ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-MOPOST001

PERFORMANCE OF AUTOMATED SYNCHROTRON LATTICE OPTIMISATION USING GENETIC ALGORITHM*

X. Zhang†, S. L. Sheehy[1], The University of Melbourne, Melbourne, Australia
[1]also at Australian Nuclear Science and Technology Organisation (ANSTO), Sydney, Australia

Abstract

Rapid advances in superconducting magnets and related accelerator technology opens many unexplored possibilities for future synchrotron designs. We present an efficient method to probe the feasible parameter space of synchrotron lattice configurations. Using this method, we can converge on a suite of optimal solutions with multiple optimisation objectives. It is a general method that can be adapted to other lattice design problems with different constraints or optimisation objectives. In this method, we tackle the lattice design problem using a multi-objective genetic algorithm. The problem is encoded by representing the components of each lattice as columns of a matrix. This new method is an improvement over the neural network based approach in terms of computational resources. We evaluate the performance and limitations of this new method with benchmark results.

INTRODUCTION

The high level goal of this work is to assess whether the 'art' of lattice design can be economically achieved by a computer without assuming existing standard lattice configurations known to accelerator physicists. This paper is a continuation of the effort to develop an automated lattice optimisation algorithm. In the previous implementation of this algorithm [1], a neural network based approach was used to generate new lattice structures. There were two main problems with the neural network based approach. Firstly, new lattices were generated piece-wise based on a feedback loop controlled by tracking the motion of a test particle. This required each successful neural network to have *a priori* encoded values to produce the exact feedback values in order to create a desirable lattice. This is possible in principle, however this system is an overly complicated mapping between the optimisation quantities and the actual machine lattice structure. Secondly, the neural networks are very sensitive to small perturbations of the node values and resulted in drastically different lattice structures. Therefore it becomes computationally expensive to optimise the neural network values for target lattice structures with a large number of elements. The work for this paper tackles both of these problems by introducing a different method for generating lattices. In addition, the newest version of the genetic algorithm was implemented to improve the performance when optimising higher dimensional objectives.

* Work supported by Australian Government Research Training Program
† xuanhaoz@student.unimelb.edu.au

LATTICE MATRIX ENCODING

One major challenge with applying any kind of optimisation algorithm is how to represent the problem and parse it to the algorithm, this is referred to as the encoding of the problem. A matrix encoding method was used in this work, such that any lattice structure with n elements and at most p attributes per element can be represented by a $p \times n$ matrix. For example, a simple FODO cell with separate function dipole magnets is represented by an 8×4 matrix as shown in Fig. 1. Each column explicitly represents a different element of the lattice including empty drift spaces (E). The rows encapsulate all the attributes needed to describe every element in the lattice, where *k0* and *k1* are the dipole and quadrupole coefficients as used by MADX [2].

Element	QF	E	B	E	QD	E	B	E
k0	0	0	1.07	0	0	0	1.07	0
k1	0.23	0	0	0	-0.23	0	0	0
Length	0.3	0.43	0.87	0.43	0.3	0.43	0.87	0.43

Figure 1: Example matrix encoding of a FODO cell with separate function dipoles.

It should be noted that this matrix encoding can include any number of attributes by adding more rows, such as higher order coefficient terms of non-linear magnets. Also, it does not matter to the optimisation algorithm where the actual starting point of the lattice is defined. This property is particularly useful when optimising for a periodic structure such as a ring.

As part of the optimisation algorithm, the aim is to probe as much of the available parameter space as possible. In other words: to explore lattice configurations that we might not think would be stable based on our experience. This calls for a method to randomly sample from the entire parameter space that is bounded by a few basic design requirements such as the total bending angle or betatron tune. Note that there is no constraint on the number of magnetic elements or the total length of arc. Using the matrix encoding method, it is possible to initialise a collection of random lattice structures by building the lattices one element at a time. This means, starting each lattice matrix from a randomly chosen element (drift, dipole, quadrupole, or combined function dipole) and randomly choosing the attribute values necessary for that element. This process is repeated by adding new columns to the lattice matrices until the total bending angle of the lattice exceeds the requirement.

ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-MOPOST001

GENETIC ALGORITHM

The principle for the multi-objective genetic algorithm (MOGA) applied in this work is based on Darwinian evolution. The fittest members of a population are most likely to survive or pass on their traits to future generations as the population evolves. The algorithm used in this work is the constrained and adaptive version of the reference-point-based many-objective evolutionary algorithm (NSGA-III) [3, 4]. Compared to algorithms applied in previous works [1], NSGA-III has improved performance at preserving population diversity for problems with higher dimensional objectives ($N > 3$) as well as reduced computation resource needs. Specifically, NSGA-III allows the user to define a set of reference points within the objective space and change the selection pressure based on the population distribution around each reference point.

A brief outline of the custom implementation in context of this work is provided below, specifics of the NSGA-III are not reproduced here.

1. Build a population of N randomly initialised lattice matrices based on the required total bending angle.
2. Evaluate the one-turn matrix, objective values, and penalty values for each new member of the population.
3. Sort the population into non-domination fronts using non-dominated sorting.
4. Initialise new generation of R randomly initialised matrices and $N-R$ offsprings from the previous generation using genetic operators.
5. Evaluate the one-turn matrix, objective values, and penalty values for each new member.
6. Rank the combined population of $2N$ members and associate each member with the closest reference point.
7. Select N members starting from the first non-domination front. For the last front to be included, select members that would maximise the diversity of associated reference points.
8. Adaptively change the reference point positions to reduce over clustering around any one point.
9. Start new generation and repeat steps 4-8.

Non-Dominated Sorting and Genetic Operators

NSGA-III and previous versions of this algorithm all rely on the idea of non-domination levels, which is a relative measure of the constraint violation and objective satisfaction of the members of a population. As a result of the lattice matrix encoding and the random initialisation process, a large portion of the initial population corresponds to unstable lattice structures. This creates a problem of how to quantify the degree of divergence between unstable lattices. One naive solution is using the trace value of the first order one-turn transfer matrix $\mathrm{Tr}(M_1)$. As our results in the next section demonstrate, $\mathrm{Tr}(M_1)$ is an effective measure of divergence. Our algorithm is able to reach stable lattice structures by favouring unstable lattices with smaller $\mathrm{Tr}(M_1)$ values.

In addition to the original NSGA-III constrained-dominate definition, we added the following. A member p is defined to constrained-dominate another member q if any of the following is true.

- If p is stable in at least one transverse plane and q is unstable in both planes.
- If both are unstable in one plane, but $\mathrm{Tr}(M_1)$ of p in the unstable plane is smaller then that of q.
- If both are unstable in both planes, but the sum of $\mathrm{Tr}(M_1)$ of p in both planes is smaller then that of q.

The *crossover* and *mutation* operators are used to create new lattice matrices from existing members for each generation of the algorithm. The crossover operator selects pairs of matrices from the parent population and creates two new matrices by swapping columns of the parent matrices at the same crossover point. A binary tournament selection process [5] is used to select parent pairs from a randomly selected subset of the population, the non-domination rank is used as the comparison criterion and the size of the subset correlates to the selection pressure. New matrices created by the crossover operator are neither guaranteed to result in better lattice structures compared to their parents nor even stable lattices, however this does not seem to have significant impact on the convergence of the algorithm and can be improved in the future by applying a repair operator [6]. One limitation of the crossover operator is that it does not allow for subtraction of elements from existing matrices, therefore the number of magnetic elements in new matrices can only increase in number. An additional lattice simplification step could be added to the algorithm later if desired.

The mutation operator only acts on a single lattice matrix at a time, it randomly changes the attribute of a randomly selected position on the matrix within the bounds of allowed values for that position. Attributes on the same column are also changed as needed to reflect the mutated attribute. For example, if the *Element* attribute was mutated from QF to CF on the column [QF, $k0 = 0$, $k1 = 0.75$, $l = 0.3$], then a new non-zero $k0$ attribute would also be generated for that column.

RESULTS

We report the results of a benchmark study carried out to check the performance of the algorithm. For this study, the optimisation objectives were the following.

- f_1: Minimise $\max(\beta_x)$.
- f_2: Minimise $\max(\beta_y)$.
- f_3: Minimise number of segments.
- f_4: Minimise number of magnetic elements.

While the constraints were set to mimic a long closed dispersion arc.

- Total bending angle, $B_{\mathrm{tot}} = 180°$.
- Horizontal tune, $Q1 = 0.836$.
- Initial(β, α, D, Dp) = Final(β, α, D, Dp).

Each constraint violation is normalised using the sigma function ($|x|/(1+|x|)$) and the total penalty value is calculated by summing the normalised constraint violations. A maximum penalty tolerance of 0.2 was used to select feasible members from the population. Note the vertical tune was left

ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-MOPOST001

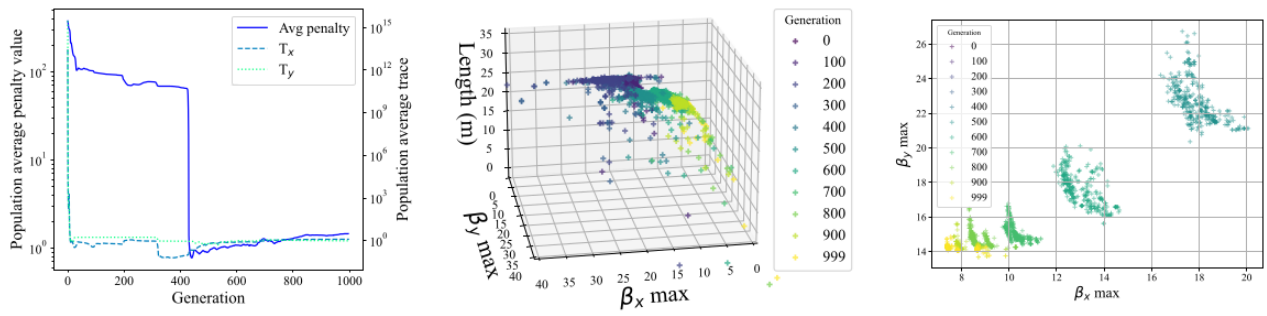

Figure 2: (left) The solid line shows the population average penalty value, the dashed and dotted lines are the population average horizontal and vertical Tr(M_1) values. (mid) Feasible members projected onto 3D objective space, and (right) 2D objective space. The colours from blue to yellow denotes the evolution from generation 1 to 1000.

unconstrained to allow an extra degree of freedom such that the algorithm can populate the Pareto front in the objective space, rather than confined to a single solution that would minimise the constraint violation for a specific vertical tune. A population of 288 random members were initialised and allowed to evolve for 1000 generations. Initially, the objective space was evenly divided into 10 partitions along each objective and a set of 286 reference points were generated to cover the 4D simplex formed by the partitions [7]. The population size was chosen to maximise the efficiency of the reference point association procedure following [4].

The algorithm was able to converge to stable lattice structures within 20 generations using the Tr(M_1) ranking method as shown on Fig. 2 (left). Interestingly, the population average penalty value on Fig. 2 (left) sharply decreases around generation 500, then slowly increases towards generation 1000. The sharp drop can be correlated to a sudden success in matching the total bending angle constraint. While the slow increase after generation 500 is the algorithm attempting to minimise the objectives while exploiting the tolerance range set for the penalty value. Figures 2 (mid) and 2 (right) shows the progression of the population getting closer to the Pareto optimum front over the generations. The distribution on the (β_x max, β_y max) plane matches our expectation of a trade-off relationship. One interesting example of the best performing member is shown in Fig. 3. Although not feasible to build, it is interesting that the algorithm has found a design with long dipole segments and a few closely packed almost alternating focusing segments in between.

The algorithm presented in this work is able to efficiently converge to stable lattice structures and populate the Pareto front of the supplied objectives. However, this feature is also the limitation of the algorithm, since it will attempt to 'cheat' the objectives by slightly tweaking the attributes of a feasible member in order to fill out the objective space. This is in contrast to the original intent of searching for completely different lattice structures to fill out the objective space. As a result, members in the first non-domination front all feature a similar structure as shown in Fig. 3.

Having prior knowledge of lattice design and accelerators more generally, there are now a number of other improvements to be made to the basic algorithm, these are the following. Initialise new lattice matrices conditionally to better represent physical requirements such as leaving extra drift space between each new magnetic element. Introduce repair operator after combination and mutation operations to improve the likelihood of creating new stable structures. Optimise the selection pressure to preserve diversity of candidate structures while maintaining a reasonable convergence rate towards stable candidates. Seeding the algorithm using several stable lattice structures or combining the converged stable candidate pool from several different random initialisations to improve diversity of structures.

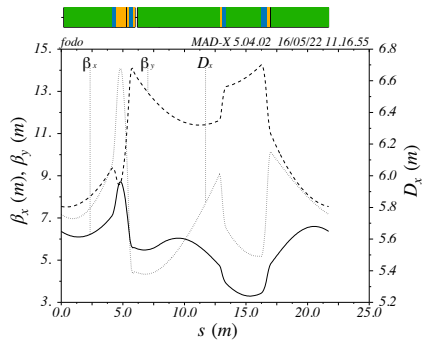

Figure 3: Lattice functions of a member in the first non-dominated front, representing one Pareto trade-off point in the objective space. Green segments are pure dipoles, yellow and blue segments are horizontally focusing and defocusing combined function dipoles respectively.

CONCLUSION

We presented an improved method to generate lattice structures using a lattice matrix encoding method. Our implementation of NSGA-III is capable of finding a Pareto front for a set of given constraints, however it is limited by the lack of a penalty function that measures the diversity of lattice layouts. We plan to address this limitation in future works and apply the algorithm to a full ring structure with slow extraction constraints.

ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-MOPOST001

REFERENCES

[1] X. Zhang, S. L. Sheehy, and E. Benedetto, "Automated Synchrotron Lattice Design and Optimisation Using a Multi-Objective Genetic Algorithm", in *Proc. 12th Int. Particle Accelerator Conf. (IPAC'21)*, Campinas, Brazil, May 2021. pp. 616–619. doi:10.18429/JACoW-IPAC2021-MOPAB182

[2] CERN - Accelerator Beam Physics Group, "MAD - Methodical accelerator design", 2020. https://mad.web.cern.ch/mad/.

[3] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints", *IEEE Trans. Evol. Comput.*, vol. 18, no. 4. pp. 577-601, 2014. doi:10.1109/TEVC.2013.2281535

[4] H. Jain and K. Deb, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach", *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602-622, 2014. doi:10.1109/TEVC.2013.2281534

[5] B. Miller and D. Goldberg, "Genetic algorithms, tournament selection, and the effects of noise", *Complex Syst.*, vol. 9, pp. 193-212, 1995.

[6] M. Ehrgott, *Multicriteria Optimization*, Second Ed., Berlin, Germany: Springer, 2005.

[7] I. Das and J. E. Dennis, "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems", *SIAM J. Optim.*, vol. 8, no. 3, pp. 631-657, 1998. doi:10.1137/S1052623496307510