9th International Particle Accelerator Conference IPAC2018, Vancouver, BC, Canada JACoW Publishing
ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-WEPAF054

ONLINE MULTI-OBJECTIVE OPTIMISATION AT DIAMOND LIGHT SOURCE

M. Apollonio*, R. T. Fielder, I. P. S. Martin, R. Bartolini[1], Diamond Light Source, Oxfordshire, U.K.
J. Rogers, Imperial College of Science, Technology and Medicine, U.K.
G. Henderson, University of Oxford, U.K.
[1]also at John Adams Institute, University of Oxford, U.K.

Abstract

At Diamond Light Source we have developed an Optimization Package currently used online to improve the performance of the machine, usually measured in terms of lifetime, injection efficiency or beam disturbance at injection. The tool is flexible in that control variables in order to optimise objectives (or their functions) can be easily specified by means of EPICS process variables (PV), making it suitable for virtually any sort of optimization. At present three different algorithms can be used to perform optimizations in a multi-objective fashion: Multi-Objective Genetic Algorithm (MOGA), Particle Swarm Optimizer (MOPSO) and Simulated Annealing (MOSA). We present a series of tests aimed at characterizing the algorithms as well as improving the performance of the machine itself.

INTRODUCTION

The optimisation of an electron storage ring (SR) presents several difficulties. The non-linear nature of the problem together with the often large number of parameters involved, makes predictions very complex. Also, the objectives may be characterized by local minima or dis-continuities, preventing the use of traditional gradient-based searches. Noise or drift of the objectives should also be taken into account, in order to reach good solutions for the machine. Finally, we often face the case of a set of many objectives in conflict between each other, where the improvement of one variable usually degrades another. At Diamond, 1D or 2D parameter scans are the typical choice with single objective problems, however such method relies on a starting point close the optimum. With more complex problems, often involving a larger number of parameters, the Robust Conjugate Direction Search technique is often used [1]. It is a powerful method with very good noise rejection, and capable of dealing with a large number of input parameters, however it is limited to a single objective search.

MULTI-OBJECTIVE OPTIMISATION

Multi-objectivity is commonly met in a variety of systems and problems, where conflicting variables need to be optimised. A classic example in a SR, is the optimization of lifetime (LT) and injection efficiency (IE), or of injection efficiency and residual peak to peak horizontal oscillations in the stored beam (PPX). In these kind of problems factorisation to a single objective usually entails a loss of information, or may be difficult to implement due to the heterogeneity of the objectives. Multi-objective optimisation (MOO) offers a set of solutions where a final choice can be done according to priorities, *e.g.* privileging LT against IE. This apparent ambiguity is addressed by maintaining a distribution of solutions, sorted into 'non-dominated fronts', for which no member is worse than another in more than one objective simultaneously. The fronts can be further sorted according to how diverse the solutions are, as this helps to identify the true global optimum for each objective. The goal of the optimisation algorithm is then to move the population towards an ideal Pareto-optimal front, after which no further improvement is possible.

MOO has been a standard in the accelerator physics community for at least a decade now, playing a crucial role in the off-line refinement of the non-linear dynamics of lattices, especially during the designing phase of new machines. Yet its on-line application seems less common, which prompted us to develop an on-line tool, the DLS-Online Optimiser (DLS-OO) using modern MOO techniques [2].

Three multi-objective algorithms are now available in the DLS-OO: a Genetic Algorithm (MOGA) [3], a Simulated Annealing (MOSA) [4] and a Particle Swarm Optimiser (MOPSO) [5]. MOGA was fully implemented in 2016, while MOSA initially tried as a Matlab version, has now been made fully compliant with the main Python code. Details of these two algorithms can be found in [2], while MOPSO, introduced in 2017, is described in the following sub-section.

Multi-Objective Particle Swarm Optimiser (MOPSO)

This is a population-based optimisation imitating the behaviour of a flock of birds hunting for food. At each iteration, position and velocity of each i individual in the parameter space are given by:

$$\begin{cases} \vec{v}_i^{k+1} = w\vec{v}_i^k + c_1 r_1(\vec{x}_i^{pbest,k} - \vec{x}_i^k) + c_2 r_2(\vec{x}^{gbest,k} - \vec{x}_i^k) \\ \vec{x}_i^{k+1} = \vec{x}_i^k + \vec{v}_i^{k+1} \qquad i = 1, \ldots N; \quad r_1, r_2 \in [0,1] \end{cases}$$

N individuals tend to follow a *leader* closest to the best global solution (weighed by the c_2 social term), with a tendency w to keep their initial velocity and a position close to their personal best finding (weighed by a cognitive term c_1). An element of randomness is introduced via r_1 and r_2. An external archiver is used to classify and rank the solutions, in the fashion described before. Details of the method can be found in [5].

* marco.apollonio@diamond.ac.uk

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-WEPAF054

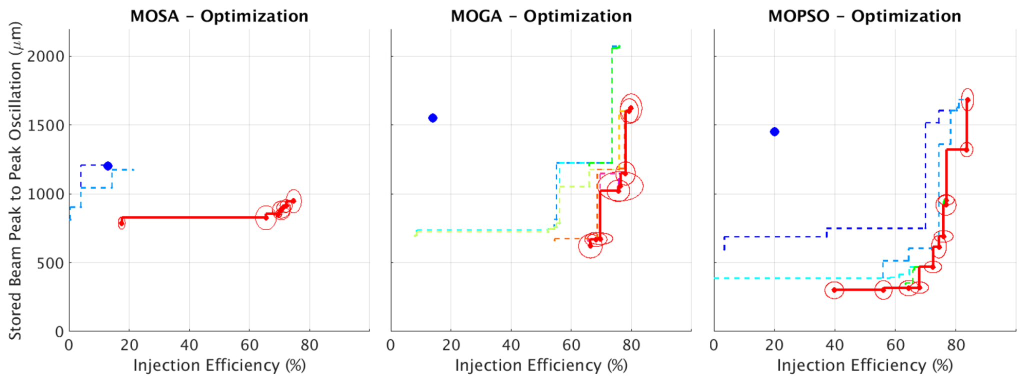

Figure 1: Front evolution for the (IE, PPX) optimisation with three different algorithms: MOSA (left), MOGA (centre) and MOPSO (right). The initial setting (blue dot) evolves through intermediate fronts (dashed lines) to the final front (red line).

OPTIMISATION PACKAGE

A thorough description of the optimisation package has been given in [2]. We highlight here its main features. The structure of the code, written in Python, consists of:

- a main user interface (UI),
- an optimiser,
- an interaction controller,
- an off-line post-processor.

Input variables and objectives are configured in the main UI, where the choice of the optimisation algorithm is made. Input variables can be grouped into families, with absolute or relative bounds for each parameter. Minimum wait times for the process variables are also defined at this stage as well as the number of samples to be measured for each objective. After this stage, the optimisation script is called, allowing to define the algorithm-specific parameters and controlling any visualization relevant to the chosen algorithm. The interaction controller is responsible for applying the input variables to the machine via EPICS and for returning each of the objective functions. It also maps between the algorithm-specific parameters (*e.g.* relative change in strength for each sextupole family) and the machine parameters (*i.e.* absolute individual magnet strengths). For each objective, mean and standard error are returned over the specified number of measurement samples after elimination of outliers. Non-dominated fronts are displayed during the optimisation after each generation. At the end of the optimisation the machine is returned to its initial state and a second interactive window is opened, giving the possibility of selecting any given point of the final front. Results are archived for post-analysis.

MACHINE TESTS

To illustrate the effectiveness of the optimiser, we performed a series of tests, here reported.

Injection Efficiency and Residual Horizontal Oscillation in the Stored Beam (IE, PPX)

In this optimisation the objective was to improve the IE and to reduce the PPX in the stored beam. Four parameters were used: the two last horizontal steering magnets in the booster transfer line and the amplitude and timing of a pinger magnet located in cell 23, used to combat the residual kicks. The initial pinger settings had been determined in a separate campaign meant to explore the possibility to use a similar device to mitigate the after-injection oscillations seen in the stored beam. For the test we initially spoiled the machine, creating a setting with a low IE (15 to 20%) and a large post-injection residual kick (1200 to 1500 μm). The MOPSO, MOGA and MOSA algorithms, were then put to the test, whose result is summarized in Fig. 1. Their initial settings were mainly chosen to equalize the time spent for each test (about 1 hr). MOGA was set with 20 individuals and 10 generations, MOPSO with 50 individuals and 5 iterations and MOSA with 5 annealings of 35 iterations each.

While we reserve a more careful optimisation of these constituting parameters for future trials, it is interesting to notice some typical features of the results. The MOSA case seems to reach a good optimised set relatively soon (after three annealings), however the variety of solutions is relatively poor, both respect to MOGA and to MOPSO. MOGA seems indeed to reach a better variety of solutions, as expected from literature. However MOPSO seems to perform even better, with a more dominating front and a larger variety of solutions. All three algorithms manage to rescue the system from an initially bad configuration bringing it basically back to the typical operating mode (82%, 1280 μm).

Injection Efficiency and Lifetime in the SR (IE, LT)

IE and LT are typically tackled by acting on the harmonic sextupoles of the SR, trying to increase the dynamic aperture and reduce the Touschek scattering rate. Six sextupole families were used as parameters for the test, whose evolution is summarized in the time trace of Fig. 2. Direct LT measurements need long settling times, making this variable impractical for an implementation in the DLS-OO. To speed up the procedure we opted for a proxy variable based on the beam loss rate monitored by a photo-multiplier tube (PMT) located after the collimators of cell 1, as routinely done at Diamond during spin depolarization measurements [6]. The beam current (I_b) keeps growing during the test, altering the LT both in a direct way and through the electron bunch

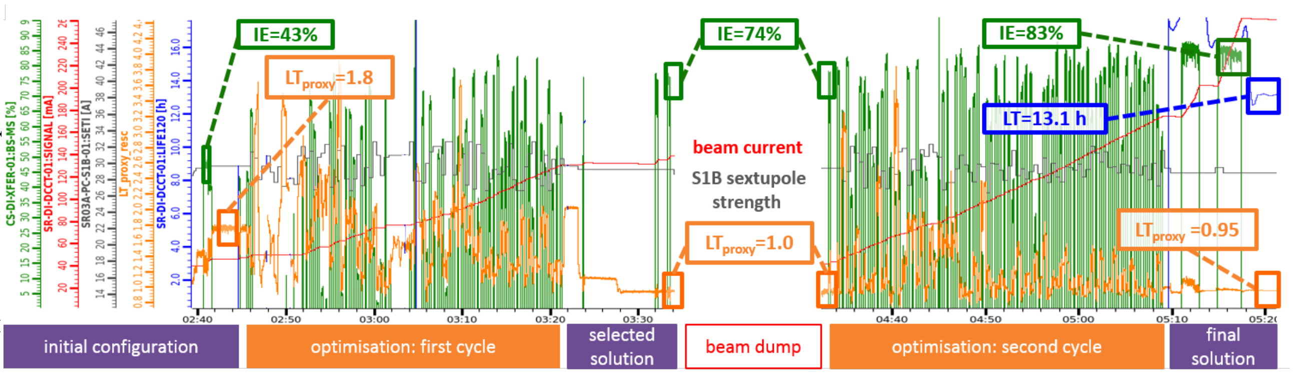

Figure 2: Objective evolution during the run to optimise IE (green trace) and LT$_{proxy}$ (orange trace). The initial settings (43%, 1.8) are marked. Two optimisation cycles were performed, interleaved by a beam-dump and recovery of the machine. At the end the initial LT of about 13 hr at 260 mA was recovered, together with the initial IE (83%).

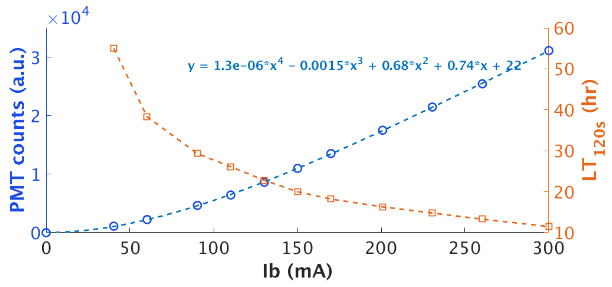

Figure 3: Rate loss as measured with a PMT (blue) and lifetime (red) versus beam current. A fit to a 4th order polynomial is used in the optimiser to parametrize the loss rate.

length, in turn a function of I$_b$. Also, the PMT is a local monitor, whereas Touschek events appear to be distributed along the SR with a pattern that might change for different lattice configurations. To tackle these issues we measure the PMT rates as a function of I$_b$ at the beginning of an optimisation run (see Fig. 3), defining an LT proxy variable as:

$$LT_{proxy} = \frac{PMT^{meas}_{rate}(I_b)}{PMT^{calib}_{rate}(I_b)} \cdot \frac{\sigma^{calib}_{y}}{\sigma^{meas}_{y}}. \qquad (1)$$

For a given machine, Eq. (1) shows that LT$_{proxy}$ should be equal to one for all currents, with deviations from unity expected when sextupoles are altered. An increase in Touschek losses in the SR would then appear as LT$_{proxy}$ > 1. The rightmost fraction compensates for undesired variations in the vertical size of the beam, being typically one when operating with an active vertical emittance feedback.

For the test we spoiled the initial configuration with (IE, LT$_{proxy}$)=(83%,1.0) corresponding to LT=13.2 hr at 260 mA, by altering the harmonic sextupole strength by 5%. This produced an initial setting with (IE, LT$_{proxy}$)=(43%,1.8), or a LT=6.6 hr. We then performed a MOPSO run with three generations and 25 individuals, picked up a solution from its final front as a new initial configuration and ran another optimisation terminating with the final front shown in Fig. 4. Also in this case we demonstrated how a spoiled machine

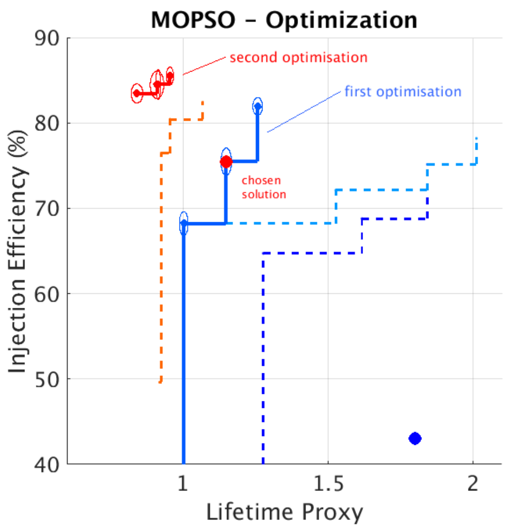

Figure 4: (IE, LT$_{proxy}$) two-step SR optimisation. The initially spoiled configuration is marked by the blue dot at (43%, 1.8). The first optimisation produced a non-dominated front (solid blue line) from which a solution was chosen (red dot) as a start for a second optimisation producing the final red front. Intermediate fronts are represented by dashed lines.

can be brought back to its standard performance. So far any attempt at reaching a better configuration has not produced the hoped results, possibly showing that our machine is already operating at its best.

CONCLUSIONS

The DLS-OO has evolved since its inception in 2016. Outlier rejection and implementation of results from previous optimisation are part of the suite. We have shown a first comparison of three diverse multi-objective algorithms, for the optimisation of (IE, PPX) and a MOPSO optimization of (IE, LT). More detailed comparative studies of the different algorithms are part of the future tests, as well as the introduction of a virtual machine in the optimisation package.

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-WEPAF054

REFERENCES

[1] X. Huang, J. Corbett, J. Safranek, J. Wu, “An algorithm for online optimization of accelerators”, *Nucl. Instr. Meth. A*, vol. 726, pp. 77–83, 2013.

[2] I. P. S. Martin *et al.*, “An Online Multi-Objective Optimisation Package”, in *Proc. Conf. IPAC2017*, Copenhagen, Denmark, May 2017, paper THPAB153, pp. 4092–4095.

[3] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II”, *IEEE Trans. Evol. Comput.*, vol. 6, No. 2, pp. 182–197, 2002.

[4] A. Suppapitnarm, K. A. Seffen, G. T. Parks, P. J. Clarkson, “A simulated annealing algorithm for multiobjective optimization”, *Engineering Optimisation*, vol. 33, No. 1, pp. 59–85, 2000.

[5] X. Pang, L. J. Rybarcyk, “Multi-objective particle swarm and genetic algorithm for the optimisation of the LANSCE linac operation”, *Nucl. Instr. Meth. A*, vol. 741, pp. 124–129, 2014.

[6] I. P. S. Martin, M. Apollonio, R. T. Fielder, G. Rehm and R. Bartolini, “ Energy Measurements with Resonant Spin Depolarisation at Diamond”, in *Proc. Conf. IPAC2011*, San Sebastian, Spain, Sept. 2011, paper TUPC159, pp. 1404–1406.