

A FLEXIBLE ONLINE OPTIMIZER FOR SPS *

T. Pulampong[†], N. Suradet, Synchrotron Light Research Institute, Nakhonratchasima, Thailand

Abstract

Siam Photon Source (SPS) machine in Thailand has been operating for more than two decades with limited diagnostic systems. It is very challenging to efficiently tune and operate the machine. With online optimization, variables and objectives are required to tune for better solutions. In this work, a flexible optimizer was developed. Objectives and variables can be freely defined based on available hardware in the form of Process Variables (PVs). Several multi-objective and Robust Conjugated Direction Search (RCDS) algorithms are provided. The online optimizer was tested on the SPS machine to improve the injection efficiency. Due to its flexibility, the optimizer can also be used for other systems.

INTRODUCTION

Achieving the ultimate goals or objectives in a complex system is not simple. In practice, model based may not be sufficient due to higher order effects, non-linearities and noise. There are several subsystems in an accelerator. Parameters tuning requires immense experience and time. Sometimes the machine condition can also be affected by the environment which adds complexity to machine operation. In recent years, online optimization is gaining more popularity. This allows live parameters tuning to achieve the desired objectives. The usefulness of such optimization is obvious especially for a system with limited diagnostics like in SPS case. However setting up an optimization can be complicated. There are several factors that have to be taken into account: programming language, optimization algorithm, connection between the optimizer and hardware, etc.

An optimizer program, in this work, was developed to ease these difficulties. The optimizer acts as a box containing optimization tools and allows users to feed some inputs and acquire some output parameters and objectives.

OPTIMIZER

The optimizer was developed to provide flexibility and easy-to-use steps for users. Python, one of the most popular programming language providing several useful packages especially optimizer, was employed. Graphical User Interface (GUI) can also be created conveniently with Tkinter package. For hardware connection and control, Epics tools can be found in PyEpics package.

For an optimization, it starts from identifying the desired objectives and variables. Then the process follows the overall flow as shown in Fig. 1:

- A list of variables and objective functions should be prepared in text files in a form of process variable (PV). This allows easy connection between those parameters

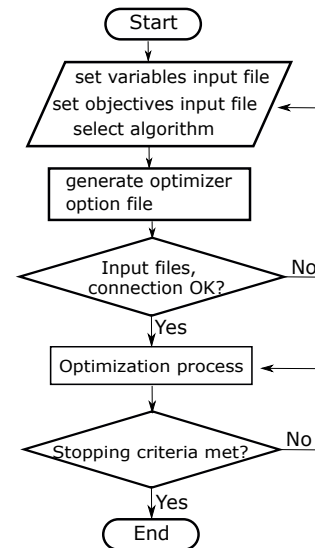


Figure 1: Overall program flow chart.

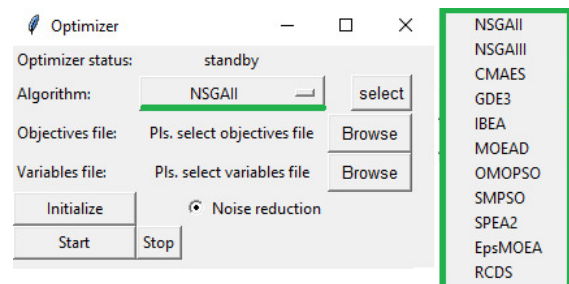


Figure 2: Optimizer user interface panel.

and the control. Upper and lower boundaries for the variables are also required to make effective and safe parameters tuning.

- An algorithm can be selected from the available list as shown in Fig. 2.
- Then the optimizer will check the input files and connection to the PVs. If there are some errors, user need to check the input files and PVs connection.
- If everything is ok, the optimization process will start and continue until the stopping criteria is met.
- The process stops when the stopping criteria is met. The maximum number of function evaluation can be set. Otherwise, the optimization can also be terminated by users any time because the data of the optimization is recorded during the process.

As shown in Fig. 2, the GUI was designed for user to set up an optimization in the described sequential steps from top to bottom. Browse buttons for objective and variable files selection are also provided. Moreover, to conveniently repeat the optimization, the optimizer will generate the required setting input files which can be executed again separately.

* Work supported by NSTDA

[†] thapakron@slri.or.th

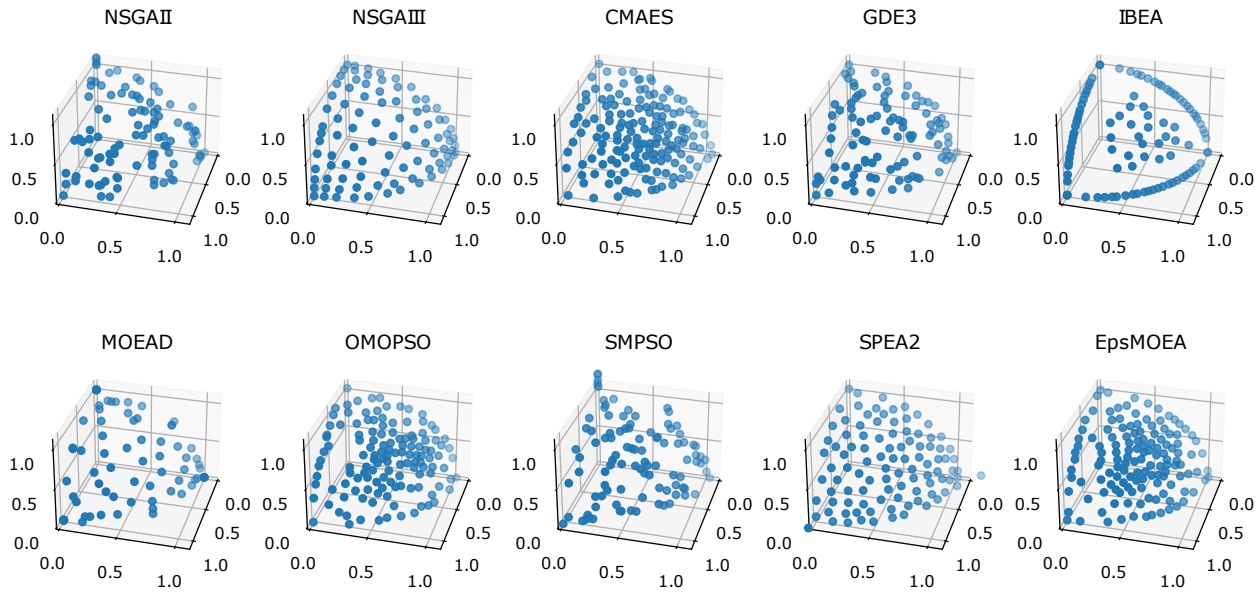


Figure 3: Optimal front from a test problem DTLZ2(3) of the provided multi-objective algorithms.

Algorithms

The optimizer provides eleven algorithms to be selected by user. Multi-objective optimization algorithms were adopted from Platypus : Multiobjective optimization in Python package [1]. In total, there are ten algorithms for multi-objective problem: NSGAI, NSGAI, CMAES, GDE3, IBEA, MOEAD, OMOPSO, SMPSO, SPEA2, and EpsMOEA. These are evolutionary algorithms. And a single objective algorithm provided is Robust Conjugate Direction Search (RCDS) [2] which is a robust algorithm against noise. The result of three objectives optimization for a test problem DTLZ2(3) for each algorithm is shown in Fig. 3. For the test problem, NSGAI and SPEA2 show good distribution between solutions while IBEA manages to highlight a clear boundary on the objectives space. NSGAI is obviously an improve version of NSGAI and works well with three objectives problem. For two objectives problem, NSGAI is still a good algorithm. To select a proper algorithm, one has to do some experiments on a test problem.

Flexibility

In order to allow flexibility to the user, algorithm, objective functions and variables can be selected. As shown in Fig. 2, there are eleven algorithms provided in a drop down algorithm option. Objective functions and variables have to be predefined in a form of text file. The input text file contains a list of PVs or functions. Epics was used to connect the optimizer and the PVs. To start the optimizer, a file generator function will write a text file containing all required parameters for the optimization. Then the file can be executed to start the optimization process. The generated optimization parameters file, objective functions file and variables file can be modified and reused again any time.

Noise reduction option is also provided. This, however, take longer time to measure objective functions for averaging. This makes the optimizer to work more efficiently against noisy measurement. To reduce the noise, in the initial stage, standard deviation (σ) of the objective functions will be measured. The data point that exceeds more than 3σ will be excluded as a noise.

The output from the optimization was also generated in a form of text file. It can be used for post-processing and data analysis later after the optimization.

SIMULATION RESULTS

From the booster design for SPS-II [3], a simulation was set to investigate how large the horizontal beam emittance can be, for good injection into the SPS-II storage ring.

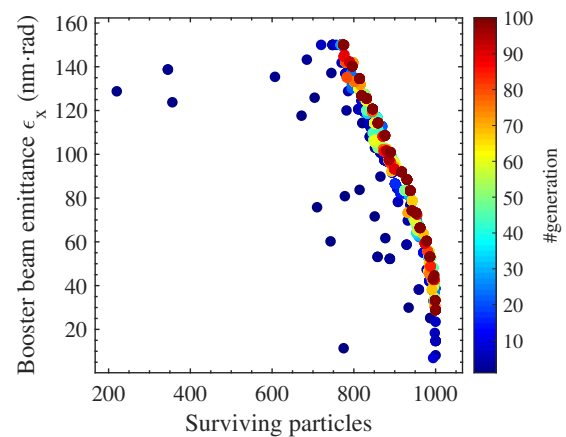


Figure 4: Optimal front of booster beam emittance and the number of surviving particles for the optimizer simulation test.

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2022). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

To test the optimizer, the injected beam distribution of a thousand particles was generated based on beam parameters (variables): horizontal beam emittance and beamsize. The generation of the injected beam introduces some noise due to the randomness. Then the beam was tracked and the number of surviving particles was recorded after a hundred turns. Fig. 4 shows the optimization solutions with a clear trade-off between booster beam emittance and surviving particles (injection efficiency). As a consequence, the maximum horizontal beam emittance for injected beam that allows good injection is below $20 \text{ nm} \cdot \text{rad}$.

ONLINE OPTIMIZATION TEST

Online optimization has been proved to be useful for SPS [4]. To test the developed flexible optimizer with the real machine, for SPS, injection from Low energy Beam Transport (LBT) into the booster was optimized. Previously, one of the bump magnets in the booster was broken and only two bump magnets remain. Now the problem is how to inject the beam efficiently with the limited bump magnets. Two objectives in this optimization were good injection efficiency and lower bump magnet excitation current (B1 and B2). Thus NSGA-II was employed for the optimization. Magnets in the LBT including the bump magnets are employed as variables. Fig. 5 shows the optimization result after twenty generations. It is observable that the solutions evolved towards lower bump magnet excitation and average booster beam current objective. A single objective optimization with RCDS was also performed to improve the injection from High energy Beam Transport (HBT) into the storage ring. From the optical fiber based beam loss monitor, the end of HBT and Septum Magnet for Injection (SMI) is critical [5]. Thus magnets at the HBT and SMI were used as variables. The evolution of the injection rate and correlation between each variable on the objective function is shown in Fig. 6. The top three strongest correlations between the magnets and injection rate are bending magnet HBTBH2 and HBT corrector HBTcor4 and HBTcor5. This can be useful for further optimization and machine tuning.

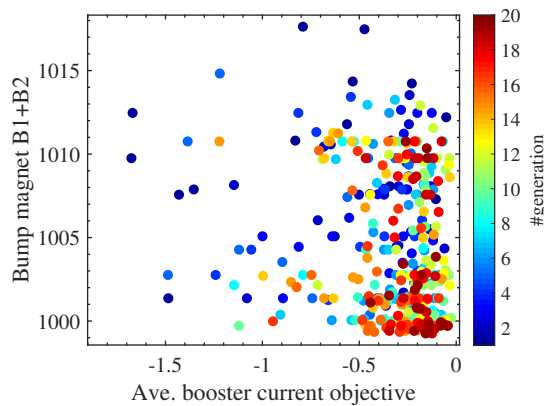


Figure 5: Bump magnet strength and booster current objective for an online multi-objective optimization.

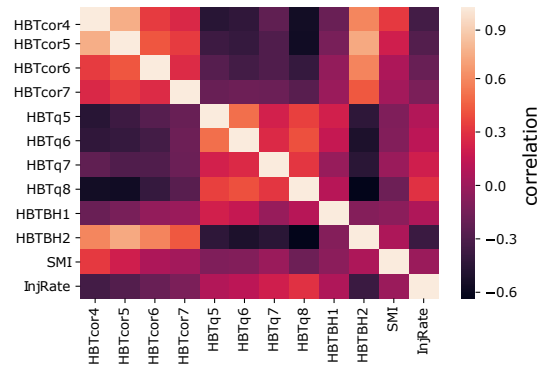
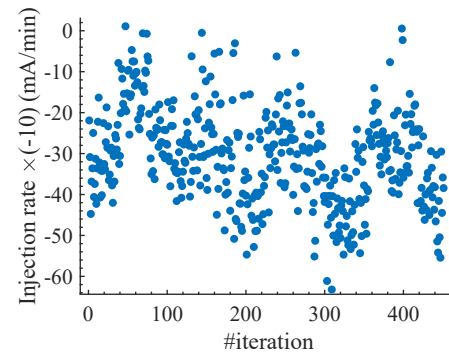


Figure 6: Injection rate objective evolution (top) and correlation between each variable (bottom) for single objective optimization.

CONCLUSION

The developed optimizer provides an easy way to set up an optimization with flexibility to select objective functions, variables and algorithm. It can be used for simulation or online optimization.

Further development of the optimizer is to provide more sophisticated method to deal with noise in the measured data. Moreover, it should have the capability to prepare the data acquired during the optimization for machine learning.

REFERENCES

- [1] <https://platypus.readthedocs.io/en/latest/>
- [2] X. Huang, J. Safranek, "Online Optimization of Storage Ring Nonlinear Beam Dynamics", *Phys. Rev. ST Accel. Beams*, vol 18, p. 084001, 2015. doi:10.1103/PhysRevSTAB.18.084001
- [3] S. Krainara, S. Klinkhieo, P. Klysubun, T. Pulampong, and P. Sudmuang, "Conceptual Design of Booster Synchrotron for Siam Photon Source II", in *Proc. IPAC'21*, Campinas, Brazil, May 2021, pp. 2795-2797. doi:10.18429/JACoW-IPAC2021-WEPAB089
- [4] T. Pulampong, P. Klysubun, S. Kongtawong, S. Krainara, and P. Sudmuang, "Online Optimisation Applications at SPS", in *Proc. IPAC'17*, Copenhagen, Denmark, May 2017, pp. 4086-4088. doi:10.18429/JACoW-IPAC2017-THPAB151
- [5] T. Pulampong, W. Phacheerak, N. Suradet, and P. Sudmuang, "Optical Fiber Based Beam loss Monitor for SPS Machine", presented at the 13th International Particle Accelerator Conf. (IPAC'22), Bangkok, Thailand, June 2022, paper MOPOPT051, this conference.