Genetic Algorithm Enhanced by Machine Learning for Dynamic Aperture Optimization

Yongjun Li

Brookhaven National Laboratory

BROOKHAVEN NATIONAL LABORATORY

13th ICAP Conference, Key West, 10/20/2018

Outline

- Existing DA optimizations: Local vs. Global
- Motivation of using ML in population-based algorithms
- Feasibility of ML in DA optimization
- Implementation of ML in MOGA
- An example - NSLS-II storage ring
- Summary

Local optimization

- Simplex, Conjugate gradient minimization (using Lagrange multiplier if constraints exist).
- Pros:
    - Easy to implement
    - Fast
- Cons:
    - Single objective, combining weighted multiple objectives
    - Trapping in local minimums
    - Single solution

Global optimization

- Population-based optimization: genetic algorithm and particle swarm (Borland, Yang, Huang, Jiao, Qiang, et. al.)
  - Pros:
    - Multiple objectives and constraints (non-dominated sorting)
    - Global minimum / completed Pareto front
  - Cons:
    - (Not so) difficult to implement
    - More computation resource needed
    - Slow (one of our motivations: Can we improve it?)

Motivation of using ML in GA

- GA has been proved useful in linear/nonlinear lattice
- No priori reason why GA needs intervention. But all these creations in our planet become possible only after **billions of years** of evolutions. Evolution (learning curve) is too slow!
  - Large search range => multiple generation + large population
  - Accumulated big data is not fully re-used and analyzed: By data mining, can we find some clues associated with beam dynamics?
- External intervention during evolution is very common

Feasibility of ML in DA optimization

- ML: learning from data to recognize **unknown** patterns:

  Given (x, y), to generalize a hypothesis y = f(x)

- DA optimization is **NOT** a typical ML problem
  - Given a lattice configuration, DA is a known function
  - There are no existing DA data before optimization

BUT…

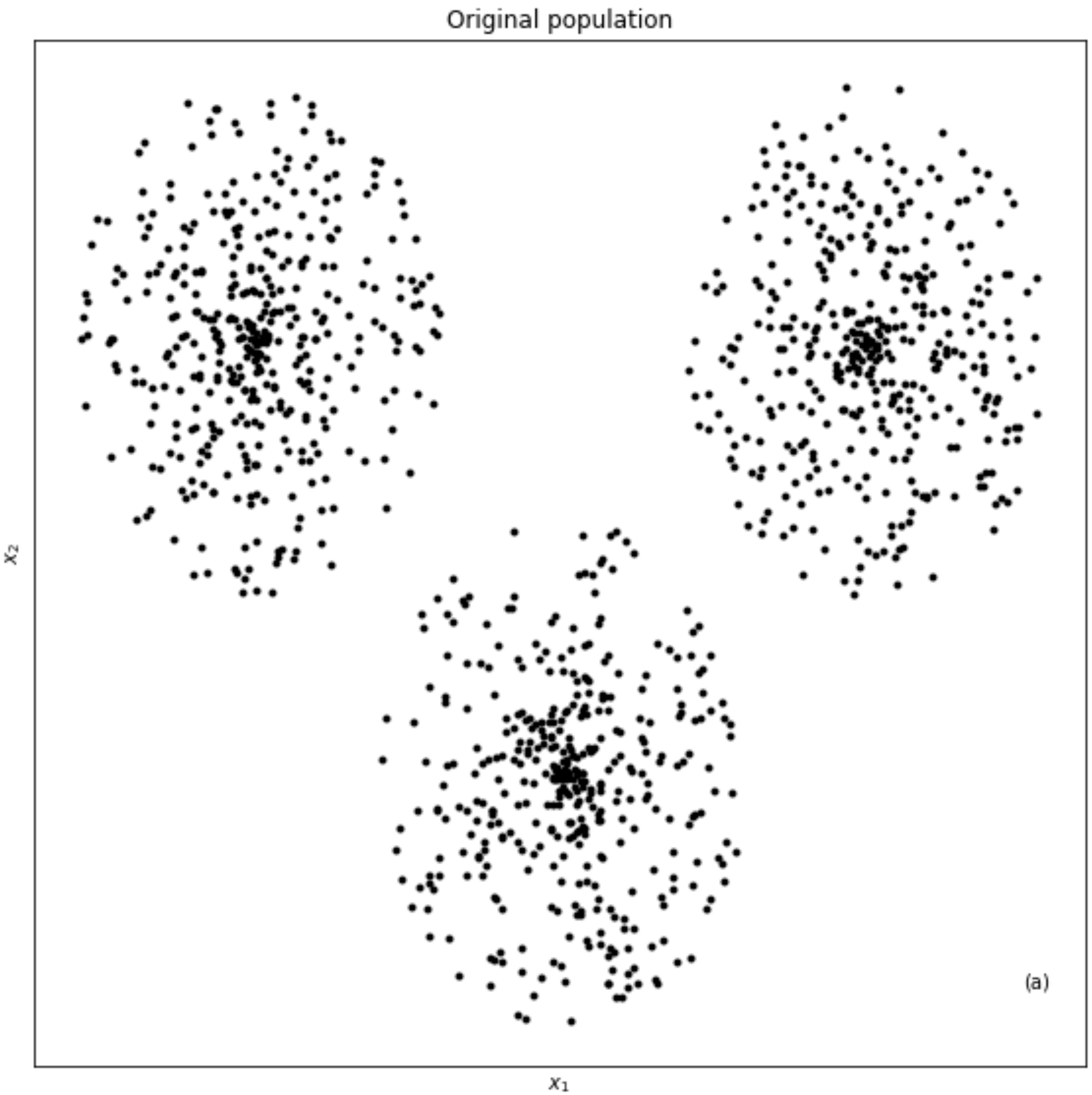
Feasibility of ML in DA optimization

- There are patterns between DA and lattice configuration
- A large data pool is generated when using population based optimization
- With ML, patterns might be able to be recognized from the data
- Applying recognized pattern to boost the evolution

Implementation of ML in MOGA

1. Initialize population randomly
2. Follow normal MOGA (cross-over and mutation) till all individuals satisfy constraints
3. Classify candidates into different clusters (N=100) using K-means algorithm
4. Compare the **average fitness** to find out a few best (n=3) clusters (elite clusters)
5. (Optional) Divide population of elite cluster into training group (95%) and testing group (5%), use supervised learning KNN algorithm to check if the learning model can predict the test group behavior.
6. Re-populate some amount (fixed or dynamically) of new population within the narrow searching space of these elite clusters, then to replace the same amount of candidates from the original population **randomly**.
7. Carry out cross-over, mutation and non-dominated sorting to next generation.
8. Repeat 3-8

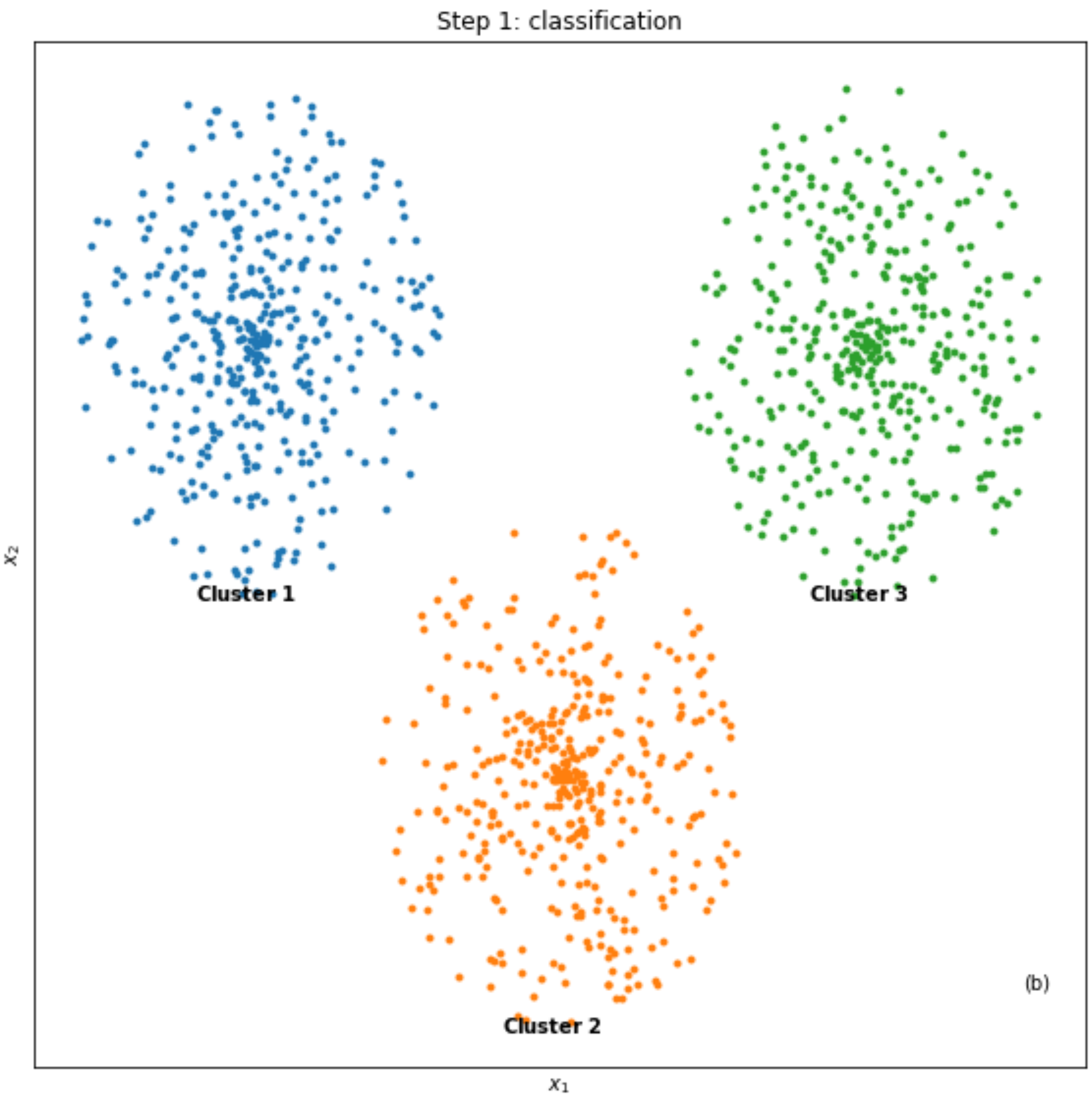
Schematic illustration of ML in GA (1)



two features (inputs): x1 and x2

Without considering the target function (DA)

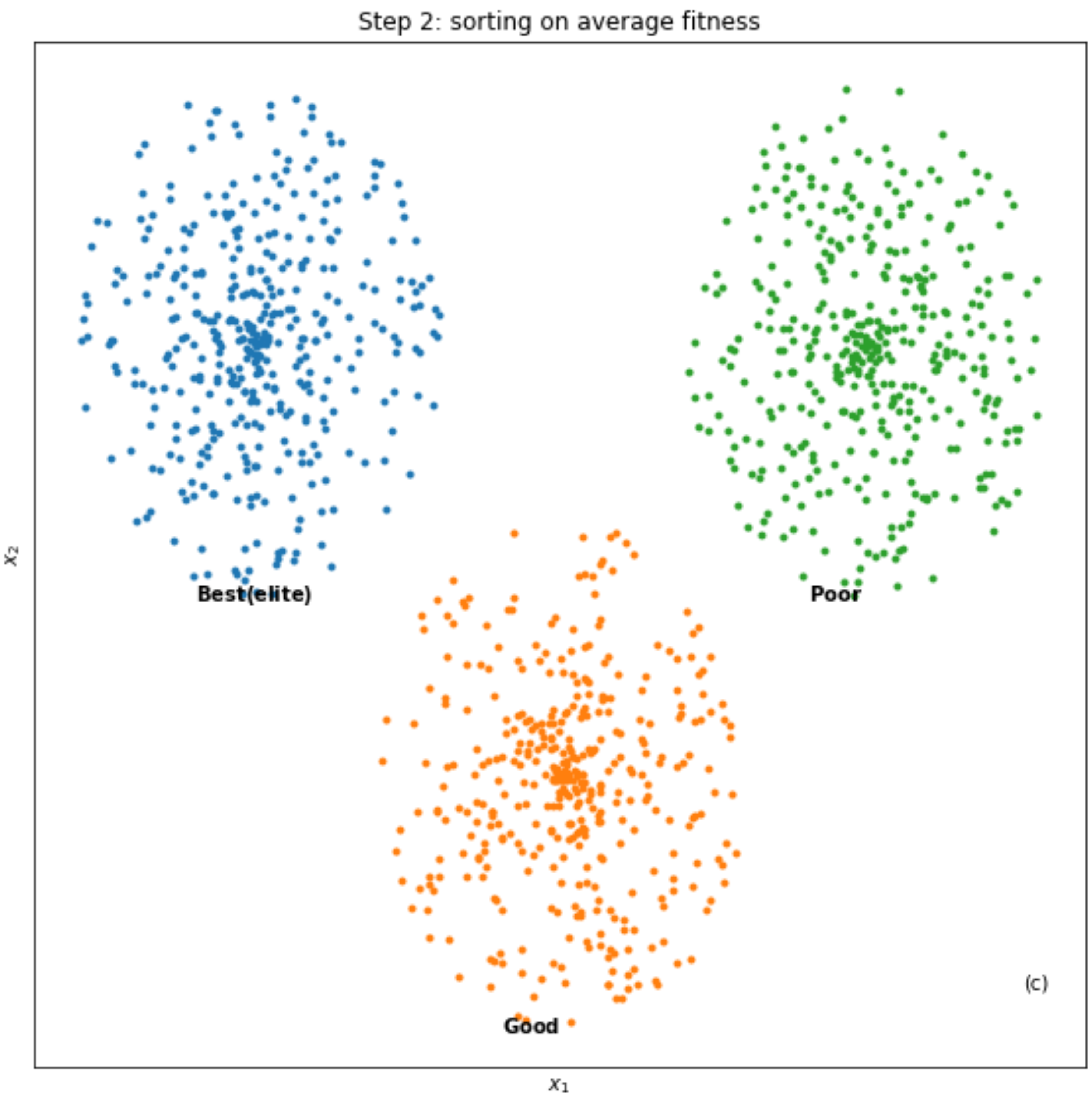
Schematic illustration of ML in GA (2)



Unsupervised learning:
Classify based on their distance in the Euclidean space

K-means algorithm (Lloyd's algorithm)

Schematic illustration of ML in GA (3)



Supervised learning: Label elite clusters based on their average fitness

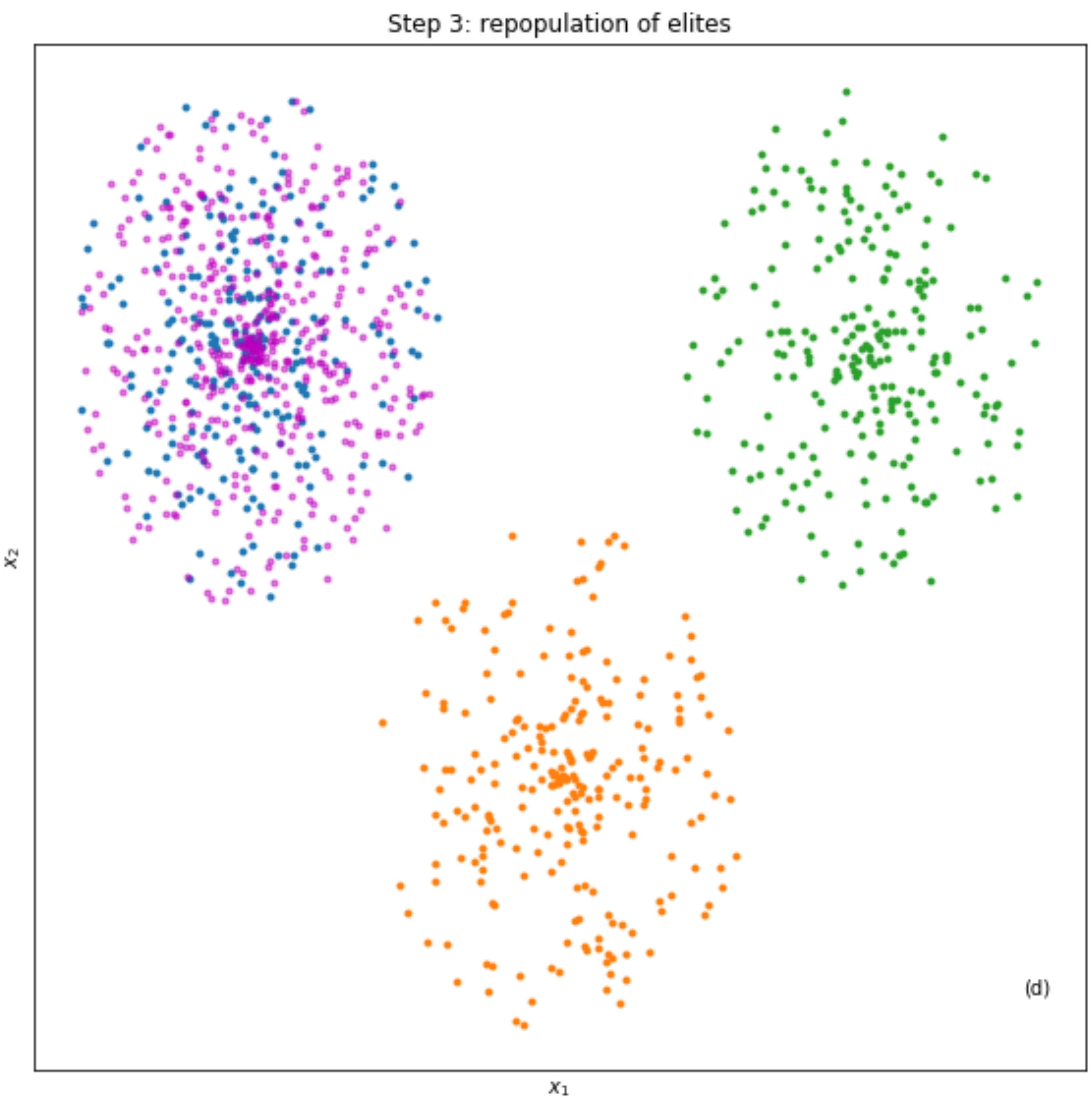
Weighted fitness:

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

$w_m = 1 \rightarrow$ average fitness

Instead of reaching an uniform crowding in the Pareto front, a more practical or "of physics" distribution can be obtained.

Schematic illustration of ML in GA (4)



Manual intervention

Supervised learning: Repopulate more potentially competitive candidates to replace randomly selected candidates

How much original candidates should be replaced?

1. **Static ratio**
2. **Dynamic ratio**

ML techniques

Unsupervised learning in **classification**

Grouping candidates into clusters based on their features (lattice settings)

K-means:
Lloyd algorithm

Supervised learning in **repopulation**

Repopulating new potentially good candidates based on the average fitnesses (dynamic aperture, measure of nonlinearity)

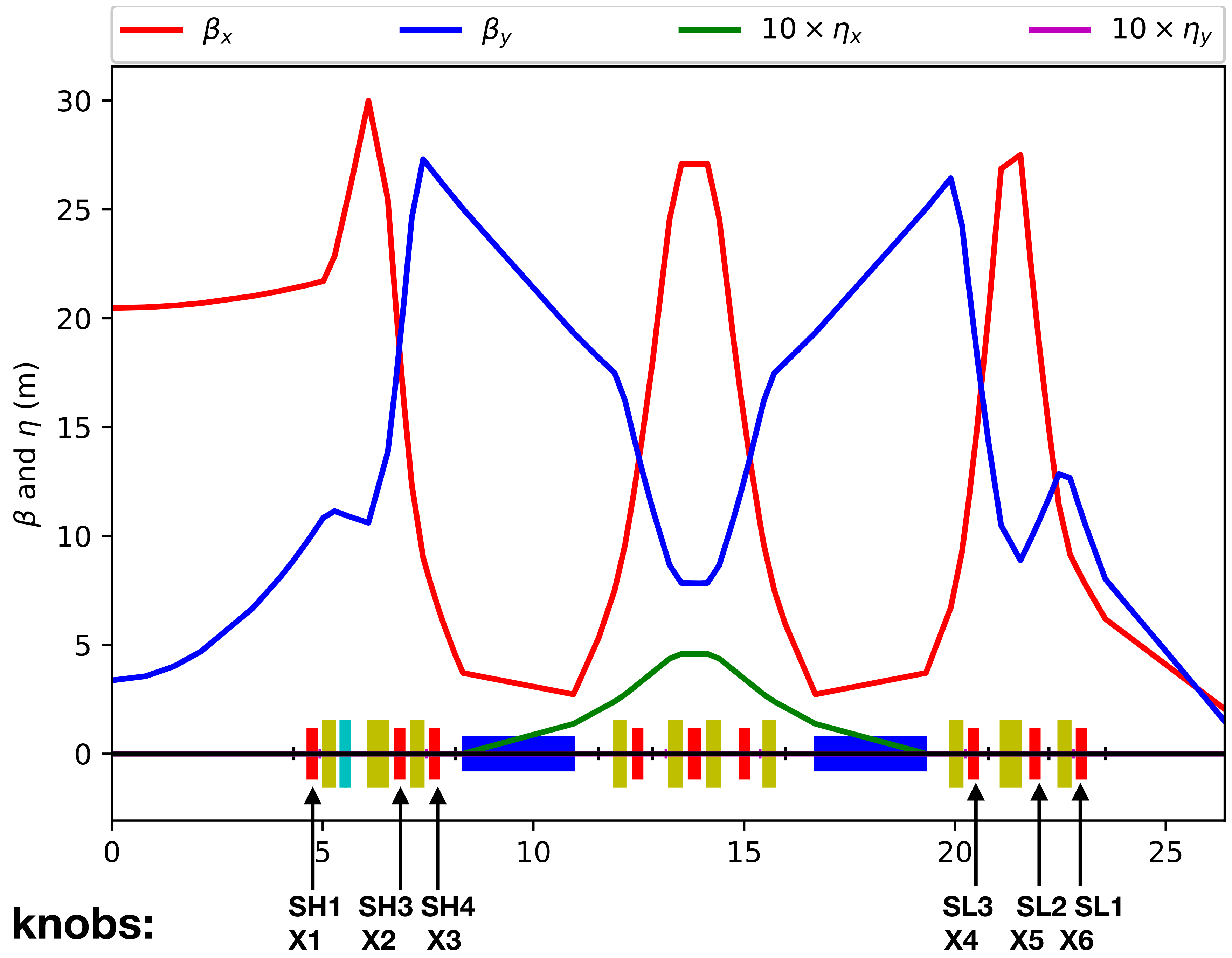
Supervised learning in **replacement**

Adjusting the amount of replaced candidates based on accuracy of prediction

KNN: K-nearest neighboring algorithm

“similarity”, “discrepancy”, are quantitatively represented by the Euclidean distance in N-dimension space

An example: NSLS-II ring

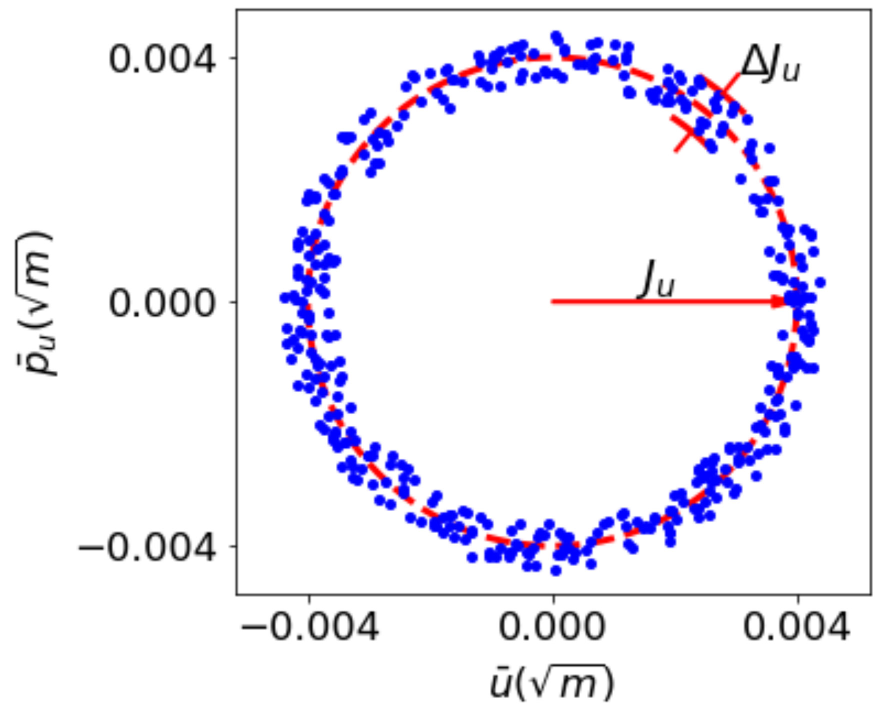


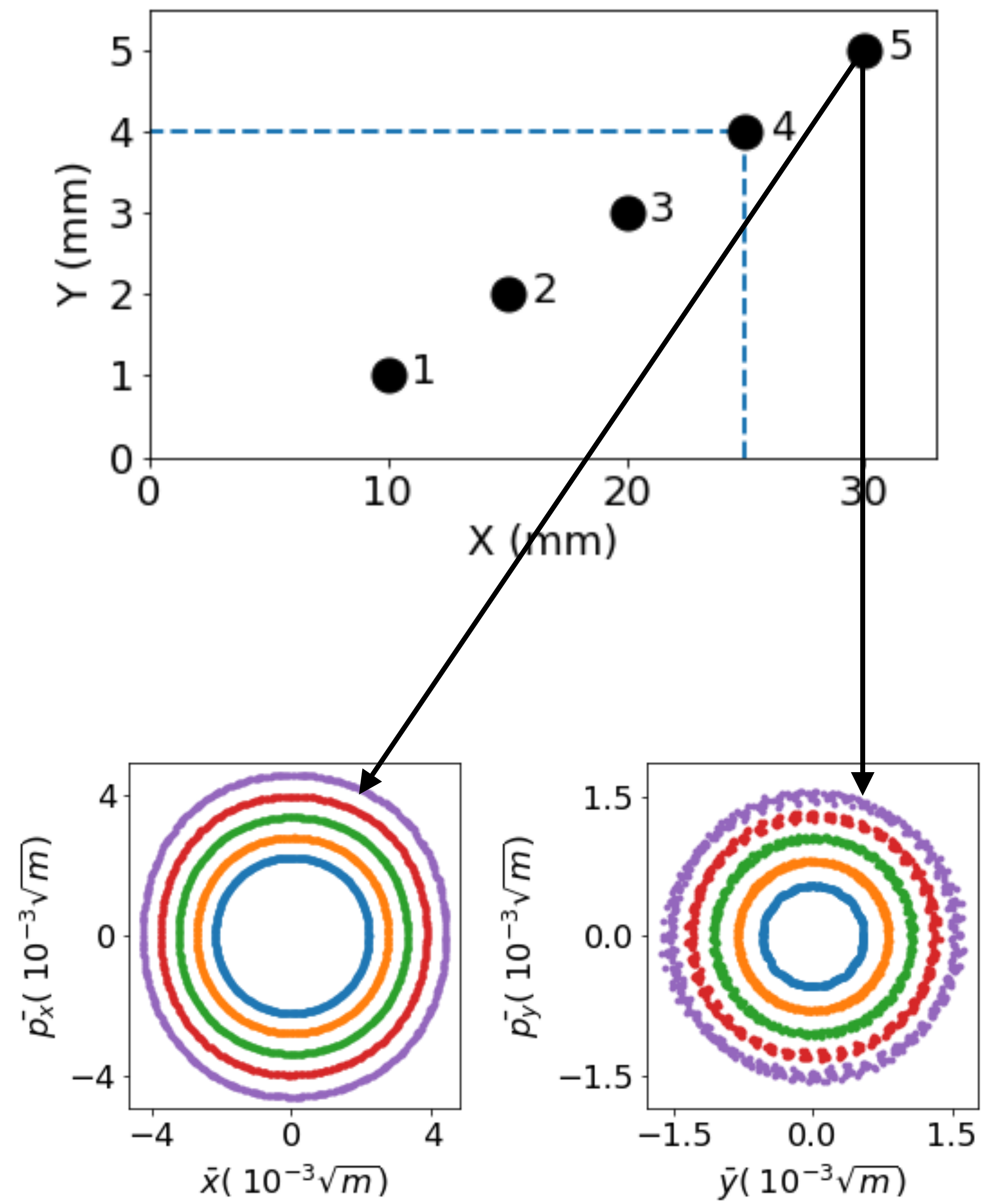
Free knobs:

Choosing optimization objectives

- Optimization objectives:
  - Tracking-based DA and Touschek lifetime (Borland)
  - Tracking-based on- and off-momentum DA (Yang)
  - Analytical nonlinear driving terms (OPAL, Li)
  - Square matrix method => new action-angle variables (Yu) => <u>**regular motion through tracking**</u>
  - …

Knobs and objectives





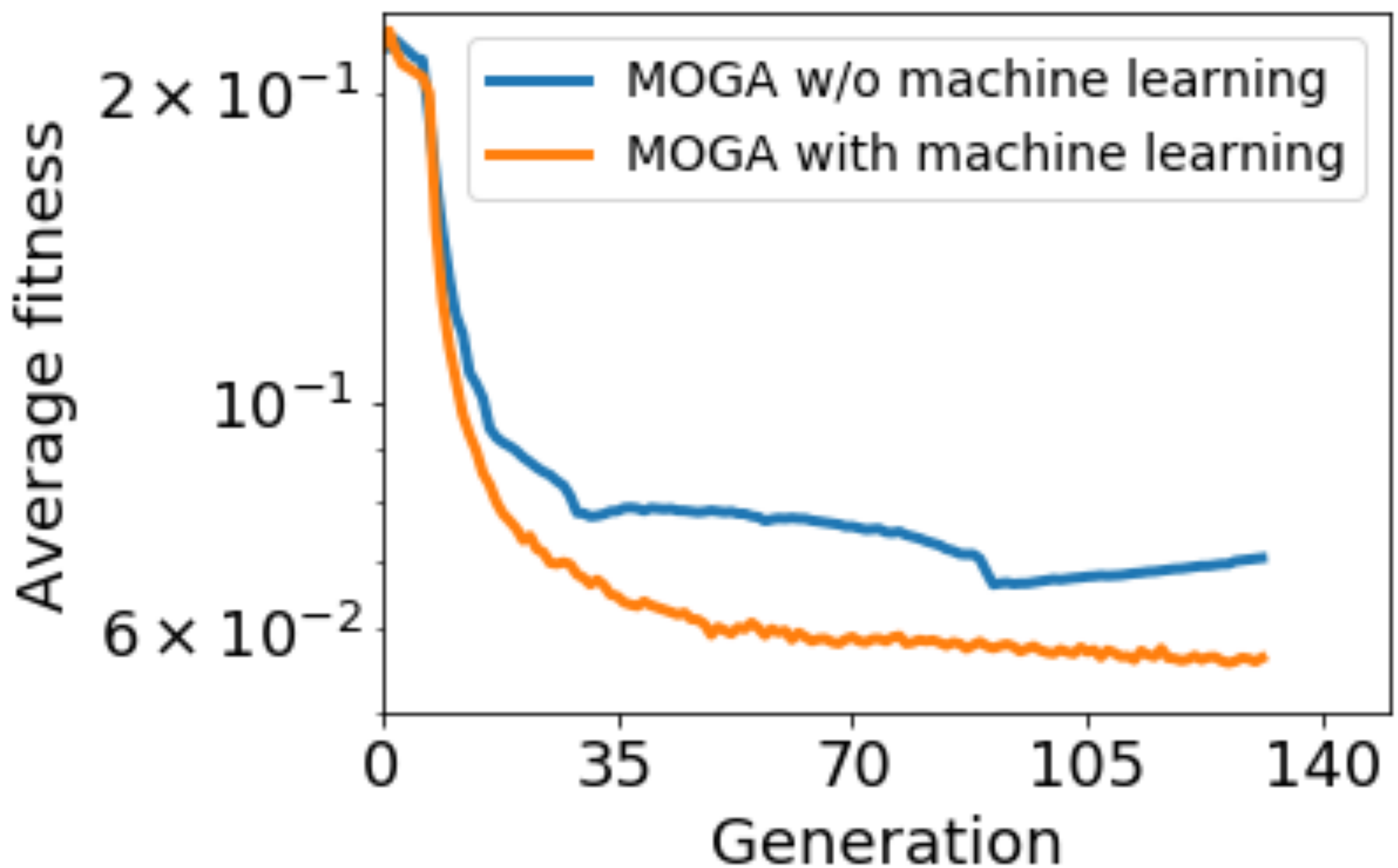
Free knobs:
6 families of harmonic sextupoles

Objectives:
5 particles dJ/J for multiple turns tracking

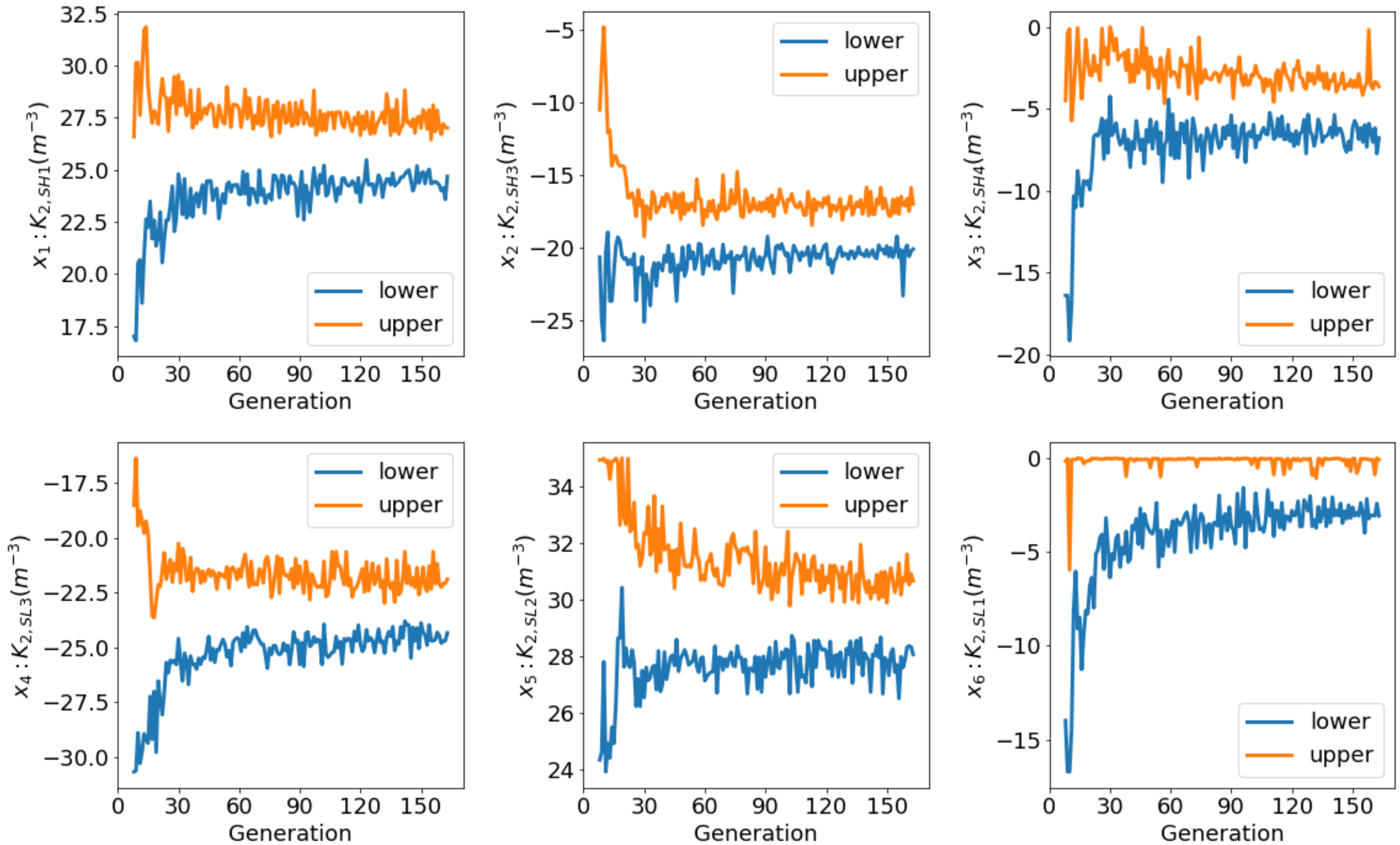
Constraints:
5 particles can survive in tracking

1. Li Hua Yu, Analysis of nonlinear dynamics by square matrix method, Phys. Rev. Accel. Beams **20**, 034001
2. Michael Borland, Private communication

Faster convergency with ML in MOGA

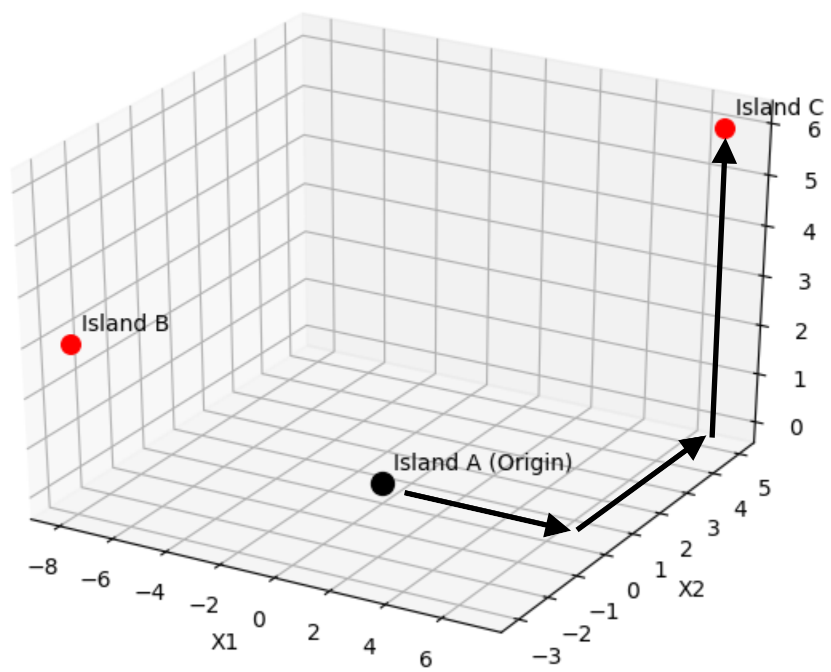


Evolution of elite ranges

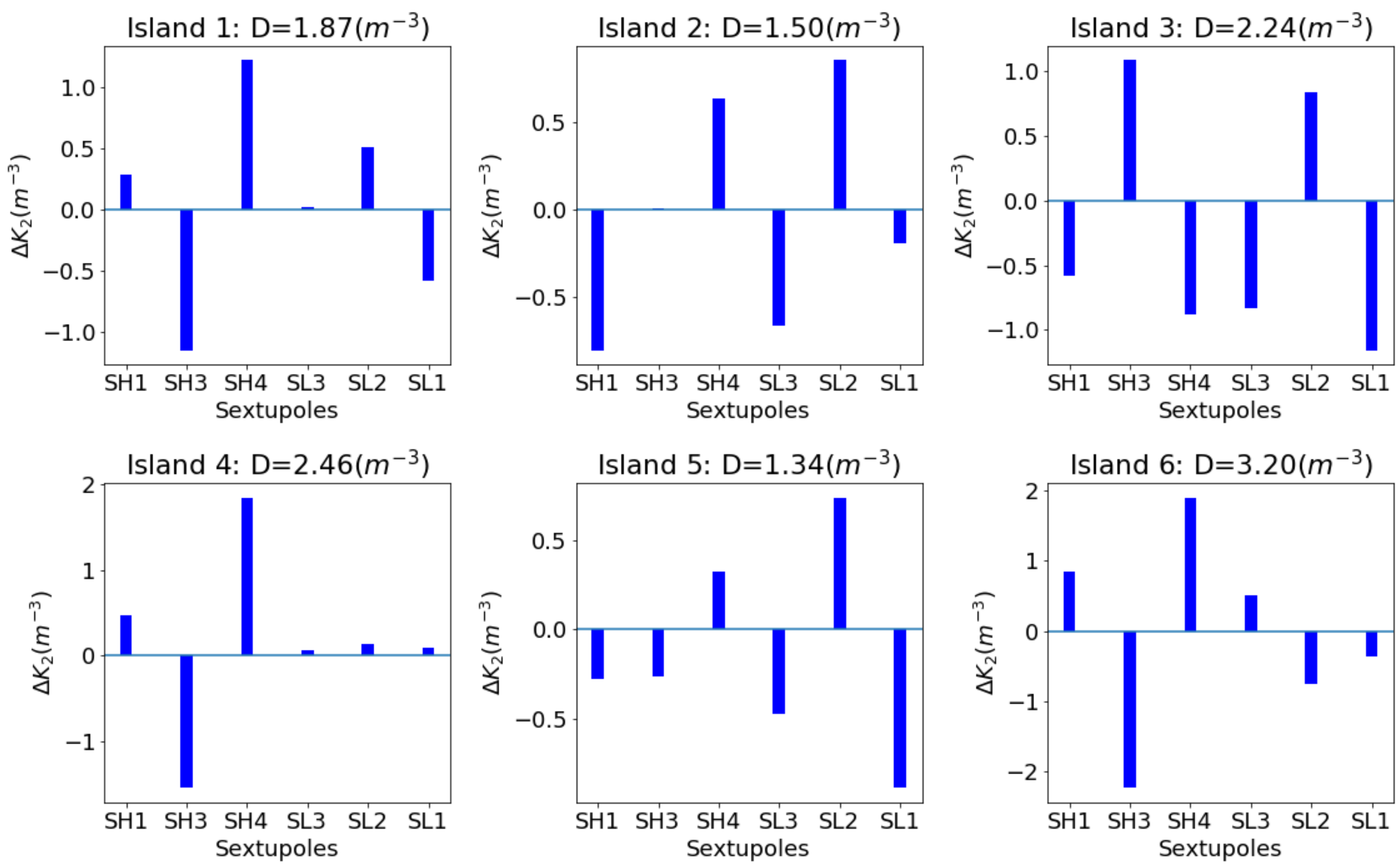


Free knobs:	**SH1**	**SH3**	**SH4**	**SL3**	**SL2**	**SL1**
	X1	**X2**	**X3**	**X4**	**X5**	**X6**

Data mining on Pareto front



Relative distances between two islands



- **Solutions are not unique (more sext knobs than needed?)**
- **Solutions are clustered into isolated islands**
- **Volumes of islands are different (Robustness of solution?)**
- **These islands might compose a structure (plane, curve?)**

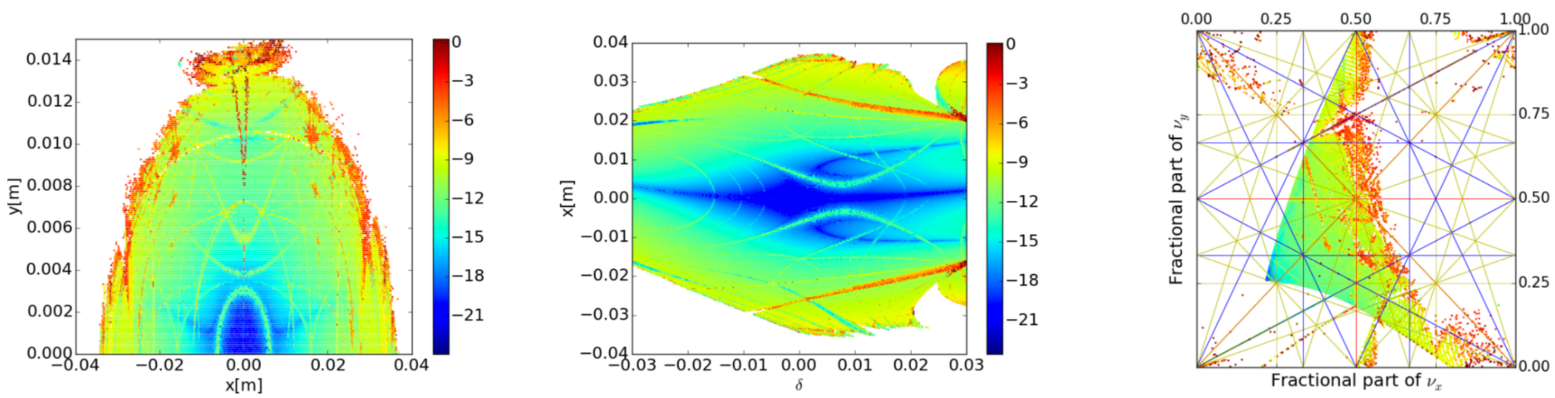
Comparison of two solutions

Table: Two well separated islands

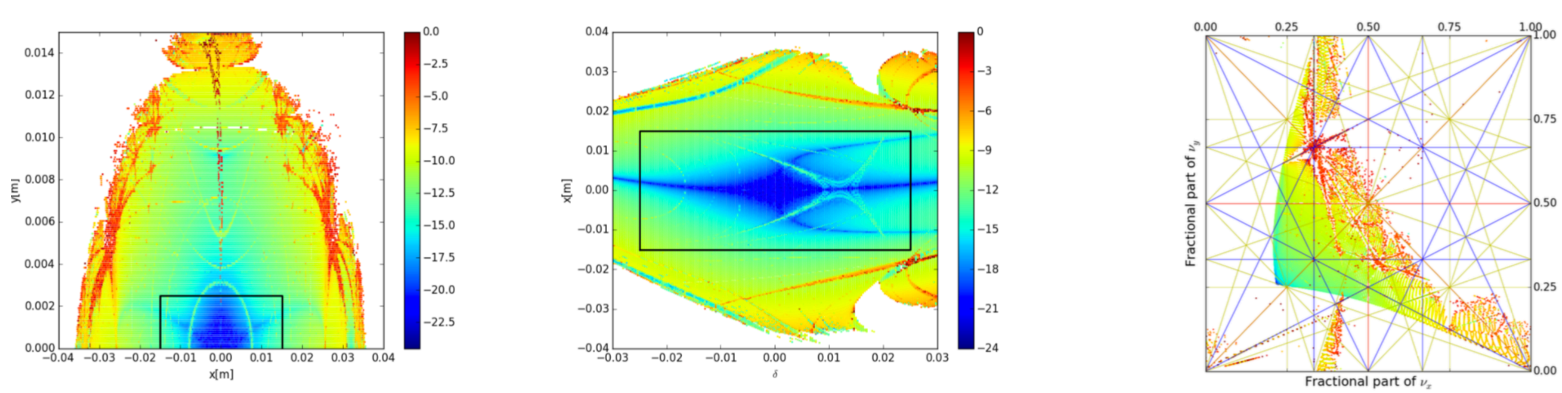
sext	parameter	island N	island 0
SH1	K2	23.97693060	26.20890568
SH3	K2	-12.94238420	-17.87663672
SH4	K2	-11.78548920	-6.39465862
SL3	K2	-23.71956290	-22.42606694
SL2	K2	30.42895540	28.54735068
SL1	K2	-1.99804991	-0.22496352

Sextupoles settings are quite different

switch off SL1 or change its polarity



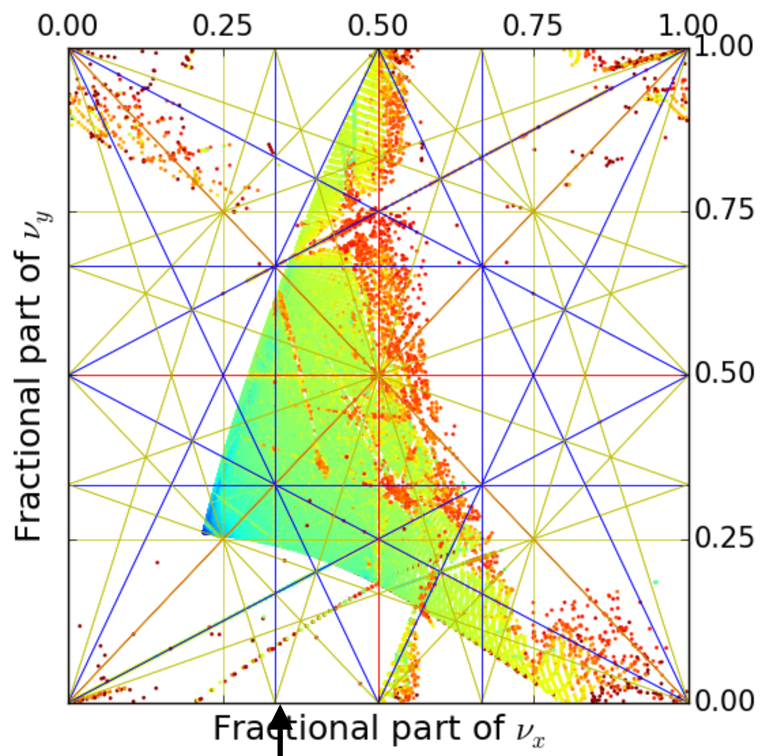
Similar nonlinear Properties



Some discussions

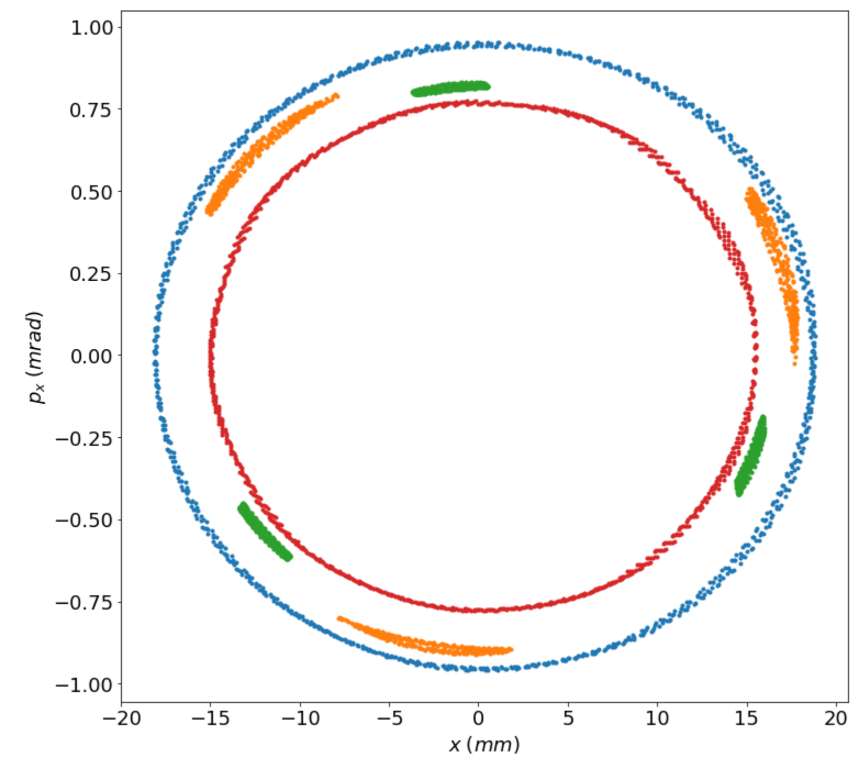
- Randomly replacement after repopulation
    - Maintain the **diversity** to achieve global optimization
- Supervised learning fails to predict the testing candidates
    - Strong nonlinearity: candidates have similar features, but different dynamic behavior
    - Robustness of solution, tight specification on magnet imperfections

Can DA cross 1/3 resonance

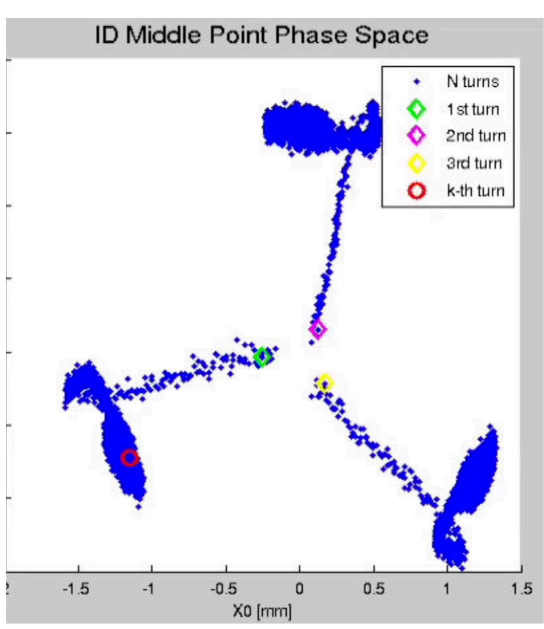


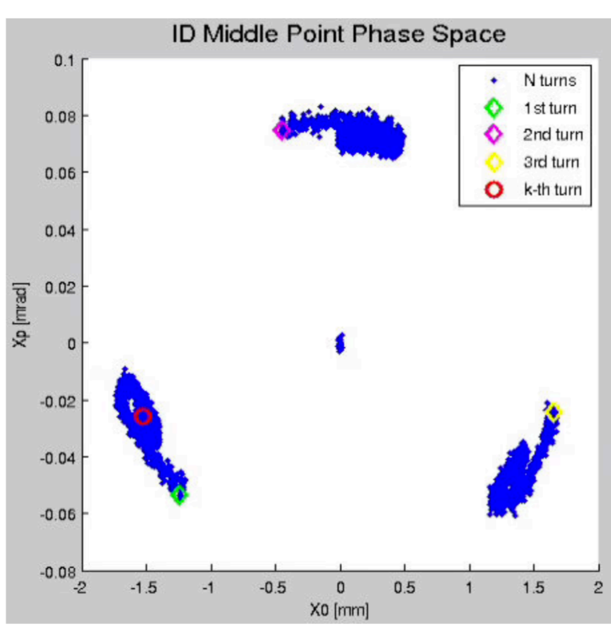
1/3 resonance

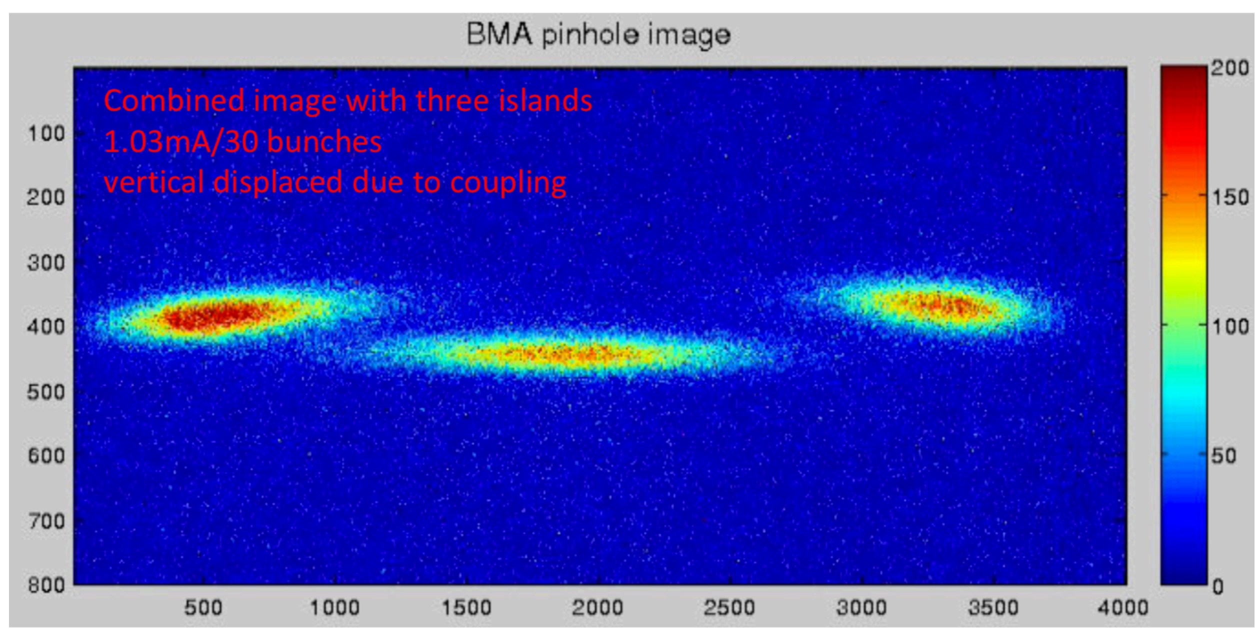
Simulations



Experimental observations







Displace beam with different methods to observe resonance trapping

Courtesy Weixing Cheng

Summary

- GA can be enhanced by ML technique in DA optimization
  - Fast convergency
  - Generating much more qualified solutions
  - Distribution of qualified solution might have some physics interpretation
  - Method itself is general for other population-based optimizer

Acknowledgements

- Coauthors: Weixing Cheng, Li Hua Yu and Robert Rainer
- Fruitful discussion with Dr. Michael Borland, Dr. Yipeng Sun, Dr. Xiaobiao Huang, and Prof. Alexander Wu Chao
- Many NSLS-II colleagues’ support and help
- ICAP’18 scientific committee