ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-THPML028

GENETIC ALGORITHMS FOR MACHINE OPTIMIZATION IN THE FAIR CONTROL SYSTEM ENVIRONMENT

W. Geithner*, Z. Andelkovic, S. Appel, O. Geithner, F. Herfurth, S. Reimann, G. Vorobyev
GSI Helmholtzzentrum für Schwerionenforschung, Darmstadt, Germany
F. Wilhelmstötter, Emarsys Information Technology and Services, Vienna, Austria

Abstract

An automated beam-setting optimization application has been implemented on top of FAIR's control system software stack based on CERN's LSA framework. The optimization functionality is built using the Jenetics software library implemented in Java. Tests of the software with beam have been performed at the CRYRING@ESR ion storage ring.

INTRODUCTION AND BACKGROUND

In recent years, with advances in machine learning and evolutionary algorithms, a number of software libraries became available, allowing researchers and application programmers to utilize these libraries for their purposes. In 2017 we successfully investigated if genetic algorithms can be applied in the context of accelerator optimization [1, 2]. Despite the promising results of this prototype, it came with some disadvantages. The prototype driving and reading device data was programmed in Python and is communicating to the low level FESA stack of the FAIR control system [3] bypassing the higher levels of the FAIR control system [4] based on the LHC Software Architecture (LSA) implemented in Java [5]. We finally decided to implement an application in JAVA built on the high-level LSA-layers and the open-source genetic-algorithm library Jenetics [6].

For testing the software with beam we used the ion source and injector of the CRYRING@ESR ion storage ring, which serves besides its main purpose as machine for atomic and nuclear physics experiments as test bench for the FAIR control system [7].

GENETIC ALGORITHM

In computer science and operations research, a genetic algorithm (GA) is inspired by the process of natural selection. Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as *mutation*, *crossover* and *selection* [8]. The optimization problem can be understood to find the global extreme (minimum or maximum) of an unknown multi-dimensional function $f(\boldsymbol{x})$, where $\boldsymbol{x} = \{x_1, ..., x_N\}$ is a vector of parameters x_i. In terms of an accelerator x_i is one setting value of a device, in terms of the control system one LSA setting parameter.

In GA the term "*gene*" defines one single parameter of the optimization problem together with a value range and current value. The Jenetics' `DoubleGene`-class was used to model devices parameters. A "*chromosome*" is a parameter-vector of K genes, where in our context K is always 1, which is a boundary condition imposed by Jenetics. A "*genotype*" is a vector of L chromosomes, where the specification (type, ranges) of the individual genes may differ from chromosome to chromosome, but not within one chromosome. In the actual implementation, we modeled a set of L accelerator devices to be controlled by the genetic algorithm as a genotype of L Jenetics' `DoubleChromosome` instances with one `DoubleGene` each. The "*genotype*" is defined as one concrete value set of chromosomes/genes, i.e. one set of LSA trim values, A "*population*" is a collection of N so called "*individuals*", where each individual is defined by its genotype. Furthermore, each individual is characterized by its "*fitness*", which is the actual function value to one genotype. In terms of accelerators, the fitness is interpreted as the readout value of a beam-diagnostic instrument. A genotype together with its fitness function is called "*phenotype*".

SOFTWARE ARCHITECTURE

A JAVA application "Device-Scanner" has been developed, which communicates through the top level software layer of the FAIR control system [5]. To achieve flexibility and extensibility, the integration of the genetic algorithm into Device-Scanner was realized via a software interface `ScanAlgorithmIF` which allows future algorithms to be integrated too. The management of the actual access to the LSA- and JAPC layer (trimming/readout) is encapsulated in the Device-Scanner framework, so the developer of an "algorithm" does not have to care about these aspects and can focus on the development of the algorithm. The Jenetics framework is called in a class `GeneticScanner` implementing `ScanAlgorithmIF`. The three most important methods of this class are `initAlgorithm(...)` which builds the genes, chromosomes and gentotype required by the genetic algorithm, furthermore it instantiates the Jenetics `Engine<DoubleGene, Double>` engine. The method `fitness(Genotype<DoubleGene> genotype)` implements the actual calls to the outside framework. It triggers setting of trims and readout of beam diagnostic instruments and returns this value to the Jenetics framework, which interprets this value as the "fitness". `performScan(...)` starts the actual generation of setting values by Jenetics and the control of the scanning process.

ENVIRONMENTS FOR TESTING

For the purposes of testing the optimization function without beam it is desirable to use a multi-dimensional function

* w.geithner@gsi.de

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-THPML028

which has only one global maximum and minimum and is valid over a large range of values. We realized these requirements by the following function, which is a derivative of Rastrigin's function [9]:

$$f(\boldsymbol{x}) = \prod_{i=1}^{N} x_i + \sum_{i=1}^{N} x_i^3 \sin^2(0.1\,\pi\, x_i) - (\prod_{i=1}^{N} x_i)\cos(0.5\pi\, x_i)$$

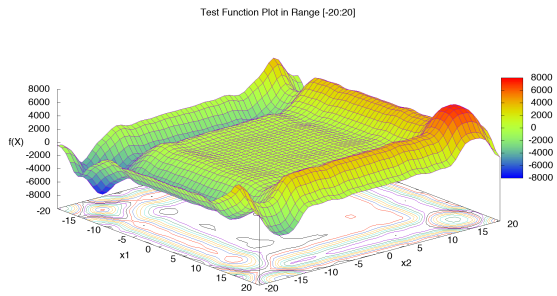

Figure 1: 2-dimensional example-plot of the function in the range $[-20 \leq x_i \geq +20]$ used for testing the implementation of the genetic algorithm without beam.

The function shown for 2 parameters in Fig. 1 serves as virtual "readout device" during the tests without beam. The code is processed in the same way as with real readout devices, the only difference is that the actual trim of settings is suppressed if the test -"device" is used.

Like in our earlier publication [2] we performed tests with beam using the local injector of the CRYRING@ESR ion storage ring comprising most important devices of an accelerator such as dipole, quadrupole magnets, electrostatic elements, an RFQ-linac, etc. The tests were performed using a 40 keV deuterium beam from the local ion source. The ion source produced an ion beam with a total intensity between 300 and 800 µA. Almost 90 per cent of this mixture was D_2^+, the remaining ten per cent dominated by D^+.

Two scenarios have been used for the test of the algorithm: 1) Find the more intense D_2^+ mass peak detected on the Faraday cup within a broad parameter range of the mass separation magnet, which also covers the less intense D^+ peak. 2) Optimize the beam intensity of a single ion species on the Faraday cup by tuning the dipole magnet in only a small scan range, but additionally tuning the electrostatic quadrupole doublet lenses before and the two magnetic steerers behind the mass separator magnet, see Fig. 2. These five devices can be controlled via 9 LSA-parameters: currents applied to the magnets, 4 voltages applied to the first electrostatic lens, and 2 voltages applied to the second electrostatic lenses (see Fig. 2).

To minimize the influence of the ion source fluctuations on the tuning result, instead of the beam intensity, the transmission through the beamline has been evaluated by normalizing the beam current measured on the Faraday cup after the mass separator magnet to the total beam current from the ion source measured with a current transformer after the source.

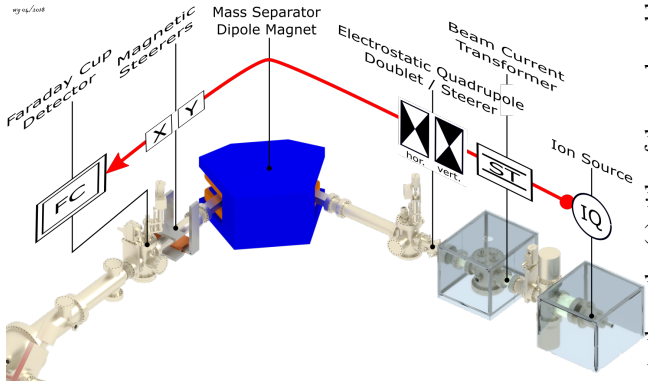

Figure 2: CRYRING@ESR ion source and low energy beamline with elements/devices used to perform the tests of the genetic algorithm with beam.

EXPERIMENTAL RESULTS

The first test scenario "Broad range of single parameter" is performed by varying the current through the dipole magnet. To qualify the performance of the genetic algorithm, three aspects are of interest: 1) transmission through the magnet, 2) time to complete the task, 3) does the genetic algorithm converge on a local or the global maximum?

The test was performed in two steps: first a mass spectrum, i.e. measured signal on the Faraday cup versus dipole current in the range of 15-30 A was taken with a resolution of 0.05 A. The peak of the D_2^+ signal has a width of about 0.2 A. In a second step, a genetic optimization was performed in the same current range.

The result of the genetic optimization is shown in Fig. 3, where two parameters are plotted against the number of steps (abscissa): current through the dipole magnet , i.e. LSA-parameter (blue, right ordinate) and transmission, i.e. the "fitness" (red, left ordinate). Obviously the algorithm converges to one current value after about 120 steps. The transmission still fluctuates which is due to ion source instabilities which cannot be mitigated by normalizing to the current transformer.

This test scenario showed that - most important of all - the genetic algorithm converged in the global maximum (= the $D_2^+ - peak$). The performance in terms of transmission was comparable to what the standard mass scan revealed. The performance in terms of time was comparable to determining the D_2^+-peak via a mass scan in the broad scan range.

The test scenario "beamline optimization" was performed in two steps: scanning of parameters in broad scan ranges, extract corrdidors of "good" settings and scan again with more narrow ranges. In the first step, the beamline devices were scanned in broad ranges: the mass separator magnet +/- 1 A around the D_2^+-peak determined earlier. The six voltages applied to the electrodes of the electrostatic quadrupole lenses in a range of +/- 500 V around previously known

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-THPML028

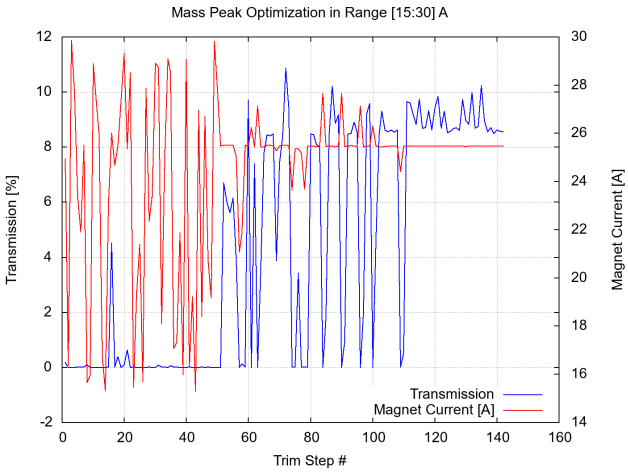

Figure 3: Plot of the transmission against number of steps in a genetic scan. Note that the plot has two y-axes: to the left the transmission in %, to the right the current to power the dipole magnet.

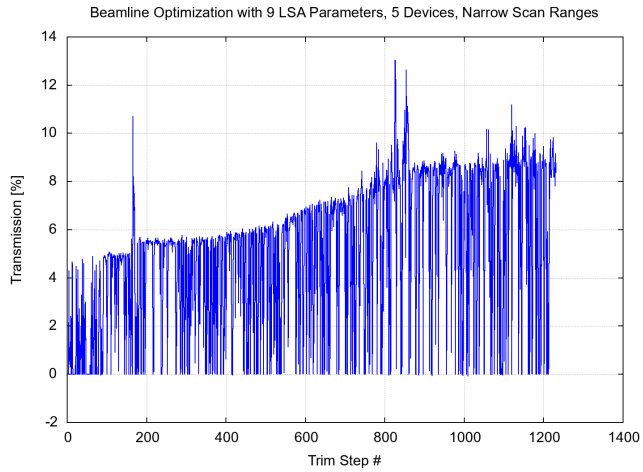

Figure 4: Converged genetic scan driving 9 LSA parameters in narrow scan ranges. The scan reached the final value after 22 generations. Note that the sparks in the plot (values above 9) are due to fluctuations of the ion source

"good" settings, which equals to 1/4th of the maximum voltage range of these devices of 0-4000 V. The two steerer magnets were scanned in the range from -1 A to +1 A, half of the maximum current. The settings of the genetic scanner were rather "aggressive" (e.g large values for mutation rate) to achieve quick exploration of the large parameter space. This approach did not converge within the given span of time and was terminated manually. Two factors influenced the result: the strategy led to large jumps in the parameter values and as a consequence transmission, making it hard for the algorithm to converge smoothly towards the maximum. Furthermore, the performance of the ion source, especially instable plasma conditions play a crucial role, as it introduces non-deterministic transmission fluctuations which cannot be coped with by the algorithm without further measures.

Nevertheless, the data collected during this scan contains valuable information as input for a second optimization step. In a "postmortem" analysis of the data, one can extract corridors of good settings for each of the parameters involved in the scan. This can be achieved by analyzing the setting values distributions from the first step show pronounced peaks at high transmission. The good setting corridors determined in this way can lateron be used to reduce the scan ranges in a second iteration of genetic scanning. Currently, this step has to be done manually but should be automated in the future. With the narrowed scan ranges and less aggressive algorithm settings (low mutation rate), the algorithm converges, but slowly as shown in Fig. 4. The number of trims send to the hardware was 1200, which corresponded to about 135 minutes of scanning.

Two factors hampered - again - quicker convergence of the algorithm: with +/- 1 A around the mass peak, the scan range of the mass separator magnet was chosen too large and the ion source caused non-deterministic fluctuations in the transmission. Admittedly the test scenarios did not consider physics input on reasonable scan ranges which would have been the case in real-life operation. Furthermore, in real-life operation it would have been possible to quicken the process by stopping the algorithm manually after if reached stable setting (approximately after half of the scan time) and to perform the histographic analysis described before.

CONCLUSION AND OUTLOOK

With the prototype implementation presented in this paper, we are able to provide an end-user application exploiting the genetic algorithm framework Jenetics to optimize unknown beamline settings through the LSA framework of the FAIR control system. Tests featuring a mass separator magnet and simple ion optical elements in the injector beamline of CRYRNG@ESR showed that the Jenetics framework is able to drive the corresponding devices in a way that the algorithm converges at a global maximum based on Faraday cup readings. The genetic scanning approach seems to be useful for beamline "exploration", e.g. determine good beamline settings in situations where estimated settings are only known from simulations, extrapolation, etc. Furthermore, the genetic algorithm has potential to reduce the "human factor" in beamline optimization, which is influenced by operator skills and experience, concentration level at the end of a shift, etc.

Looking forward, we envisage the following steps to improve the Jenetics-driven beamline optimization: systematic studies of the influence of the various parameters parameterizing the Jenetics algorithm need to be performed to determine optimal settings for the purpose of optimizing beamline settings most efficiently and effectively. Furthermore, the algorithm shall be applied to the more complex environment of the storage ring as such including higher level beam physics parameters such as beam energy matching of the ring, tune, chromaticity, etc.

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-THPML028

REFERENCES

[1] S. Appel, O. Boine-Frankenheim, and F. Petrov, "Injection optimization in a heavy-ion synchrotron using genetic algorithms", *Nucl. Instrum. Methods A*, vol. 852, pp. 73-79, 2017.

[2] S. Appel, S. Reimann, V. Chetvertkova, W. Geithner, F. Herfurth, U. Krause, M. Sapinski, and P. Schütt, "Automatized optimization of beam lines using evolutionary algorithms", in *Proc. IPAC'17*, May 2017, Copenhagen, Denmark, pp. 3941-3944, 2017, https://doi.org/10.18429/JACoW-IPAC2017-THPAB096

[3] T. Hoffmann, "FESA - The front-End Software architecture at FAIR", in *Proc. PCaPAC'08*, Ljubljana, Slovenia, Oct. 2008, pp. 183-185, 2008.

[4] R. Bär, C. Betz, D. Beck, J. Fitzek, U. Krause, S. Jülicher, M. Thieme, and R. Vincelli, "News from the FAIR control system under development", in *Proc. PCaPAC'14*, Oct. 2014, Karlsruhe, Germany, pp. 37-39, 2014.

[5] G. Kruk, S. Deghaye, M. Lamont, M. Misiowiec, and W. Sliwinski, "LHC Software Architecture (LSA) - evolution toward LHC beam commissioning", in *Proc. ICALEPCS'07*, Oct. 2007, Knoxville, TN, USA, pp. 307-309, 2007.

[6] F. Wilhelmstötter, "Jenetics advanced genetic algorithm", http://jenetics.io

[7] W. Geithner, Z. Andelkovic, D. Beck, H. Brauning, A. Brauning-Demian, H. Danared, C. Dimopoulou, M. Engstrom, S. Fedotova, O. Gorda, F. Herfurth, R. Hess, A. Kallberg, C. Kleffner, N. Kotovskiy, I. Kraus, M. Lestinsky, S. Litvinov, F. Nolden, A. Reiter, T. Sieber, M. Steck, and G. Vorobyev, "Status and outlook of the CRYRING@ ESR project", *Hyperfine Interact.* p. 238, 2017.

[8] M. Mitchell, *"An Introduction to Genetic Algorithms"*, MIT Press Cambridge, ISBN:0-262-13316-4, 1996.

[9] L. A. Rastrigin, "Extremal control systems", in *Theoretical Foundations of Engineering Cybernetics Series*, Nauka, Moscow, 1974.