SLAC-PUB-15939

Nonlinear dynamics optimization with particle swarm and genetic algorithms for SPEAR3 emittance upgrade

Xiaobiao Huang, James Safranek

SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025

Abstract

Nonlinear dynamics optimization is carried out for a low emittance upgrade lattice of SPEAR3 in order to improve its dynamic aperture and Touschek lifetime. Two multi-objective optimization algorithms - a genetic algorithm and a particle swarm algorithm are used for this study. The performance of the two algorithms are compared. The result shows that the particle swarm algorithm converges significantly faster to similar or better solutions than the genetic algorithm and it does not require seeding of good solutions in the initial population. These advantages of the particle swarm algorithm may make it more suitable for many accelerator optimization applications.

Keywords: Nonlinear beam dynamics, optimization, SPEAR3

1. Introduction

SPEAR3 is a third generation light source that was rebuilt on the footprint of its predecessors in 2003-2004. Its present horizontal emittance is 10 nm (including radiation damping effect of insertion devices). In order to significantly improve the photon beam brightness, we have carefully studied the lattice potential of this machine [1], from which a lattice that would reduce the emittance to nearly 6 nm was developed and tested in experiment. However, the tested lattice is not yet suitable for user operation due to the reduced dynamic aperture. Although a proposed upgrade of the septum magnet that decreases the separation between the injected beam and the stored beam will substantially alleviate the problem of dynamic aperture deficit, it

Email address: xiahuang@slac.stanford.edu (Xiaobiao Huang)

Preprint submitted to Nuclear Inst. and Methods in Physics Research, A April 1, 2014

Work supported in part by US Department of Energy under contract DE-AC02-76SF00515.

is still necessary to improve the dynamic aperture through nonlinear lattice optimization.

Typically the goals of nonlinear dynamics optimization of a lattice are to improve injection efficiency and Touschek lifetime, which translate to large dynamic aperture and momentum acceptance. The optimizing variables, or the "knobs", are usually the sextupole magnets in the lattice. Because a lattice with a large dynamic aperture does not necessarily also have a large momentum acceptance, it is desirable to optimize both objectives with multi-objective optimization algorithms and choose the best solution from the solutions in the Pareto front afterward [2]. Both the dynamic aperture and the momentum acceptance of a storage ring lattice are difficult to optimize because they are strongly nonlinear functions of the optimizing variables and there are usually many local minima that make the terrain of objective functions in the optimizing variable space very corrugated and non-smooth. Advanced algorithms are needed to reach the global minimum for the nonlinear beam dynamics optimization problem.

Multi-objective genetic algorithms (MOGA) are a powerful method for the optimization of multi-variable, multi-objective problems with a complex terrain. They have been widely used in the accelerator community in many optimization problems [4, 5, 6, 7, 8], in particular, in the optimization of nonlinear beam dynamics of storage rings [6, 8]. A MOGA algorithm has been used in our SPEAR3 nonlinear dynamics optimization problem. Recently the multi-objective particle swarm optimization (MOPSO) method has been demonstrated to be a useful technique in the optimization of accelerator problems [9], where it was shown that MOPSO has much faster convergence than MOGA for the problems studied. Because nonlinear dynamics optimization is usually time consuming, high efficiency is very desirable for the optimization method. We have applied MOPSO to SPEAR3 nonlinear dynamics optimization and compared its performance to that of MOGA.

In Section 2 we will first briefly describe the SPEAR3 emittance upgrade lattice and the nonlinear dynamics optimization problem. Then the application of both MOGA and MOPSO to this problem is shown in Section 3. Conclusion is given in Section 4.

2. The SPEAR3 emittance upgrade lattice

The SPEAR3 lattice consists of 18 double-bend-achromat (DBA) cells, 14 of which are standard ones and the rest are matching cells. After a

thorough examination of the potential of the standard cell with the global search of stable solution technique [3], we concluded that the only approach of reducing emittance is to increase the horizontal tune and the dispersion leak [1]. Two upgrade lattice options have been extensively studied both in simulation and in experiments. Their key lattice parameters are listed in Table 1 in comparison to the present nominal lattice. Lattice option I increases the horizontal tune by one unit and keeps the dispersion leak at 0.1 m. The effective emittance, which includes the contribution of the finite dispersion function at the insertion devices and the beam momentum spread, is reduced from the present value of 10.3 nm to 7.2 nm. Lattice option II increases the horizontal tune to 15.32 and increases the dispersion leak to 0.12 m, which leads to a further reduction of emittance. The effective emittance of option II is 6.7 nm.

Table 1: Parameters for SPEAR3 lattices

Parameter	nominal	option I	option II
ν_x	14.13	15.13	15.32
ν_y	6.22	6.22	6.18
D_x (m)	0.1	0.1	0.12
μ_x/cell	0.779	0.829	0.838
ϵ (nm) w/ID	9.7	6.8	6.1
$\epsilon_{\rm eff}$ (nm) w/ID	10.3	7.2	6.7

Both lattice options were tested on the machine experimentally. There are two main difficulties that prevent us from delivering user beam in these lattices without some hardware upgrade. Both are related to beam injection. First, one of the three injection kickers is too weak to form a closed kicker bump of the required magnitude for the new lattices due to the change of the horizontal phase advance per standard cell (see Table 1 for μ_x/cell). The three kickers are placed in adjacent standard straight sections with the middle one (K2) sharing a straight section with the injection septum. Because the present horizontal phase advance between the first (K1) and the third (K3) kickers is nearly 3π, only a small kick is required from K2 to form a closed kicker bump. However, as the phase advance between the kickers is increased, a significantly larger kick is needed from K2 for the new lattices, which exceeds its present strength limit. To solve this problem, we plan to move

the septum magnet closer to the stored beam in order to reduce the size of the kicker bump from 22 mm to 13.5 mm. In addition, an upgrade of K2 pulser circuit is planned to increase the maximum kick by 50%.

Second, the dynamic apertures of the new lattices are not sufficient for high injection efficiency. The new lattices have smaller dynamic aperture because the tune shifts with oscillation amplitude have significantly increased due to the the change of horizontal phase advance between the sextupole magnets. For example, the calculated coefficient $\partial\nu_x/\partial\epsilon_x$ for the nominal lattice and the two new lattice options are 1800, 7700 and 11900, respectively, where $\epsilon_x = 2J_x$ and $J_x = [x^2 + (\alpha_x x + \beta_x x')^2]/\beta_x$ is the Courant-Snyder invariant for the horizontal motion. The large tune shifts with amplitude cause the beam to reach lower order resonances at smaller oscillation amplitude and therefore reduce the dynamic aperture. The measured dynamic aperture for the new lattices are 2 to 3 mm smaller than the nominal lattice. Consequently even after a considerable effort to experimentally optimize injection the highest injection efficiency for lattice options I and II are only 55% and 30%, respectively.

To accommodate the reduced dynamic aperture, an upgrade of the septum magnet is planned to reduce the septum wall thickness from approximately 6 mm to 2.5 mm. This would significantly reduce the requirement of dynamic aperture and may enable user operation with the lattice options listed in Table 1. However, in order to have some safety margin, it is still desirable to further optimize the nonlinear dynamics of these lattices.

The experimental, beam-based optimization of lattice option II included scans of horizontal and vertical phase advances per cell with the working points fixed. It was found that the dynamic aperture (and hence injection efficiency) depends on the horizontal phase advance sensitively (Figure 1). The lattice we chose corresponds to the highest injection efficiency. However, for this lattice the vertical beta function at the middle of matching straight sections is $\beta_y = 4.0$ m, instead of the more desirable standard value of 2.5 m. The lattice that gives $\beta_y = 2.5$ m at the matching straight sections (which has $\mu_x = 0.842$ for a standard cell) has poor dynamic aperture such that injection cannot be established. So another strong reason that requires us to optimize the nonlinear beam dynamics is to enable user operation of the lattice that satisfies the vertical beta function requirement at the matching straight sections. This is discussed in the next section.

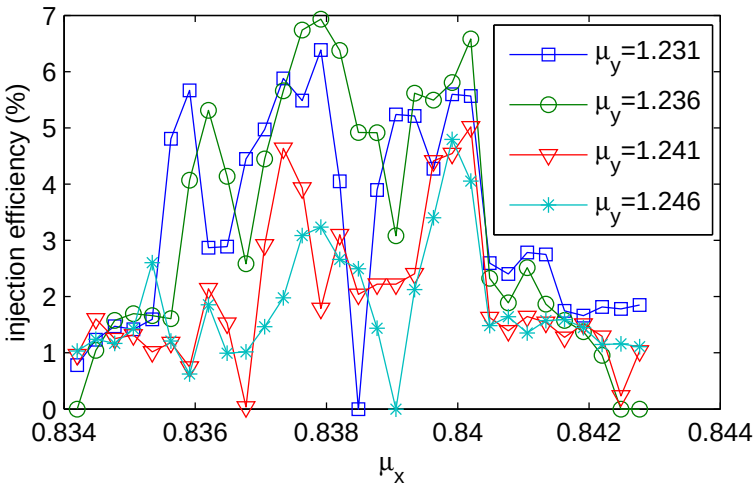

Figure 1: Injection efficiency vs. horizontal phase advance per cell. Not all injection conditions were optimized for the scan.

3. Nonlinear dynamics optimization

3.1. Optimization objectives and tracking setup

We only discuss the nonlinear dynamics optimization study for the lattice with the working point (15.32, 6.18) and dispersion leak of $D_x = 0.12$ m which satisfies the requirement of $\beta_y = 2.5$ m at the matching straight sections. This is the most desirable lattice for user operation and has not been successfully implemented due to its poor nonlinear dynamics performance.

SPEAR3 has two families of sextupole magnets (SF and SD) in each cell for chromaticity correction. All SF and SD magnets in standard and matching cells are powered in series separately. So there are four individual sextupole controls (i.e., *knobs*). We have experimentally optimized the sextupole knobs in matching cells while using the standard cell sextupoles to keep chromaticities fixed [1]. To achieve better nonlinear dynamics performance, we need more sextupole knobs that can effectively adjust the nonlinear dynamics behavior. We have studied the approach of adding sextupole magnets to the machine and found that it is not cost-effective. In the end we decided to add new power supplies to split the sextupole series into smaller groups for additional knobs. Among the several splitting options, the one we favored may be represented as "xyaazaayx", where each letter stands for a cell in one half of the ring and the other half is its mirror reflection. Sextupoles SF and SD

in the x, y and z cells are free variables and those in the a cells are powered in series for chromaticity correction. There are six free sextupole variables in total. A proper range is assigned to each of the free variables according to the power supply limit. In our implementation of the optimization codes each parameter is normalized so that its numerical range is from 0 to 1.0. In other words, the parameter space is a unit hyper-cube.

The nonlinear dynamics performance of a storage ring lattice may be characterized by the dynamic aperture and the momentum acceptance (also called momentum aperture). Therefore these parameters are used as the optimization objective functions. Both dynamic aperture and momentum aperture are evaluated with particle tracking simulation. In the lattice model a 3 mm vertical physical aperture is placed at the septum. For dynamic aperture evaluation particles distributed on 19 rays extending from the origin of the x-y plane on and above the mid-plane are launched and tracked for 5000 turns. The angles between the rays are distributed so that there are enough particles near the mid-plane. We found it was necessary to track a large number of turns because in both simulation and experiment (as shown in Figure 2) there can be considerable loss of the injected beam even after a few thousand turns.

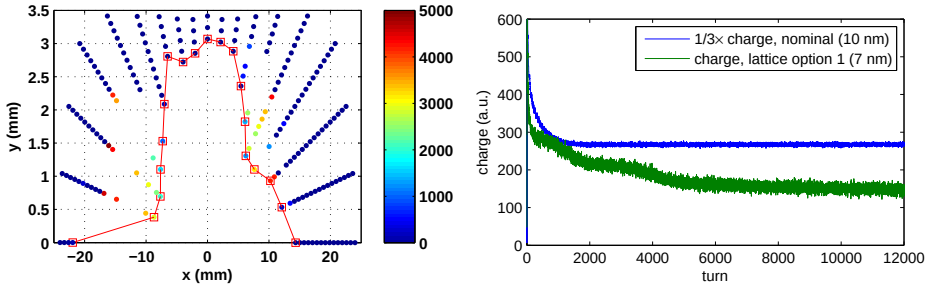

Figure 2: Dynamic aperture as indicated by particles lost in tracking (left) and the measured beam current vs. the number of turns after injection for lattice option I (right). Color code in the left plot indicates the number of turns when particles get lost. The beam current was measured with turn-by-turn beam position monitor (BPM) sum signal in the right plot. Part of the initial decrease of the sum signal is due to the BPM response to changing bunch length.

The dynamic aperture is measured with the area under the curve that connects the lost particles closest to the origin on each ray. For the nominal SPEAR3 lattice the horizontal phase space ellipse at the septum is asymmetric at large oscillation amplitude. The intersection on the negative side

($x < 0$) is bigger than the positive side. This facilitates injection because the injected beam comes from the negative side (inside the ring). We would like to encourage this asymmetry for the new lattices. Therefore the dynamic aperture objective is defined as the area under the negative side of the curve plus one half of the area of the positive side.

The momentum aperture as determined by transverse beam dynamics tends to be small at locations with large dispersion invariant, such as in the arc of a DBA cell. In fact the momentum aperture is usually limited by transverse dynamics only in the arcs (elsewhere it is limited by the RF bucket height). Therefore the momentum aperture objective is set as the average momentum aperture at the central quadrupoles of the 7 standard DBA cells in one half of the ring. At each of these locations particles with initial energy error from -0.035 to -0.01 and 0.01 to 0.035 with a step size of 0.001 are launched and tracked for 1000 turns. It is determined that tracking 1000 turns is sufficient for the momentum aperture after comparison is made between results of tracking with 1000 turns and 5000 turns. The momentum aperture is determined from the initial energy error of the surviving particles. Because the optimization algorithms are implemented for minimization, a negative sign is added to both objective functions to effectively maximize them.

Evaluation of the objective functions described above is time consuming. It is critical to use an efficient particle tracking code in the calculation. The optimization of nonlinear beam dynamics requires the evaluation of thousands of solutions. Since both the MOGA and MOPSO algorithms make program decisions in each step after a population of solutions are evaluated, it is natural to simultaneously evaluate many solutions on cluster computers. For these purposes, a standalone tracking code was developed and used in this study.

3.2. Setup for the MOGA algorithm

The widely used MOGA algorithm NSGA-II [11] is adopted in our study. This algorithm manipulates a population of solutions with a fixed size over a number of generations. At each generation the crossover and mutation operations are performed to generate new solutions (i.e. children). After the new solutions are evaluated, they are combined with the old population (i.e., parents) and a non-dominated sorting is conducted to select the best solutions to pass on to the next generation. Crowding distance is used in the selection of solutions with the same rank to promote diversity in the surviving

population. The selection operation essentially applies a pressure that drives the population to evolve toward the best solutions in the parameter space (which are called the Pareto front).

A population size of 100 is used. The crossover probability ρ_c of 0.8 and 0.95 have been tried. In the case below for comparison with MOPSO the crossover probability is 0.8 because this case is considerably better than the case with $\rho_c = 0.95$. This can be seen in Figure 6 left plot where the solutions of MOPSO totally dominate the MOGA with $\rho_c = 0.95$ case while they do that only partially to the MOGA with $\rho_c = 0.8$ case. The indices that control the random number distribution in crossover and mutation are set to $\mu_c = 20$ and $\mu_m = 40$, respectively. Similar performance was found for a case with the mutation parameter changed to $\mu_m = 20$. Two approaches have been used to generate the initial population. The first one is completely random. Each parameter of the initial solutions is drawn from a uniform distribution between 0 and 1. Because this approach does not perform well, we adopted a second approach in which the nominal solution and 49 other solutions around it are seeded to generate the initial population. The nominal solution corresponds to the condition when all SF/SD sextupoles in standard cells are set to equal values, respectively. Parameters for the other 49 solutions are drawn from a Gaussian distribution with a standard deviation of 0.15 that is centered on the nominal solution. The rest of the initial solutions are random from a uniform distribution.

3.3. Setup for the MOPSO algorithm

The MOPSO algorithm is similar to MOGA in that it also manipulates a population of solutions with random operations. Here each solution in the population is considered a moving particle in the parameter space. The position of each particle is updated at every iteration by adding an amount called its *velocity*, i.e.,[9]

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}, \tag{1}$$

where $\mathbf{x}_i^t$ and $\mathbf{v}_i^t$ are vectors that represent the position and the velocity of the i'th particle at iteration t, respectively. The velocity is calculated as the weighted sum of three terms,

$$\mathbf{v}_i^{t+1} = w\mathbf{v}_i^t + c_1 r_1(\mathbf{p}_i^t - \mathbf{x}_i^t) + +c_2 r_2(\mathbf{g}^t - \mathbf{x}_i^t), \tag{2}$$

where the terms on the right-hand side represent the previous velocity, the distance between the present position and the position of the best solution

for the history of this particle (i.e., *personal best*, $\mathbf{p}_i^t$) and the distance between the present position and a global best solution, $\mathbf{g}^t$, respectively. For multi-objective problems the global best solution is not unique and it is thus randomly selected from the archived global best solutions. Parameters w, $c_{1,2}$ and $r_{1,2}$ control the behavior of the algorithm. Mutation operation is also performed to a small fraction of randomly selected solutions. After the initial position and velocity distributions are given, the particle population then start to move in the parameter space.

We implemented the MOPSO algorithm described in Ref. [9] with small modifications. As in Ref. [9], we set the parameters $w = 0.4$, $c_1 = c_2 = 1.0$. Several ways of choosing parameters r_1 and r_2 are tried, including setting them to random numbers drawn from the uniform distribution in $(0, 1)$ or a Gaussian distribution centered at $\mu = 0.5$ with standard deviation $\sigma = 0.15$ each iteration, and setting them to constants $r_1 = r_2 = 0.5$ or $r_1 = r_2 = 1.0$. It was found that the case with $r_1 = r_2 = 1.0$ yields the best performance and this is the case presented below. The mutation probability is set to 16.7% (which is 1/the number of optimization variables). We update the global best archive by simply combining the previous archive and the new solutions, performing non-dominated sorting and keeping the best solutions up to the quota. The crowding distance criterion is used to select solutions from the middle front of the combined population that can only partially enter the global best archive. And, because the archive is already sorted with non-dominated sorting, no sorting with crowding distance is needed before updating the velocity and position data of the particles. Both approaches of generating initial population as described in the previous subsection are tried. The initial velocity distribution is drawn from a uniform distribution in $[0, 0.1]$. The population size is 100. The algorithm is run for 100 iterations.

3.4. Comparison of results

The performance of a multi-objective optimization algorithm may be measured by the objective functions of the final population and the speed of convergence. For each of the two algorithms we studied, several setup conditions have been used. For example, the initial population can be purely random or can be seeded with the nominal solution and a compact random distribution around it. It is found that the MOGA algorithm depends significantly on the initial distribution. If initial seeds are not provided, it converges prematurely to a set of solutions that are far worse in both objectives than the solutions

found by the run with seeds. The MOPSO algorithm, however, is not dependent on the initial distribution. Figure 3 shows the objective functions of the best solutions at the final generation (after 10100 lattice evaluations) for MOGA and MOPSO, with or without initial seeds.

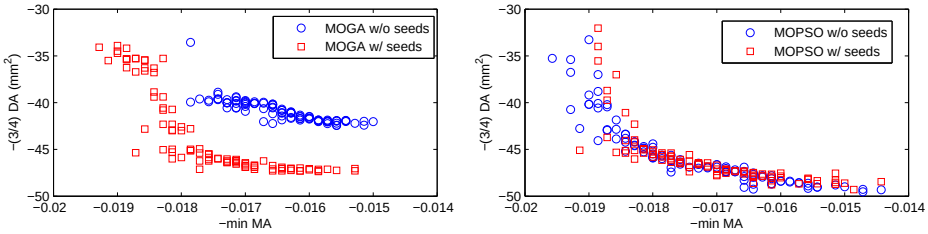

Figure 3: Comparison of objective functions after 10100 lattice evaluations for MOGA (left) and MOPSO (right) with or without initial seeds.

To better compare the history of the optimization runs, we define weighted sums of the two objective functions after normalization as follows

$$F_w = \sum_{\text{population}} w_1 \frac{f_1 - f_1^{\text{target}}}{s_1} + (1 - w_1) \frac{f_2 - f_2^{\text{target}}}{s_2}, \tag{3}$$

where w_1 and $1 - w_1$ are weights for the objective functions, f_1 and f_2 the momentum aperture and dynamic aperture objective functions, respectively, the target values $f_1^{\text{target}} = -0.02$ and $f_2^{\text{target}} = -50$ mm^2 and the scaling constants $s_1 = 0.02$ and $s_2 = 50$ mm^2. Five combinations of weights are chosen, with $w_1 = 1$, 0.75, 0.5, 0.25 and 0, respectively. Figure 4 shows the history of the weighted objective with these weight combinations for MOGA (left) and MOPSO (right), with or without initial seeds. The objectives make quick gains initially and slow down after 10 or 20 generations for both algorithms. For MOGA algorithm, the run with initial seeds clearly reached a better set of solutions after about the 30th generation. However, the initial seeds do not seem to make a difference for MOPSO except at the very beginning. In general, it is desirable for an algorithm to reach final results without initial seeds because in such cases the search of the parameter space is unbiased.

Next we compare the performance of MOGA and MOPSO. Figure 5 shows the objective functions for the two algorithms at four stages. Figure 6 left plot shows the percentage of MOPSO solutions in the 100 best solutions when the best solutions of the two algorithms at each generation are combined

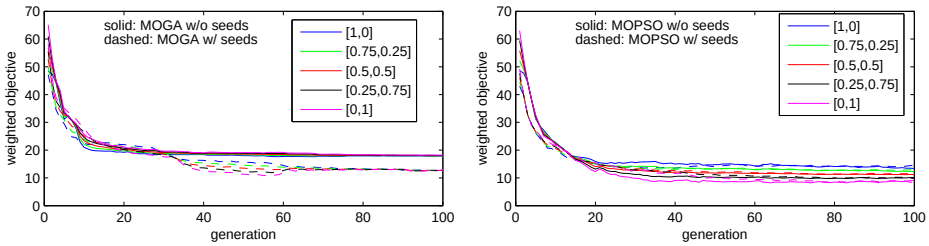

Figure 4: Comparison of the history of weighted objective functions in 100 generations for MOGA (left) and MOPSO (right) with or without initial seeds.

and selected with non-dominated sorting. Clearly MOPSO converges to the final solutions at a much faster pace than MOGA. At generation 20, the MOGA population falls behind the MOPSO population significantly while the latter is already very close to the final population. We think the reason for the slow convergence of MOGA is its lack of diversity in the population. To characterize the diversity of the population, we calculated the average parameter space crowding distance for the best solution population of each algorithm. Here the parameter space crowding distance is defined for each solution as the distance (i.e., the 2-norm of the difference vector) between the solution and its nearest neighbor in the population in the parameter space. It is worth noting that the global best solution archive of MOPSO is not used to spawn new solutions. Nonetheless, the average crowding distance of the global best archive still serves as a good indication of the diversity of the new solutions. Figure 6 right plot shows the average parameter space crowding distance of MOGA and MOPSO. The MOGA population becomes closely packed in the parameter space quickly after the launch, which makes it very ineffective in searching for better solutions because the new solutions it generates are very similar to the existing solutions.

By comparing the two bottom plots in Figure 5 we may conclude that MOPSO converges to roughly the same solutions in the objective space in 20 generations as MOGA does in 100 generations. Fast convergence is extremely useful if an algorithm is used for online optimization of machine performance [12, 13], in which case the lattice evaluation is carried out with measurement. Because online experiments cannot be conducted in parallel as in simulation and each measurement takes considerable time, high efficiency of the algorithm is essential for an experimental optimization to finish in reasonable time.

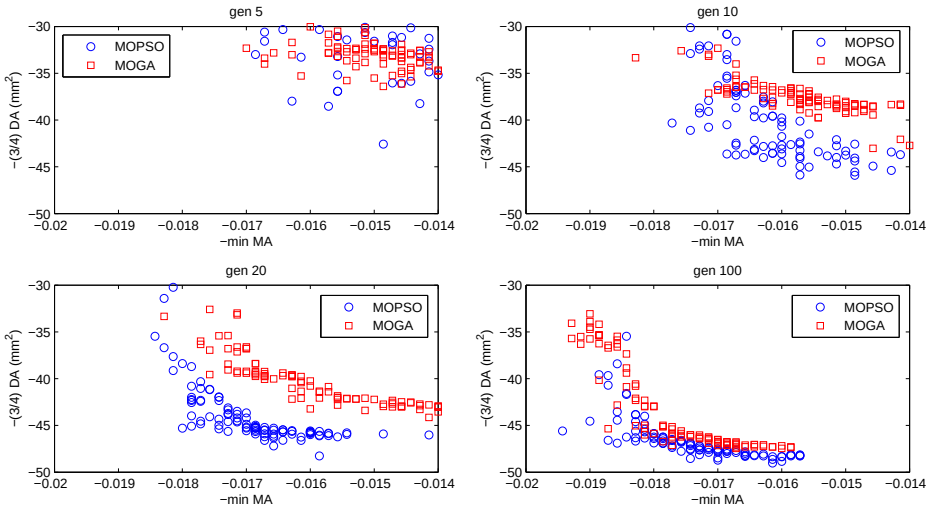

Figure 5: Comparison objective functions of the populations for MOGA (with initial seeds) and MOPSO (without initial seeds) at generation 5, 10, 20 and 100.

3.5. Nonlinear dynamics performance of the best solutions

The final solutions found by the optimization algorithms are significantly better than the initial solution in both dynamic aperture and momentum aperture. A best solution was chosen from the final population of MOPSO as a trade-off between the two objective functions. The dynamic aperture and momentum aperture for this solution are compared to the initial solution in Figure 7. The color code of the left plot indicates the number of turns when the particles are lost in tracking for the optimized lattice. There is a substantial improvement in the dynamic aperture. We also ran dynamic aperture tracking with random optics errors in the lattice with multiple random seeds to generate an rms beta beat of 1% and 0.2% linear coupling. The average dynamic aperture is -12.9 mm on the negative side, which would be sufficient for user operation with full capture of the injected beam after the septum upgrade is complete. The momentum aperture in the right plot are obtained with tracking of 5000 turns. The corresponding Touschek lifetime for the optimized solution would be 6.9 hrs under the user operation mode with a 500 mA beam and a 0.2% linear coupling. The calculated Touschek lifetime for the initial solution is only 3.2 hrs.

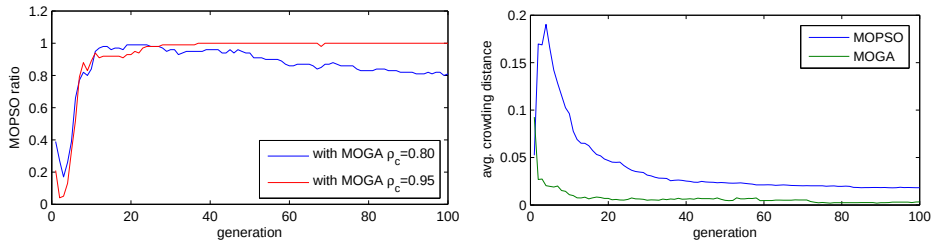

Figure 6: Left: The percentage of MOPSO solutions in the best 100 solutions selected by combining the MOGA and MOPSO solutions at each generation and applying non-dominated sorting. The results with respect to the MOGA cases with crossover probability of $\rho_c = 0.80$ and 0.95 are shown. Right: The average parameter space crowding distance of the global best archive for MOPSO and MOGA ($\rho_c = 0.80$).

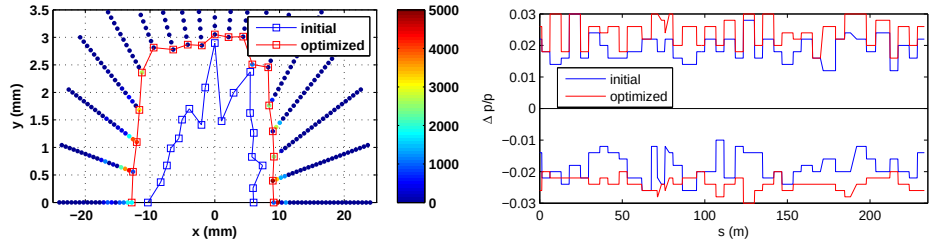

Figure 7: Comparison of the dynamic aperture (left) and momentum aperture (right) for a selected final lattice solution and the initial lattice.

4. Conclusion

We developed new lattices for SPEAR3 to significantly reduce its effective emittance from the present level. Because of the strong interaction between the chromatic sextupoles, these lattices have poor nonlinear dynamic performance before optimization of the sextupole strength distribution. The multi-objective optimization algorithms MOGA and MOPSO are applied to the nonlinear dynamic optimization problem for the SPEAR3 upgrade study and their performance are compared. We found that MOPSO has two distinct advantages. First, the results of MOPSO are not dependent on the initial distribution of solutions, which allows a thorough, unbiased search of the parameter space. On the other hand, MOGA requires seeding of good solutions in the initial distribution of solutions in order to converge to the true global optima. Second, MOPSO converges significantly faster than MOGA. It reaches the region of the final solutions of MOGA with a factor of five fewer lattices evaluations than MOGA. These advantages of MOPSO, as demon-

strated in Ref. [9] and here, make it much more effective than MOGA in many applications. In particular, the fast convergence makes MOPSO much more useful than MOGA in online optimization in which lattice evaluation is done with measurement on the machine. The advantages of MOPSO over MOGA are probably due to the high diversity in the solutions of the former because in MOPSO new solutions are not generated from existing good solutions.

Acknowledgment

The study is supported by DOE Contract No. DE-AC02-76SF00515.

[1] X. Huang, Y. Nosochkov, J. Safranek, L. Wang, THPC073, Proceedings of IPAC 2011, San Sebastian, Spain (2011).

[2] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, Wiley & Sons, LTD (2001).

[3] D. Robin, W. Wan, F. Sannibale and V. Suller, Phys. Rev. ST Accel. Beams **11**, 024002 (2008).

[4] I. Bazarov and c. K. Sinclair, Phys. Rev. ST Accel. Beams **8**, 034202 (2005).

[5] L. Yang, D. Robin, F. Sannibale, C. Steier, W. Wan, Nucl. Instrum. Methods. Phys. Res., A, **609** (2009) 50-57.

[6] M. Borland, *et. al.*, TH6PFP062, Proceedings of PAC'09, Vancouver, BC, Canada (2009).

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

[8] L. Yang, Y. Li, W. Guo and S. Krinsky, Phys. Rev. ST Accel. Beams **14**, 054001 (2011)

[9] X. Pang, L.J. Rybarcyk, Nucl. Instrum. Methods. Phys. Res., A, **741** (2014) 124-129.

[10] L. Wang, X. Huang, Y. Nosochkov, J. Safranek, TUPPC096, Proceedings of IPAC12, New Orleans, Louisiana (2012).

[11] K. Deb, IEEE Trans. on Evolutionary Computation, vol 6, no 2, (2002).

[12] X. Huang, J. Corbett, J. Safranek, J. Wu, Nucl. Instrum. Methods. Phys. Res., A, **726** (2013) 77-83.

[13] K. Tian, J. Safranek, Y. Yan, Phys. Rev. ST Accel. Beams **17**, 020703 (2014)