ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-MOPAB182

AUTOMATED SYNCHROTRON LATTICE DESIGN AND OPTIMISATION USING A MULTI-OBJECTIVE GENETIC ALGORITHM*

X. Zhang†, S. L. Sheehy, The University of Melbourne, Melbourne, Australia
E. Benedetto[1], TERA Foundation, Novara, Italy
[1]also at CERN, Geneva, Switzerland

Abstract

As part of the Next Ion Medical Machine Study (NIMMS), we present a new method for designing synchrotron lattices. A step-wise approach was used to generate random lattice structures from a set of feedforward neural networks. These lattice designs are optimised by evolving the networks over many iterations with a multi-objective genetic algorithm (MOGA). The final set of solutions represent the most efficient and feasible lattices which satisfy the design constraints. It is up to the lattice designer to choose a design that best suits the intended application. The automated algorithm presented here randomly samples from all possible lattice layouts and reaches the global optimum over many iterations. The requirements of an efficient extraction scheme in hadron therapy synchrotrons impose stringent constraints on the lattice optical functions. Using this algorithm allows us to find the global optimum that is tailored to these constraints and to fully utilise the flexibilities provided by new technology.

INTRODUCTION

The Next Ion Medical Machine Study (NIMMS) coordinated by CERN aims to design the next generation hadron therapy system including a compact synchrotron and an efficient beam delivery method for future clinical use [1]. One option for the compact synchrotron design is to use curved Alternating Gradient Canted-Cosine-Theta (AG-CCT) superconducting magnets [2]. This type of magnet allows for layered superconductor coil windings [3, 4], making it possible to create strong combined-function magnets.

Typically, synchrotron lattices are designed around the main bending magnet's capabilities. Long development time is needed to perform any sort of comprehensive comparison of suitable lattice designs and fine tune the lattice to accomodate often conflicting requirements on the optical functions. As a result of using the AG-CCT magnet design, there is also greater flexibility on the lattice layout and opens opportunities for more exotic designs. It would be unfeasible to perform a full comparison of all possible lattice layouts manually. Any such design also carries the risk of very quickly becoming obsolete with new developments in magnet technology.

In this paper, we propose an automated global search and optimisation algorithm that is able to probe the entire feasible parameter space of lattice layouts, and provide optimised design candidates for each specific application. This algorithm can be used as a general tool to assist in the design process of any new synchrotron or beam line.

LATTICE GENERATION

A feedforward neural network represents a transformation from one set of parameters about a given system (input) to another set of parameters (output) which provides meaningful information about the system. It consists of many layers of inter-connected nodes that propagate the input values via non-linear transformations. This transformation can be represented as

$$\mathbf{N}_k = \phi(\mathbf{W}_k \cdot \mathbf{N}_{k-1} + \mathbf{B}_k), \tag{1}$$

where $\mathbf{N}_k = (N_k^1, N_k^2, ..., N_k^n)$ is the k^{th} layer with n nodes; $\mathbf{N}_{k-1} = (N_{k-1}^1, N_{k-1}^2, ..., N_{k-1}^m)$ is the $(k-1)^{\text{th}}$ layer with m nodes; $\mathbf{W}_k$ is the $(n \times m)$ weight matrix of the k^{th} layer, and $\mathbf{B}_k = (B_k^1, B_k^2, ..., B_k^n)$ is the bias value of the k^{th} layer. The activation function, ϕ, is applied to each node to normalise the node values within a finite range and to ensure the overall transformation from input to output is non-linear. The choice of activation function can affect the overall performance of the neural network, the Softsign function $(x(1 + |x|)^{-1})$ is used in this study. There are also numerous methods for initialising a neural network, we initialise the networks with zero biases, and the weights at each layer form a uniform distribution in the range $(-n^{-1/2}, n^{-1/2})$, where n is the number of nodes in the previous layer. See [5, 6] for more details on neural network design.

In order to perform an unbiased search of the global parameter space, we combine first-order particle tracking and a neural network, to generate lattices at random. Starting from an initial particle position in 6D phase-space $\mathbf{X} = (x, p_x, y, p_y, s, \delta p)$, this position vector is fed into the neural network input values. A new lattice segment is generated from the information provided by the network's output layer. Then, the initial particle position vector is propagated through the new segment using transfer matrices. This process of new segment generation is repeated until the desired beam transport distance is achieved.

There are five nodes in the output layer, they are defined as follows.

1. Probability of a dipole component, $\mathscr{P}(k_0)$.
2. Dipole field strength, k_0.
3. Probability of a quadrupole component, $\mathscr{P}(k_1)$.
4. Quadrupole field strength, k_1.
5. Length of the new segment, l_{seg}.

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

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-MOPAB182

The magnet type of each new lattice segment is determined by randomly sampling from a uniform distribution (*Rand*) and comparing it with the probability values from nodes 1 and 3. If both $\mathcal{P}(k_0)$ and $\mathcal{P}(k_1)$ are less than *Rand*, then the new segment is set as an empty drift. A schematic of the neural network structure and lattice generation process is shown in Fig. 1.

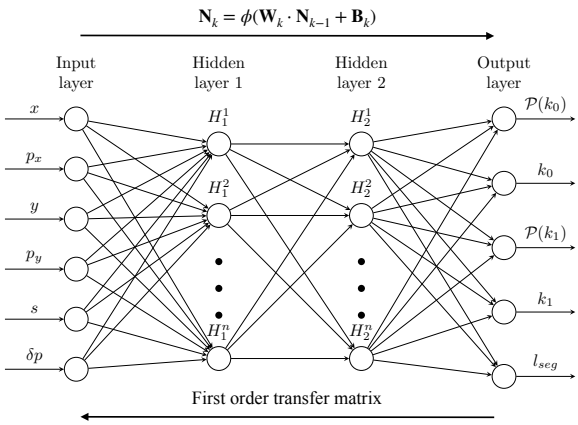

Figure 1: Schematic of the neural network structure and the propagation of test particle coordinates.

In principle, this algorithm is able to produce lattice designs that account for both transverse and longitudinal beam dynamics as well as non-linear effects. Currently, only first-order transfer matrices are used, and only one particle with initial position $\mathbf{X}_0(x, p_x, y, p_y, s, \delta p) = (0.5\ m, 0.5, 0.5\ m, 0.5, 0\ m, 0)$ is tracked.

Each completed lattice is compiled and evaluated using MADX [7], a macro is used to search for the periodic Twiss functions of the lattice. This is one of the current limitations of the algorithm, typically a long computation time is required to find the periodic solution. Other methods that could remove the need to invoke MADX are being investigated, such as modifying the neural network input to use an ensemble of particle positions or transport the Twiss parameters.

MULTI-OBJECTIVE GENETIC ALGORITHM

We apply the Constrained Non-dominated Sorting Genetic Algorithm (Constrained NSGA-II) to train the neural networks. This is a type of artificial evolutionary algorithm that was inspired by evolutionary processes in nature, where the fittest candidate has the highest probability to survive and pass on their genetic information to future generations.

In context with our task, each neural network is a candidate fighting for survival, the network weights ($\mathbf{W}$) are the genetic information. Two neural networks can combine to produce new offsprings, this is achieved by performing *cross-over* and *mutation* operations on the networks' genes. By combining the rows of the weight matrices ($\mathbf{W}_k$) of each network into a 1D vector, it is simple to apply the cross-over and mutation operations. A *single point cross-over* slices and swaps the 1D string of both networks at a common position, thus creating two new strings that can be re-assembled back to neural networks. A *binary mutation* inverts the weight value of a random node on the network.

The goal of multi-objective optimisation is to find the best solutions that satisfy the objectives within the whole feasible parameter space. Only key concepts of NSGA-II are described here, please refer to [8, 9] for more details.

The multi-objective optimisation process can be described as searching for a set of solutions $\mathbf{x} = (x_1, x_2, ..., x_n)$ that satisfies some constraints

$$c_i(\mathbf{x}) = 0 \quad \text{or} \quad c_j(\mathbf{x}) \geq 0, \tag{2}$$

and optimises (minimises or maximises) some objective functions

$$f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x})). \tag{3}$$

A solution x is said to *dominate* another solution ($x' \in \mathbf{x}$) if and only if

$$\forall i \in (1, 2, ..., m) : f_i(x) \leq f_i(x'), \quad \text{and} \tag{4}$$

$$\exists j \in (1, 2, ..., m) : f_j(x) < f_j(x'). \tag{5}$$

In addition, a solution x *constrained-dominate* another solution x' if any of the following is true.

- Solution x satisfies all the constraints $c(x)$ and solution x' does not.
- Both solutions violate at least one constraint, but x has a smaller overall constraint violation.
- Both solutions satisfy all the constraints and x dominates x'.

Finally, the crowding-distance (i_{distance}) of a particular solution is defined as the average distance of all neighbouring points along each of the objective functions. It is used to measure the diversity of solutions surrounding a particular solution in the objective function space.

A schematic diagram of the NSGA-II algorithm is shown in Fig. 2. The steps of Constrained NSGA-II optimisation applied to our task are as follows.

1. The algorithm starts from a population of P neural networks that have randomly initialised network weights. For each network, a corresponding lattice is generated according to the method described in the previous section.
2. The whole population is sorted and given a rank using the constrained-dominate definition.
3. $P/2$ pairs of networks are selected using binary tournament selection [10], the selection criterion is based on the non-dominated rank and crowding distance of the candidates.
4. For each selected pair, a single point cross-over is performed to create two offsprings, this increases the total population size to $2P$.
5. Each offspring also has a chance to undergo one binary mutation on one of its weights.

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-MOPAB182

6. The full population with $2P$ members is sorted again, the top P members are selected to proceed to the next iteration from step 3.

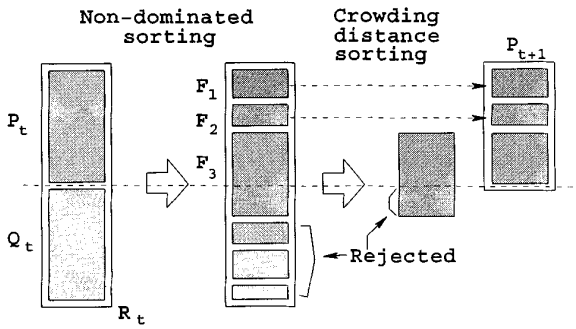

Figure 2: Schematics of NSGA-II [8]. The initial population is denoted by P_t and the offsprings are denoted by Q_t. The groups F_1, F_2, F_3 represent non-dominated ranks.

PRELIMINARY RESULTS

In this benchmark study, we test the performance of the automated algorithm by creating a short 6 meter long beam line. The dipole field strength is capped at 1.5 T for easier comparison with normal conducting FODO cells. The following constraints are applied:

1. Stability criterion, $\mathbf{Tr}\,(\mathscr{M}_{\text{one-turn}}) < 2$.
2. Horizontal tune, $Q1 = 0.1672$.
3. Vertical tune, $Q2 = 0.172$.
4. Total bending angle, $B_{\text{arc}} = 20°$.
5. Initial$(\beta_x, \alpha_x, \beta_y, \alpha_y)$ = Final$(\beta_x, \alpha_x, \beta_y, \alpha_y)$.

As well as three optimisation objective functions:

- f_1: Minimise max(β_x).
- f_2: Minimise max(β_y).
- f_3: Minimise number of segments.

Neural networks with 2 hidden layers and 10 nodes per hidden layer are used. An initial population of 1000 random neural networks were used and allowed to evolve for 100 iterations.

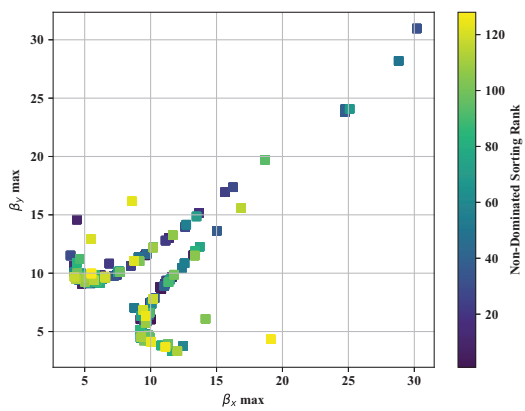

Figure 3: Non-dominated front of the three objective functions after 100 iterations, projected onto the max (β_x), max (β_y) plane.

Even with a relatively small population size and iteration steps, the algorithm is able to produce stable lattices as well as provide an indication of the border of the feasible region, this is clearly visible on Fig. 3. Figure 4 shows the lattice structure and optical functions of the top ranking candidate at the end of 100 iterations. This demonstrates that it is possible to automatically produce convergent lattice structures with our algorithm. It is also important to note the top ranking design features an unexpected lattice arrangement with a reasonable number of magnets, and it offers optical functions that are competitive with baseline FODO cells.

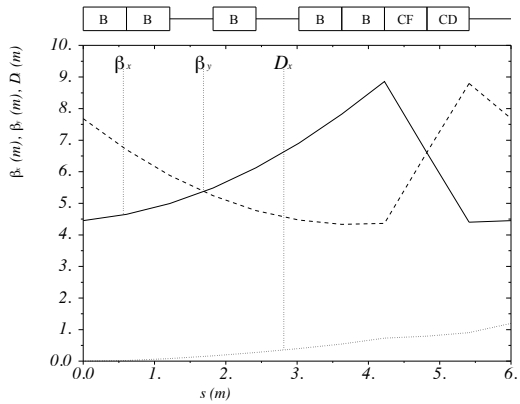

Figure 4: Lattice functions of the top ranking candidate in our benchmark study. The element labels B, CF and CD refers to pure dipole magnets, combined function dipole with horizontal focusing and combined function dipole with horizontal de-focusing respectively.

CONCLUSION

We present a novel method to automatically generate lattice structures using neural networks. The networks are trained with a multi-objective genetic algorithm. Our preliminary benchmark result shows the automated algorithm is capable of producing convergent and sensible lattice layouts. The promising results from our benchmark study provides motivation for extending the algorithm to create closed ring lattices with parameters and constraints of the NIMMS compact synchrotron. Future work will incorporate constraints on beam optics to achieve slow extraction and optimisation of superconducting magnet aperture.

ACKNOWLEDGEMENTS

The stepwise lattice design approach was inspired by the wonderful video from the channel Art of Engineering [11].

REFERENCES

[1] M. Vretenar *et al.*, "The Next Ion Medical Machine Study at CERN: towards a next generation cancer research and therapy facility with ion beams", presented at the 12th Int. Particle Accelerator Conf. (IPAC'21), Campinas, Brazil, May 2021, paper MOPAB413, this conference.

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-MOPAB182

[2] E. Benedetto *et al.*, "A carbon-ion superconducting gantry and a synchrotron based on canted cosine theta magnets", submitted for publication. arXiv:2105.04205

[3] L. Brouwer *et al.*, "Design of an achromatic superconducting magnet for a proton therapy gantry", *IEEE Trans. Appl. Supercond.*, vol. 27, no. 4, pp. 1–6, Jun. 2017. doi:10.1109/TASC.2016.2628305

[4] L. Brouwer *et al.*, "Design and test of a curved superconducting dipole magnet for proton therapy", *Nucl. Instrum. Methods Phys. Res. Sect. A*, vol. 957, p. 163414, Mar. 2020. doi:10.1016/j.nima.2020.163414

[5] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", in *Proc. AISTATS'10*, Sardinia, Italy, May 2010, pp. 249-256.

[6] J. Branke, "Evolutionary algorithms for neural network design and training", in *Proc. INWGA'95*, Vaasa, Finland, Jan. 1995, pp. 145-163.

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

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182-197, Apr. 2002. doi:10.1109/4235.996017

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

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

[11] Art of Engineering, "Can A. I. design the perfect roller coaster?", https://youtu.be/4l5MGqrAItU