

Genetic algorithm enhanced by machine learning in dynamic aperture optimization

Yongjun Li,^{*} Weixing Cheng, Li Hua Yu, and Robert Rainer
Brookhaven National Laboratory, Upton, New York 11973, USA

 (Received 16 January 2018; published 29 May 2018)

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 National Synchrotron Light Source II (NSLS-II) Storage Ring. During the evolution processes employed by the genetic algorithm, the population is classified into different clusters in the search space. The clusters with top average fitness are given “elite” status. Intervention on the population is implemented by repopulating some potentially competitive candidates based on the experience learned from the accumulated data. These candidates replace randomly selected candidates among the original data pool. The average fitness of the population is therefore improved while diversity is not lost. Maintaining diversity ensures that the optimization is global rather than local. The quality of the population increases and produces more competitive descendants accelerating the evolution process significantly. When identifying the distribution of optimal candidates, they appear to be located in isolated islands within the search space. Some of these optimal candidates have been experimentally confirmed at the NSLS-II storage ring. The machine learning techniques that exploit the genetic algorithm can also be used in other population-based optimization problems such as particle swarm algorithm.

DOI: 10.1103/PhysRevAccelBeams.21.054601

I. INTRODUCTION

Population-based optimization techniques, such as evolutionary (genetic) [1–16] and particle swarm [17–19] algorithms, have become popular in modern accelerator design. They are effective design tools for both linear and nonlinear system optimization. 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 non-linearity 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, multiobjective genetic algorithm (MOGA) [30] is a suitable optimization tool to compromise among these objectives simultaneously.

A general model for multiobjective optimization is (i) given a set of free variables x_n within the range $x_n \in [x_n^L, x_n^U]$, $n \in [1, N]$; (ii) subject to some constraints $c_j(x_n) \geq 0$, $j \in [1, J]$, and $e_k(x_n) = 0$, $k \in [1, K]$; (iii) 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 that mimics biological evolution [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, as in nature, 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

^{*}yli@bnl.gov

Published by the American Physical Society under the terms of the *Creative Commons Attribution 4.0 International license*. Further distribution of this work must maintain attribution to the author(s) and the published article’s title, journal citation, and DOI.

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. Commonly, the algorithm terminates when either a maximum number of generations has been reached, or a satisfactory fitness level has been achieved for the population.

The goal of multiobjective 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 nondominated or “Pareto optimal.” In these cases a nondominated 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. For a given application, in an unconstrained optimization, more desirable solutions (based on some unspecified metric or criteria) can be dominated by less desirable solutions, and understandably, the unconstrained optimization may fail to produce solutions that meet the unspecified metrics. In this instance, the unconstrained optimization is not an accurate model of the problem to be solved. If constraints are provided, however, 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 multiobjective, nondominated sorting with employment of the genetic algorithm forms the basis of the “MOGA” method.

Although MOGA has proven to be effective, it 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 non-linear 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]. The enormity of the dimensions of the search space also presents an issue. Modern accelerators have many magnets that are tuned independently. Each of them can be continuously tuned within a wide range, limited only by the capacity of their power supply or their field saturation. Compounding the issue, a storage ring’s lattice is also a highly nonlinear system, in which a small difference in the magnet configuration can result in vastly different behavior in regard to beam dynamics. Given the dimensionality provided by the genetic algorithm, and although tens of thousands of candidates make up the population of the search space of each generation, the population density is still low. Optimal candidates might also be located within a narrow space. The outcomes

of satisfactory candidates rely on some random processes, such as initial seeding, mutation, and the probability of crossover. In order to determine at least some of optimal candidates, either a large population is needed, or a long time frame for evolution, or both. MOGA are therefore usually limited by processing power and/or computation resources.

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 selecting candidates that are trapped in local minima. An effective intervention step would be able to significantly speed up the evolution in the desired direction. To do so, some machine learning techniques are introduced to traditional MOGA methods to intervene on the natural process. The intervention process is implemented automatically. First, it classifies the candidates in the search space, and second, it increases the ratio of potential elites among the population without loss of diversity. This method was demonstrated by optimizing the NSLS-II storage ring’s dynamic aperture. The convergence of average fitness of the population evolved much faster with intervention. Since sufficient elites exist, their distribution in the search space can also be studied.

To further explain the enhanced MOGA method and its application at NSLS-II, the remaining sections are outlined as follows: Section II describes the technical details of MOGA enhanced by intervention. Section III introduces the application of this method on the NSLS-II ring. Here, both the tracking simulation and the experimental results of an optimal solution obtained with this technique are also demonstrated. A brief summary is given in Sec. IV.

II. 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

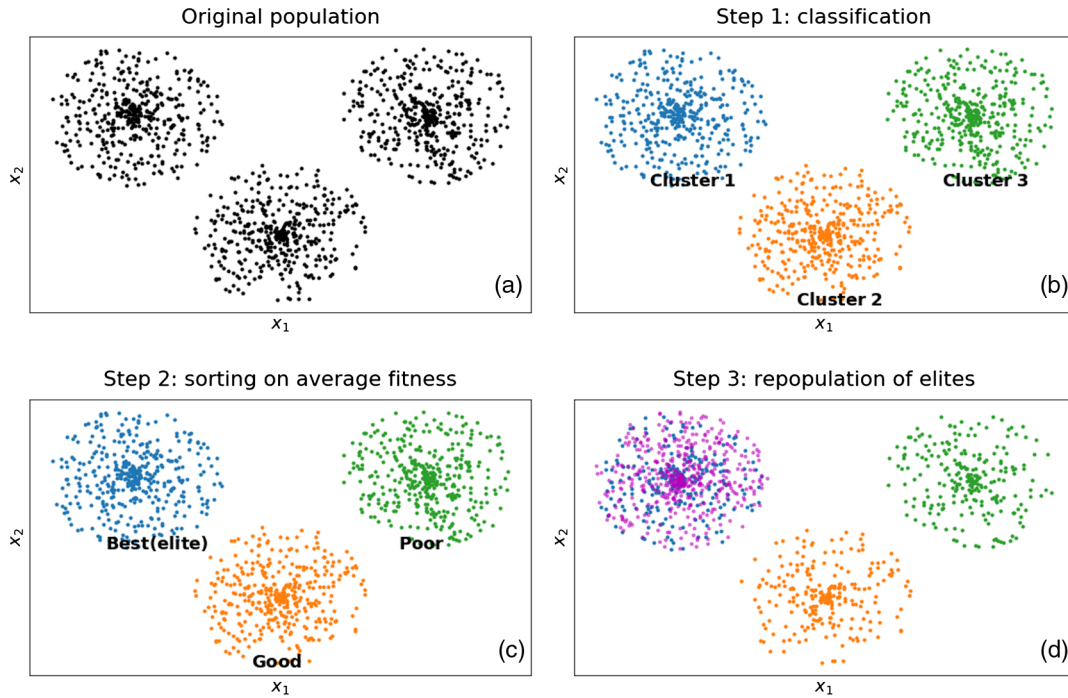


FIG. 1. Schematic illustration of intervention using machine learning techniques. Here, a search space of two free variables is assumed. The distribution of the original population is shown in the subplot (a). The candidates are classified into three colored clusters with the K-means algorithm in the 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.

are classified into different clusters in the search space based on a parameter D ,

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

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. The number of clusters should be determined based on the size of the population and the nonlinearity of the system itself. In general, a highly nonlinear system needs more clusters to narrow down the range of each cluster for better performance.

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

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

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 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. After the replacement, the number within the population remains the same, but the

percentage of elites among the total population is increased. 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. The newest generation, after intervention, produces descendants which start the next iteration. While the next generation undergoes the same intervention, the elite range for the following repopulation of descendants will be dynamically redefined 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 nondominated 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 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. If discrepancy is large enough for a particular generation,

intervention on the evolution process can be skipped entirely.

In the previous discussion, the size of each population was quite large. Therefore machine learning could be implemented within a generation. In the case of small population sizes, however, an alternative method still exists for machine learning. Rather than using the candidates in a single generation, accumulated candidates can be used from multiple generations to define the elite range for repopulation.

III. 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 three families of chromatic sextupoles in both the horizontal and vertical planes. 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

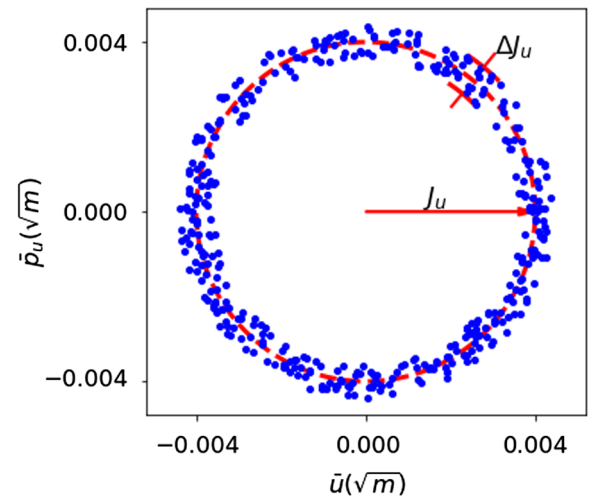


FIG. 2. The root means squared (rms) spread of action from a constant is used as an optimization objective. The dashed circle represents a constant linear action at different angles. The dots are the normalized turn-by-turn coordinates.

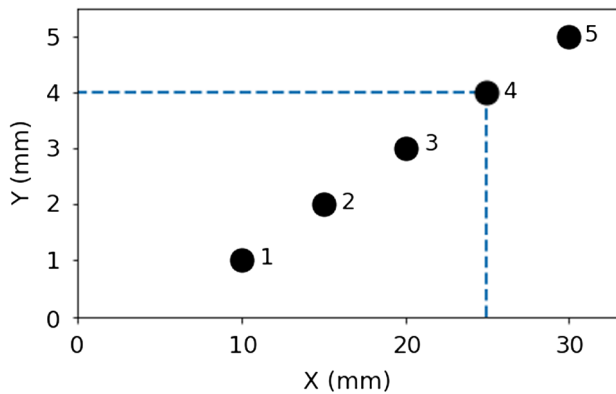


FIG. 3. Five initial particle coordinates in the $x - y$ plane with their conjugate momenta $p_{x,y} = 0$ used for tracking. The turn-by-turn data are used to evaluate the spread of their linear actions. The dashed line is the size of the desired dynamic aperture. The 5th particle is chosen beyond the 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.

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.

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 $K_2 \in [0, \pm 40] \text{ m}^{-3}$ (Here \pm is chosen depending on its polarity). Initially a population total of 5000 is cast, and maintains the population size during the evolution process. The search space density of a population of 5000 in a six-dimensional space is extremely low. 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. Each candidate’s resulting fitness is evaluated with a symplectic particle tracking simulation code and all data are archived in a database. 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 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

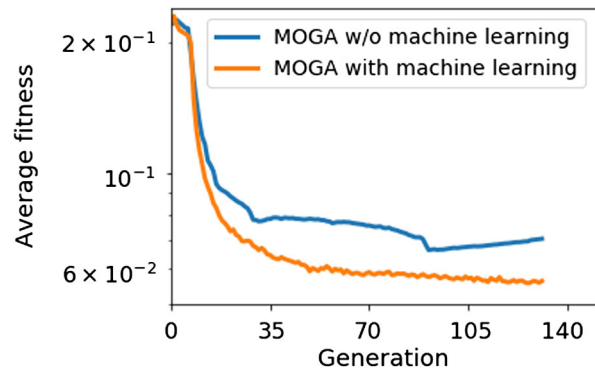


FIG. 4. Comparison of the evolution of average fitness with and without machine learning for 135 generations. Without machine learning, the evolution process can sometimes stop, or even regress. On the other hand, the fitness convergence becomes faster and steadier with the introduction of machine learning.

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. An optional dynamic replacement method can be employed here as well, as described previously.

With intervention, a fast convergence in the average fitness has been observed during evolution. Figure 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. Although each generation requires extra time to implement machine learning, the overall evolution is actually sped up. More importantly, the amount of competitive candidates is significantly increased, which allows for analysis of the distribution of optimal candidates in the search space.

With each generation, all candidates are reclassified. 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 *SLI* sextupole’s elite range shifts toward zero (limited by its polarity). The sextupole *SLI*’s strength K_2 in many optimal candidates is observed to be very weak. 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

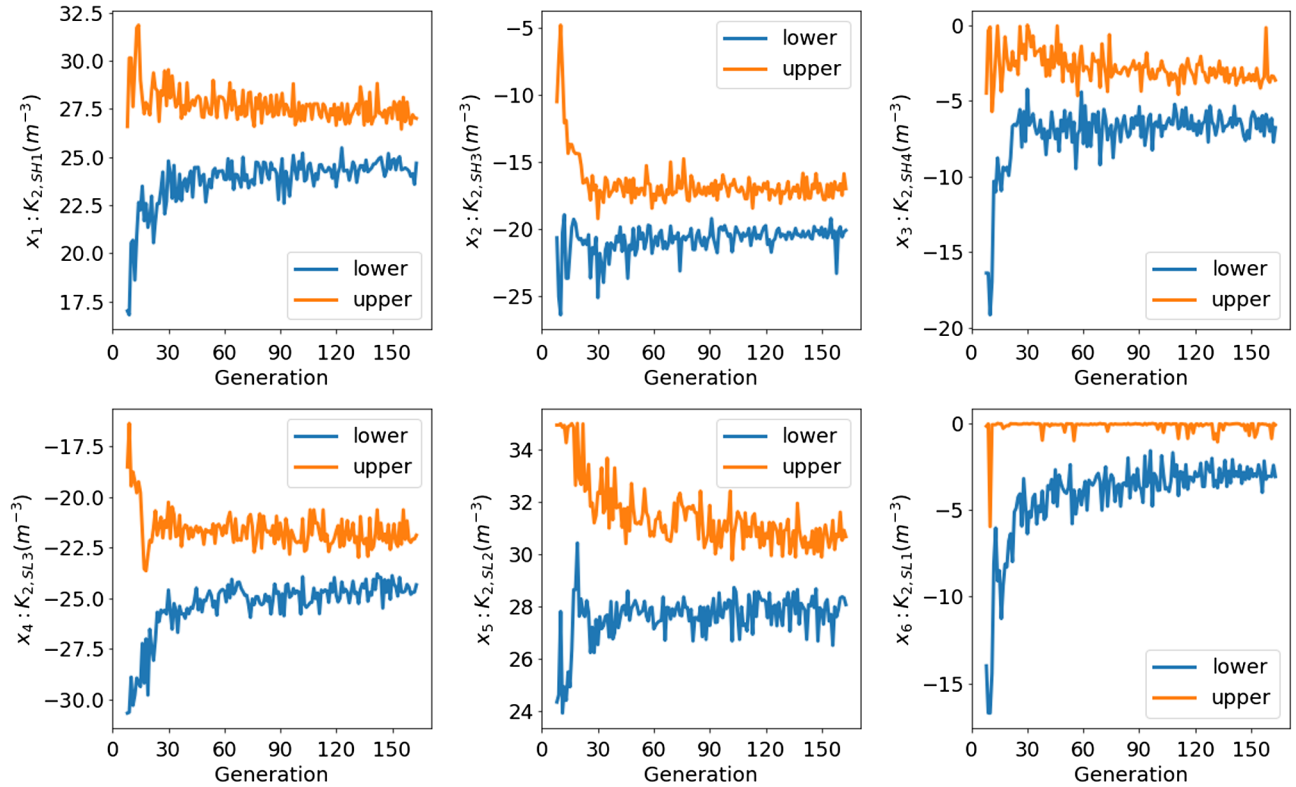


FIG. 5. Variation of the ranges for generating new elite population in six-dimensional search space (sextupole's K_2) along the evolution. The elite ranges fluctuate, but gradually converge toward much narrower ranges.

like an isolated island in the search space. The island volumes, defined as

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

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. It can sometimes be useful to know the distribution of the optimal candidates for a complicated nonlinear system. For example, in certain scenarios, undetectable random errors might degrade performance of a particular nonlinear lattice. Other optimal candidates belonging to different islands, however, might not be as sensitive to the same errors. This means more viable lattice options for machine operations are available. From the traditional MOGA population, very few good candidates may result, but the density of good candidates

among the population is too low to analyze their distribution, unless given an extremely long evolutionary period and/or a huge population.

The performance of the optimization has been tested on a parallel cluster equipped with more than 200 Intel® Xeon® 2.2–2.3 GHz CPU cores. With a population of 5000, and a fixed fraction of 20% repopulated candidates per generation. It takes about 10 hours to run 120 generations for the NSLS-II ring. A large fraction of replacement means a more aggressive expectation convergence, which may result in overshooting during optimization. The population size, fraction of replacement, and the final performance eventually depend on the complexity of the machine lattice.

The following paragraphs describe the detailed tracking results with the simulation code ELEGANT [36] and the experimental observations at the NSLS-II ring. From many optimal candidates obtained thus far, one solution is arbitrarily chosen and used as the 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 for 2048 turns. 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 purpose of using these initial conditions is to demonstrate that the optimized

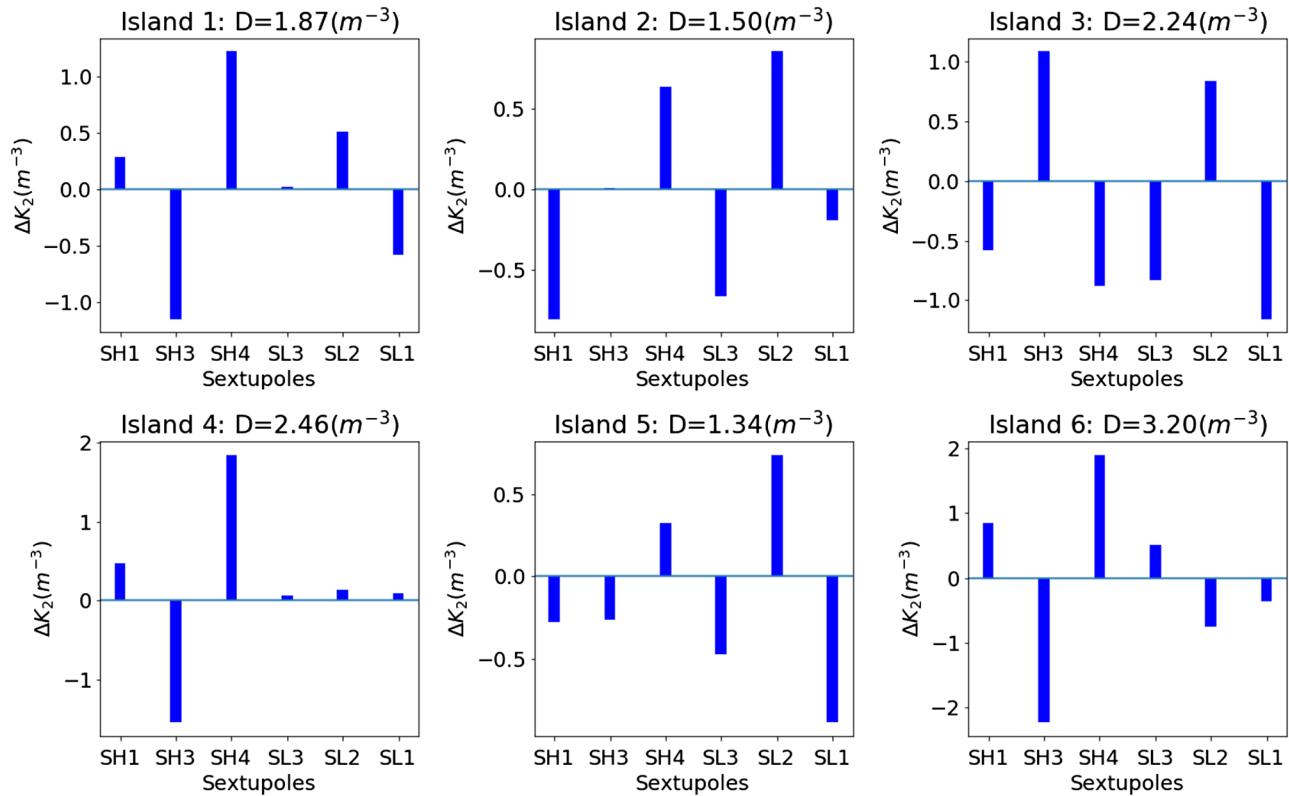


FIG. 6. Relative distances of six neighbors from one elite candidate, which is used as the reference here in the search space. The reference sextupole settings are given as $K_{2,SH1} = 26.20891 \text{ m}^{-3}$, $K_{2,SH3} = -17.87664 \text{ m}^{-3}$, $K_{2,SH4} = -6.39466 \text{ m}^{-3}$, $K_{2,SL3}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

sextupole settings can cover the dynamic aperture area of interest, which is independent from these particular initial conditions. In other words, the choice of initial conditions for the optimization is not unique. The simulations indicate that the rms spread of actions J gradually increase with the initial amplitudes, but still remain small in both the horizontal and vertical planes as expected. The maximum rms spread with the initial condition $x = 20 \text{ mm}$ and $y = 3 \text{ mm}$ (the outer ring in the plot) is around 3%, indicating that the motion is quite regular.

A detailed frequency map analysis has been carried out for both the on-momentum dynamic aperture (Figs. 8

and 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

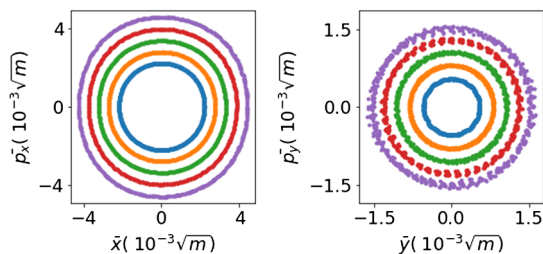


FIG. 7. Simulated multiturn trajectories in the phase space (left: the horizontal plane, right: the vertical plane) for five different initial conditions. The maximum spread for the initial condition $x = 20 \text{ mm}$, $p_x = 0 \text{ mrad}$ and $y = 3 \text{ mm}$, $p_y = 0 \text{ mrad}$ is around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

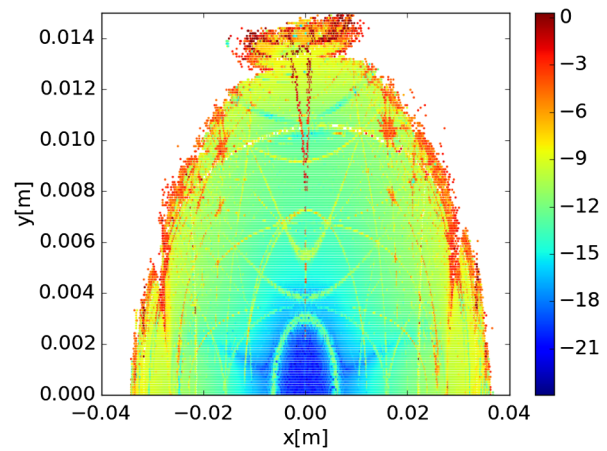


FIG. 8. Dynamic aperture for on-momentum particles. The color represents the tune diffusion obtained by turn-by-turn tracking simulation. Diffusion [37] is defined as the difference of tunes $\Delta\nu$ extracted from the different time durations $\text{Diff} = \log_{10} \sqrt{\Delta\nu_x^2 + \Delta\nu_y^2}$.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

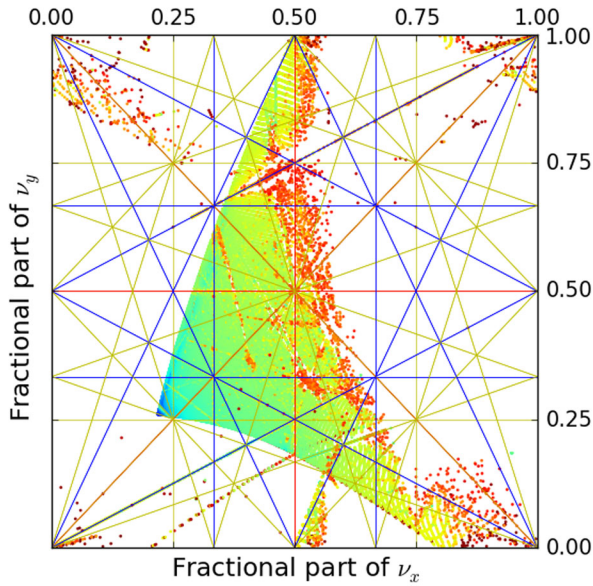


FIG. 9. Frequency map corresponding to the on-momentum dynamic aperture in the x - y planes. A large tune-shift-with-amplitude is observed in this lattice. The third order resonance line can be crossed stably (without obvious diffusion).

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. As the detailed nonlinear properties of this lattice are not relevant to the optimization technique discussed here, it will require more detailed discussion outside the scope of this paper.

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

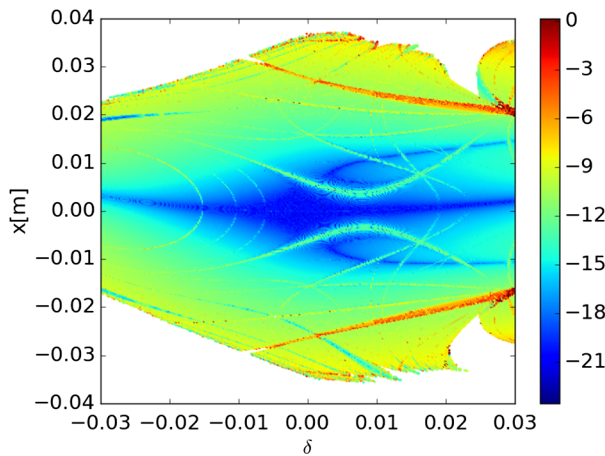


FIG. 10. Frequency map of energy (momentum) acceptance in the $x - \delta$ planes. Here $\delta = \frac{\Delta p}{p}$ is the particles' relative momentum deviation.

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.

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 possible to $1/3 \approx 0.3333 \dots$ (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

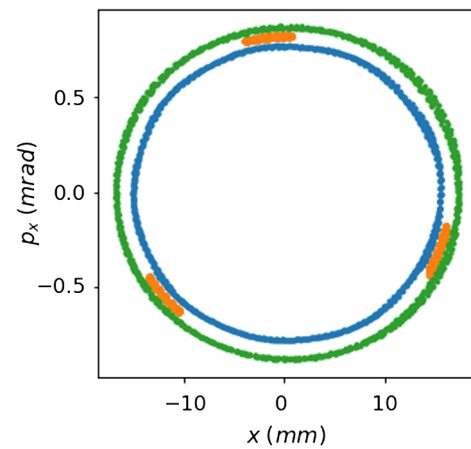


FIG. 11. Simulated turn-by-turn trajectories in phase space. A large tune-shift-with-amplitude coefficient shifts the tune away from the third order resonance quickly when the betatron amplitude is slightly off. The stop-band width for this lattice is narrow, which means particle motion is stable even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

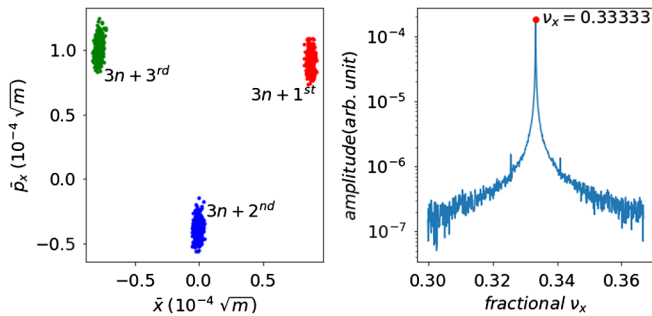


FIG. 12. Left: measured beam turn-by-turn trajectories in the phase space with two neighboring beam position monitors (BPM). Three isolated islands are observed with a phase advance of $\frac{2\pi}{3}$ in-between as expected. Right: the FFT spectrum confirms that the beam remains stable on the third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

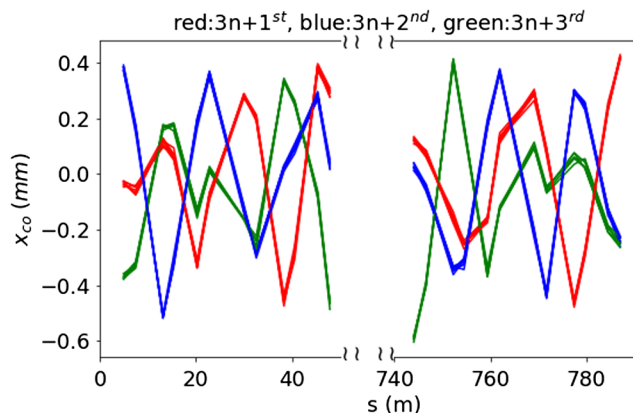


FIG. 13. Measured closed orbit at $\nu_x = 1/3$ with BPM turn-by-turn data. The periodicity of the closed orbit became $1/3$ rather than 1 . In other words, the closed orbit repeated itself every 3 turns.

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].

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

IV. 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. Greater density of elite candidates allows for study of the distribution of optimal candidates (resulting dynamic aperture) in the search space. Therefore, more options are provided for viable magnet settings. An optimal candidate's robustness can be roughly estimated by determining the dimension of the island to which it belongs in the search space. 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.

Thus far, the optimizer is driven by the simulated data to realize an offline optimization. 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].

ACKNOWLEDGMENTS

We would like to thank our NSLS-II colleagues for supporting this study. One of authors (Y. L.) 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.

- [1] L. Yang, Y. Li, W. Guo, and S. Krinsky, Multiobjective optimization of dynamic aperture, *Phys. Rev. ST Accel. Beams* **14**, 054001 (2011).
- [2] L. Yang, D. Robin, F. Sannibale, C. Steier, and W. Wan, Global optimization of an accelerator lattice using multi-objective genetic algorithms, *Nucl. Instrum. Methods Phys. Res., Sect. A* **609**, 50 (2009).
- [3] M. Borland, L. Emery, V. Sajaev, and A. Xiao, Multi-objective optimization of a lattice for potential upgrade of the advanced photon source, *Conf. Proc. C* **110328**, 2354 (2011).
- [4] Y. Li and L. Yang, Multi-objective dynamic aperture optimization for storage rings, *Int. J. Mod. Phys. A* **31**, 1644019 (2016).
- [5] C. Sun, D. S. Robin, H. Nishimura, C. Steier, and W. Wan, Small-emittance and low-beta lattice designs and optimizations, *Phys. Rev. ST Accel. Beams* **15**, 054001 (2012).
- [6] L. Wang, X. Huang, Y. Nosochkov, J. A. Safraneck, and M. Borland, Optimization of the dynamic aperture for SPEAR3 low-emittance upgrade, *Conf. Proc. C* **1205201**, 1380 (2012).
- [7] M. P. Ehrlichman, Genetic algorithm for chromaticity correction in diffraction limited storage rings, *Phys. Rev. Accel. Beams* **19**, 044001 (2016).
- [8] K. Tian, J. Safraneck, and Y. Yan, Machine based optimization using genetic algorithms in a storage ring, *Phys. Rev. ST Accel. Beams* **17**, 020703 (2014).

- [9] W. Gao, L. Wang, and W. Li, Simultaneous optimization of beam emittance and dynamic aperture for electron storage ring using genetic algorithm, *Phys. Rev. ST Accel. Beams* **14**, 094001 (2011).
- [10] Colwyn Gulliford, Adam Bartnik, and Ivan Bazarov, Multiobjective optimizations of a novel cryocooled dc gun based ultrafast electron diffraction beam line, *Phys. Rev. Accel. Beams* **19**, 093402 (2016).
- [11] C. Gulliford, A. Bartnik, I. Bazarov, and J. Maxson, Multiobjective optimization design of an rf gun based electron diffraction beam line, *Phys. Rev. Accel. Beams* **20**, 033401 (2017).
- [12] A. He, F. Willeke, L. H. Yu, L. Yang, T. Shaftan, G. Wang, Y. Li, Y. Hidaka, and J. Qiang, Design of low energy bunch compressors with space charge effects, *Phys. Rev. ST Accel. Beams* **18**, 014201 (2015).
- [13] I. V. Bazarov and C. K. Sinclair, Multivariate optimization of a high brightness dc gun photoinjector, *Phys. Rev. ST Accel. Beams* **8**, 034202 (2005).
- [14] R. Bartolini, M. Apollonio, and I. P. S. Martin, Multiobjective genetic algorithm optimization of the beam dynamics in linac drivers for free electron lasers, *Phys. Rev. ST Accel. Beams* **15**, 030701 (2012).
- [15] A. Hofler, B. Terzić, M. Kramer, A. Zvezdin, V. Morozov, Y. Roblin, F. Lin, and C. Jarvis, Innovative applications of genetic algorithms to problems in accelerator physics, *Phys. Rev. ST Accel. Beams* **16**, 010101 (2013).
- [16] B. Terzić, A. S. Hofler, C. J. Reeves, S. A. Khan, G. A. Krafft, J. Benesch, A. Freyberger, and D. Ranjan, Simultaneous optimization of the cavity heat load and trip rates in linacs using a genetic algorithm, *Phys. Rev. ST Accel. Beams* **17**, 101003 (2014).
- [17] X. Huang and J. Safranek, Nonlinear dynamics optimization with particle swarm and genetic algorithms for SPEAR3 emittance upgrade, *Nucl. Instrum. Methods Phys. Res., Sect. A* **757**, 48 (2014).
- [18] Y. Jiao and G. Xu, Optimizing the lattice design of a diffraction-limited storage ring with a rational combination of particle swarm and genetic algorithms, *Chin. Phys. C* **41**, 027001 (2017).
- [19] J. Li (private communication).
- [20] A. J. Dragt, Lie methods for nonlinear dynamics with applications to accelerator physics (2011), <https://www.physics.umd.edu/dsat/dsatliemethods.html>.
- [21] É. Forest, *Beam Dynamics: A New Attitude and Framework*, The Physics and Technology of Particle and Photon Beams (Hardwood Academic CRC Press, Amsterdam, 1998), Vol. 8.
- [22] M. Berz, Modern map methods in particle beam physics, *Adv. Imaging Electron Phys.* **108**, 1 (1999).
- [23] C. X. Wang, Explicit formulas for 2nd-order driving terms due to sextupoles and chromatic effects of quadrupoles, Report No. ANL/APS/LS-330, 2012 (unpublished).
- [24] L. H. Yu, Analysis of nonlinear dynamics by square matrix method, *Phys. Rev. Accel. Beams* **20**, 034001 (2017).
- [25] Y. Li, L. H. Yu, and L. Yang, Optimize nonlinear beam dynamical system with square matrix method, [arXiv:1706.02195](https://arxiv.org/abs/1706.02195).
- [26] Y. Li and L. H. Yu, Using square matrix to realize phase space manipulation and dynamic aperture optimization, in *Proceedings, 2nd North American Particle Accelerator Conference (NAPAC2016): Chicago, Illinois, USA, 2016* (2017), p. TUPOB54, <https://accelconf.web.cern.ch/AccelConf/napac2016/papers/tupob54.pdf>.
- [27] Y. Li and L. H. Yu, Applying square matrix to optimize storage ring nonlinear lattice, in *Proceedings, 8th International Particle Accelerator Conference (IPAC 2017): Copenhagen, Denmark, 2017* (2017) p. WEPIK122, <https://accelconf.web.cern.ch/AccelConf/ipac2017/papers/wepik122.pdf>.
- [28] M. Borland and Y. P. Sun (private communication).
- [29] Y. Sun and M. Borland, Comparison of nonlinear dynamics optimization methods for APS-U, in *Proceedings, 2nd North American Particle Accelerator Conference (NAPAC2016): Chicago, Illinois, USA, October 9-14, 2016* (2017) p. WEPOB15, <https://accelconf.web.cern.ch/AccelConf/napac2016/papers/wepob15.pdf>.
- [30] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *Trans. Evol. Comp.* **6**, 182 (2002).
- [31] H. Yoshida, Construction of higher order symplectic integrators, *Phys. Lett.* **150A**, 262 (1990).
- [32] D. E. Goldberg and J. H. Holland, Genetic algorithms and machine learning, *Mach. Learn.* **3**, 95 (1988).
- [33] F. Pedregosa *et al.*, Scikit-learn: Machine learning in Python, *J. Mach. Learn. Res.* **12**, 2825 (2011).
- [34] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from Data* (AMLBook, New York, 2012), Vol. 4.
- [35] BNL, <https://www.bnl.gov/nsls2/project/PDR/>.
- [36] M. Borland, elegant: A flexible sdds-compliant code for accelerator simulation, Report No. ANL/APS/LS-287, 2000.
- [37] D. Robin, C. Steier, J. Laskar, and L. Nadolski, Global Dynamics of the Advanced Light Source Revealed through Experimental Frequency Map Analysis, *Phys. Rev. Lett.* **85**, 558 (2000).
- [38] A. W. Chao (private communication).
- [39] A. W. Chao and M. Month, Particle trapping during passage through a high-order nonlinear resonance, *Nucl. Instrum. Methods* **121**, 129 (1974).
- [40] P. Goslawski, F. Kramer, M. Ruprecht, A. Jankowiak, M. Ries, and G. Wüstefeld, Status of transverse resonance island nuckets as bunch separation scheme, in *Proceedings, 8th International Particle Accelerator Conference (IPAC 2017): Copenhagen, Denmark, 2017* (2017) p. WEPIK057, <https://accelconf.web.cern.ch/AccelConf/ipac2017/papers/wepik057.pdf>.
- [41] A. Huschauer, A. Blas, J. Borburgh, S. Damjanovic, S. Gilardoni, M. Giovannozzi, M. Hourican, K. Kahle, G. Le Godec, O. Michels, G. Sterbini, and C. Hernalsteens, Transverse beam splitting made operational: Key features of the multiturn extraction at the cern proton synchrotron, *Phys. Rev. Accel. Beams* **20**, 061001 (2017).
- [42] C. Sun, G. Portmann, M. Hertlein, J. Kirz, and D. S. Robin, Pseudo-Single-Bunch with Adjustable Frequency: A New Operation Mode for Synchrotron Light Sources, *Phys. Rev. Lett.* **109**, 264801 (2012).
- [43] W. Cheng, Y. Li, and K. Ha, Techniques for transparent lattice measurement and correction, *J. Phys. Conf. Ser.* **874**, 012082 (2017).

- [44] Y. Li, W. Cheng, K. Ha, and R. Rainer, Transparent lattice characterization with gated turn-by-turn data of diagnostic bunch train, *Phys. Rev. Accel. Beams* **20**, 112802 (2017).
- [45] X. Huang and J. Safranek, Online optimization of storage ring nonlinear beam dynamics, *Phys. Rev. ST Accel. Beams* **18**, 084001 (2015).
- [46] X. Huang (private communication).