ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-ICAP2018-SAPAF01

GENETIC ALGORITHM ENHANCED BY MACHINE LEARNING IN DYNAMIC APERTURE OPTIMIZATION

Yongjun Li,* Weixing Cheng, Li Hua Yu, Robert Rainer,
Brookhaven National Laboratory, Upton, New York 11973

Abstract

With the aid of machine learning techniques, the genetic algorithm has been enhanced and applied to the multi-objective optimization problem presented by the dynamic aperture of the NSLS-II storage ring.

INTRODUCTION

Population-based optimization techniques, such as evolutionary (genetic) [1–16] and particle swarm [17–19] algorithms, have become popular in modern accelerator design. Optimization of a nonlinear lattice's dynamic aperture usually has multiple objectives, such as the area and the profile of the dynamic aperture, energy acceptance, beam lifetime [1, 3], and nonlinear driving terms (NDT) [4] etc. Dynamic aperture and energy acceptance can be evaluated through direct single-particle tracking simulations. NDTs can be extracted analytically from the one-turn-map for a given nonlinear lattice configuration [20–23]. Recent studies have found that the spread from a constant of the action obtained with the square matrix method [24–27] represents a kind of nonlinearity measure of a lattice, which can be treated as an optimization objective as well. Another optimization objective, which is deduced from the square matrix method and used in this paper, is the spread of linear action $J_{x,y}$ from a constant. The spread is numerically computed from simulated turn-by-turn data [28, 29]. Based on the number of objectives presented in this application, multi-objective genetic algorithm (MOGA) [30] is a suitable optimization tool to compromise among these objectives simultaneously.

A general model for multi-objective optimization is:

- given a set of free variables x_n within the range $x_n \in [x_n^L, x_n^U]$, $n \in [1, N]$;
- subject to some constraints $c_j(x_n) \geq 0$, $j \in [1, J]$, and $e_k(x_n) = 0$, $k \in [1, K]$;
- simultaneously minimize a set of objective functions $f_m(x_n)$, $m \in [2, M]$.

Here x_n^L, and x_n^U are the lower and the upper boundaries of the n^{th} free variables. N, J, K and M are non-negative integers. Note for simplicity, clarity, and without loss of generality, all constraints are lower bounds, and all objectives are minimized.

A genetic algorithm (GA) is a type of evolutionary algorithm. It can be used to solve both constrained and unconstrained optimization problems based on a natural selection process [30]. Each candidate has a set of free variables which it inherits from its parents and is mutated at random corresponding to a certain probability. Each candidate's free variables x_n can be regarded as an N-dimensional vector $\mathbf{x}$. Their ranges $[x_n^L, x_n^U]$ define a volume of an N-dimensional "search space". The evolution is an iterative process. The new population from each iteration is referred to as a "generation". The process generally starts with a population that is randomly generated and the fitness of the individuals is evaluated. Individuals with greater fitness are randomly selected, and their genomes are modified to form the next generation. The average fitness of each generation therefore increases with each iteration of the algorithm. The goal of multi-objective optimization (MO) is to optimize functions simultaneously. These functions are sometimes related and their objectives may conflict. In these events, trade-offs are considered among the objectives. In non-trivial MO problems the objectives conflict such that none can be improved without degrading others in value and are referred to as non-dominated or "Pareto optimal". In these cases a non-dominated sorting algorithm can be used to judge if one candidate is better than another [30]. In the absence of constraints or preferences, however, all Pareto optimal candidates are equally valid and given the same rank. If constraints are provided, the rank of each individual accounts for the constraints, and qualified candidates are guaranteed to dominate unqualified ones. Each qualified candidate has M fitness values f_m, which compose another M-dimensional "fitness space". The combination of multi-objective, non-dominated sorting with employment of the genetic algorithm forms the basis of the "MOGA" method. MOGA has some limitations in its application to modern storage ring optimization. In general, the application of MOGA on dynamic aperture optimization can be driven by either direct particle tracking, or analytical calculation of nonlinear characterization. It is time-consuming to evaluate the fitness quantitatively, as seen with the calculation of a large-scale storage ring's dynamic aperture using the symplectic integrator [31].

Although there is no a priori reason why the genetic evolution process needs external intervention, examples without it such as the evolution of biological life on earth or planetary formation in the solar system, were only possible after billions of years [32]. One reason why natural evolution is comparatively slow is that the percentage of elite candidates among the whole population is low. A brute force method for speeding up evolution is to narrow down the search ranges around good candidates found early in the evolution process. This decreases diversity, however, and could lead to a trapping in local minima. An effective intervention step would be able to significantly speed up the evolution in the desired

* Email: yli@bnl.gov

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-ICAP2018-SAPAF01

direction. To do so, some machine learning techniques are introduced to traditional MOGA methods to intervene on the natural process.

MOGA ENHANCED BY MACHINE LEARNING

During the evolution process, MOGA produces a large data pool. It is possible to reuse the data with machine learning techniques to intervene on the evolution process. Here an intervention method is introduced which is schematically illustrated in Fig. 1. It includes the classification of the search space (unsupervised learning), sorting based on the average fitness and repopulation of potential elite candidates (supervised learning). Starting with randomly distributed individuals, the initial population is allowed to produce descendants via the traditional genetic algorithm. Once all candidates satisfy some desired constraints, sufficient data is accumulated to intervene on the evolution process using machine learning techniques. For each following generation, all populations are classified into different clusters in the search space based on a parameter D,

$$D = \sqrt{\sum_{n=1}^{N} (x_{1,n} - x_{2,n})^2}, \tag{1}$$

which represents the "Euclidean distance" between two candidates $\mathbf{x}_1$ and $\mathbf{x}_2$ in the search space. The classification was performed with the K-means algorithm [33] as shown in the subplot (b) of Fig. 1.

After classifying candidates into different clusters, a statistical analysis is carried out on each cluster to evaluate their average or weighted fitness F, which reads as

$$F = \sum_{m=1}^{M} w_m f_m(x_n). \tag{2}$$

Here w_m is the weight on the m^{th} fitness value of f_m. As mentioned previously, our optimization has multiple objectives. Within each generation, most of the candidates belong to the same rank on the Pareto front. Although they are equally good (they exhibit no dominance) and a lot of candidates have one or two good fitness values, the rest have poor fitness. They can survive through many generations unless a constraint is imposed. These types of candidates, however, often have poor trade-offs with conflicting objectives. Weighted fitness F as a measure for implementing machine learning is therefore introduced. If all weights w_m are chosen to be $1/M$, F becomes the average fitness.

The weighted fitness of individuals in each cluster are then evaluated and sorted as illustrated in the subplot (c) in Fig. 1. A few of the better clusters are then selected and labeled with the "elite" status. Some arbitrary number of new candidates (for example, 20% of the total population) are repopulated uniformly and randomly within the narrow "elite range" of these elite clusters within the search space. Since these newly populated candidates share some similarities in

Figure 1: Schematic illustration of intervention using machine learning techniques. Here, a search space of two free variables is assumed. The distribution of the original population is shown in the subplot (a). The candidates are classified into three colored clusters with the K-means algorithm in the subplot (b). The average fitness of each cluster has been evaluated, sorted, and given a status labeled with "Best (elite)", "Good", and "Poor" respectively in the subplot (c). In the subplot (d), some potential competitive candidates (marked as the magenta dots) are repopulated inside the range of the "Best (elite)" cluster and then are used to replace the same amount of candidates from the original data pool. After the replacement, the post-population densities of the "Good" and the "Poor" cluster become low. In reality, there may not exist obvious boundaries to separate each cluster and cluster classification is not unique either.

the search space with the elite candidates thus far, they are expected to be more competitive in regard to survivability. From the original population, the same amount of candidates are randomly selected, to be replaced by the newly populated candidates. The average fitness within each generation should therefore increase respectively. This could potentially improve the probability of producing more competitive descendants favored by the optimization goals. While the next generation undergoes the same intervention, the elite range for the following repopulation of descendants will be dynamically re-defined by its own elite clusters. Note that the average fitness is used to define the elite range for repopulation. These repopulated candidates are not guaranteed the "privilege" of being "winners" in each generation. The final candidates still need to be selected through the non-dominated sorting. Considering that general fitness could have different scales in each dimension, they may need to be normalized within a similar range, usually $\in [0, 1]$, prior to averaging them [34].

Thus far the proportion of the replacement at each generation is set to a constant value. This is referred to as the Static Replacement Method (SRM). For the SRM, the proportion of replacement is arbitrary, but it is necessary to maintain diversity among the candidates to avoid traps at local minima. When the search space is too large, the distances D between candidates within the same cluster are far. In this case, it is likely that intervention would mislead evolution because the

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-ICAP2018-SAPAF01

expectation on the "elite range" may not be accurate. An optional dynamic replacement method can be used to mitigate this issue. To judge how likely an "elite range" can produce competitive candidates, a supervised machine learning technique is adopted. First, the candidates of an elite cluster are divided into a training set (usually around 90-95% of its population) and a testing set (the residual 5-10%). With the training set data, a learning model (hypothesis) H using the K-nearest neighboring (KNN) regression algorithm [33] is created. The model is used to predict the testing set's fitness (prediction). A comparison of the prediction and each individual's evaluated fitness value can determine the accuracy of the prediction. The comparison is quantitatively measured by a parameter "discrepancy" S in the fitness space,

$$S = \frac{1}{M}\sqrt{\sum_{m=1}^{M}\frac{|f_m - h_m|^2}{f_m^2}}. \quad (3)$$

Here, h_m is the m^{th} fitness value predicted from the learning model H and f_m is the actual fitness value. In this case, f is evaluated from a lattice characterization code. $S = 0$ means they are exactly same. A large S indicates a large discrepancy between the hypothesis model and the actual value. Based on the average discrepancy of the testing set, the replacement proportion for the population can be dynamically adjusted on a generation basis.

MOGA APPLICATION AT NSLS-II

The NSLS-II storage ring lattice [35] is used as an example to demonstrate the application of this method. The goal is to optimize the dynamic aperture of the operational lattice. The linear chromaticity is corrected to +2 by chromatic sextupoles. The free "tuning knobs" are six families of harmonic sextupoles with fixed polarities.

The spreads of the linear actions $J_{x,y}$ computed from turn-by-turn particle tracking simulation are chosen as the optimization objectives. The linear action J_u is defined as

$$J_{u,i} = \beta_u u_i^2 + 2\alpha_u u_i p_{u,i} + \gamma_u p_{u,i}^2 = \bar{u}_i^2 + \bar{p}_{u,i}^2, \quad (4)$$

where $u_i = (x, y)_i$ and $p_{u,i} = p_{(x,y),i}$ are the turn-by-turn coordinates in the horizontal and vertical planes respectively. $\bar{u} = \frac{1}{\sqrt{\beta_u}}u$, $\bar{p}_i = \frac{1}{\sqrt{\beta_u}}(\alpha_u u + \beta_u p_u)$ are a pair of normalized canonically conjugated coordinates, and α and β are the linear lattice optics Twiss parameters at the observation point. In the presence of nonlinear magnets, the linear actions have some spread from constants, as illustrated in Fig. 2. Typically the spread gradually increases with betatron oscillation amplitude. In order to obtain a sufficient dynamic aperture, control of the nonlinearity of motion for particles starting from different initial conditions (amplitudes) is needed. Here, five sets of initial conditions are chosen as shown in Fig. 3. The objectives are ten spreads of actions under different sextupole settings (each initial condition has both $\Delta J_{x,rms}/J_x$ and $\Delta J_{y,rms}/J_y$). For each candidate, the constraint is that all five particles can survive for multiple turns. All objectives outlined thus far are required to be equally important to ensure that there are no "holes" (particle loss) inside the dynamic aperture.

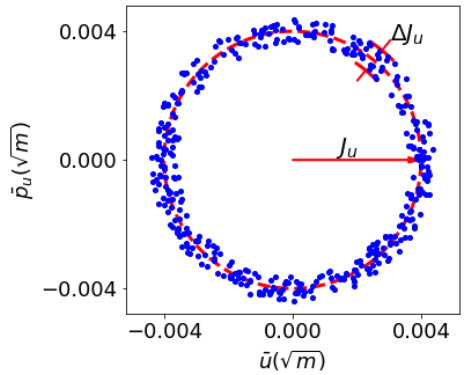

Figure 2: The root means squared (rms) spread of action from a constant is used as an optimization objective. The dashed circle represents a constant linear action at different angles. The dots are the normalized turn-by-turn coordinates.

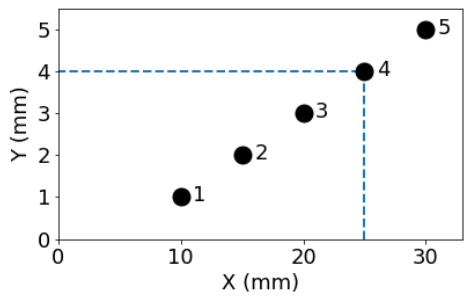

Figure 3: Five initial particle coordinates in the x-y plane with their conjugate momenta $p_{x,y} = 0$ used for tracking. The turn-by-turn data are used to evaluate the spread of their linear actions. The dashed line is the size of the desired dynamic aperture. The 5^{th} particle is chosen beyond the desired dynamic aperture in order to obtain a safe margin. The choice of the initial coordinates is not arbitrary. It may depend on the local optics functions, and physical aperture, etc.

To begin with, a random distribution is chosen in which the entire population is uniformly distributed within ranges limited by field saturation of the magnets and power supply capacity. In the NSLS-II ring, the search space at each sextupole dimension is $K_2 \in [0, \pm 40]\,\mathrm{m}^{-3}$ (Here $\pm$ is chosen depending on its polarity). Initially a population total of 5,000 is cast. For the first several generations, many candidates cannot survive under 5 initial conditions for dozens of turns. Therefore, the initial population evolves under the initial constraint of self-survival. After the evolution of 6-7 generations, all candidates can survive, but with very poor average fitness (see Fig. 4). Thus far sufficient data may have

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-ICAP2018-SAPAF01

already been accumulated to allow the optimizer to learn from the history.

The K-means algorithm is then applied (using an unsupervised learning technique) to classify the total population into $N = 100$ clusters in the search space and each individual cluster's average fitness is evaluated and sorted. The top three elite clusters are selected, based on their average fitness, to define an "elite" search range. Within this range, 20% of the total population is uniformly repopulated by random candidates. After this intervention (repopulation), the optimizer enters the next iteration.

With intervention, a fast convergence in the average fitness has been observed during evolution. Fig. 4 compares the evolution of the average fitness of MOGA with and without machine learning. Without machine learning, the improvement of fitness relies heavily on random crossover and mutation and global evolution can sometimes stop, or even regress. With the implementation of machine learning, however, the fitness convergence becomes not only faster, but much more steady. More importantly, the amount of competitive candidates is significantly increased, which allows for analysis of the distribution of optimal candidates in the search space.

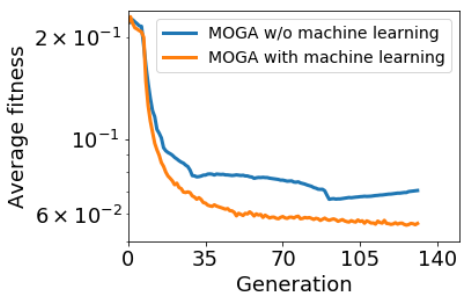

Figure 4: Comparison of the evolution of average fitness with and without machine learning for 135 generations. Without machine learning, the evolution process can sometimes stop, or even regress. On the other hand, the fitness convergence becomes faster and steadier with the introduction of machine learning.

With each generation, all candidates are re-classified. The elite ranges for repopulation also vary as shown in Fig. 5. The ranges are observed to fluctuate, but gradually converge during the evolution. For some free variables, the ranges converged quickly to a small range of optimal values. For example, the *SL1* sextupole's elite range shifts toward zero (limited by its polarity). This sextupole can therefore either be removed from the lattice, or have its polarity changed to see if machine performance can be further improved.

In the final generation's population, most of the candidates are found on the Pareto front. Among them, many have good average fitness. They are reclassified in the search space to study their distribution. These candidates appear to belong to many distinct groups. Each group is like an isolated island

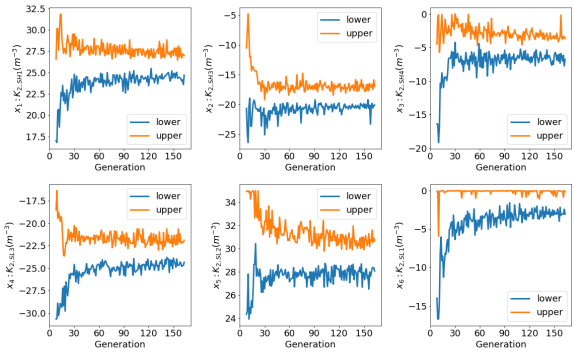

Figure 5: Variation of the ranges for generating new elite population in six-dimensional search space (sextupole's K_2) along the evolution. The elite ranges fluctuate, but gradually converge toward much narrower ranges.

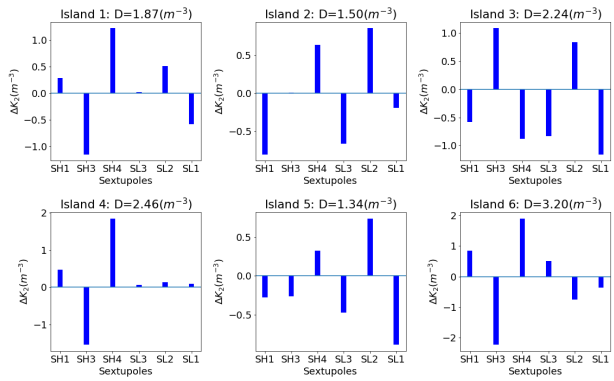

Figure 6: Relative distances of six neighbors from one elite candidate, which is used as the reference here in the search space. The reference sextupole settings are given as $K_{2,SH1} = 26.20891\,\mathrm{m}^{-3}$, $K_{2,SH3} = -17.87664\,\mathrm{m}^{-3}$, $K_{2,SH4} = -6.39466\,\mathrm{m}^{-3}$, $K_{2,SL3} = -22.42607\,\mathrm{m}^{-3}$, $K_{2,SL2} = 28.54735\,\mathrm{m}^{-3}$, $K_{2,SL1} = -0.22496\,\mathrm{m}^{-3}$.

in the search space. The island volumes, defined as

$$V = \prod_{n=1}^{N} (x_n^u - x_n^l), \tag{5}$$

are quite different. Here, N is the number of dimensions of the search space, and x_n^u and x_n^l are their upper and lower boundaries in the n^{th} dimension. In general, optimal candidates in large islands are more robust and therefore less impacted by errors than candidates in small islands because average fitness in large islands is less sensitive to the variation in search parameters. In Fig. 6, one island's coordinates are chosen as the origin to illustrate the relative distance to the six neighboring islands. All candidates in these islands yield decent dynamic apertures, but the sextupole settings are quite different.

The following paragraphs describe the detailed tracking results with the simulation code "ELEGANT" [36] and the experimental observations. From many optimal candidates obtained thus far, one solution is chosen and used as the

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-ICAP2018-SAPAF01

origin in Fig. 6, to carry out machine studies. Fitness is determined in regard to the spreads of linear actions through numerical simulations. Tracking 5 particles with different initial conditions is carried out. Their normalized conjugate position-momentum coordinates are shown in Fig. 7. Here, 5 initial conditions are used that differ from those used in the optimization setting (see Fig. 3). The maximum rms spread with the initial condition $x = 20$ mm and $y = 3$ mm (the outer ring in the plot) is around 3%, indicating that the motion is quite regular.

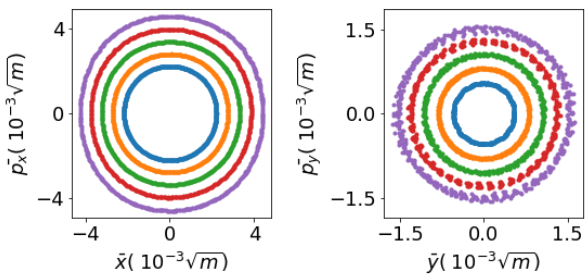

Figure 7: Simulated multi-turn trajectories in the phase space (left: the horizontal plane, right: the vertical plane) for five different initial conditions. The maximum spread for the initial condition $x = 20$ mm, $p_x = 0$ mrad and $y = 3$ mm, $p_y = 0$ mrad is around 3%.

A frequency map analysis has been carried out for both the on-momentum dynamic aperture (Fig. 8 and Fig. 9), and the off-momentum acceptance (Fig. 10). The dynamic aperture has small nonlinear diffusion [37]: up to 35 mm in the horizontal plane, and 13 mm in the vertical plane as shown in Fig. 8. In the meantime, this particular candidate has large tune-shift-with-amplitude coefficients, which can trap many resonance lines into a very thin stop-band width [38,39] (see Fig. 9). The robustness of this candidate has been confirmed by including the realistic NSLS-II magnet errors.

In this example, after the on-momentum dynamic aperture is optimized, the energy acceptance appears to be sufficient in the view of beam lifetime (see Fig. 10). The same observation holds for other optimal candidates. For the NSLS-II storage ring, it would appear that the two objectives, dynamic aperture and energy acceptance, may not conflict with each other. Should the dynamic aperture and energy acceptance conflict as optimization objectives in other synchrotrons, it is possible to include some off-momentum particle's actions as the optimization objectives.

After testing several evolved candidates on the NSLS-II storage ring, located on different islands within the search space, all yield sufficient dynamic aperture and energy acceptance, and therefore sufficient beam lifetime, for nominal operating conditions. A brief discussion of one particular experimental study period follows. During this time, beam was brought to third order tune resonance $3\nu_x = 100$ with the same lattice used for the tracking simulation with interesting results.

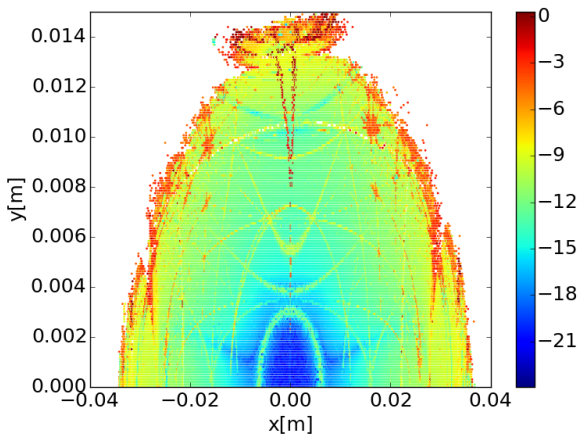

Figure 8: Dynamic aperture for on-momentum particles. The color represents the tune diffusion obtained by turn-by-turn tracking simulation. Diffusion [37] is defined as the difference of tunes $\Delta\nu$ extracted from the different time durations Diff=$\log_{10}\sqrt{\Delta\nu_x^2 + \Delta\nu_y^2}$. A cool color means the motion is less chaotic and vice versa.

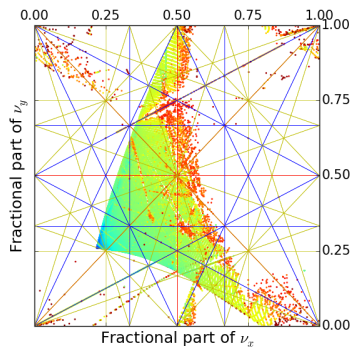

Figure 9: Frequency map corresponding to the on-momentum dynamic aperture in the $x - y$ planes. A large tune-shift-with-amplitude is observed in this lattice. The third order resonance line can be crossed stably (without obvious diffusion).

The simulated frequency map of the on-momentum dynamic aperture in Fig. 9 indicates that the third-order resonance $3\nu_x = 100$ was safely covered within the dynamic aperture, with no obvious diffusion (nonlinearity) observed in the tune space. The turn-by-turn particle tracking simulation further shows that the third order resonance has a very narrow stop bandwidth, which can "trap" particles once their trajectories are located inside the islands in the phase space (Fig. 11). During the study period, the machine's horizontal tune ν_x was set to 33.332. A short bunch train of 25 buckets was displaced to a particular amplitude using a pulse magnet (pinger). The amplitude of displacement chosen was ≈ 0.4 mm, measured at the center of the straight section where $\beta_x = 21$ m. This particular amplitude allowed the beam horizontal fractional tune to approach as close as

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-ICAP2018-SAPAF01

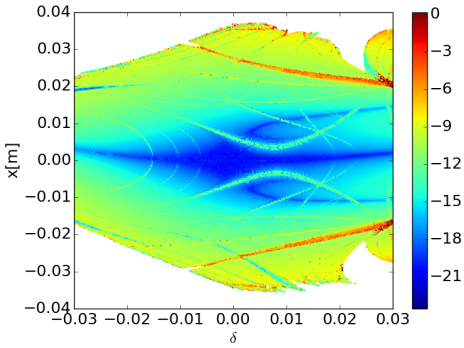

Figure 10: Frequency map of energy (momentum) acceptance in the $x-\delta$ planes. Here $\delta = \frac{\Delta p}{p}$ is the particles' relative momentum deviation.

possible to $1/3 \approx 0.3333\cdots$ (the right subplot in Fig. 12). The beam turn-by-turn trajectories were then observed to be trapped in three isolated islands in the phase space (the left subplot in Fig. 12). The beam began to circulate around the ring on a closed, stable orbit with the periodicity $1/3$ (see Fig. 13). A similar study was demonstrated and reported on in [40, 41].

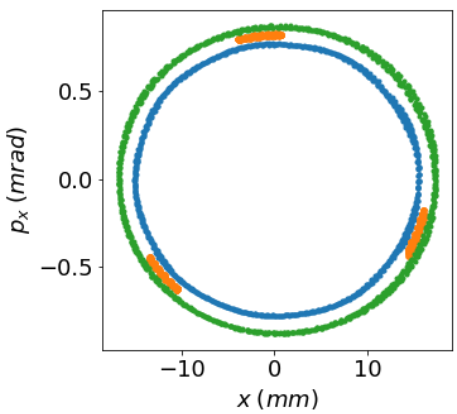

Figure 11: Simulated turn-by-turn trajectories in phase space. A large tune-shift-with-amplitude coefficient shifts the tune away from the third order resonance quickly when the betatron amplitude is slightly off. The stop-band width for this lattice is narrow, which means particle motion is stable even if its tune sits on the resonance.

The closed orbit with a $1/3$ periodicity repeats itself every 3 turns as illustrated in Fig. 13. It has some potentially interesting applications in dynamics and time-of-flight experiments [42]. For example, using a bunch-by-bunch excitation technique [43, 44], selected bunches can be displaced in this closed orbit while keeping the rest of the bunches in the original central orbit. Thus each synchrotron radiation port can deliver up to four distinct x-ray beams. The x-ray beams can have different horizontal positions and angles, and particularly different, distinct time structures. This technique

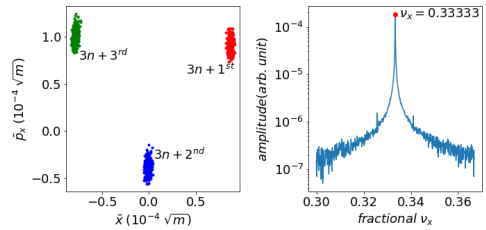

Figure 12: Left: measured beam turn-by-turn trajectories in the phase space with two neighboring beam position monitors (BPM). Three isolated islands are observed with a phase advance of $\frac{2\pi}{3}$ in-between as expected. Right: the FFT spectrum confirms that the beam remains stable on the third order resonance.

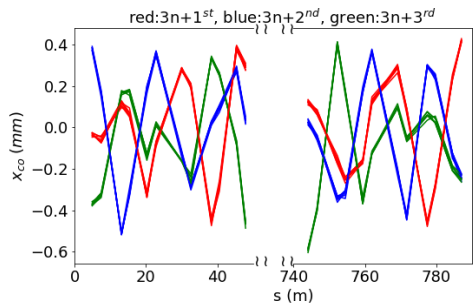

Figure 13: Measured closed orbit at $\nu_x = 1/3$ with BPM turn-by-turn data. The periodicity of the closed orbit became $1/3$ rather than 1. In other words, the closed orbit repeated itself every 3 turns.

and its implications, however, are beyond the scope of this paper and require further development and study.

SUMMARY

The evolution process of the genetic algorithm is significantly sped up when enhanced by machine learning and applied to the NSLS-II storage ring's dynamic aperture. Intervention via machine learning not only speeds up evolution, but increases the number of elite candidates in the data pool. The quality of some optimal candidates obtained with this technique have been confirmed experimentally on the NSLS-II ring and by simulation. This technique can be applied to other population-based optimization problems such as particle swarm algorithms. Extending it to an online mode would be a next logical step and would be driven by a real storage ring's TbT data [45, 46].

ACKNOWLEDGEMENTS

We would like to thank our NSLS-II colleagues for supporting this study. One of authors (YL) would like to thank Dr. Michael Borland, Dr. Yipeng Sun, Dr. Xiaobiao Huang, and Prof. Alexander Wu Chao for the fruitful discussion. This work was supported by Department of Energy Contract No. DE-SC0012704.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-ICAP2018-SAPAF01

REFERENCES

[1] L. Yang *et al.*, *Phys. Rev. ST Accel. Beams* 14, 054001 (2011)

[2] L. Yang *et al.*, *NIM-A* 609, 50-57 (2009)

[3] M. Borland *et al.*, *Conf. Proc.* C110328, 2354-2356 (2011)

[4] Y. Li *et al.*, *Int. J. Mod. Phys.* A31, 1644019 (2016)

[5] C. Sun *et al.*, *Phys. Rev. ST Accel. Beams* 15, 054001 (2012)

[6] L. Wang *et al.*, *Conf. Proc.* C1205201, 1380-1382 (2012)

[7] M. Ehrlichman, *Phys. Rev. Accel. Beams* 19, 044001 (2016)

[8] K. Tian *et al.*, *Phys. Rev. ST Accel. Beams* 17, 020703 (2014)

[9] W. Gao *et al.*, *Phys. Rev. ST Accel. Beams 14*, 094001 (2011)

[10] C. Gulliford *et al.*, *Phys. Rev. Accel. Beams* 19, 093402 (2016)

[11] C. Gulliford *et al.*, *Phys. Rev. Accel. Beams* 20, 033401 (2017)

[12] A. He *et al.*, *Phys. Rev. ST Accel. Beams* 18, 014201 (2015)

[13] I. Bazarov *et al.*, *Phys. Rev. ST Accel. Beams* 8, 034202 (2005)

[14] R. Bartolini *et al.*, *Phys. Rev. ST Accel. Beams* 15, 030701 (2012)

[15] A. Hofler *et al.*, *Phys. Rev. ST Accel. Beams* 16, 010101 (2013)

[16] B Terzic *et al.*, *Phys. Rev. ST Accel. Beams* 17, 101003 (2014)

[17] X. Huang *et al.*, *NIM-A* 757, 48-53 (2014)

[18] Y. Jiao *et al.*, *Chin. Phys.* C41, 027001 (2017)

[19] J. Li, *Private communication* (2017)

[20] A. Dragt, `https://www.physics.umd.edu/dsat/dsatliemethods.html` (2011)

[21] E. Forest, *Beam Dynamics: A New Attitude and Framework,* Hardwood Academic/CRC Press (1998)

[22] M. Berz, *Modern map methods in particle beam physics,* Academic Press (1999)

[23] C. Wang, *ANL/APS/LS* 330 (2012)

[24] L. Yu, *Phys. Rev. Accel. Beams* 20, 034001 (2017)

[25] Y. Li*et al.*, *arXiv*:1706.02195 (2017)

[26] Y. Li*et al.*, *NAPAC*'2016 TUPOB54 (2017)

[27] Y. Li*et al.*, *IPAC*'2017 WEPIK122 (2017)

[28] M. Borland and Y. Sun, *Private communication* (2017)

[29] Y. Sun*et al.*, *NAPAC*'2016 WEPOB15 (2017)

[30] K. Deb*et al.*, *Trans. Evol. Comp* 6, 182–197 (2002)

[31] H. Yoshida, *Phys. Lett.* A150, 262–268 (1990)

[32] D. Goldberg *et al.*, *Machine learning* 3, 95-99 (1988)

[33] F. Pedregosa *et al.*, *J. Machine Learning Res.* 12, 2825–2830 (2011)

[34] A. Yaser *et al.*, *Learning from data,* AMLBook (2012)

[35] BNL, `https://www.bnl.gov/nsls2/project/PDR/` (2013)

[36] M. Borland, *ANL/APS/LS* 287 (2000)

[37] D. Robin *et al.*, *Phys. Rev. Lett.* 85, 558–561 (2000)

[38] A. Chao, *Private communication* (2017)

[39] A. Chao *et al.*, *NIM* 121, 129-138 (1974)

[40] P. Goslawski *et al.*, *IPAC*'2017, WEPIK057 (2017)

[41] A. Huschauer *et al.*, *Phys. Rev. Accel. Beams* 20, 061001, (2017)

[42] C. Sun *et al.*, *Phys. Rev. Lett.* 109, 264801 (2012)

[43] W. Cheng *et al.*, J. *Phys. Conf. Ser.* 874, 012082 (2017)

[44] Y. Li *et al.*, *Phys. Rev. Accel. Beams* 20, 112802 (2017)

[45] X. Huang *et al.*, *Phys. Rev. ST Accel. Beams* 18, 084001 (2015)

[46] X. Huang, *Private communication* (2017)