Enlarging Dynamic and Momentum Aperture by Particle Swarm Optimization

Zhenghe Bai, Lin Wang, Weimin Li

USTC/NSRL

Sept. 2011

Introduction

□ Obtaining adequate dynamic aperture (DA) and momentum aperture (MA) is crucial for

—high injection efficiency

—long beam lifetime

□ Minimizing nonlinear driving terms

□ Artificial Intelligence (AI) algorithms

—Genetic Algorithms (GAs)

—Particle Swarm Optimization (PSO)

□ Two important problems in the nonlinear optimization using AI algorithms

—how to quantitatively describe the objective functions?

—how to reduce the computation time?

Simulation Parameters

□ DA simulation

—code: *elegant*

—command: *bunched_beam*

—initially **1600** tracking particles are uniformly distributed

—one can find some small unstable islands

—but the calculation consumes much time

—50~100 turns

□ MA simulation

—transverse MA

—several tens of turns

□ After optimization, the optimal solutions are checked with detailed tracking and analysis.

Quantitative Criterion of DA

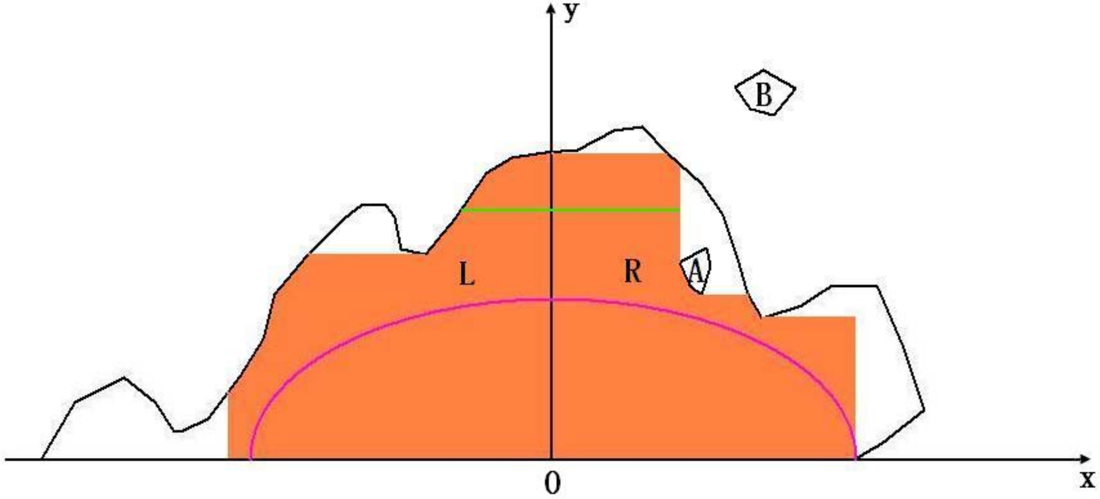
- □ ***A*** is an unstable island

  B is a stable island

- □ The quantitative criterion:

  $$D = P - k_1 \times |L - R| + k_2 \times E$$

  $$(0 \le k_1 \le 1, k_2 \ge 0)$$



- □ ***P*** represents the area of the orange pyramid-like shape

  L represents the left area of the *P* (*x*<0), and ***R*** the right (*x*>0)

  E represents the area inside the pink half-ellipse

- □ The minimum height is restricted → the left "tail" is clipped off

  The maximum height is restricted → the area below the green line

PSO Algorithm

□ Particle Swarm Optimization (PSO) was proposed in 1995, motivated by social behavior of organisms such as bird flocking.

□ In PSO, each potential solution is called “particle”. Each particle updates its velocity and position according to its own experience, and the experience of neighboring particles.

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times r_1 \times (\underline{p_{id}(t)} - x_{id}(t)) + c_2 \times r_2 \times (\underline{p_{gd}(t)} - x_{id}(t))$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1)$$

$\boldsymbol{p_{id}}$: the particle’s personal best position found so far

$\boldsymbol{p_{gd}}$: the best position found so far in the neighborhood of the particle

□ The neighborhood is the entire swarm → the global version of PSO

The neighborhood is a subset of the swarm → the local version of PSO

Constraint-handling Technique

□ The objective function is $-D$ (D is the quantitative value of DA), and the PSO algorithm minimizes the objective function.

□ The constraints are corrected chromaticity settings.

□ The weighted sum of constraint violations is treated as an additional objective function.

Strategy for Reducing DA Computation Time

□ The DA calculation has a special characteristic. For one solution,

—if its several-turn DA is good, its hundreds-of-turn DA **may be** good;

—if its several-turn DA is bad, its hundreds-of-turn DA **must be** bad.

□ We proposed the "**More Turns and Fewer Turns**" (MTFT) strategy for reducing DA computation time.

□ ***D_best*** is the best DA quantitative value in the swarm found so far, which is tracked for more turns.

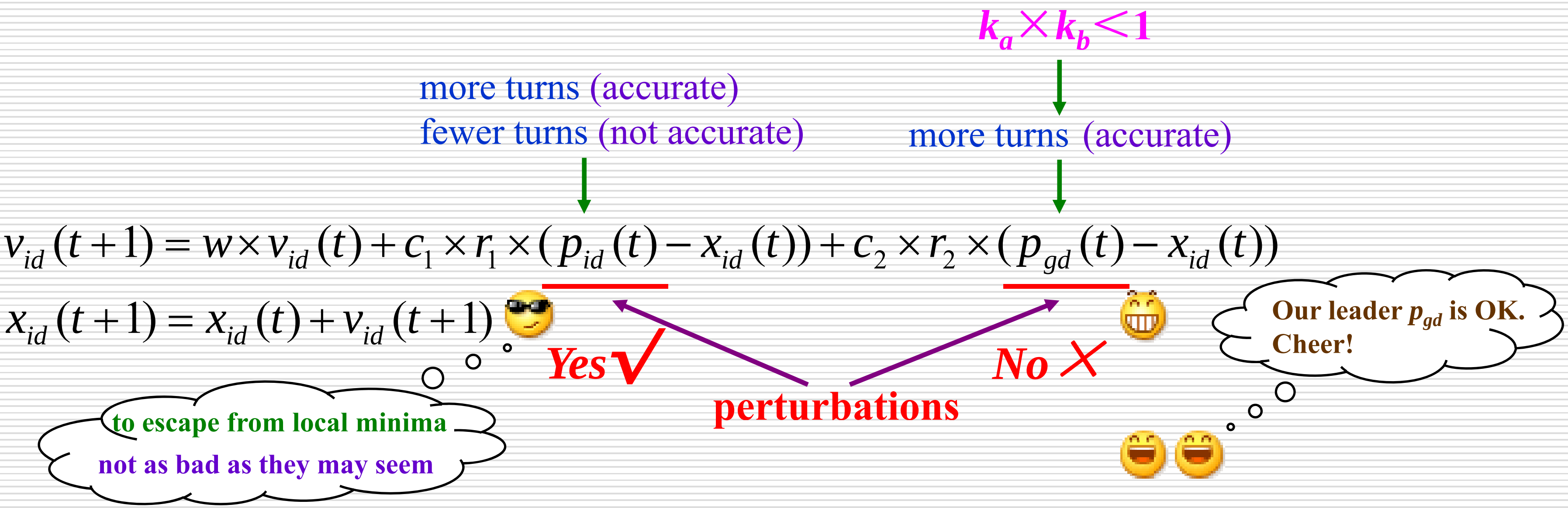
k_a is a positive constant **about 1**; k_b is a constant in the range $0<k_b<1$; $k_a\times k_b<1$.

□ The DA of one solution is first tracked for fewer turns, and its DA quantitative value for fewer turns, ***D_i***, is obtained.

—If $D\_i<k_a\times D\_best$, we assign $k_b\times D\_i$ as its DA quantitative value for more turns;

—If $D\_i\geq k_a\times D\_best$, its DA is then tracked for more turns to get its DA quantitative value for more turns.

Strategy for Reducing DA Computation Time



$k_a \times k_b < 1$

more turns (accurate)

fewer turns (not accurate)

more turns (accurate)

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times r_1 \times (p_{id}(t) - x_{id}(t)) + c_2 \times r_2 \times (p_{gd}(t) - x_{id}(t))$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1)$$

Yes √

perturbations

No ×

Our leader p_{gd} is OK. Cheer!

to escape from local minima

not as bad as they may seem

- □ ***D_1***: the DA quantitative value of p_{id}; ***D_2***: the DA quantitative value of p_{gd}.
- □ If *D_i*<*D_1* (of course *D_i*<*D_2*), the equation of motion for the particle will not change whether the particle's DA is tracked for fewer turns or more turns.
- □ Because $k_a \times k_b < 1$, *D_2* is always accurate, but *D_1* sometimes is not so accurate. Thus, some perturbations are introduced into the PSO algorithm.
- □ The perturbations may help escape from local minima, so they are not as bad as they may seem.

Strategy for Reducing DA Computation Time

- □ ***$N_{\_M}$***: the number of the more turns; ***$N_{\_F}$***: the number of the fewer turns.
- □ Theoretically the computation time can be reduced by a factor of ***$N_{\_M}/N_{\_F}$*** at most.

  But in some our DA optimizations, from the point of view of the iteration number, it is more than ***$N_{\_M}/N_{\_F}$***.
- □ That means that using the MTFT strategy, at the same iteration the number of the particles satisfying the constraints is usually smaller than not using the strategy (because of the perturbations).

  We assume that the performance of constrained PSO is not sensitive to the number of the particles satisfying the constraints. Thus the iteration number becomes important.
- □ So, using the strategy, **the computation time can be reduced much more**. And, as analysed and assumed above, in our DA optimization, we found that **the PSO algorithm using our proposed MTFT strategy can get as good results as not using the strategy**.

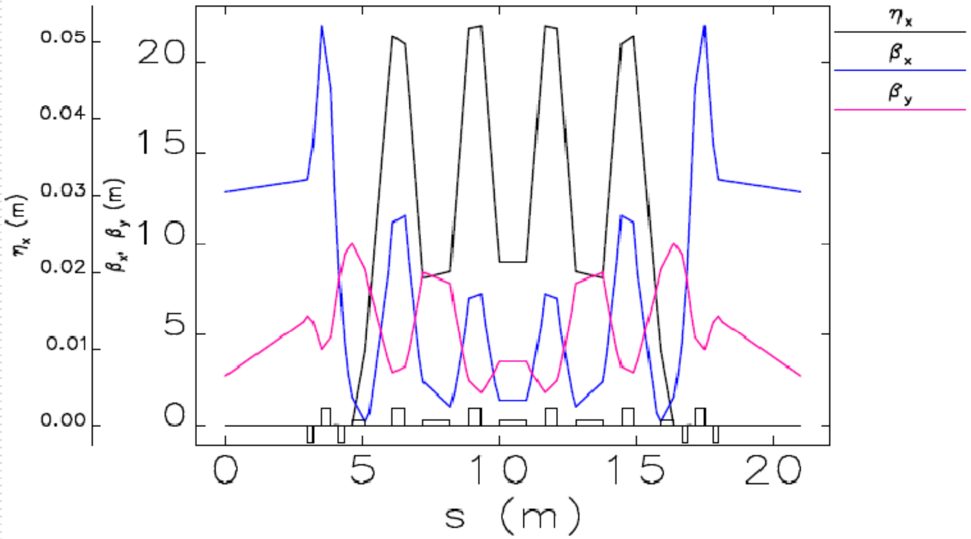
Other Strategies for DA Optimization

- □ The global version of PSO converges very fast, but is easily trapped into local minima. We use the local version to optimize the DA.

- □ To make some relatively good solutions as good “attractors” at the beginning of the algorithm, we proposed the pre-initialization strategy.
- □ Before initialization, the same size population as in initialization is repeatedly initialized lots of times. In this process, the dynamic apertures of the solutions satisfying the chromaticity constraints are tracked for only one turn or several turns.
- □ Then, at initialization, the positions of the best solutions obtained in pre-initialization are respectively assigned to the positions of the same number of particles.
- □ The process of pre-initialization is quick.

Optimization of DA of an FBA Lattice

- □ The FBA lattice has 20 super-periods, with an ultra-low emittance of 69pm•rad at 1.5GeV.



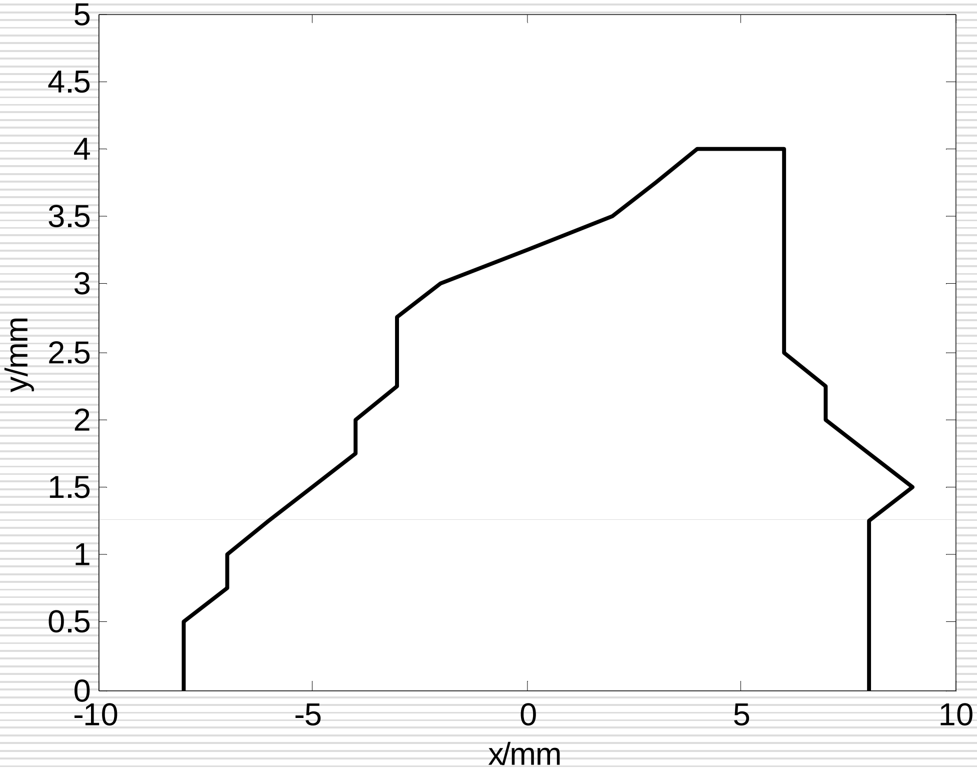
- □ In our optimization related to DA, we choose 50 or 100 turns as the more turns, and 4 or 5 turns as the fewer turns.
- □ The coefficient k_b is always 0.8, and the other k_a is slightly greater than or equal to 1.

Optimization of DA of an FBA Lattice

- □ First we use seven variables of chromatic sextupole strengths to optimize the DA.
- □ We optimized the DA without lattice errors, using the local version of PSO with a population size of 10000 and 30 iterations, on a common PC computer (2.50GHz CPU).
- □ After optimization, two sextupole strengths are basically zero. So we can employ five families. The variable range for optimal solutions is obtained.
- □ Then we add two harmonic sextupoles to further optimization. So there are seven variables. The searching range of chromatic sextupole strength is set to the range obtained above. Lattice errors are included.
- □ The local version of PSO with a population size of 400 ran for 75 iterations on the same PC computer.

Optimization of DA of an FBA Lattice

- □ Even though with 1600 tracking particles, either of these two works was done **in only three days** on the PC computer (2.50GHz CPU).
- □ It strongly shows the power of our proposed MTFT strategy for reducing computation time.



Joint Optimization of DA and MA

- □ The quantitative criterion of MA :

  $$M = M_p + |M_n| + k_3 \times M_{\min} \quad (k_3 \geq 0)$$

- □ M_p represents the minimum value of MA along the lattice in the positive momentum deviation direction;

  $|M_n|$ represents the minimum absolute value of MA in the negative direction;

  M_{min} represents the minimum between M_p and $|M_n|$.

  Let M_e be the enough value of MA. If the value of M_p or $|M_n|$ is greater than M_e, the value of M_e is assigned to M_p or $|M_n|$.

- □ For one solution, if its DA is not good, we will not choose this solution even though its MA is good. So, in such a case, it is not necessary to calculate the MA.

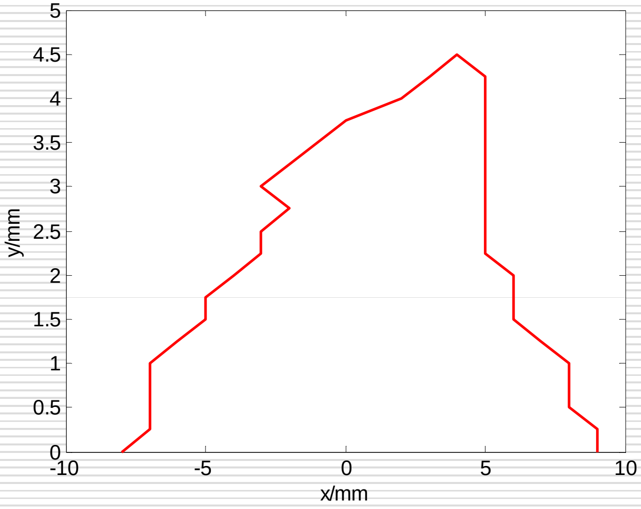
- □ MA is only calculated by tracking in the region of good DA.

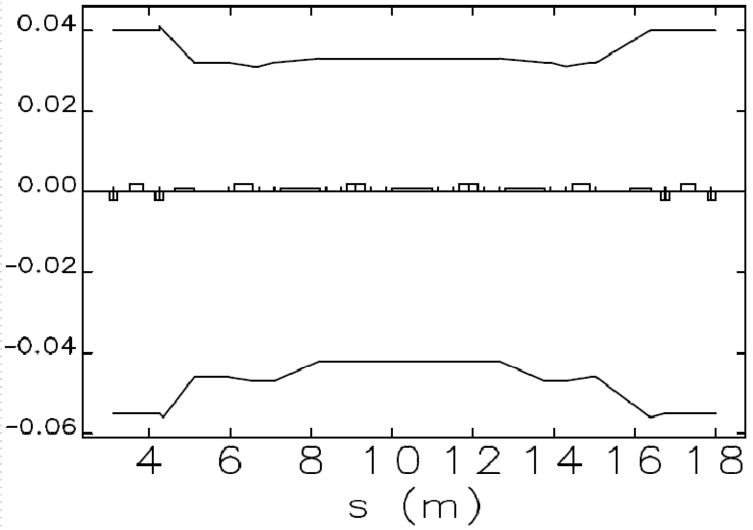
Joint Optimization of DA and MA

- □ we also use the MTFT strategy to reduce DA computation time.

  If one solution’s MA is calculated by tracking, its DA must be tracked for more turns.

- □ To quickly find optimal solutions, we used the global version of PSO. The algorithm with a population size of 600 ran for 30 iterations on the desktop computer, and it consumed **about one week**.





Conclusions

- □ The quantitative criteria for DA and MA are effective.
- □ The methods for reducing the computation time are powerful.
- □ PSO is a very helpful AI algorithm.