A Parallel General Purpose Multi-Objective Optimization Framework, Applied to Beam Dynamic Studies

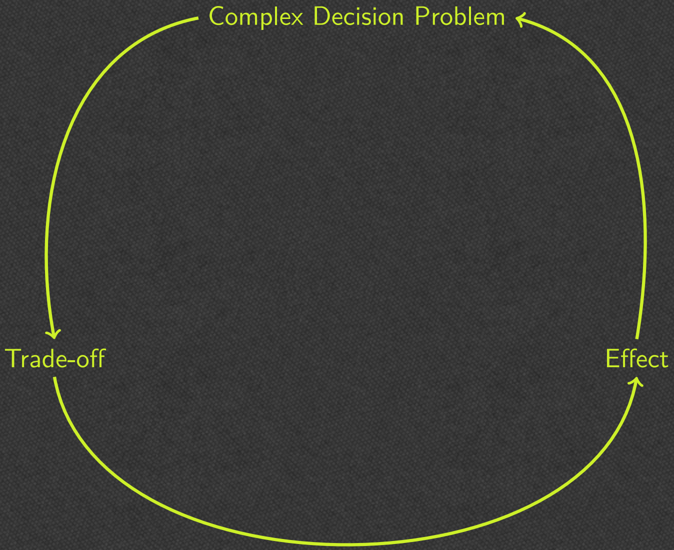
Yves Ineichen[1,2,3], Andreas Adelmann[2], Costas Bekas[3], Alessandro Curioni[3], Peter Arbenz[1]

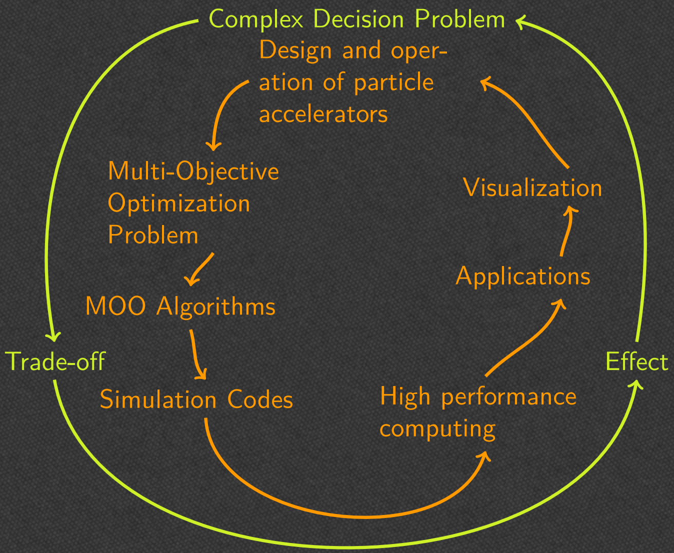
[1]ETH Zürich
Department of Computer Science
CH-8092 Zürich, Switzerland

[2]Paul Scherrer Institut
Accelerator Modelling and Advanced Simulations
CH-5234 Villigen, Switzerland

[3]IBM Research-Zurich
CH-8803 Rüschlikon, Switzerland

21st August 2012


Complex Decision Problem
Trade-off
Effect


Complex Decision Problem
Design and operation of particle accelerators
Multi-Objective Optimization Problem
MOO Algorithms
Simulation Codes
High performance computing
Applications
Visualization
Trade-off
Effect

Multi-Objective Optimization

[a short introduction]

Multi-Objective Optimization Problem

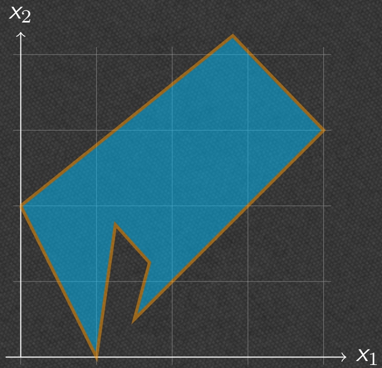
Objectives

Design variables

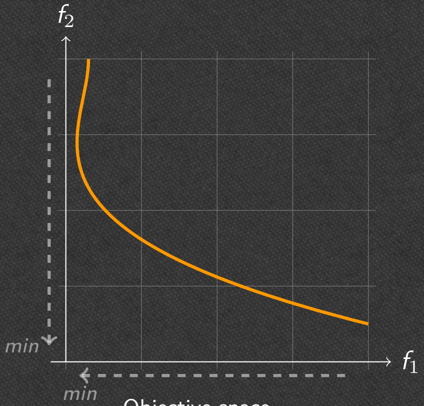
$$
\begin{aligned}
\text{min} \quad & f_m(\mathbf{x}), & m = 1 \dots M \\
\text{s.t.} \quad & g_j(\mathbf{x}) \geq 0, & j = 0 \dots J \\
& x_i^L \leq \mathbf{x} = x_i \leq x_i^U, & i = 0 \dots n
\end{aligned}
$$

Constraints

Mapping design to objective space

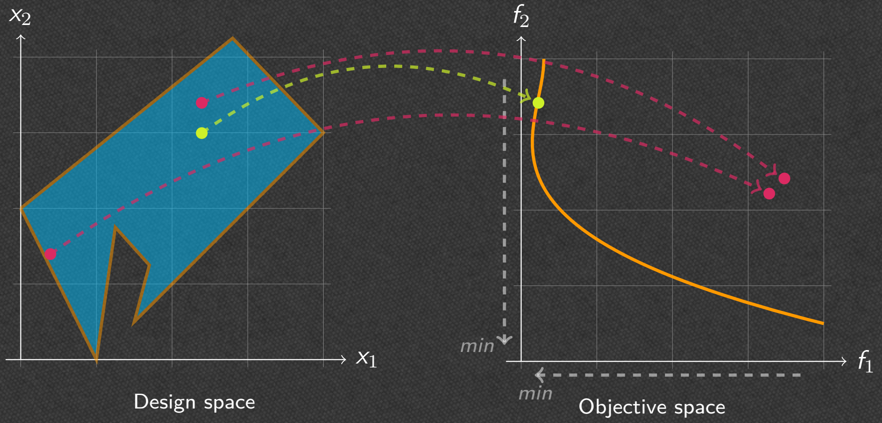


Design space

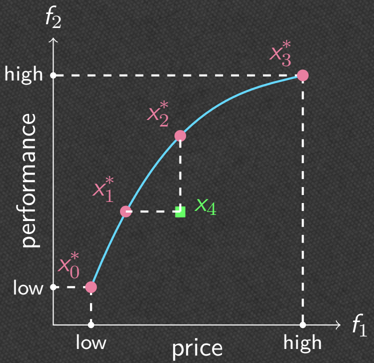


Objective space

Mapping design to objective space

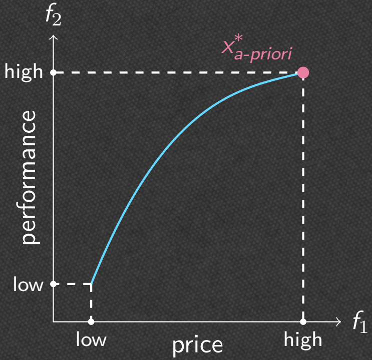


Optimality?



- conflicting objectives:
  minimize price
  maximize performance
- red points are "equally optimal": cannot improve one point without hurting at least one other solution
  → Pareto optimality
- blue curve is called Pareto front
- x_4 is dominated by x_1^* and x_2^*

Preference Specification



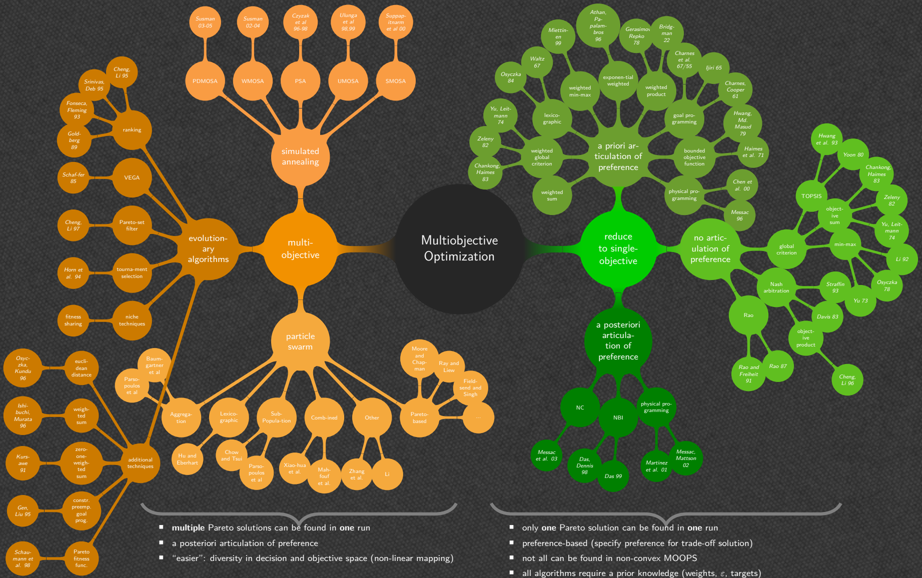
a-priori preference:
e.g. performance $\gg$ price $\rightarrow x^*_{a\text{-}priori}$

a-posteriori preference: $\rightarrow$ Pareto front

- provides deeper understanding of solution space
- visualizes how choice affects design space

Evolutionary Algorithms

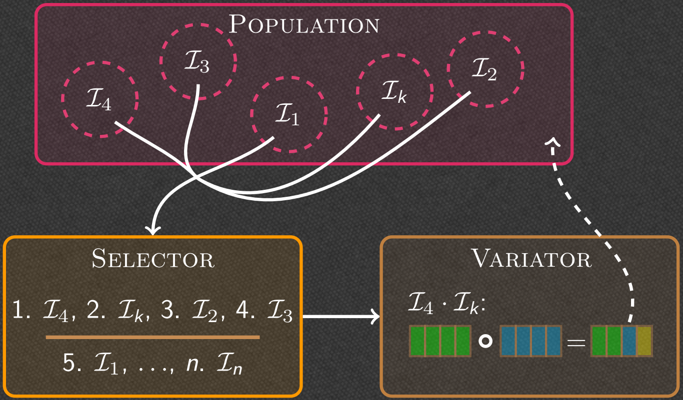
[how can we solve multi-objective optimization problems?]



- **multiple** Pareto solutions can be found in **one** run
- a posteriori articulation of preference
- "easier": diversity in decision and objective space (non-linear mapping)

- only **one** Pareto solution can be found in **one** run
- preference-based (specify preference for trade-off solution)
- not all can be found in non-convex MOOPS
- all algorithms require a prior knowledge (weights, ε, targets)

Evolutionary Algorithms



Ranking individuals

Non-dominated sorting genetic algorithm (NSGA-II) *initialization*:

- count how many solutions n_p dominate solution p
- store all solutions p dominates in set S_p
- set $k \leftarrow 0$

Repeat while there exists solutions with $n_p > 0$:

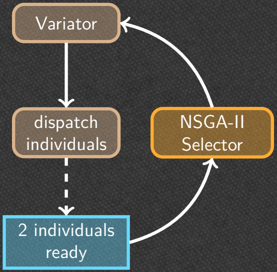
- for all solutions p with $n_p = 0$:
  - store solution in k-th non-dominated front
  - visit all members i of S_p and reduce n_i by one
- $k \leftarrow k + 1$

➡ Order relation corresponds to index in set of non-dominated fronts

---

A fast and elitist multiobjective genetic algorithm: NSGA-II, K. Deb et. al., IEEE Transactions on Evolutionary Computation, 6(2):182–197, Apr. 2002.

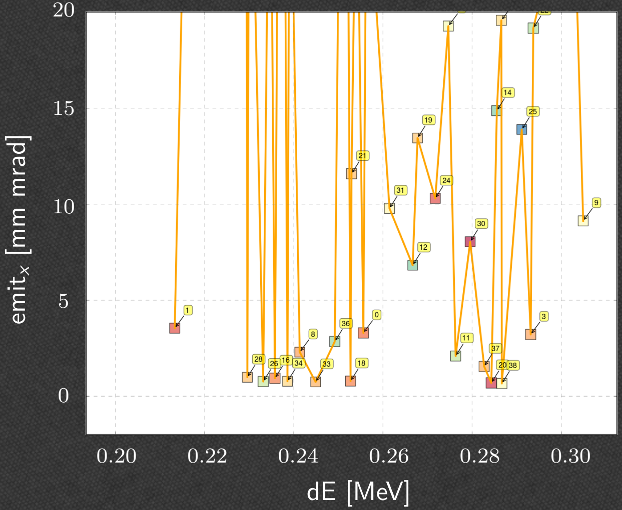
Evolutionary Algorithms



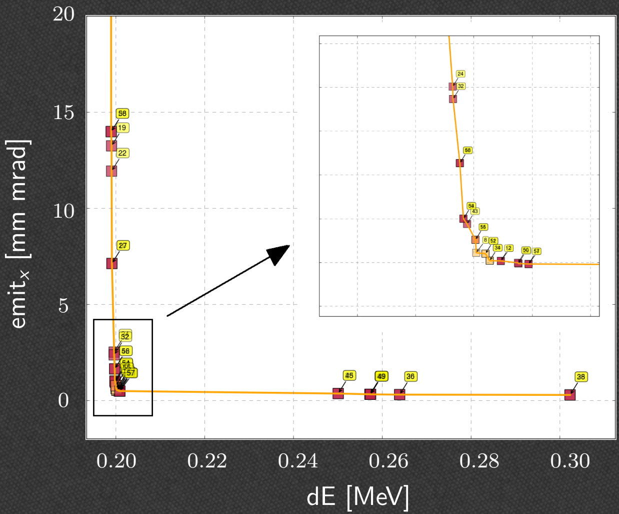
- PISA
  - Finite state machine
  - NSGA-II selector
  - access to many other selectors
- "Continuous generations"
- Independent bit mutations
- Various crossover polices

A Platform and Programming Language Independent Interface for Search Algorithms: `http://www.tik.ee.ethz.ch/pisa/`

First Population



649th Population

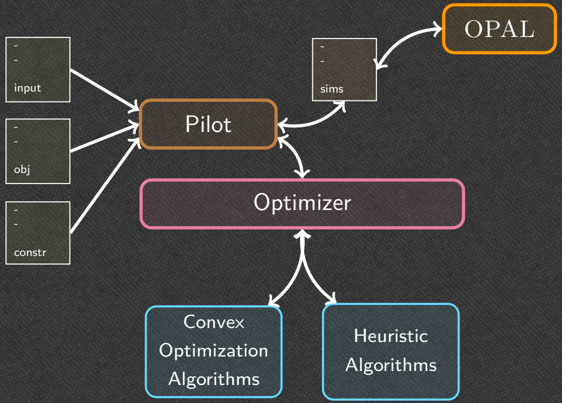


Now scientist/operator can specify PREFERENCE

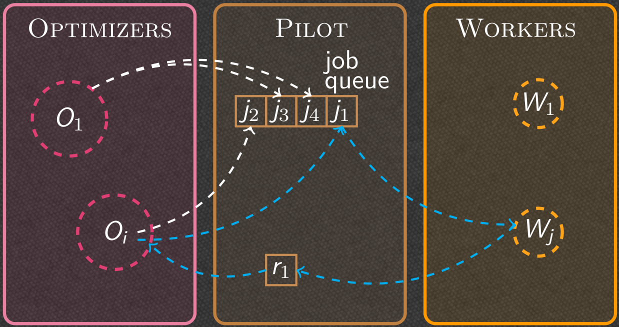
The Framework

[how can we facilitate solving multi-objective problems?]

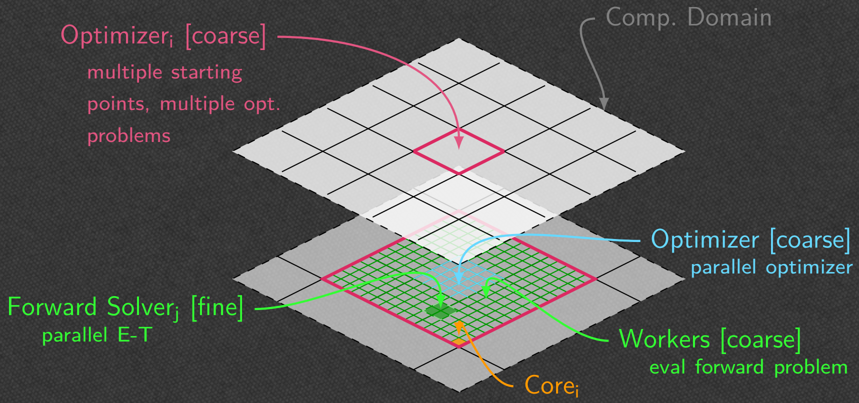
Multi-Objective Optimization Framework



Master/Worker Model



Master/Worker Model



Application

[how can we use the framework?]

Ingredients

$1\times$ optimization algorithm, $1\times$ forward solver and
$1\times$ specification of optimization problem, e.g., annotating the simulation input file:

```
//d1: DVAR, VARIABLE="D_LAG_RGUN",
      LOWERBOUND="-0.1", UPPERBOUND="0.1";
//d2: DVAR, VARIABLE="D_LAG_B01",
      LOWERBOUND="-0.1", UPPERBOUND="0.1";
//obj1: OBJECTIVE, EXPR="energy*-1";
//obj2: OBJECTIVE, EXPR="dE * meas_error("file", "rms_x")";
//objs: OBJECTIVES = (obj1, obj2);
//dvars: DVARS = (d1, d2);
//constrs: CONSTRAINTS = ();
//opt: OPTIMIZE, OBJECTIVES=objs, DVARS=dvars,
       CONSTRAINTS=constrs;
```

Forward solver typically is expensive to run..

Maxwell's Equation in the Electrostatic approximation

1,2 or 3D Field Maps & Analytic Models $(\mathbf{E}, \mathbf{B})_{\text{ext}}$

Poisson Problem s.t. BC's

$$\Delta\phi'_{sc} = -\frac{\rho'}{\epsilon_0}$$
$$\rightarrow (E, B)_{SC}$$

Electro Magneto Optics

$$\mathbf{H} = \mathbf{H}_{\text{ext}} + \mathbf{H}_{\text{sc}}$$

N-Body Dynamics

If $\mathbf{E}(\mathbf{x}, t)$ and $\mathbf{B}(\mathbf{x})$ are known, the equation of motion can be integrated:

- Boris-pusher (adaptive version soon!)
- Leap-Frog
- RK-4

and we require a massive number of forward solves per optimization

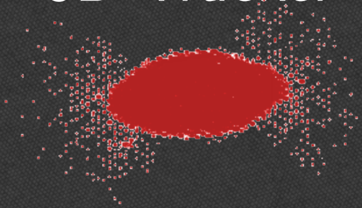
Object Oriented Parallel Accelerator Library (OPAL)

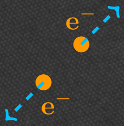
OPAL is a tool for **precise** charged-particle optics in large accelerator structures and beam lines including 3D space charge.

- built from the ground up as a parallel application
- runs on your laptop as well as on the largest HPC clusters
- uses the MAD language with extensions
- written in C++ using OO-techniques, hence OPAL is easy to extend
- nightly regression tests track the code quality

OPAL: `https://amas.psi.ch/OPAL`

3D Tracker





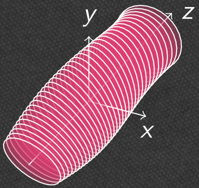
repulsive force of charged particles

- Huge # of macro particles (100'000 – 100'000'000)
- Computing space-charge is expensive
- Load balancing difficult
- Lots of synchronization points

Slow but "high resolution" forward solver

The Object Oriented Parallel Accelerator Library (OPAL), Design, Implementation and Application, A. Adelmann et. al.

Envelope Tracker



- # slices $\ll$ # macro particles
- Analytical space-charge
- Slices distributed in contiguous blocks
- Load imbalance of at most 1 slice
- Low number of synchronization points

➡ Fast but "low resolution" forward solver

A fast and scalable low dimensional solver for charged particle dynamics in large particle accelerators, Y. Ineichen et. al.

Usage

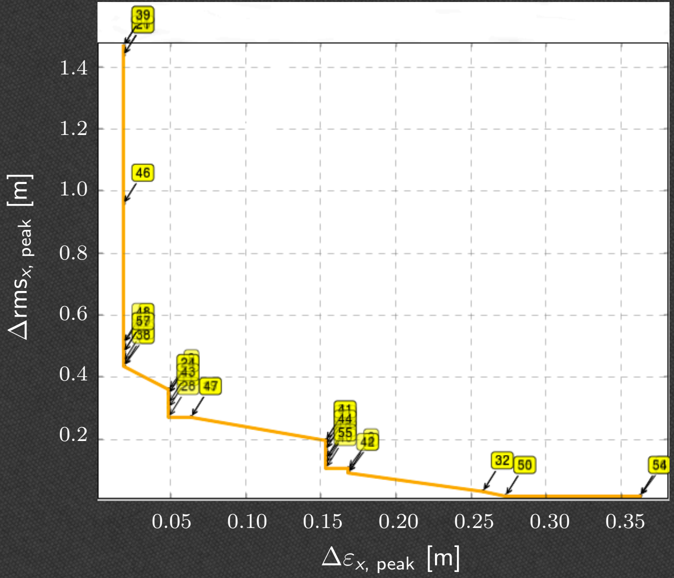
[Ferrario Matching Point]

Optimization Problem

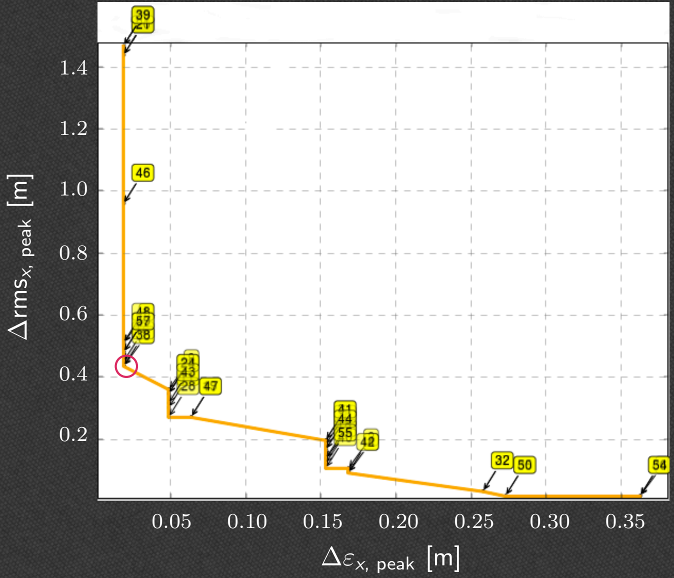
$$\text{min} \quad [\varepsilon_x, \Delta \text{rms}_{x,\text{peak}}, \Delta \varepsilon_{x,\text{peak}}]$$

```
//min_ex:      OBJECTIVE, EXPR="emit_x";
//peak_rms_x: FROMFILE,  FILE="rms_x-err.dat";
//peak_e_x:   FROMFILE,  FILE="emit_x-err.dat";
```

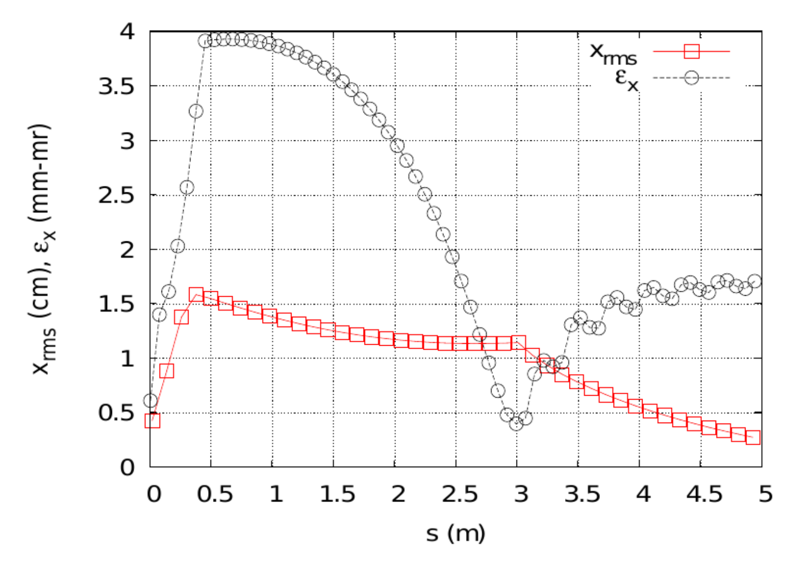
```
//sig_x:    DVAR, VARIABLE="SIGX", LOWERBOUND="0.000250",
                                   UPPERBOUND="0.000250";
//sol_ks:   DVAR, VARIABLE="MSOL10_i", LOWERBOUND="110",
                                       UPPERBOUND="120";
//lag_gun:  DVAR, VARIABLE="D_LAG_GUN", LOWERBOUND="0.0",
                                        UPPERBOUND="0.05";
//volt_gun: DVAR, VARIABLE="RACC_v", LOWERBOUND="25",
                                     UPPERBOUND="40";
```



Pareto front after 1'000 generations (approx. 20 minutes on 16 cores)



Pareto front after 1'000 generations (approx. 20 minutes on 16 cores)



Simulation results for individual 38

Conclusions

Multi-Objective Optimization Problems

- omnipresent in many fields in research and design
- important in understanding problem and trade-off solutions
- expensive to solve

Framework

- closes the gap between theory and user
- combining OPAL and EA results in a viable MOOP solver for beam dynamics
- HPC necessary to compute Pareto front in meaningful timeframe

Outlook

- Investigate other multi-objective optimization algorithms
- Run real optimization problem on massive number of cores
- Visualization of results

This project is funded by:

IBM Research – Zurich

Paul Scherrer Institut

Acknowledgements

- OPAL developer team
- SwissFel team
- Sumin Wei

Backup

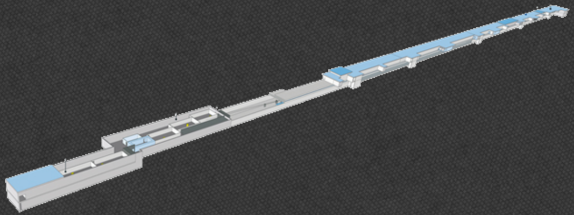
Optimization Problem

$$
\begin{aligned}
\text{min} \quad & [\text{energy spread}, \text{emittance}] \\
\text{s.t.} \quad & \partial_t f(\mathbf{x}, \mathbf{v}, t) + \mathbf{v} \cdot \nabla_x f(\mathbf{x}, \mathbf{v}, t) + \frac{q}{m_0} (\mathbf{E}_{\text{tot}} + \mathbf{v} \times \mathbf{B}_{\text{tot}}) \cdot \nabla_v f(\mathbf{x}, \mathbf{v}, t) = 0 \\
& \nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t}, \qquad \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \\
& \nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}, \qquad \nabla \cdot \mathbf{B} = \mathbf{0} \\
& \rho = -e \int f(\mathbf{x}, \mathbf{v}, t)\; d^3\mathbf{p}, \qquad \mathbf{J} = -e \int f(\mathbf{x}, \mathbf{v}, t) \mathbf{v}\; d^3\mathbf{p} \\
& \mathbf{E} = \mathbf{E}_{\text{ext}} + \mathbf{E}_{\text{self}}, \qquad \mathbf{B} = \mathbf{B}_{\text{ext}} + \mathbf{B}_{\text{self}} \\
& \dots
\end{aligned}
$$

Envelope Equations

$$\frac{d^2}{d^2 t} R_i + \beta_i \gamma_i^2 \frac{d}{dt}(\beta_i R_i) + R_i \sum_j K_i^j = \frac{2c^2 k_p}{R_i \beta_i} \times$$
$$\left(\frac{G(\Delta_i, A_r)}{\gamma_i^3} - (1 - \beta_i^2) \frac{G(\delta_i, A_r)}{\gamma_i} \right) + \frac{4\varepsilon_n^{\mathrm{th}} c}{\gamma_i} \frac{1}{R_i^3}$$
$$\frac{d}{dt}\beta_i = \frac{e_0}{m_0 c \gamma_i^3} \left(E_z^{\mathrm{ext}}(z_i, t) + E_z^{\mathrm{sc}}(z_i, t) \right)$$
$$\frac{d}{dt} z_i = c\beta_i$$

SwissFel [1]: Switzerland's X-ray free-electron laser Project at PSI



- big project: $>$ 100 people, expensive, 700 m long
- 1 Ångström
- to reach target it is of crucial importance to attain "good" beam properties (e.g. narrow beam/small phase space volume)

[1] http://www.psi.ch/swissfel/

SwissFel [1]: Switzerland's X-ray free-electron laser Project at PSI

Calls for optimization of Injector

- Several conflicting objectives
- Key technology: multi-objective optimization

- big project: $>$ 100 people, expensive, 700 m long
- 1 Ångström
- to reach target it is of crucial importance to attain “good” beam properties (e.g. narrow beam/small phase space volume)

[1] `http://www.psi.ch/swissfel/`