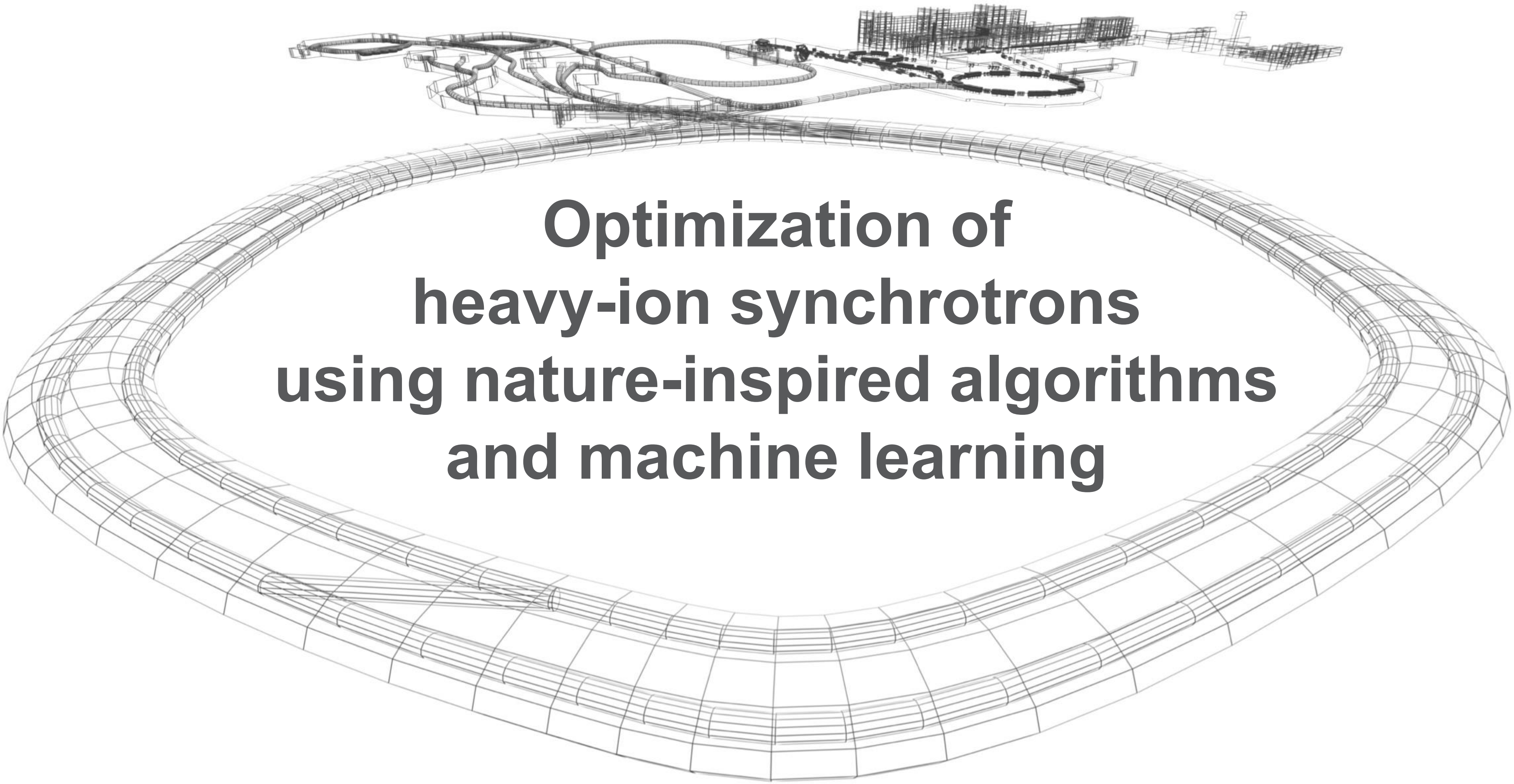


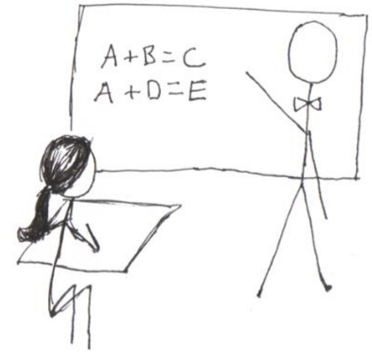
Optimization of heavy-ion synchrotrons using nature-inspired algorithms and machine learning

Dr. Sabrina Appel, Accelerator Physics Department, GSI, Darmstadt

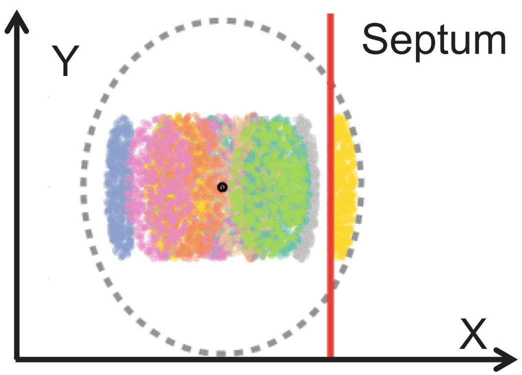
Outline

- **Nature-inspired optimization**
  - Evolutionary algorithm
  - Particle swarm optimization

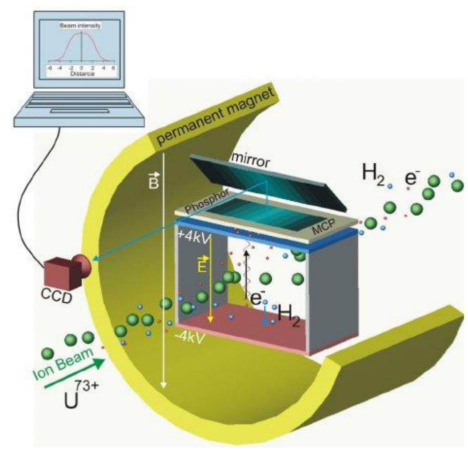
- **Machine Learning**
  - Linear Regression
  - Artificial Neural Networks



- **Example optimization problem:**
  - Multi-Turn Injection



- **Example Machine Learning**
  - Beam profile reconstruction

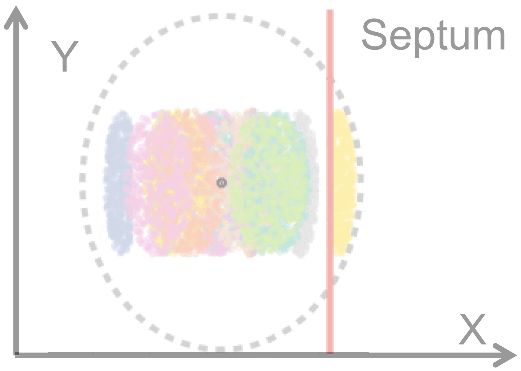


Outline

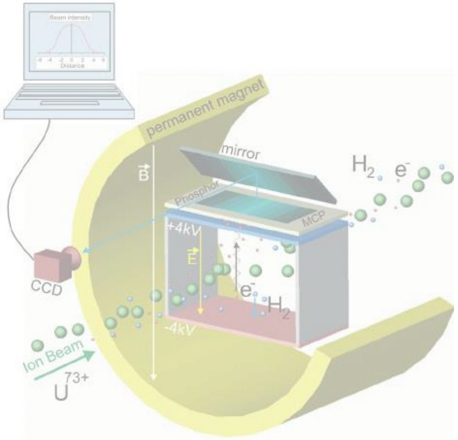
- **Nature-inspired optimization**
  - Evolutionary algorithm
  - Particle swarm optimization

- **Machine Learning**
  - Linear Regression
  - Artificial Neural Networks

- **Example optimization problem:**
  - Multi-Turn Injection



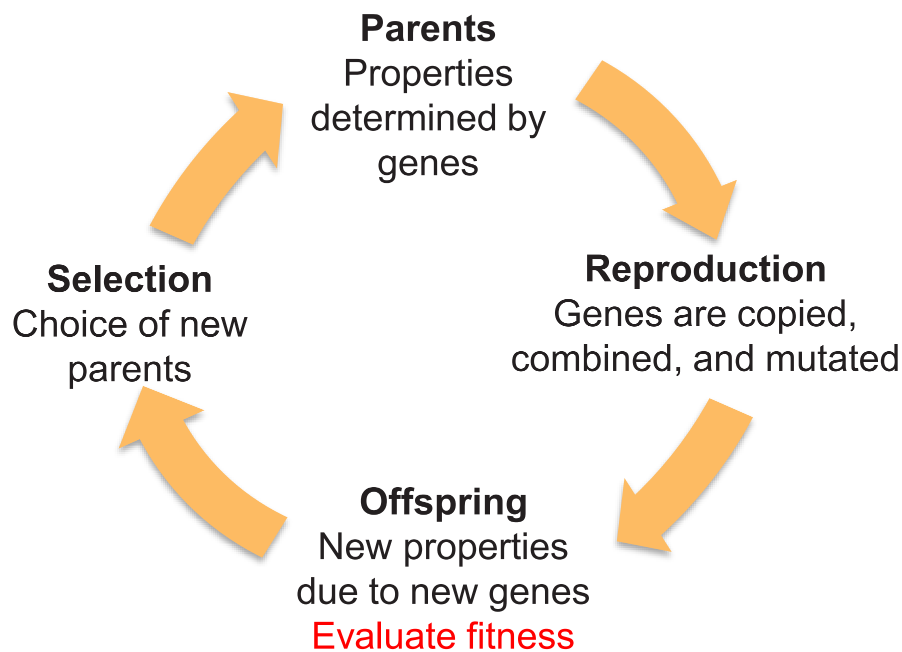
- **Example Machine Learning**
  - Beam profile reconstruction



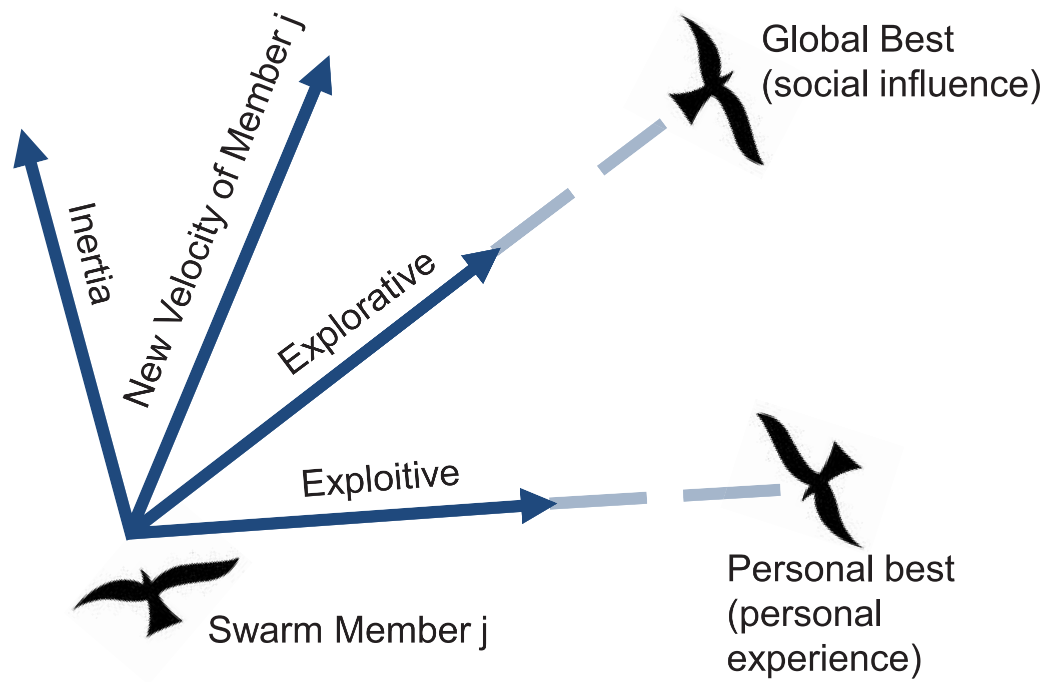
Nature-inspired optimization

- Search for solutions using techniques such as mutation, selection and crossover
- Nature-inspired algorithms are smart parameter scans
- The fitness measures how good an individual is adapted

Genetic algorithms



Particle swarm algorithms

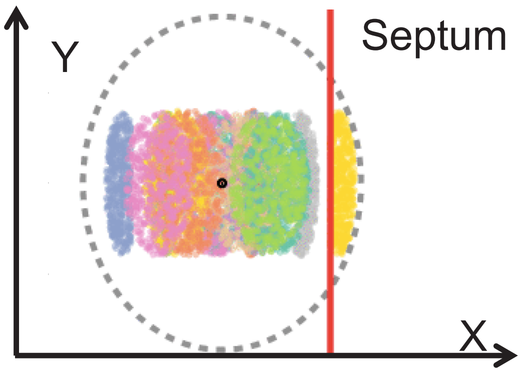


Outline

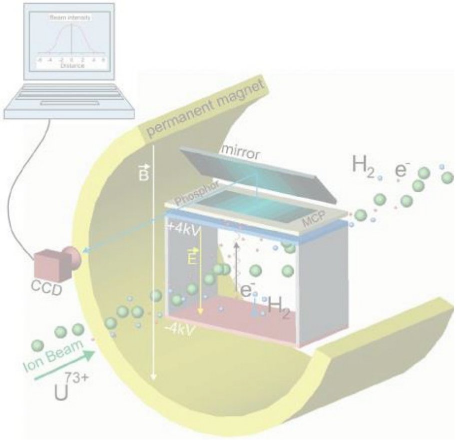
- **Nature-inspired optimization**
  - Evolutionary algorithm
  - Particle swarm optimization

- **Machine Learning**
  - Linear Regression
  - Artificial Neural Networks

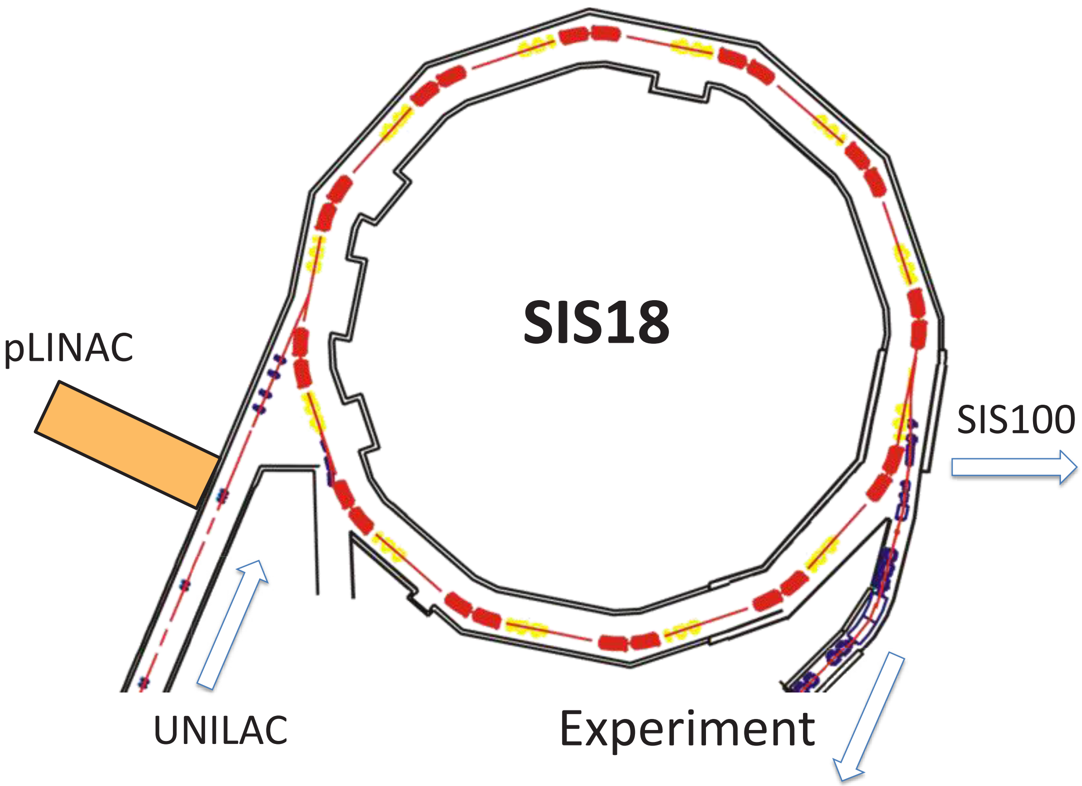
- **Example optimization problem:**
  - Multi-Turn Injection



- **Example Machine Learning**
  - Beam profile reconstruction



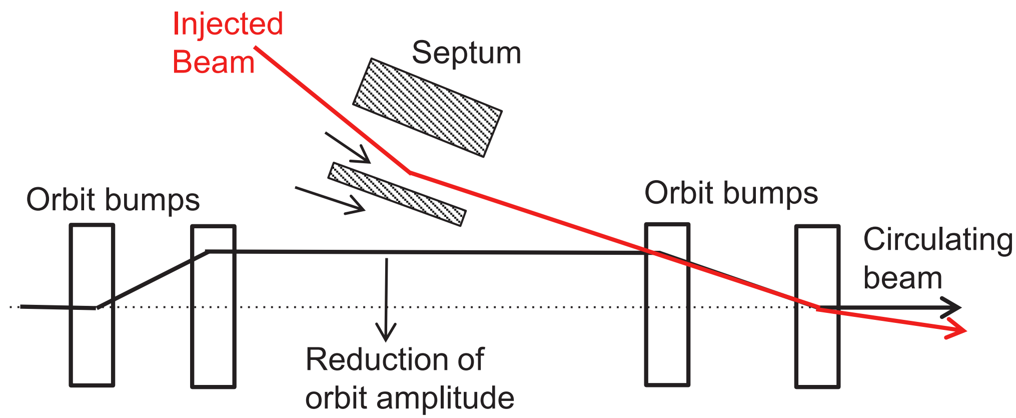
Heavy-ion synchrotron SIS18

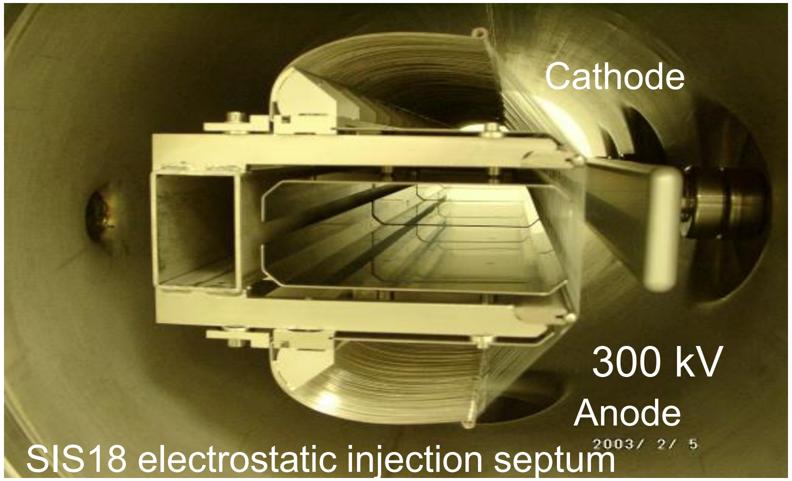


- SIS18 will serve as a booster for SIS100.
- MTI bottleneck to reach intense beams for FAIR.
- Loss-induced vacuum degradation is key intensity-limiting factor.
- Injector upgrade
  - pLINAC: New injector for protons.
  - UNILAC: Replacing of post-stripper section.
- GA optimization has been performed to define interface parameters.

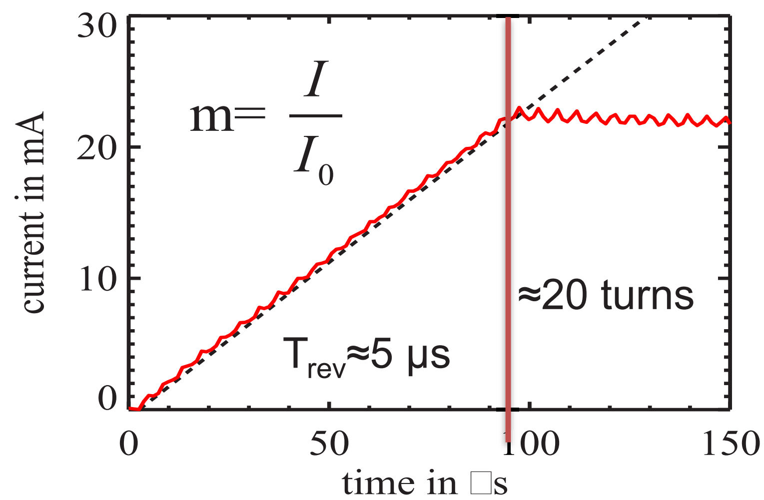
Model: Multi-turn injection

MTI has to respect Liouville's theorem:
Injected beams only in free space





Measured MTI performance in SIS18

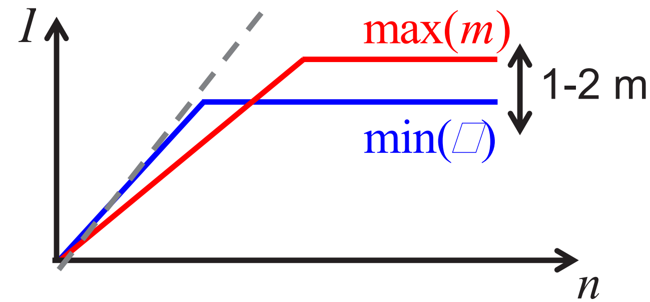


Gain factor should be high as possible $m = \dfrac{I}{I_0}$

Injection loss should be low as possible $\Box = \dfrac{I_{loss}}{n I_0}$

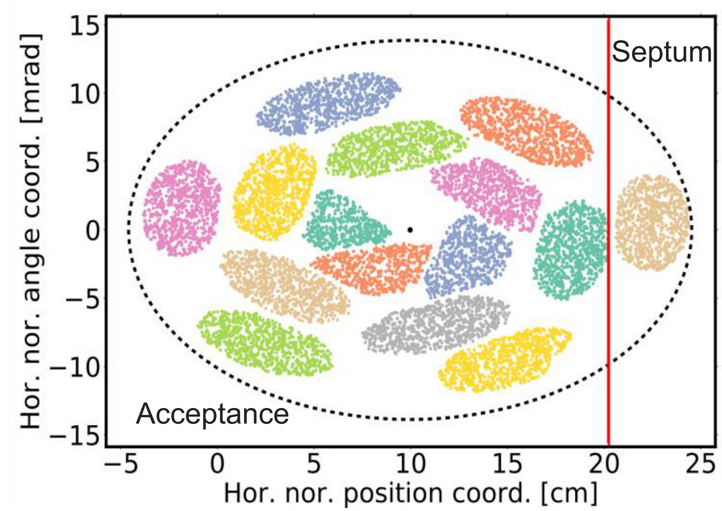
MTI into SIS18: Model

- **Multi-objectives:**
  - Gain factor (maximize) $I = mI_0$
  - Beam loss (minimize) $\square = \frac{I_{loss}}{nI_0}$
  - Emittance ε_x



- **Constraints:**
  - Position of septum, machine acceptance

Model in simulation code



- **Parameters:**
  - Position of incoming beam at septum
  - Initial bump amplitude and its decreasing
  - Injected turns
  - Horizontal tune and emittance

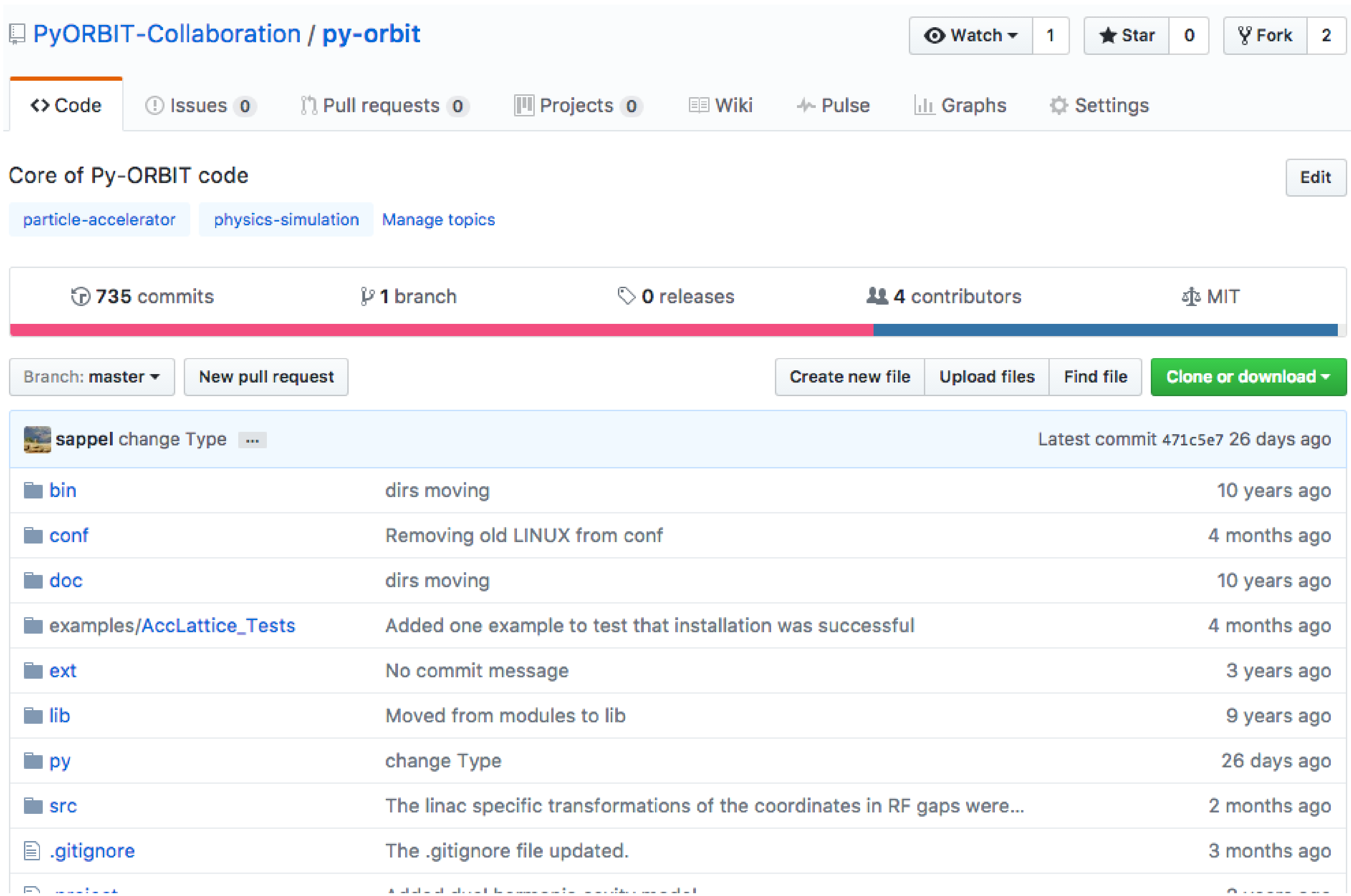
pyorbit

https://github.com/PyORBIT-Collaboration

Open-source hosting

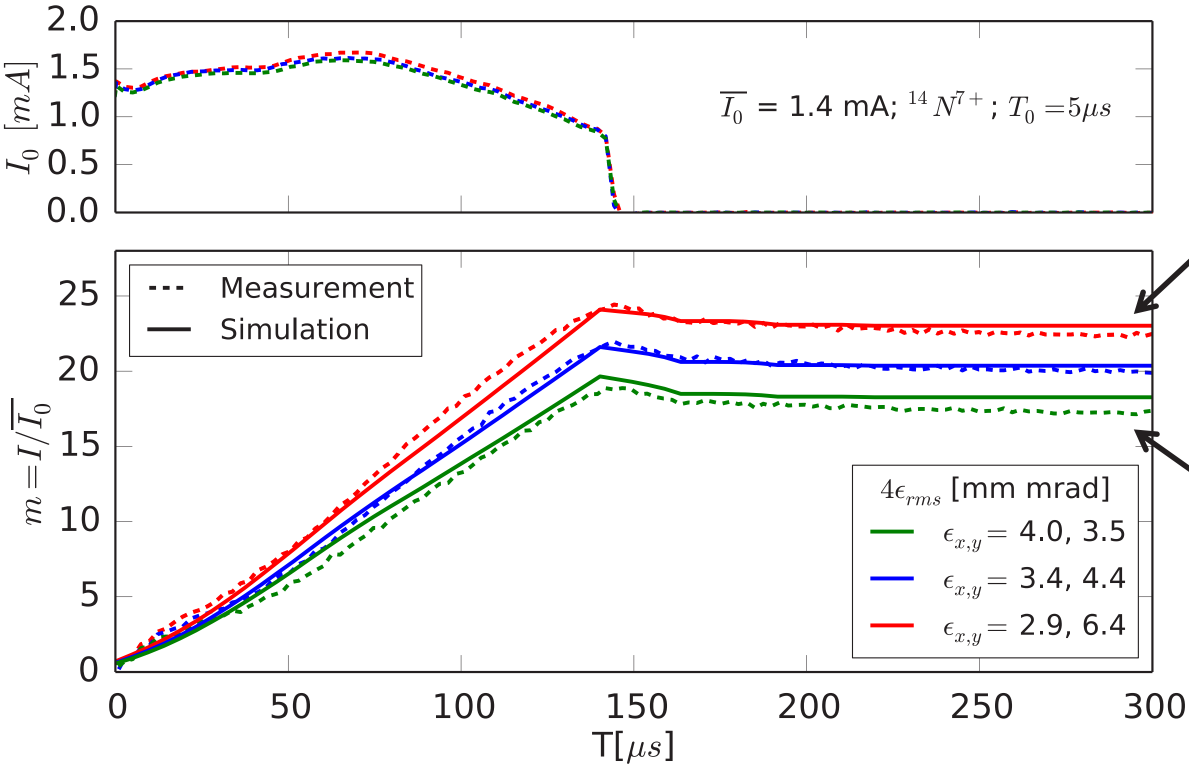
PyORBIT-Collaboration

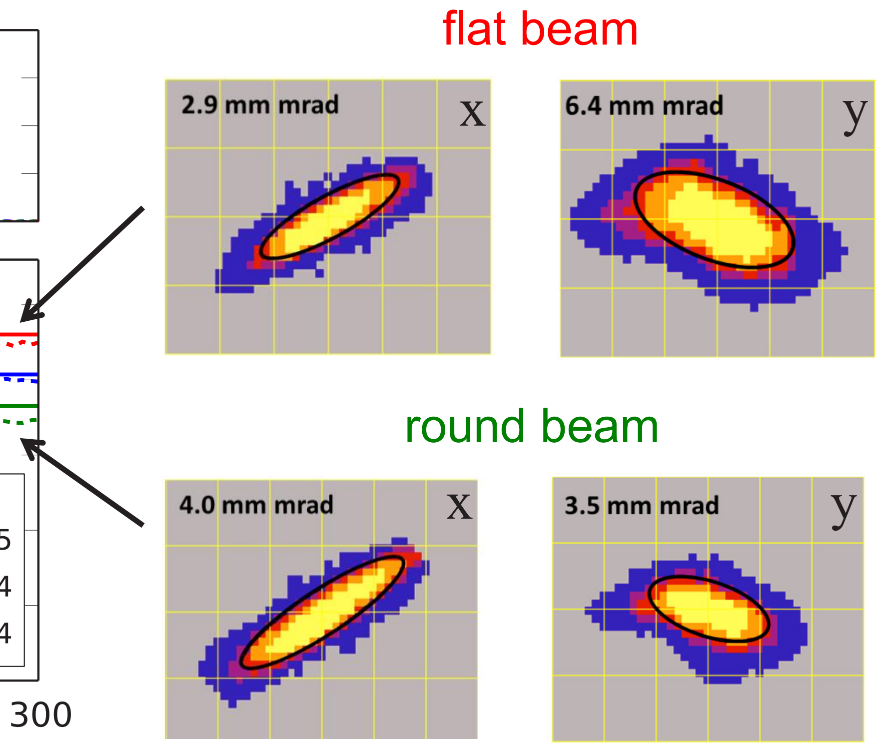
SNS, CERN, GSI, J-PARC



Implementation and validation

MTI performance has been measured as a function of injector emittance.
Round-to-flat transformation with EMTEX Beam line.





Excellent agreement between simulation and measurement!

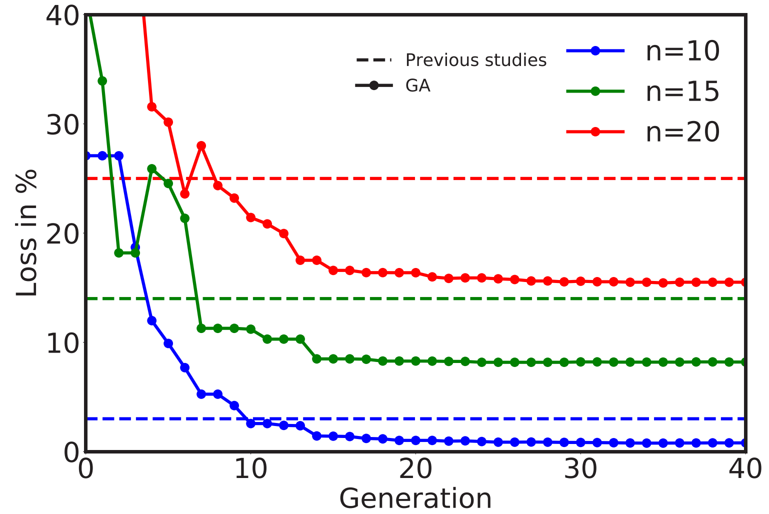
L. Groening et al: Phys. Rev. Lett. 113 264802 (2014), S. Appel et al: Nucl. Instrum. Methods A 866 (2017), pp. 36-39

Optimization results

Optimization of loss

Genetic algorithms can improve MTI.

Especially for longer injection GA discovers a much better solution.

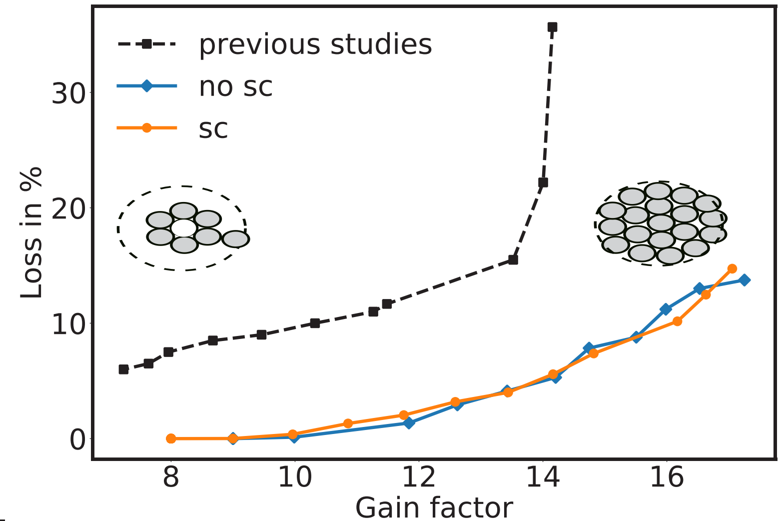


Optimization of loss and gain factor

Dependence of gain factor on loss.

Loss-free injection could be found.

Space charge results in a similar PA front, but with different injection settings.



MOPSA shown similar result with fast convergence.

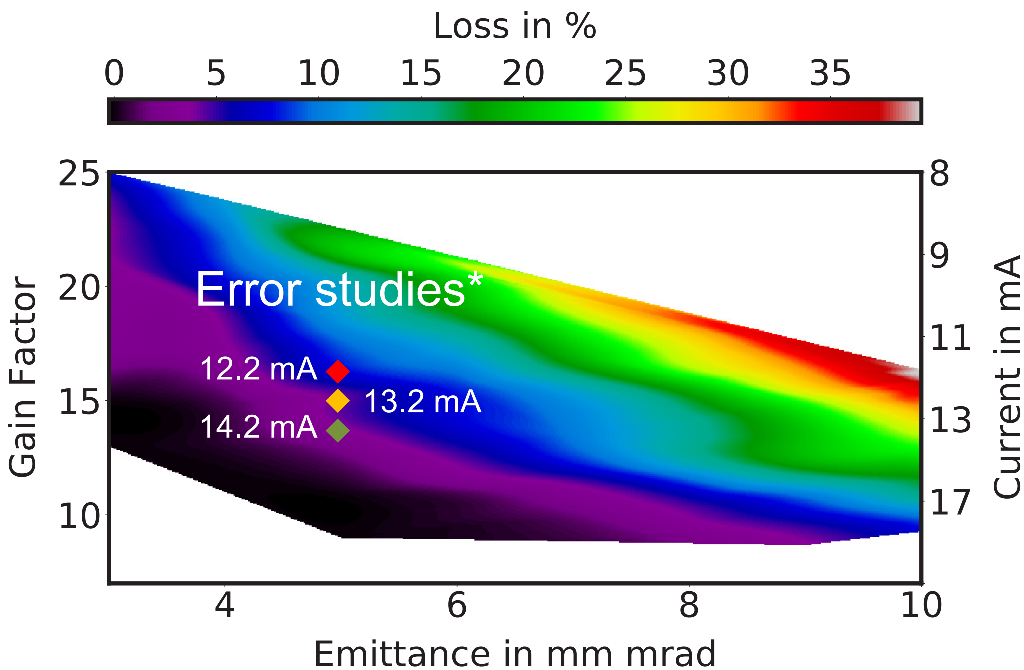
Optimization results

Optimization of loss, gain factor and beam emittance (injector)

Dependence of interface parameter

$$B = \frac{I}{\square} \qquad m(\square) = \frac{N}{I} q f_0$$

allows to define a frame, in which the required beam parameter can be matched at best.

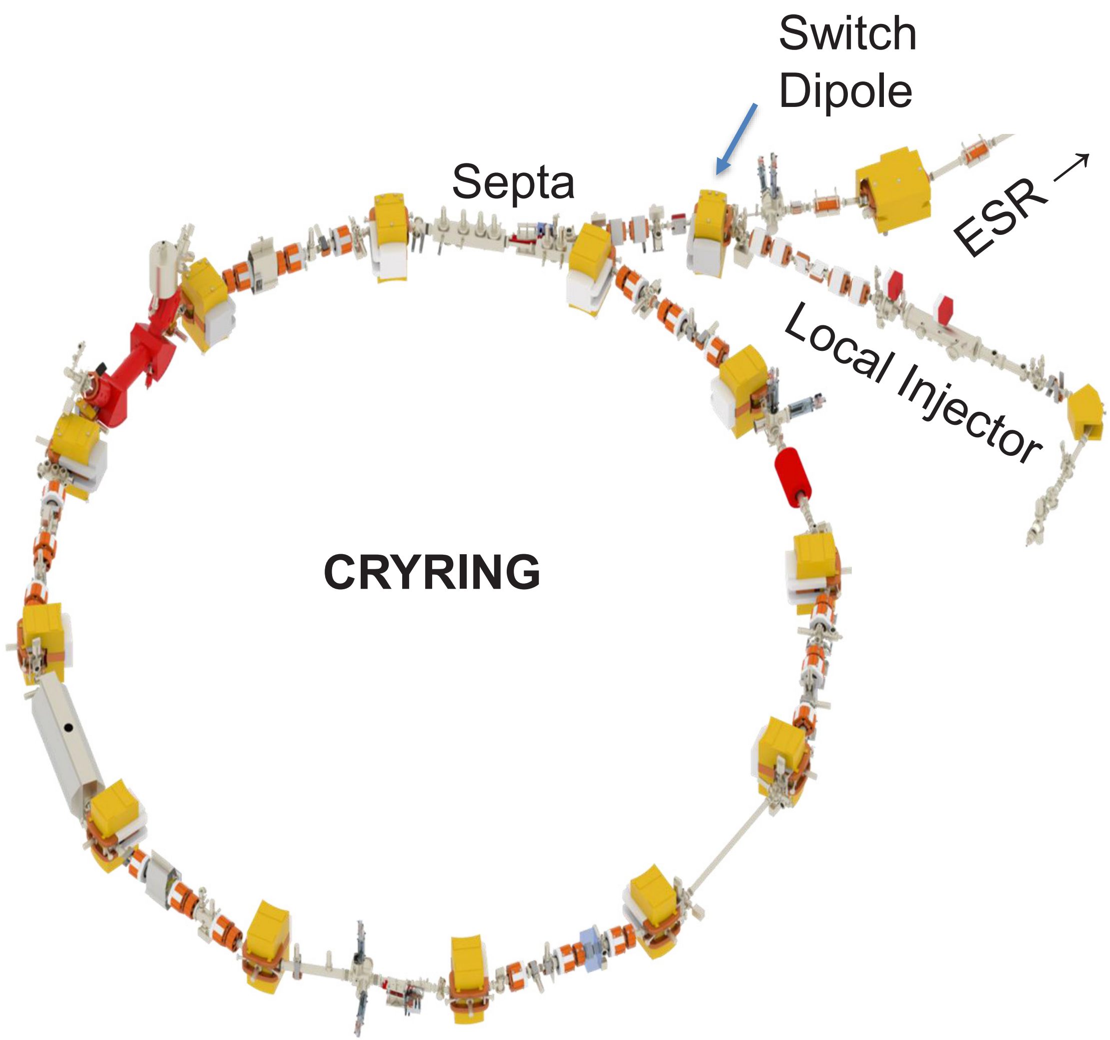


3D Pareto front for proton injector has generated also.
pLINAC: Relaxed situation, generous beam parameter margin

S. Appel et al: Nucl. Instrum. Methods A 852 (2017), pp. 73-79
C. Kleffner, LINAC2018, THPO046 (2018)

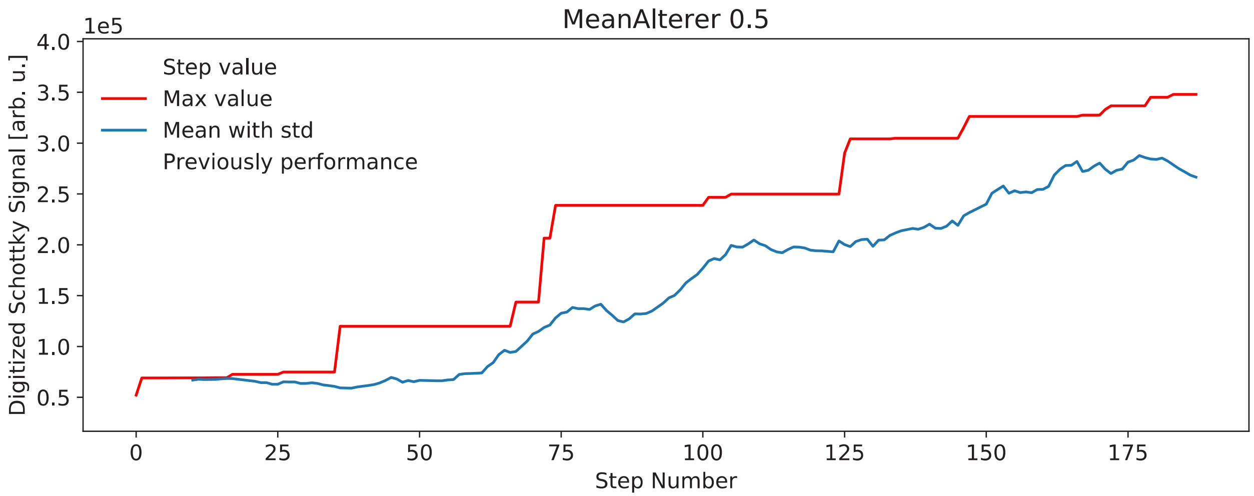
*A. Rubin, Beam dynamics design of the new FAIR post-stripper linac, GSI Accelerator Seminar, 14.05.17

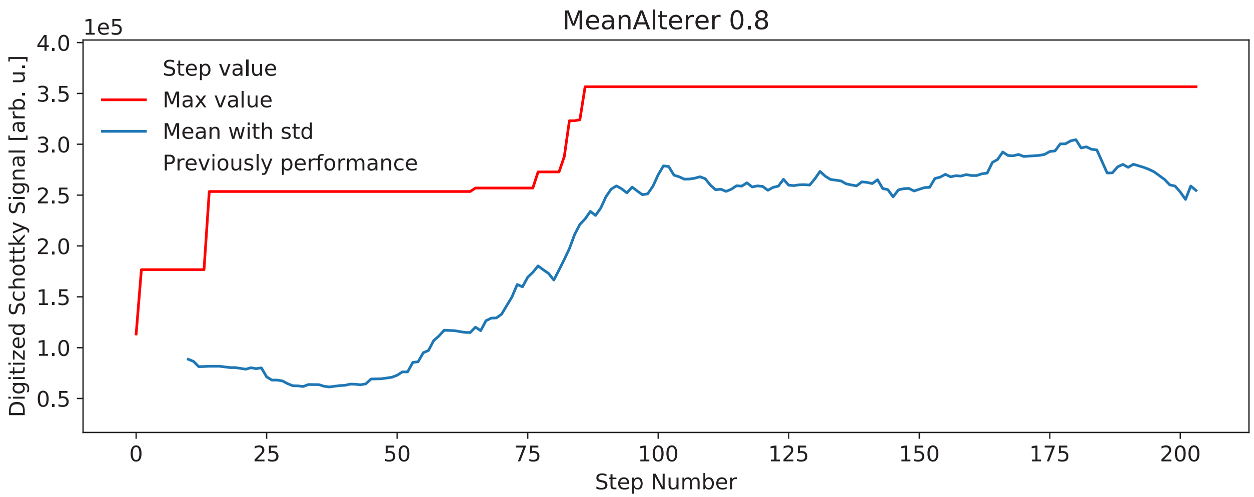
CRYRING@ESR



- Swedish in-kind contribution to FAIR
- CRYRING@ESR can be used stand-alone for testing novel technical developments.
- Control system is Java based.
- Jenetics end-user ready software library implementing an genetic algorithm in Java.
- Choice to use Jenetics was obvious although faster algorithm are known.

CRYRING@ESR: Online optimization





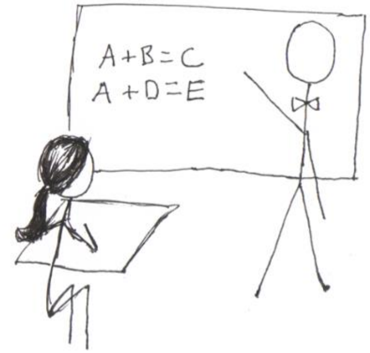
Large tournament size has chosen to reach fast convergence.

~ 90 minutes

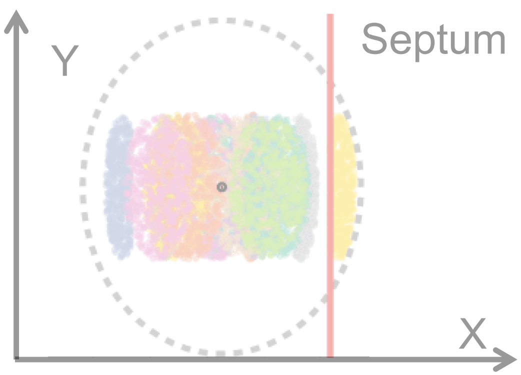
Outline

- **Nature-inspired optimization**
  - Evolutionary algorithm
  - Particle swarm optimization

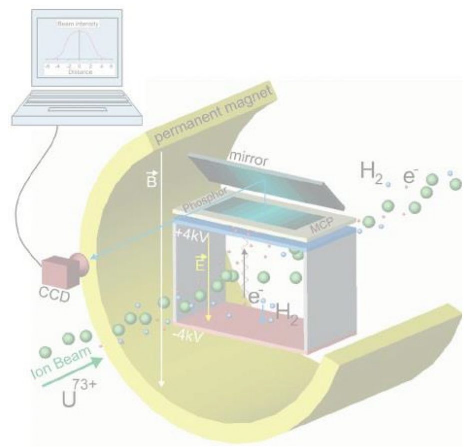
- **Machine Learning**
  - Linear Regression
  - Artificial Neural Networks



- **Example optimization problem:**
  - Multi-Turn Injection

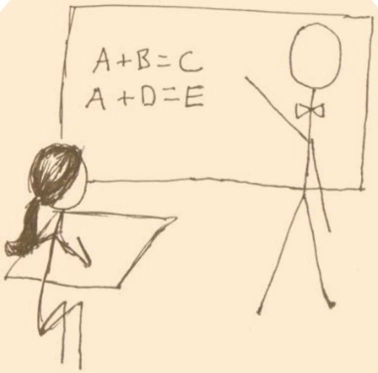


- **Example Machine Learning**
  - Beam profile reconstruction



Machine Learning

Supervised Learning



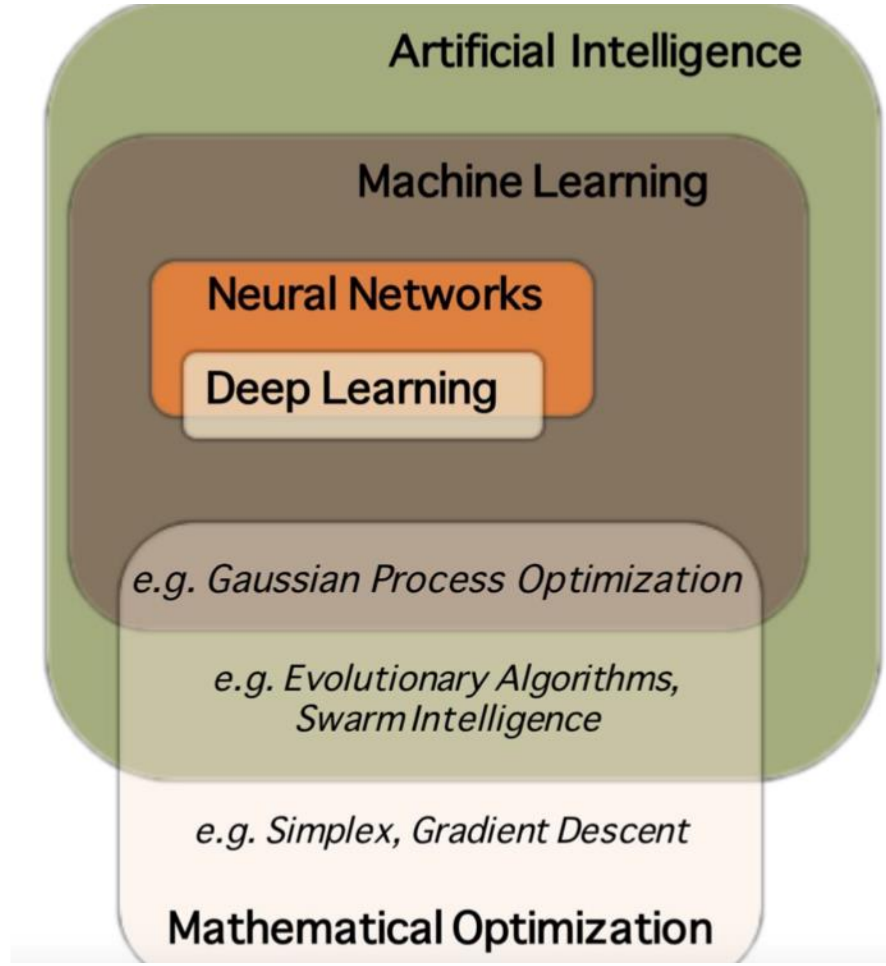
learn known input/ output pairs

Unsupervised Learning

no labeled data -> infer structure

Reinforcement Learning

interact with the environment -> adjust behavior based on reaction



Source: Auralee Edelen, ICFA Workshop on ML for Particle Accelerators, SLAC, 27.02 - 02.03.2018

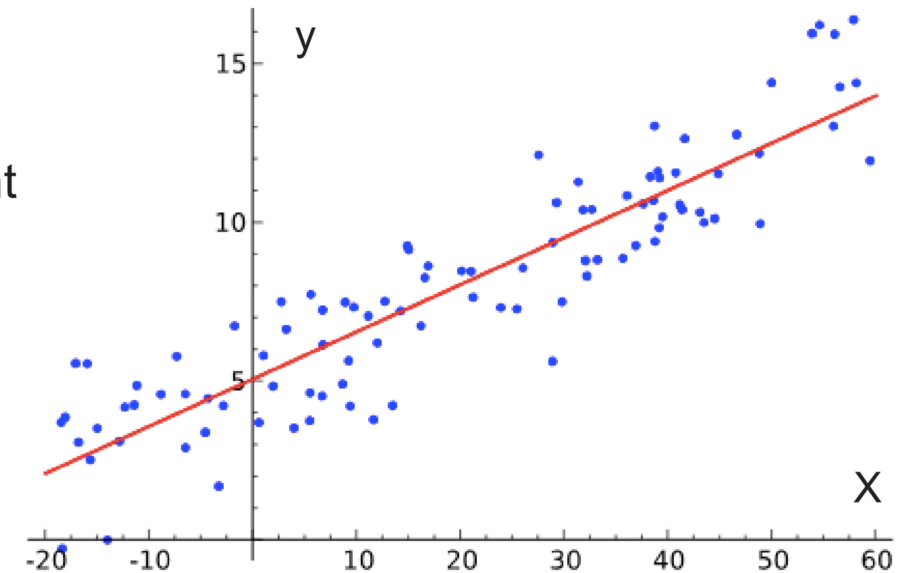
Machine Learning

Machine Learning – algorithms which can learn and make predictions on data, without explicit programming.

ML covers many different algorithms with varying complexity from linear approximation to biologically inspired Artificial Neural Networks.

- Linear approach modelling.
- Relationship between scalar dependent variable and explanatory variables.

$$y = W^T x + b$$



Least squares approach is often used for fitting linear regression models.

Artificial Neural Network

- Biologically inspired.
- The perceptron is a simplified model of a biological neuron.

Perceptron parameters:

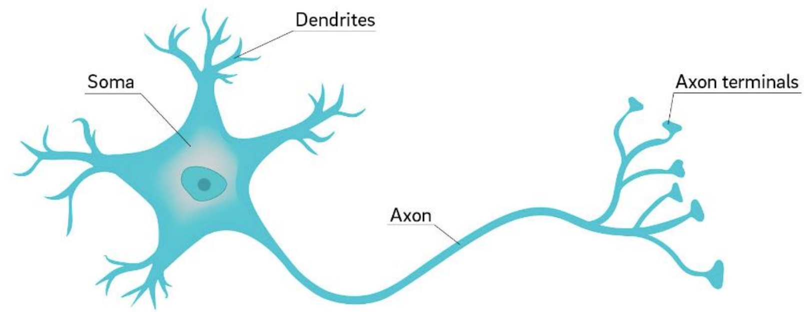
- Weights from the inputs (X) and bias (b).
- g is the activation function, a step-like function with a threshold.

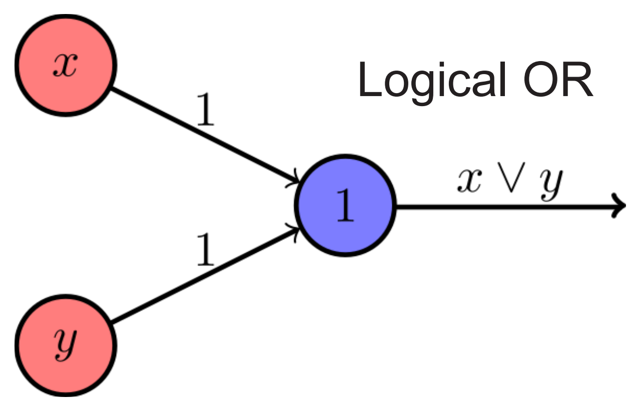
Output

$$o = g\left(\sum_{k=0}^{N} x_k W_k + b\right)$$

Neuron

David Baillot/ UC San Diego

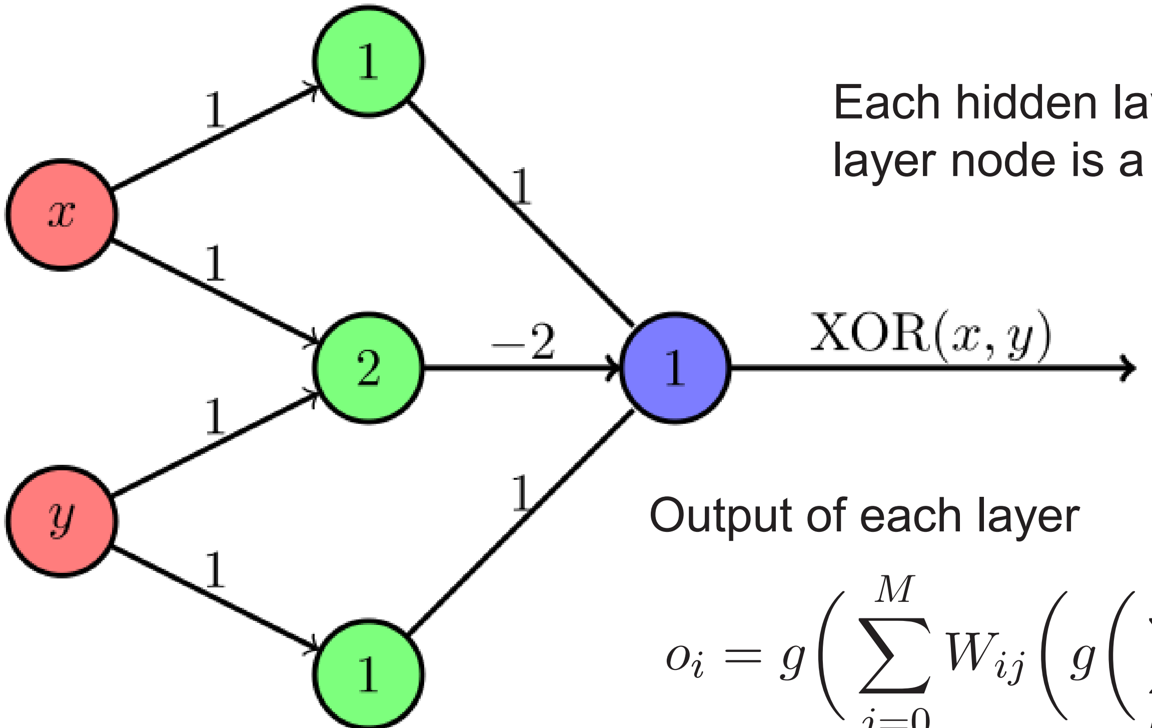




MartinThoma, Wikipedia

Artificial Neural Network

MartinThoma, Wikipedia



Each hidden layer and output layer node is a perceptron.

Output of each layer

$$o_i = g\left(\sum_{j=0}^{M} W_{ij}\left(g\left(\sum_{k=0}^{N} x_k W_{jk} + b_j\right)\right) + b_i\right)$$

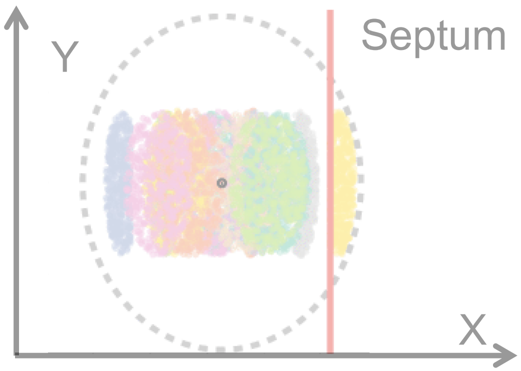
Adding “hidden” layer(s) allow non-linear target functions to be represented

Outline

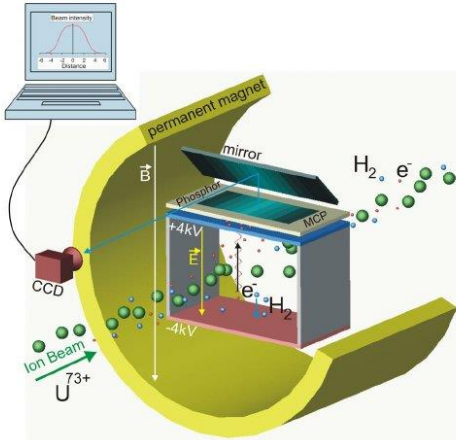
- **Nature-inspired optimization**
  - Evolutionary algorithm
  - Particle swarm optimization

- **Machine Learning**
  - Linear Regression
  - Artificial Neural Networks

- **Example optimization problem:**
  - Multi-Turn Injection

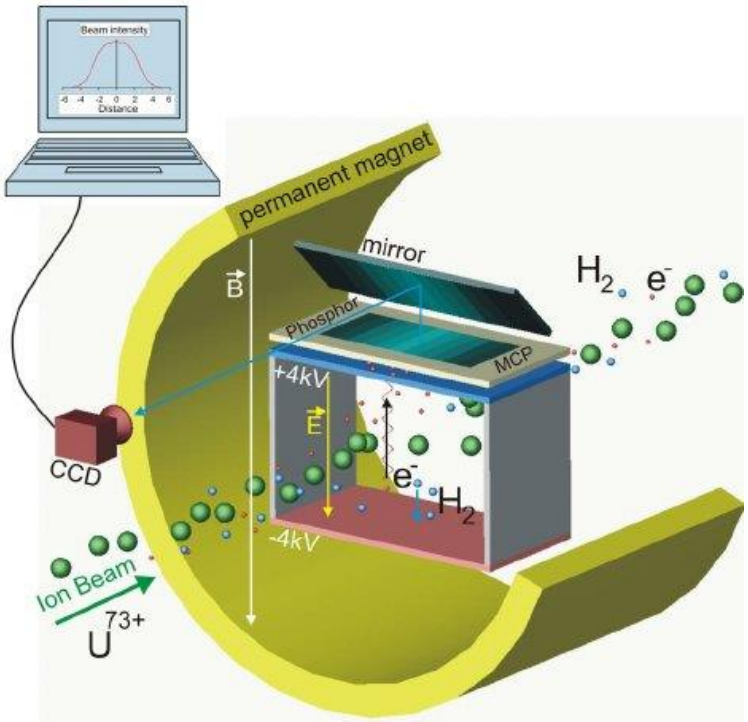


- **Example Machine Learning**
  - Beam profile reconstruction



IPM (Ionization Profile Monitors)

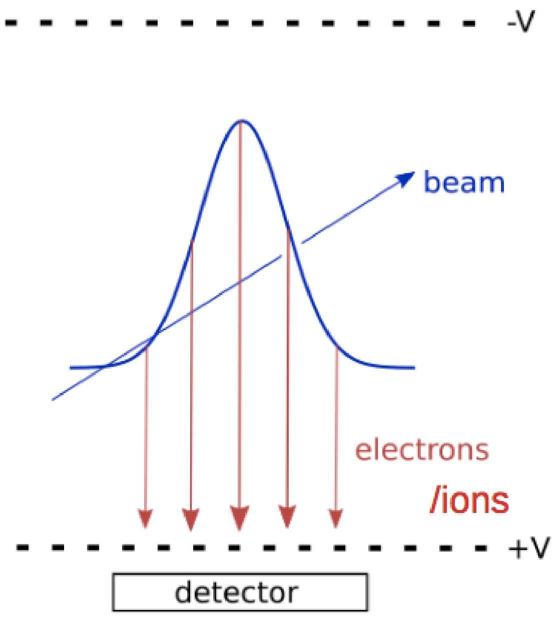
- For optimization and control knowledge of beam parameters is a key ingredient.
- IPM has been constructed first in Argonne National Laboratory in 1967.
- Measures transverse profile of a particle beam.
- Rest gas (pressure 10^{-8} mbar) is ionized by the beam.
- Electric field is used to transport electrons/ions to a detector.
- If electrons are used – additional magnetic field is usually applied to confine their movement.



Profile Distorsion in IPM

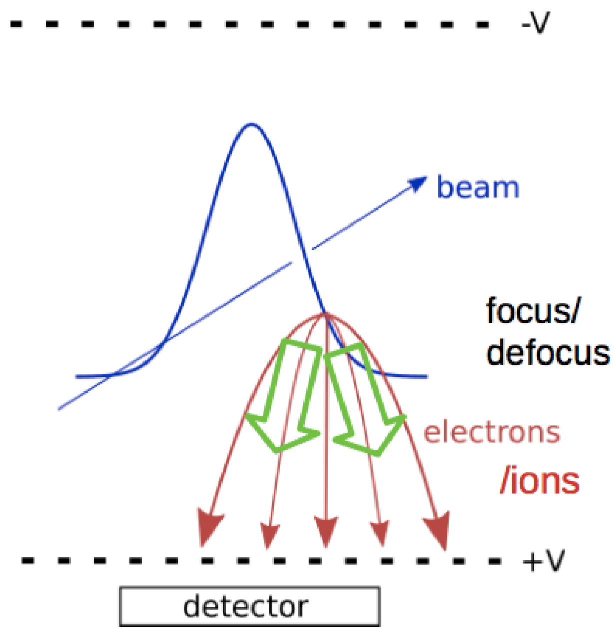
Ideal case

- Particles are moving on straight lines towards the detector



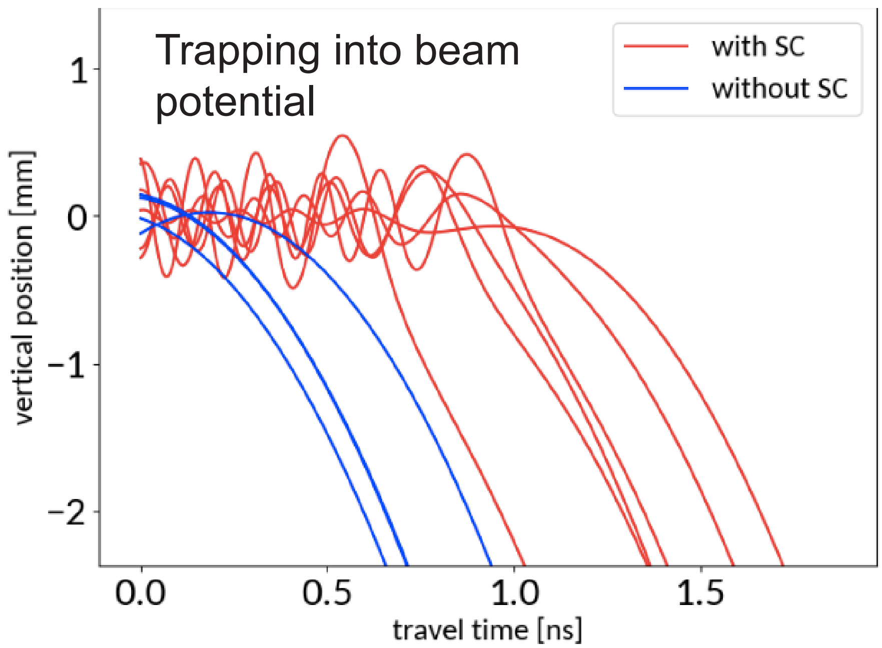
Real case

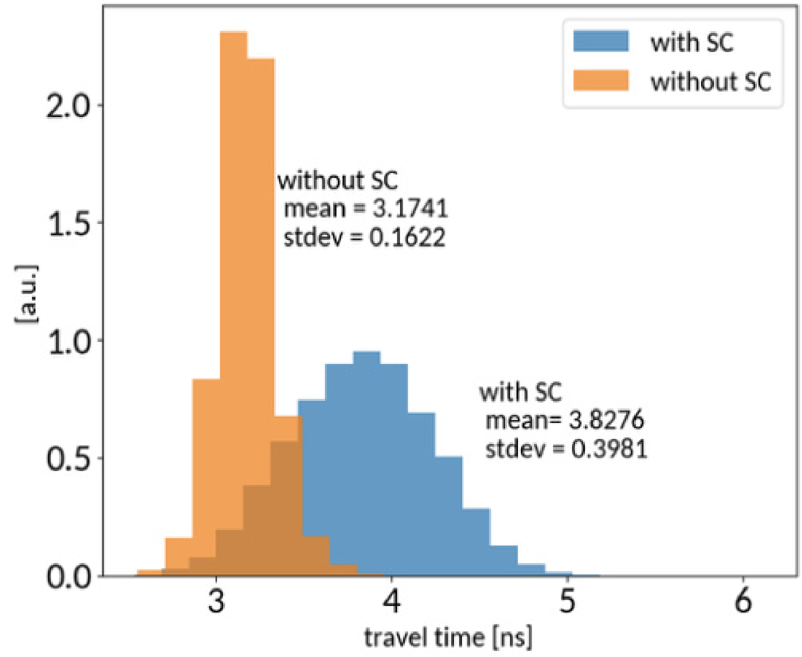
- Particle trajectories are influenced by initial momenta and by the interaction with the beam field



... instrumental effects come on top!

Profile Distorsion in IPM

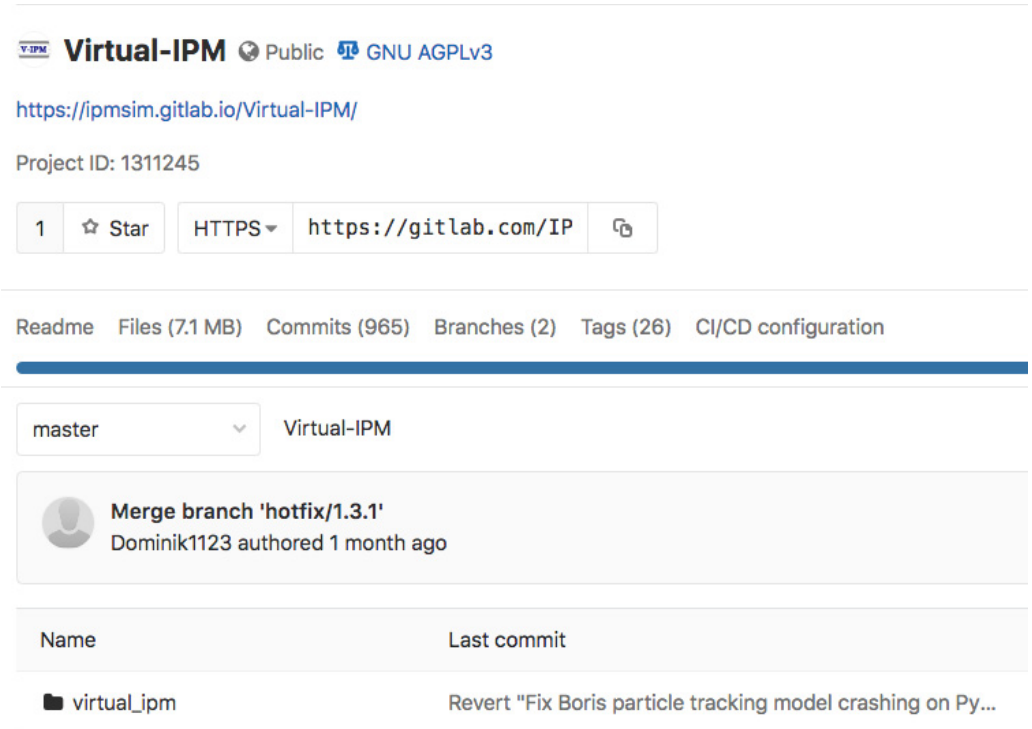




Electrons are trapped in bunch field for the time when bunch passes.
They make several oscillations around bunch center. Complex movement!

Several attempts have been made to correct or describe such effects,
but no sufficient analytic procedure was found yet.

Virtual-IPM program



Open-source hosting

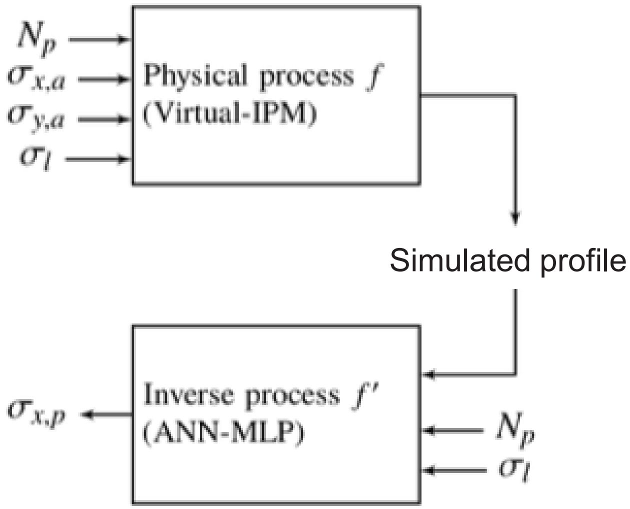
Code on gitlab:
https://gitlab.com/IPMsim/Virtual-IPM

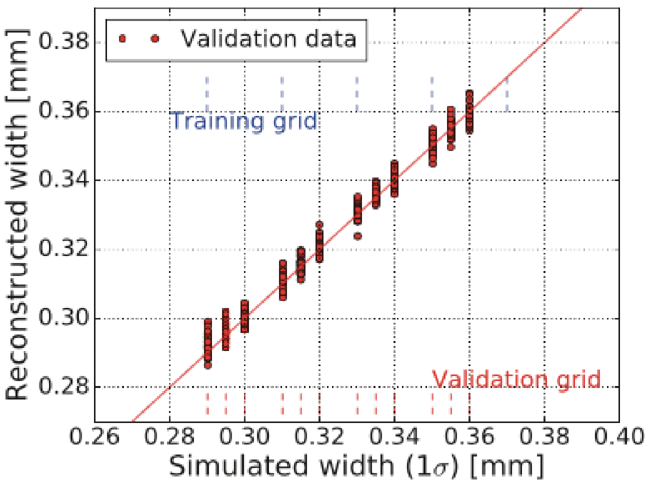
Available as python module:
https://pypi.org/project/virtual-ipm

- After looking for a proper program: Decision to write Virtual-IPM.
- Written in Python with modern, modular architecture.
- Covers: IPM, BIF, gas jets.

Profile correction using ANN

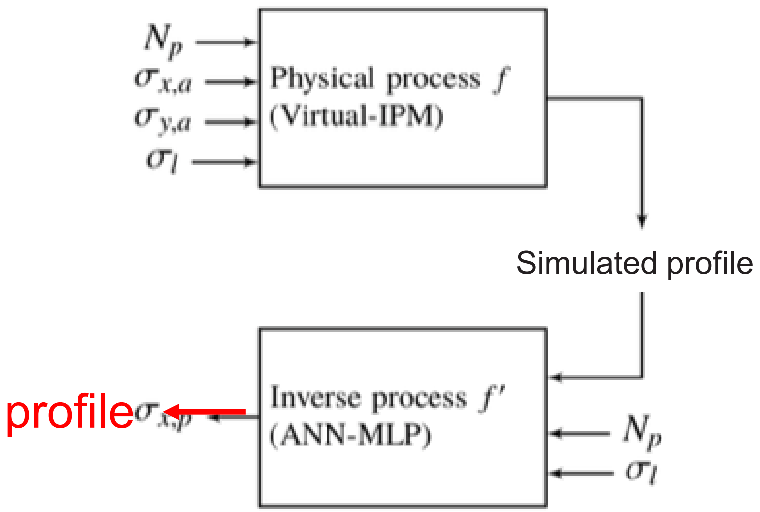
R. Singh, et al, Proc. of IBIC17 (WEPCC06)

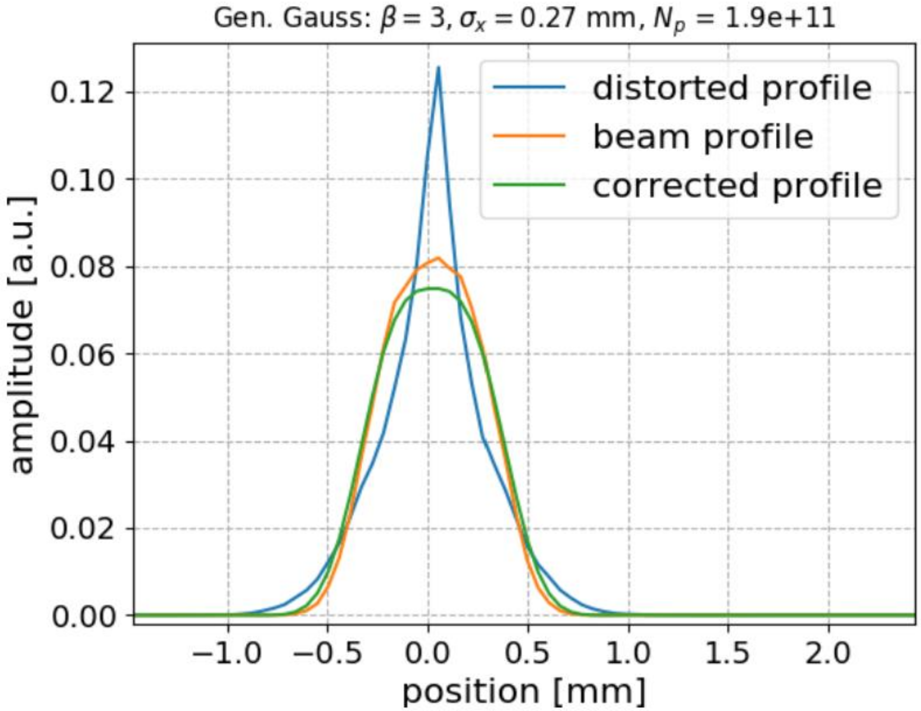




- Virtual-IPM was used for simulating the movement of electrons for a typical LHC case.
- Value of beam size restored with 1% accuracy!
- Good performance with noise.
- Even simple linear regression model showed very promising results for beam width reconstruction.

Profile correction using ANN





Results for Gaussian profiles: Very good profile shape reconstruction.

SIS100: Profile distortion for some beams a profile distortion is expected to be visible and will require a similar correction procedure."

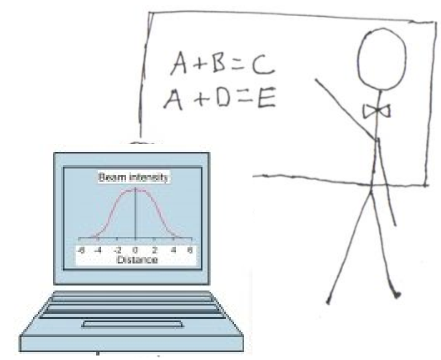
M. Sapinski et al., in Proc. of HB2018, THA2WE02.

Summary

- **Nature-inspired optimization**

  - Multi-object optimization: Identification of injector brilliance range.
  - Reach after ~1.5 hours of online optimization time previous transmission.
  - Potential to reduce the manpower requirements.

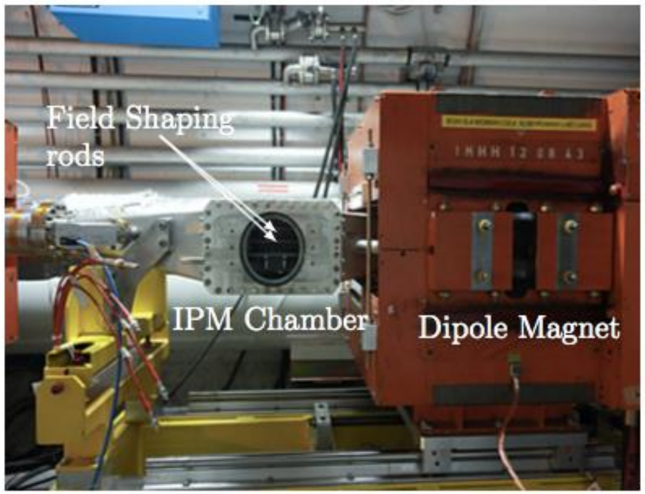
- **Machine Learning**
  - First investigations, using simulated data, yield promising results.
  - Method has a potential to extend usability and reduce cost of IPMs for high brightness beams.
  - The application of machine learning to longitudinal Schottky signals is under investigation.

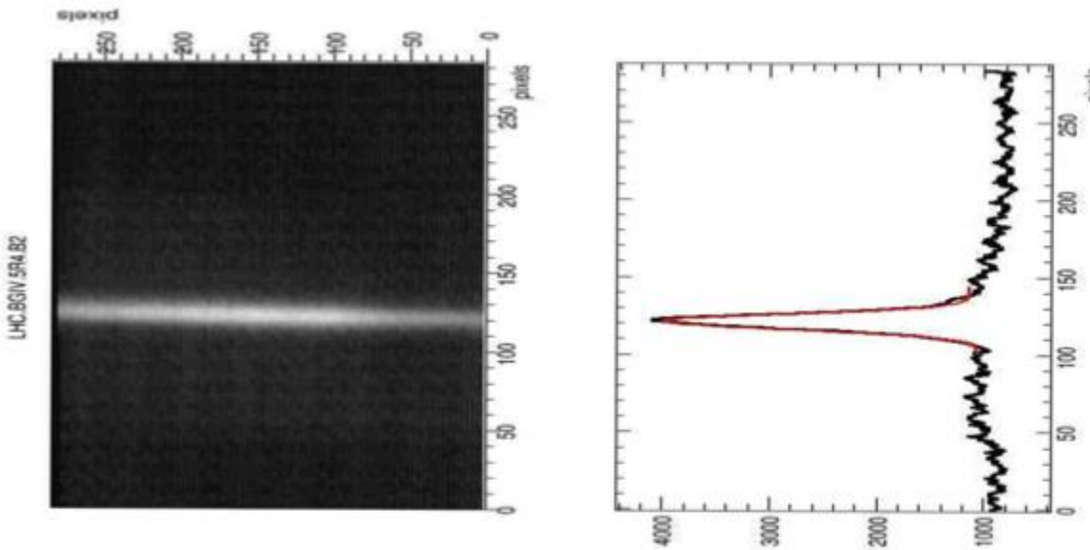


Thank you for your attention

IPM (Ionization Profile Monitors)

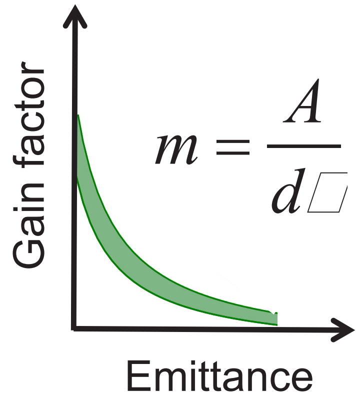
IPM installation at LHC





Injector brilliance depending

EMittance Transfer EXperiment (EMTEX)

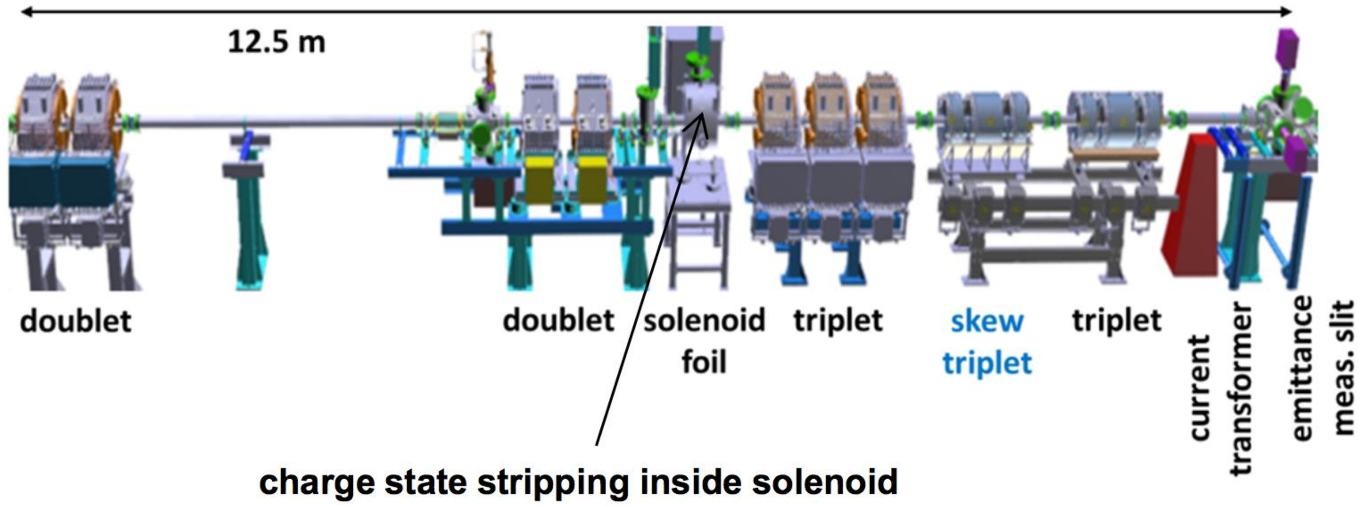


Re-partitioning of beam emittances increase efficiency

Beam flatness amount is controlled by solenoid field

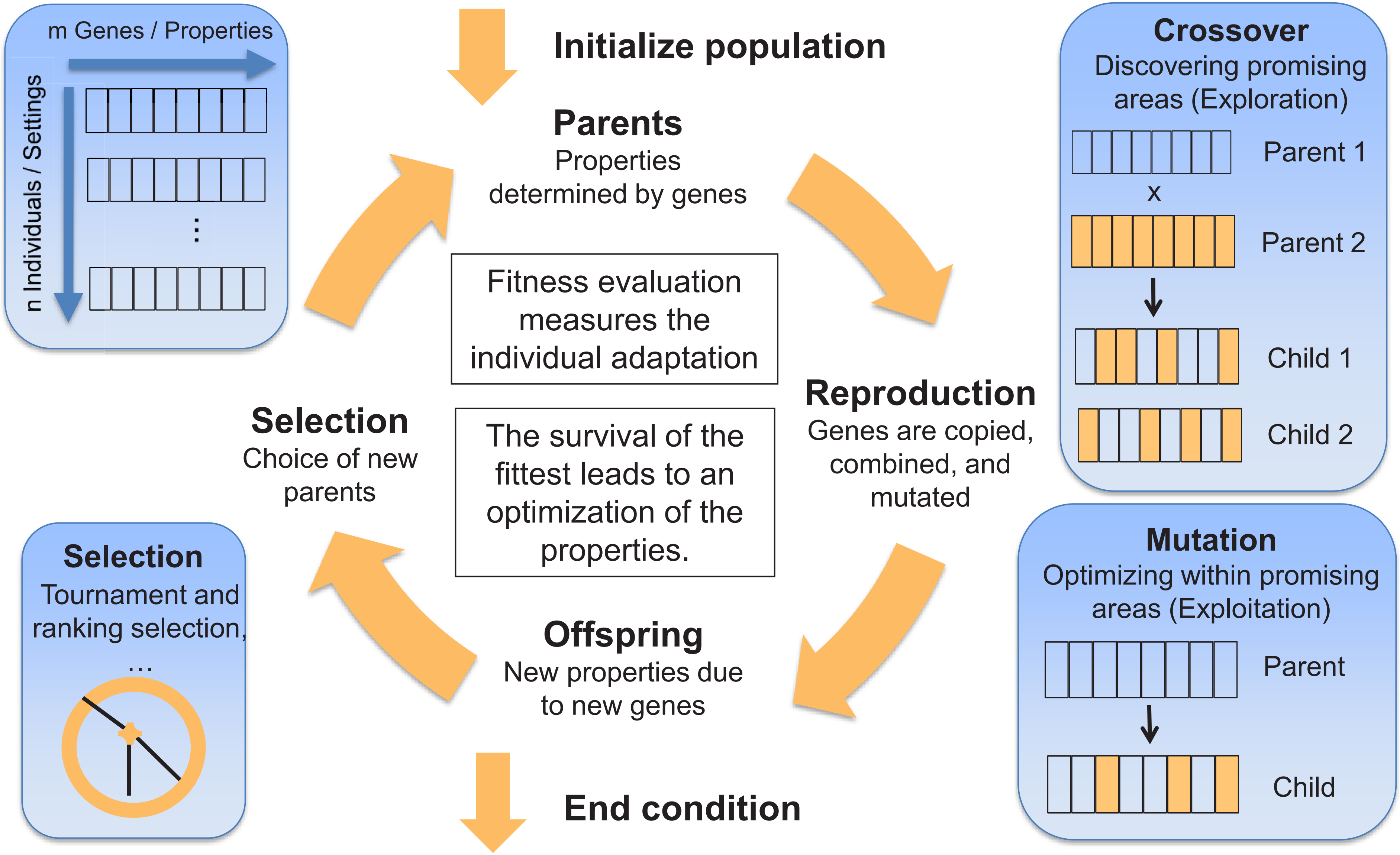
Twiss-parameters are preserved

EMTEX Beam line



L. Groening: Phys. Rev. ST Accel. Beams 14 064201 (2011)
C. Xiao et al: Phys. Rev. ST Accel. Beams 16 044201 (2013)
L. Groening et al: Phys. Rev. Lett. 113 264802 (2014)
S. Appel et al: Nucl. Instrum. Methods A 866 (2017), pp. 36-39

Genetic Algorithms



Particle swarm algorithms

Inspiration from the "graceful but unpredictable choreography of a bird flock"

Position

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

Velocity update

$$v_i(t+1) = wv_i(t) + r_1 C_1 (P_i^l - x_i) + r_2 C_2 (P^g - x_i)$$

Inertia   Local search   Global search

x_i Each individual particle position refers to a point in the variable space

w Inertia weight reflects effect of particle current motion

P_i^l Personal best; analogous to "nostalgia"

C_1 Cognitive parameter is contribution of particle personal experience

P^g Global best is the best position ever for entire swarm

C_2 Social parameter reflects publicized knowledge or social norms

r_1, r_2 Stochastic elements of the algorithm