaneous a small focused beam at the experiment target. This mean, for long slow extraction time for the HADES experiment of 10 s, an online optimization could be carry out in round 10 minutes.

CONCLUSION AND OUTLOOK

A python programming interface to MADX has been developed and used together with derivative-free algorithm to optimize the HADES beam line. To allow simultaneous minimization of the beam loss and a small focused beam at the experiment target a weighting factor approach for multi-objective reduction has been used. An optimized HADES optic has been found and an experimental proof of this result is planned. A comparison study between BOBYQA and Bayesian optimization has been carried out. The *Py-BOBYQA* seems better studied for online optimization of beam lines as good initial values are available, a faction of the beam line elements are optimized and external boundary conditions is not an issue. As next step, external boundary conditions like machine and radiation protection as to be taken into account. Such an optimization problem prefer must likely the adaptive and safe Bayesian optimization from [18]. The so called SafeOp guaranteed that the performance of the system never falls below a critical value.

REFERENCES

- [1] P. Spiller *et al.*, *FAIR Technical Design Report*, 2008.
- [2] S. Reimann *et al.*, “Building an Operation Team for FAIR Nearly from Scratch”, presentation at WAO10, 2016.
- [3] S. Ratschow *et al.*, “The High Energy Beam Transport System for FAIR”, in *Proc. 11th European Particle Accelerator Conf. (EPAC'08)*, Genoa, Italy, Jun. 2008, paper THPP104, pp. 3608–3610.
- [4] S. Appel *et al.*, “Injection optimization in a heavy-ion synchrotron using genetic algorithms”, *Nucl. Instrum. Methods A*, vol. 852, pp. 73–79, 2017.
- [5] S. Appel, W. Geithner, S. Reimann, M. Sapinski, R. Singh, and D. M. Vilsmeier, “Optimization of Heavy-Ion Synchrotrons Using Nature-Inspired Algorithms and Machine Learning”, in *Proc. ICAP'18*, Key West, Florida, USA, Oct. 2018, pp. 15–21. doi:10.18429/JACoW-ICAP2018-SAPAF02
- [6] W. Geithner *et al.*, “Genetic Algorithms for Machine Optimization in the Fair Control System Environment”, in *Proc. 9th Int. Particle Accelerator Conf. (IPAC'18)*, Vancouver, Canada, Apr.-May 2018, pp. 4712–4715. doi:10.18429/JACoW-IPAC2018-THPML028
- [7] A. Konak *et al.*, “Multi-objective optimization using genetic algorithms: A tutorial”, *Reliab. Eng. Syst. Saf.*, 91 (9), pp. 992–1007, 2006.
- [8] S. Dutta, *Optimization in Chemical Engineering*, Cambridge University Press, 2016.
- [9] M. W. McIntire, T. M. Cope, D. F. Ratner, and S. Ermon, “Bayesian Optimization of FEL Performance at LCLS”, in *Proc. 7th Int. Particle Accelerator Conf. (IPAC'16)*, Busan, Korea, May 2016, pp. 2972–2975. doi:10.18429/JACoW-IPAC2016-WEPOW055
- [10] M. J. D. Powell, “The BOBYQA algorithm for bound constrained optimization without derivatives”, technical report DAMTP 2009/NA06, University of Cambridge (2009).
- [11] X. Huang *et al.*, “An algorithm for online optimization of accelerators”, *Nucl. Instrum. Methods A*, vol. 726, pp. 77–83, 2013.
- [12] C. Cartis *et al.*, “Improving the Flexibility and Robustness of Model-Based Derivative-Free Optimization Solvers”, arXiv:1804.00154, 2018, <https://pypi.org/project/Py-BOBYQA/>.
- [13] J. Snoek *et al.*, “Practical Bayesian Optimization of Machine Learning Algorithms”, in *Proc. NIPS'12*, 2012.
- [14] P. I. Frazier, “A Tutorial on Bayesian Optimization”, arXiv:1807.02811, 2018.
- [15] F. Noqueira, “Py-BOBYQA: A python implementation of global optimization with gaussian processes”, <https://github.com/fmfn/BayesianOptimization>
- [16] M. Wolf, “Automatized optimization of accelerators – Comparison of processes”, Bachelor-Thesis, TU-Darmstadt, 2018.
- [17] *The MAD-X Program, User's Reference Manual*, L. Deniau (editor), Version 5.04.02, 2018, <http://mad.web.cern.ch/mad/>.
- [18] J. Kirschner *et al.*, “Adaptive and Safe Bayesian Optimization in High Dimensions via One-Dimensional”, arXiv1902.03229, 2019.