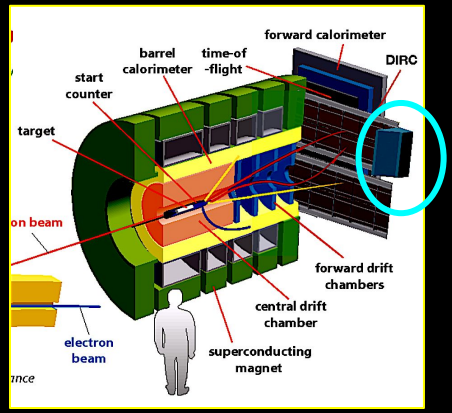
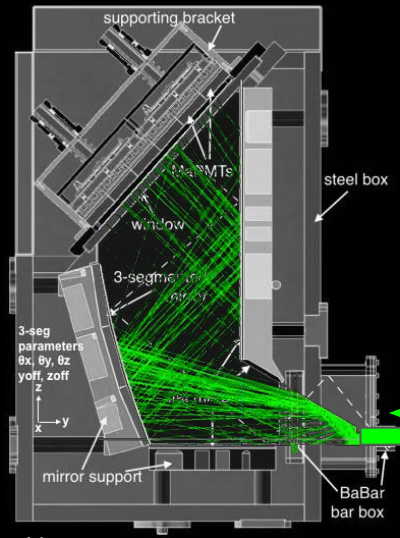
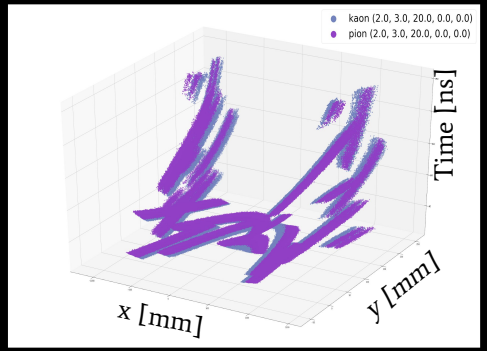
GlueX DIRC Alignment



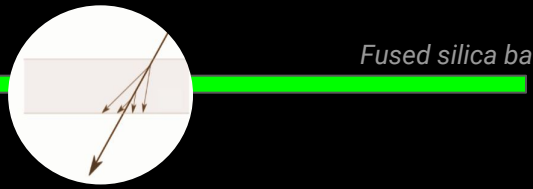
GlueX View

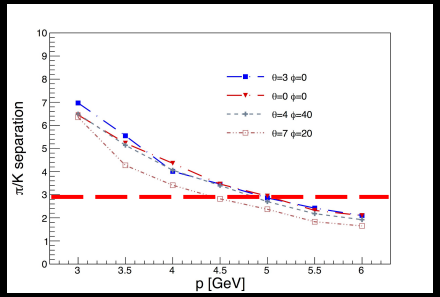


Optical box



3D Readout





π/K separation with DIRC

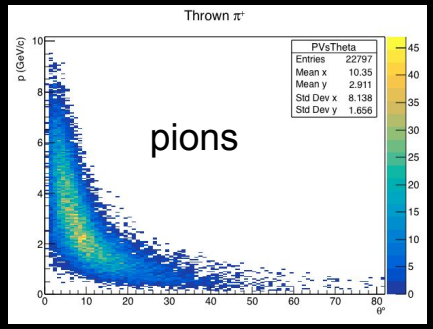
3D (x,y,t) readout allows to separate spatial overlaps.

Patterns take up significant fractions of the PMT in x,y and are read out over 50-100 ns due to propagation time in bars.

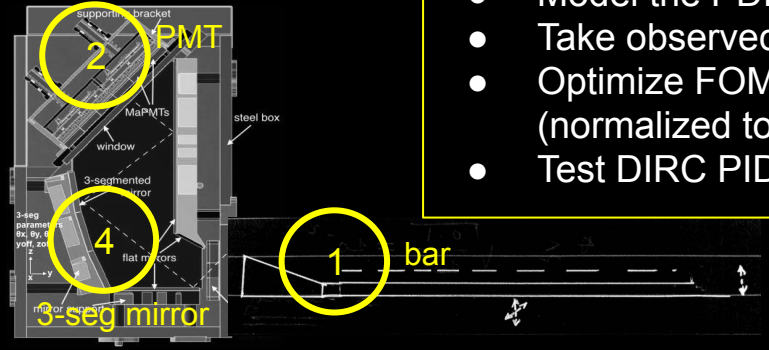
H12700 PMTs have a time resolution of O(200 ps) and read-out electronics giving time information in 1 ns buckets.

Approach

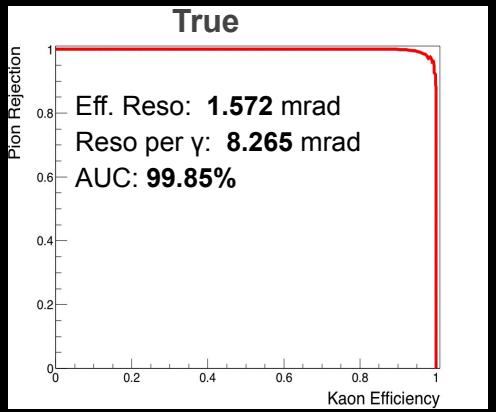
- Select high purity sample of particles at low P (well identified by GlueX PID w/o DIRC)
- Model the PDF as a function of the offsets
- Take observed hits to build Likelihood
- Optimize FOM = logL (normalized to a default alignment)
- Test DIRC PID on larger momentum P

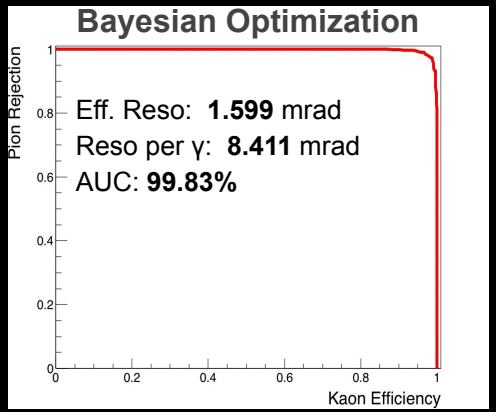


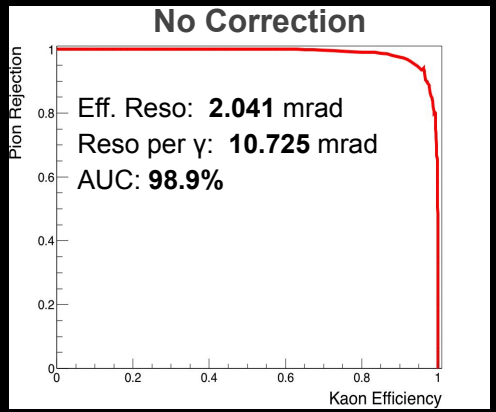
Main alignment parameters



Pion rejection vs Kaon efficiency at large P







True:

3-seg mirror:
θx,θy,θz=(0.25,0.50,0.15) deg,
y = 0.50 mm;
bar: z = 2.00 mm;
PMT: (r,θ)=(1.50 mm,1.00 deg)

BO-reversed engineered:

3-seg mirror:
θx,θy,θz=(0.25, 0.58, 0.12) deg,
y = 0.59 mm;
bar: z = 2.08 mm;
PMT: (r,θ)=(1.87 mm, 1.35 deg)

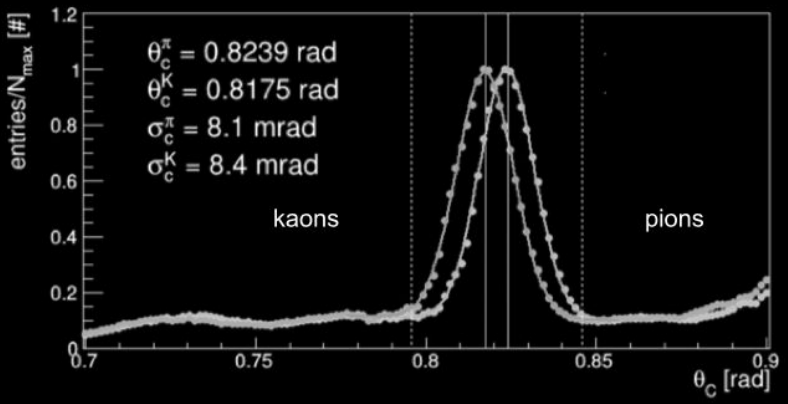
Reconstruction Algorithms and PID

LUT-based geometrical

R. Dzhygadlo et al. Nucl. Instr. And Meth. A, 766:263 (2014)

1. Creation of the LUT: store directions at the end of the radiator for each hit pixel

2. Direction from the LUT for the hit pixels are combined with the track directions (from tracking)

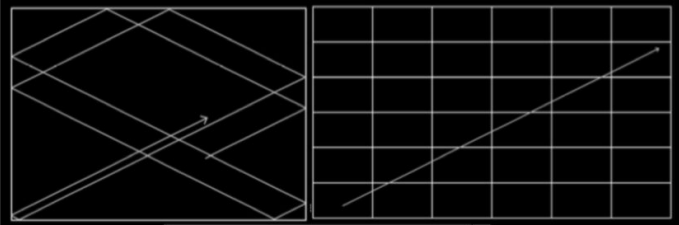


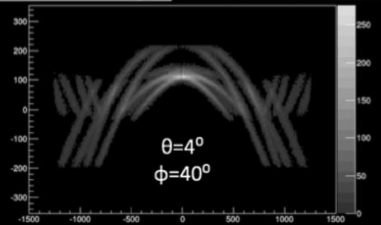
KDE-based

J. Hardin and M. Williams, JINST 11.10 (2016)

Fast tracing mapping straight lines through a tiled plane

1. Generation - 2. Traces through bars - 3. Traces through expansion volume





$$P(x) \approx \sum_{i}^{n} K(x - s_i)$$

https://github.com/jmhardin/FasDIRC

basically a trade-off memory/CPU usage

faster reconstruction/hit pattern

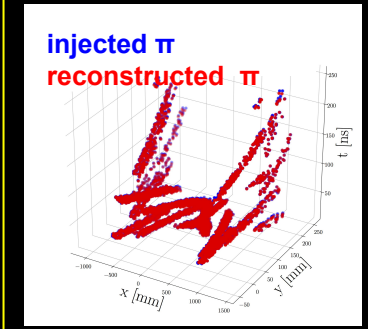
better resolution in regions with high overlap

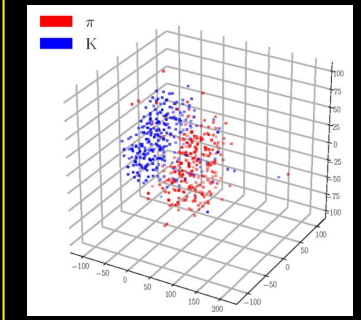
Hyperparameters tuning: DeepRICH

CF and J. Pomponi
Machine Learning: Science and Technology 1.1 (2020): 015010

DeepRICH

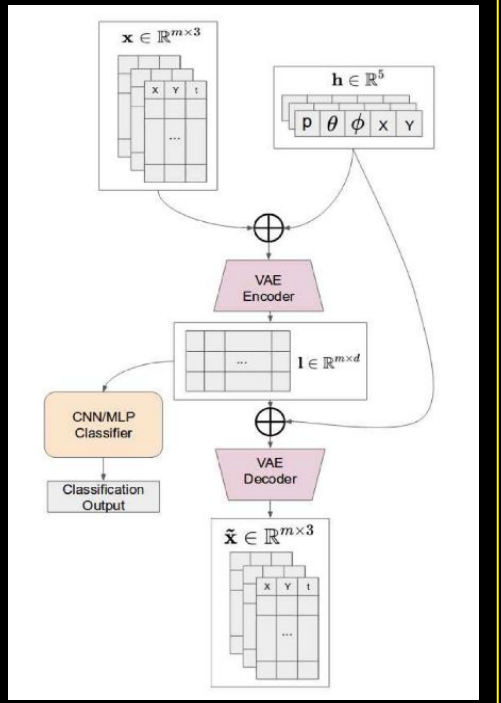
injected π
reconstructed π





latent space
t-SNE used for 3D visualization

injected



reconstructed

Hyperparameters

Table 2. List of hyperparameters tuned by the BO. The tuned values are shown in the outermost right column. The optimized test score is about 92%.

symbol	description	range	optimal value
NLL	λ_r	$[10^{-1},10^{2}]$	0.784
CE	λ_c	$[10^{-1},10]$	1.403
MMD	λ_v	$[1,10^{3}]$	1.009
LATENT_DIM	latent variables dimension	[10,200]	16
var_MMD	σ in $\mathcal{N}(0,\sigma)$	[0.01,2]	0.646
Learning Rate	learning rate	[0.000 1, 1]	$6.6{\cdot}10^{-4}$

DeepRICH Performance

Table 3. The area under curve (%), the signal efficiency to detect pions ε_S and the background rejection of kaons ε_B corresponding to the point of the ROC that maximizes the product $\varepsilon_S{*}\varepsilon_B$. The corresponding momenta at which these values have been calculated are also reported. This table is obtained by integrating over all the other kinematic parameters (i.e. a total of ∼6k points with different θ, ϕ, X, Y for each momentum).

	DeepRICH			FastDIRC		
Kinematics	AUC	ε_S	ε_B	AUC	ε_S	ε_B
4 GeV/c	99.74	98.18	98.16	99.88	98.98	98.85
4.5 GeV/c	98.78	95.21	95.21	99.22	96.33	96.32
5 GeV/c	96.64	91.13	91.23	97.41	92.40	92.47

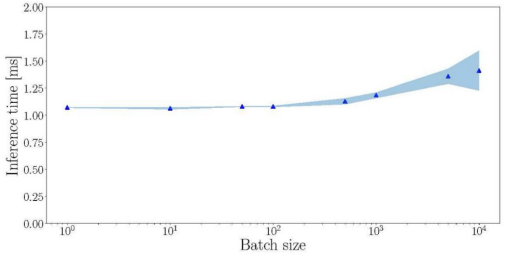


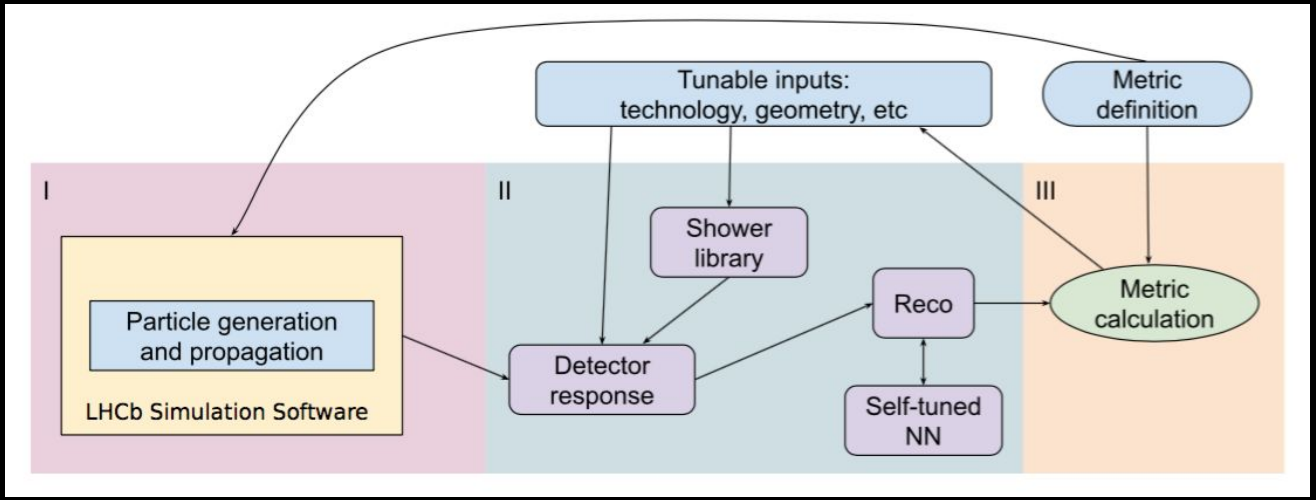
Figure 9. After training, the inference time is almost constant as a function of the batch size, meaning that the effective inference time—*i.e.*, the reconstruction time per particle—can be lower than a μs, the architecture being able to handle 10^4 particles in about 1.4 *ms* in the inference phase. Notice that the corresponding memory size in the inference phase is approximately equal to the value reported in table 4.

Other Applications

ML-assisted approach to calorimeter R&D

- Advanced detector R&D for both new and ongoing experiments in HEP requires per- forming computationally intensive and detailed simulations as part of the detector-design optimisation process.
- ML can substitute the most computationally intensive steps while retaining the GEANT4 accuracy to details.
- In [1] they focus on the Phase II Upgrade of the LHCb Calorimeter under the requirements on operation at high luminosity. The optimization pipeline looks like the following:

[1] A. Boldyrev et al (Yandex), arXiv:2005.07700v1 [physics.ins-det] 2020

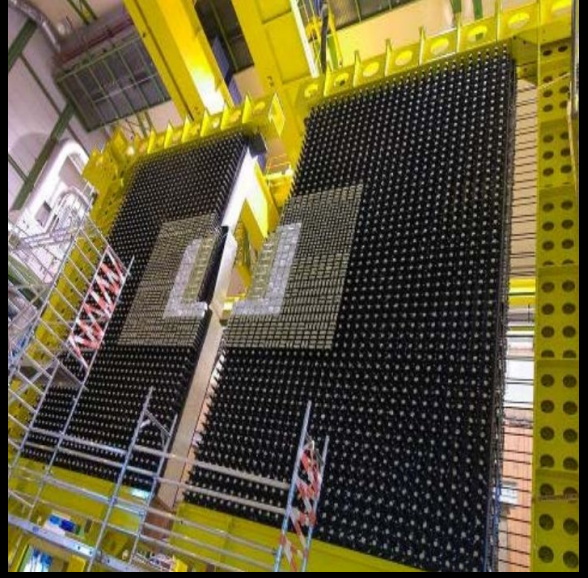


- Notice similarities with workflows discussed before.
- BO can be used to self-tune parameters.
- ML (decision trees) intervenes in tuning the reconstruction.

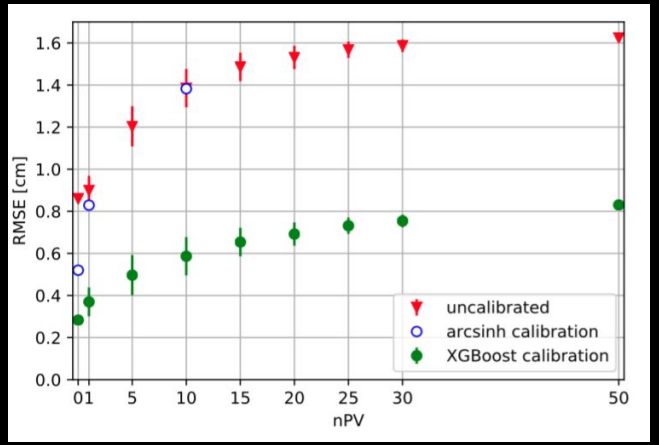
ECAL LHCb

$$\frac{\sigma_E}{E} = \frac{(8 \div 10)\%}{\sqrt{E(GeV)}} \oplus 0.9\%$$

- 4 mm thick scintillator tiles and 2 mm thick lead plates (Shashlik technology) ~25 X_0 (1.1 λ_I); Moliere radius ~ 36 mm;
- Modules 121.2 x 121.2 mm^2, 66 Pb +67 scintillator tiles;
- Segmentation: 3 zones 3 module types, Inner (9 cells per module), Middle (4), Outer (1). Total of 3312 modules, 6016 cells, (7.7 x 6.3) m^2, ~100 tons.



- Take advantage of segmentation / modularity (see discussion on characterization of the detector design problem) and create a Geant4 standalone simulation for 30x30 cells of size 20.2x20.2 mm^2 which can be rearranged in the inner, middle and outer ECAL modules.
- Used a signal sample $B^0_s \rightarrow J/\psi(\mu^+\mu^-)\pi^0(\gamma\gamma)$ and the LHCb minimum bias sample as background.
- Studies as a function of pile-up (PU) and number of primary vertices (nPV).
- Calibration (spatial and energy) optimized using XGBoost and BO for fine-tuning of parameters at the simulation and reconstruction steps.



Multi-Objective Optimization

- So far we have been discussing of optimization driven by a single objective
- The design can be actually driven by multiple objectives: an optimal design should be the result of a simultaneous optimization of multiple figures of merits (FoMs), taking into account, e.g, efficiency, resolution, distinguishing power between different particle types, as well as costs for the realization.
- In this context, the detector design can be considered as a complex combinatorial problem where AI-based approaches are clearly the most suited tools to deal with such complexity. In terms of computing resources, Geant-based simulations consume processor time, while AI is dealing with complicated regression problems.
- MOO is an active field of research in AI which has experienced in recent years a remarkable growth of applications like in social systems [1], material discovery [2], and multi-task learning problems thanks to the increased computational power available [3].

[1] G.-G. Wang, X. Cai, Z. Cui, G. Min, and J. Chen, “High performance computing for cyber physical social systems by using evolutionary multi-objective optimization algorithm,” IEEE Transactions on Emerging Topics in Computing, 2017.
[2] A. M. Gopakumar, P. V. Balachandran, D. Xue, J. E. Gubernatis, and T. Lookman, “Multi-objective optimization for materials discovery via adaptive design,” Scientific reports, vol. 8, no. 1, pp. 1–12, 2018.
[3] O. Sener and V. Koltun, “Multi-task learning as multi-objective optimization,” Advances in Neural Information Processing Systems, vol. 31, pp. 527–538, 2018

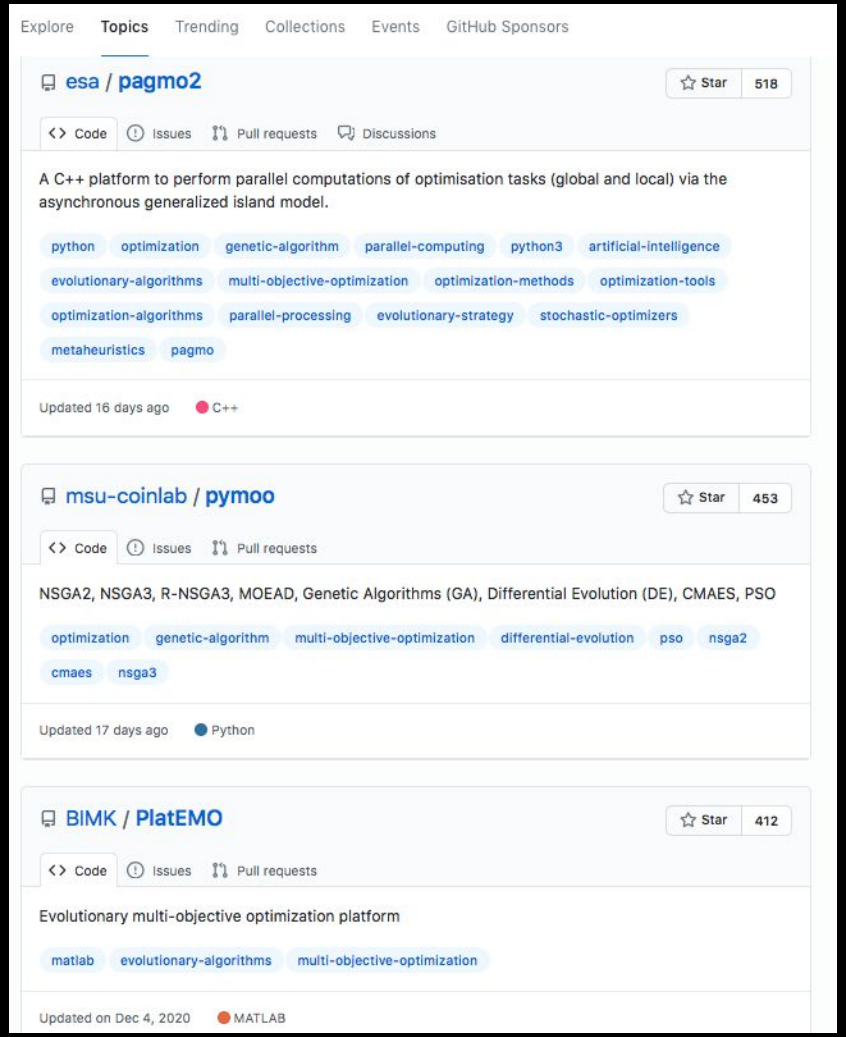
Frameworks

- Notice that MOO with dynamic/evolutionary algorithms (see, e.g., [1-3]) are probably the most utilized approaches on github, followed by more recent developments on multi-objective bayesian optimization (see, e.g., [4-7]). Using them has the advantage of having an entire community developing those tools.

  https://github.com/topics/multi-objective-optimization

- Agent-based approaches to MOO are also possible (see, e.g., [8]), but won't be discussed here.

- Remarkably these approaches can accommodate mechanical and geometrical constraints during the optimization process.



Frameworks

- Notice that MOO with dynamic/evolutionary algorithms (see, e.g., [1-3]) are probably the most utilized approaches on github, followed by more recent developments on multi-objective bayesian optimization (see, e.g., [4-7]). Using them has the advantage of having an entire community developing those tools.

  https://github.com/topics/multi-objective-optimization

- Agent-based approaches to MOO are also possible (see, e.g., [8]), but won't be discussed here.

- Remarkably these approaches can accommodate mechanical and geometrical constraints during the optimization process.

[1] J. J. Durillo and A. J. Nebro, "jMetal: A Java framework for multi-objective optimization," Advances in Engineering Software, vol. 42, no. 10, pp. 760–771, 2011.

[2] F.-A. Fortin, F.-M. De Rainville, M.-A. G. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," The Journal of Machine Learning Research, vol. 13, no. 1, pp. 2171–2175, 2012.

[3] J. Blank and K. Deb, "pymoo: Multi-objective Optimization in Python," IEEE Access, vol. 8, pp. 89497–89509, 2020

[4] M. Laumanns and J. Ocenasek, "Bayesian optimization algorithms for multi-objective optimization," in International Conference on Parallel Problem Solving from Nature, pp. 298–307, Springer, 2002.

[5] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy, "Botorch: Programmable bayesian optimization in pytorch," arXiv preprint arXiv:1910.06403, 2019.

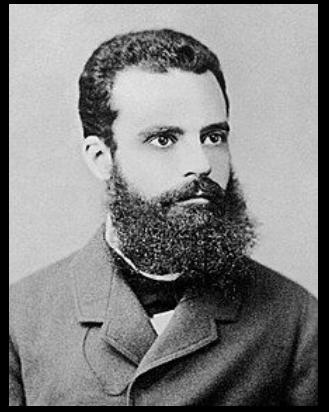
[6] P. P. Galuzio, E. H. de Vasconcelos Segundo, L. dos Santos Coelho, and V. C. Mariani, "MOBOpt—multi-objective Bayesian optimization," SoftwareX, vol. 12, p. 100520, 2020.

[7] A. Mathern, O. S. Steinholtz, A. Sjöberg, M. Önnheim, K. Ek, R. Rempling, E. Gustavsson, and M. Jirstrand, "Multi-objective constrained Bayesian optimization for structural design," Structural and Multidisciplinary Optimization, pp. 1–13, 2020.

[8] R. Yang, X. Sun, and K. Narasimhan, "A generalized algorithm for multi-objective reinforcement learning and policy adaptation," in Advances in Neural Information Processing Systems, pp. 14636–14647, 2019

MOO and Pareto Efficiency

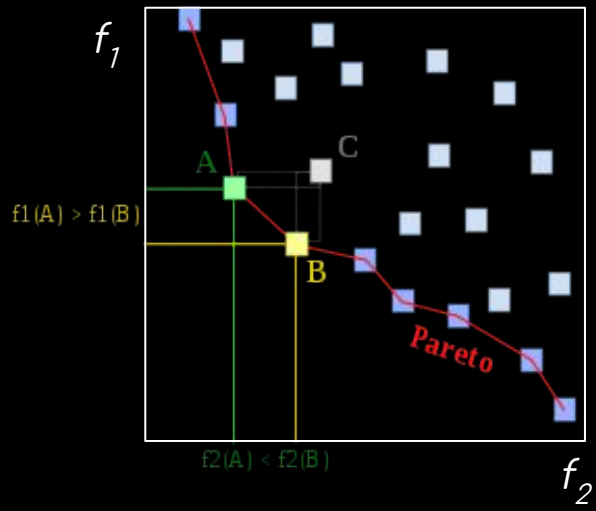
- The problem becomes challenging when the objectives are of conflict to each other, that is, the optimal solution of an objective function is different from that of the other. For example improving the resolution of a detector could imply increasing the costs for its realization.
- In solving such problems, with or without constraints, they give rise to a trade-off optimal solutions, popularly known as Pareto-optimal solutions.



V. Pareto, 1848-1923

- Due to the multiplicity in solutions, these problems were proposed to be solved suitably using evolutionary algorithms which use a population approach in its search procedure.
- Starting with parameterized procedures in early nineties, the so-called evolutionary multi-objective optimization (EMO) algorithms is now an established field of research.

Point *C* is not on the Pareto frontier because it is dominated by both point *A* and point *B*.



Evolutionary Optimization

[1] Deb, Kalyanmoy. "Multi-objective optimisation using evolutionary algorithms: an introduction." *Multi-objective evolutionary optimisation for product design and manufacturing*. Springer, London, 2011. 3-34.

- Evolutionary optimization (EO) algorithms use a population based approach in which more than one solution participates in an iteration and evolves a new population of solutions in each iteration.
- The reasons for the popularity of EOs are many:

  (i) do not require any derivative information

  (ii) relatively simple to implement

  (iii) flexible and have a widespread applicability.
- The use of a population of solutions to solve multi-objective optimization problems an EO procedure seems a “natural” choice.
- The MOO problems give rise to a set of Pareto-optimal solutions which need a further processing to arrive at a single preferred solution. To achieve the first task, the use of population in an iteration helps an EO to simultaneously find multiple non-dominated solutions, which portrays a trade-off among objectives, in a single simulation run.

MO-based solutions are helping to reveal important hidden knowledge about a problem – a matter which is difficult to achieve otherwise [1].

Evolutionary Optimization

EO principles differ from classical approaches in many ways:

- An EO procedure does not typically use gradient information in the search process. EO methodologies are direct search procedures.
- An EO procedure uses a population approach in an iteration, and has some advantages: (i) parallel processing power; (ii) allows EO to find multiple optimal solutions; (iii) provides EO with the ability to normalize decision variables (as well as objective and constraint functions) within an evolving population using the population-best minimum and maximum values.
- An EO uses stochastic operators. This allows an EO algorithm to negotiate multiple optima and other complexities better and provide them with a global perspective in their search.

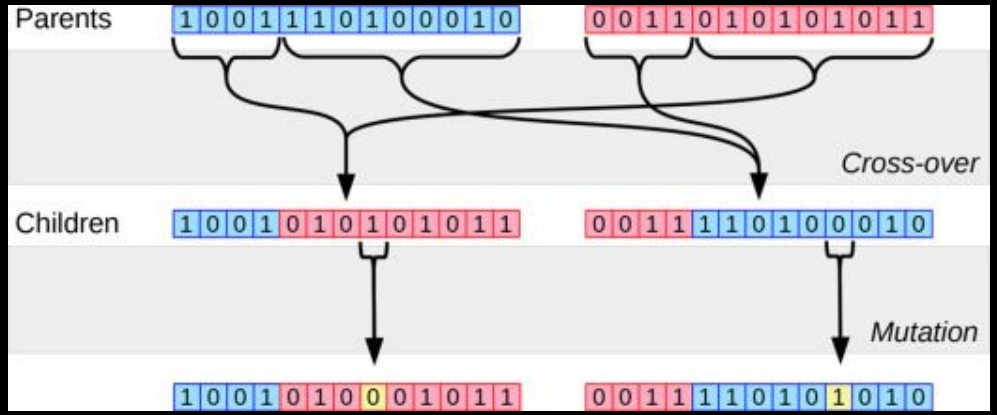
The initialization usually involves a random creation of solutions. it is highlighted that for solving complex real-world optimization problems, a customized initialization is helpful in achieving a faster search.

A selection is made to form an intermediate mating pool. A simple approach, called *tournament* selection, consists in picking two solutions at random from the population and the better of the two is kept, etc.

The variation operator is a collection of a number of operators (such as crossover, mutation etc.) which are used to generated a modified population.

Genetic Algorithm

- The purpose of the crossover operator is to pick two or more solutions (parents) from the mating pool and create one or more solutions by exchanging information among the parent solutions.



- This is applied with a crossover probability ($\mathbb{P}c \in [0,1]$), indicating the proportion of population members participating to the operation. The remaining proportion is simply copied to the modified (child) population.
- Each child solution, created by the crossover operator, is then perturbed in its vicinity by a mutation operator with a probability $\mathbb{P}m$, usually set as 1/n, where n is the number of variables (on average, 1 variable is mutated per solution). For real-parameter optimization, a simple Gaussian probability distribution with a predefined variance can be used with its mean at the child variable value.
- The elitism operator combines old with newly created population and chooses to keep the better solutions from the combined populations. It makes sure that an algorithm has a monotonically non-degrading performance.
- Finally the user of an EO needs to choose some termination criteria.

Crossover Operators

[1] https://engineering.purdue.edu/~sudhoff/ee630/Lecture04.pdf

- Actually a variety of types of crossovers [1]: Single point crossover, Linear crossover, Blend crossover, Simulated binary crossover (SBX).
- SBX is an efficient crossover for real variables, which mimics the crossover of binary encoded variables. It uses probability density function that simulates the single-point crossover in binary-coded GAs.

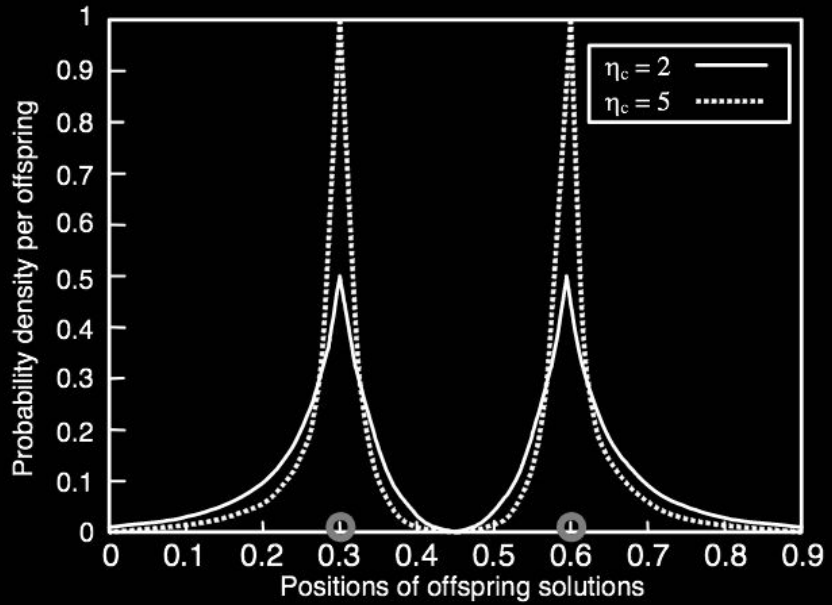
SBX Algorithm:

- Select parents x1 and x2
- Generate random u ∈ [0,1)
- Calculate β (η_c is the distribution index)

$$\beta = \begin{cases} (2u)^{\frac{1}{\eta_c+1}}, & \text{if } u \le 0.5 \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases}$$

Compute offspring as:

$$x_1^{new} = 0.5[(1+\beta)x_1 + (1-\beta)x_2]$$
$$x_2^{new} = 0.5[(1-\beta)x_1 + (1+\beta)x_2]$$



Large ηc tends to generate children closer to the parents
Small ηc allows the children to be far from the parents

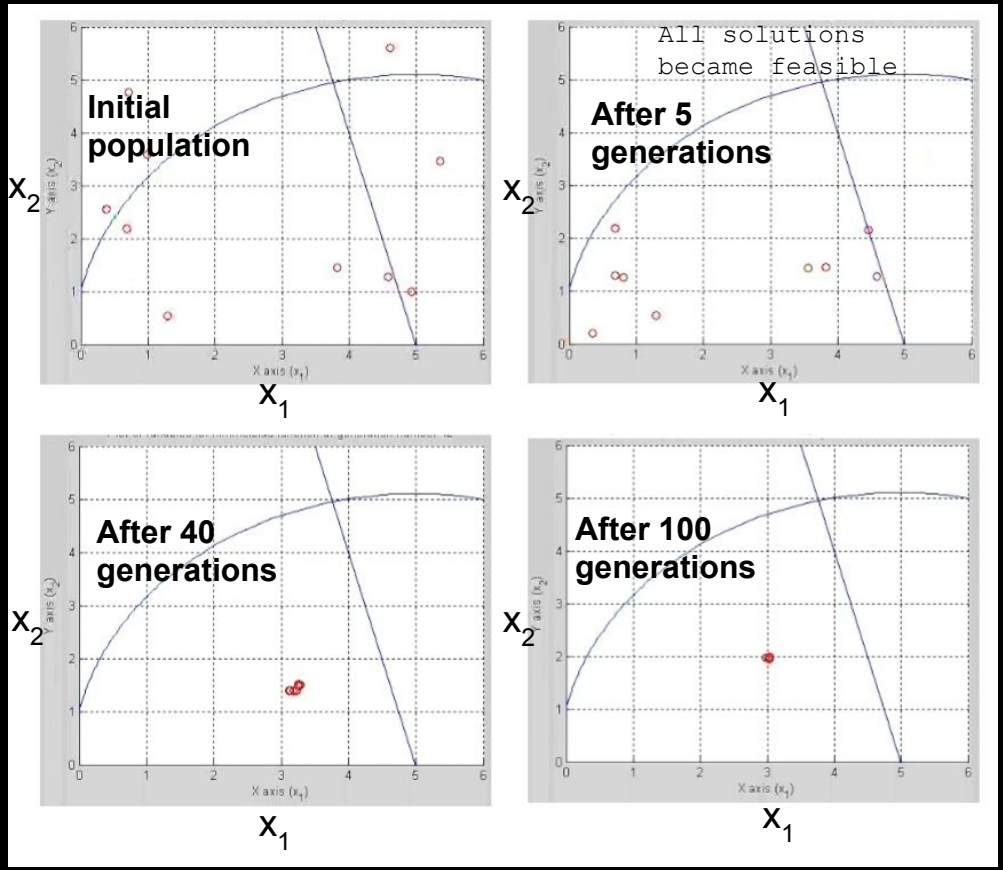
Phases of Evolution

Typically two phases of an EO:

- First, the GA exhibits a more global search by maintaining a diverse population, discovering potentially good regions of interest.
- Second, a more local search takes place by bringing the population members closer together.

Toy example: 1 objective, 2 constraints

$$\begin{aligned} \text{Minimize} \quad & f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \\ \text{subject to} \quad & g_1(x) \equiv 26 - (x_1 - 5)^2 - x_2^2 \geq 0, \\ & g_2(x) \equiv 20 - 4x_1 - x_2 \geq 0, \\ & 0 \leq (x_1, x_2) \leq 6. \end{aligned}$$



Evolutionary MOO (EMO)

[1] Deb, Kalyanmoy. *Multi-objective optimization using evolutionary algorithms*. Vol. 16. John Wiley & Sons, 2001.
[2] Blank, Julian, and Kalyanmoy Deb. "pymoo: Multi-objective Optimization in Python." *IEEE Access* 8 (2020): 89497-89509

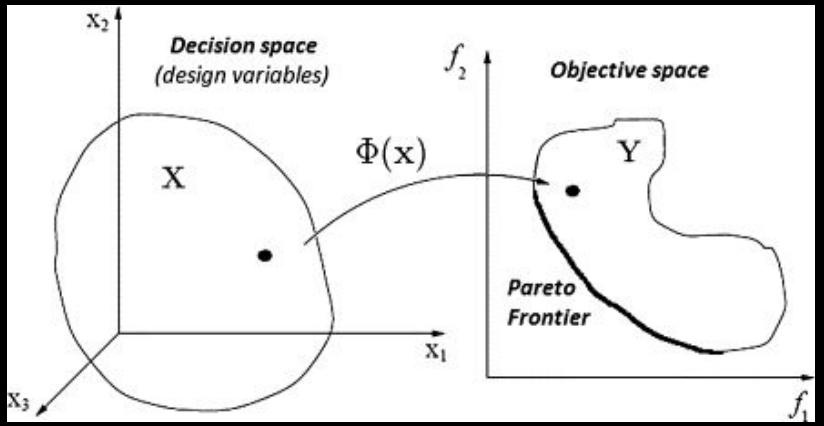
- In the following --- and in the hands-on session --- we will refer to the multi-objective optimization based on evolutionary algorithms [1], and in particular pymoo [2], written in Python, which also includes visualization and decision making tools.
- The definition of a generic MOO problem can be formulated as:

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

- M objective functions *f(x)* to optimize. By construction, pymoo performs minimization so a function to maximize needs a minus sign.
- There can be J inequalities *g(x)*
- There can be K equality constraints *h(x)*
- There are N variables x_i with lower and upper boundaries.

Evolutionary MOO (EMO)

- The solutions satisfying the constraints and variable bounds constitute a feasible decision variable space $S \subset R^n$, which corresponds to our design space.



- One of the striking differences between single-objective and multi-objective optimization, is that in the latter the objective functions constitute a multi-dimensional space called objective space, $Z \subset R^M$.
- The optimal solutions in multi-objective optimization can be defined from a mathematical concept of partial ordering. In the parlance of multi-objective optimization, the term "domination" is used for this purpose.
- All points which are non-dominated by any other member of the set are called the non-dominated points. One property of any two such points is that a gain in an objective from one point to the other happens only due to a sacrifice in at least one other objective (trade-off).

Non-Dominated Front

Do a pair-wise comparison using the above definition.

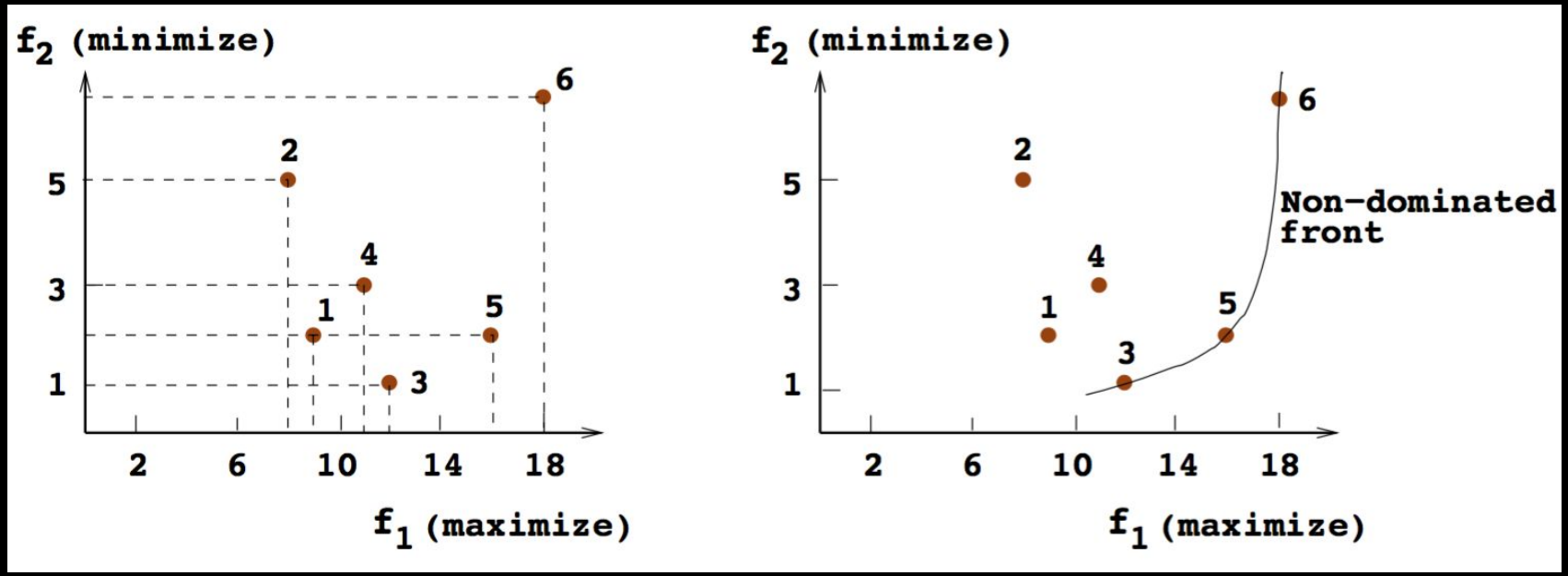


Fig. A set of points and the first non-domination front are shown.

Non-Dominated Front

[1] Kung HT, Luccio F, Preparata FP. On finding the maxima of a set of vectors. Journal of the Association for Computing Machinery. 1975;22(4):469–476

[2] Jensen, Mikkel T. "Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms." *IEEE Transactions on Evolutionary Computation* 7.5 (2003): 503-515.

- This trade-off property between the non-dominated points makes the practitioners interested in finding a wide variety of them before making a final choice.
- The computational effort needed to select the points of the non-domination front from a set of N points is O(N log N) for 2 and 3 objectives, and O(N $(\log N)^{M-2}$) for M > 3 objectives [1,2].
- If the given set of points for the above task contain all points in the search space (assuming a countable number), the points lying on the non-domination front, by definition, do not get dominated by any other point in the objective space, hence are Pareto-optimal points (together they constitute the Pareto-optimal front) and the corresponding pre-images (decision variable vectors) are called Pareto-optimal set of solutions.
- Evolutionary MOO attempts to satisfy the following principles:

  1. Find a set of solutions which lie on the Pareto-optimal front
  2. Find a set of solutions which are diverse enough to represent the entire range of the Pareto-optimal front.

Choice of Solution

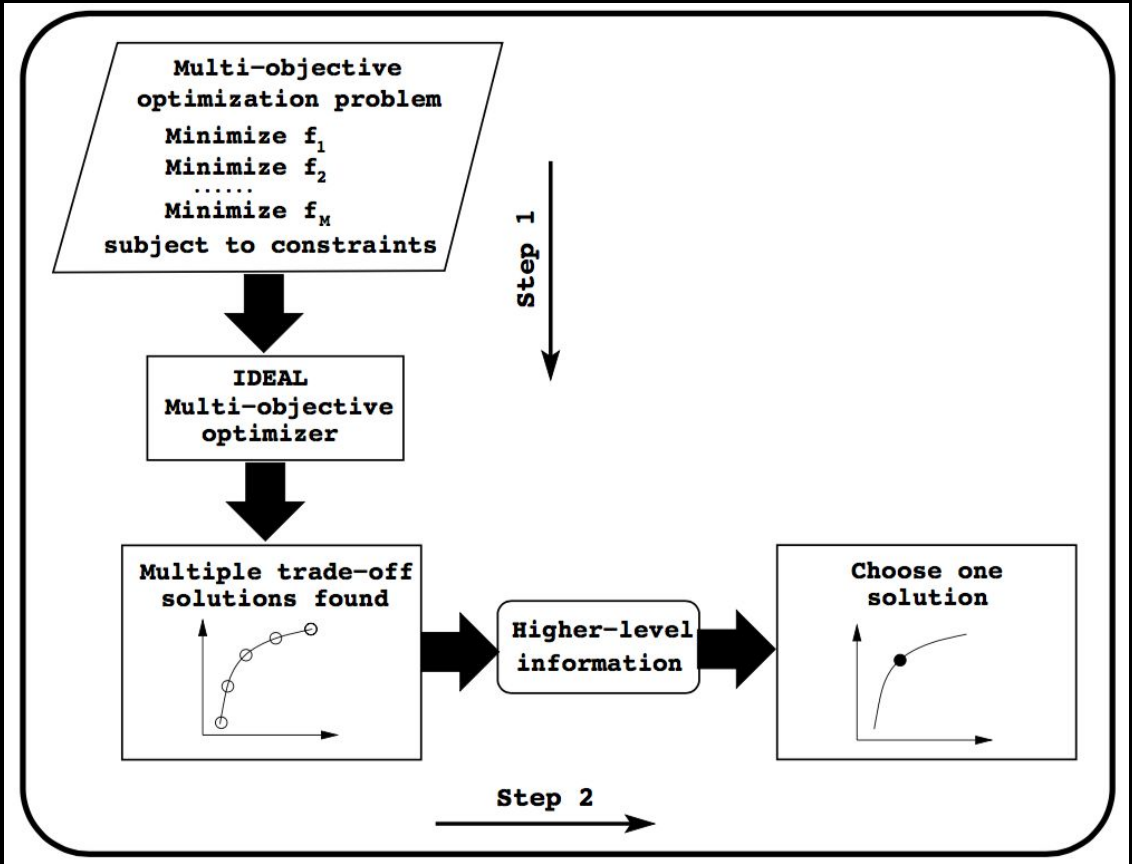
- Since a number of solutions are optimal, the obvious question arises: which of these optimal solutions must one choose?
- Answers this typically involves higher-level information which is often non-technical, qualitative and experience-driven. One has to evaluate the pros and cons of each of these solutions.
- So in MOO the effort must be made in finding the set of trade-off optimal solutions by considering all objectives to be important. Then operate a choice.
- Therefore Evolutionary Multi-Objective Optimization can be summarized as:

```
Step 1: Find multiple non-dominated points as close to the Pareto-optimal
front as possible, with a wide trade-off among objectives.

Step 2 Choose one of the obtained points using higher-level information.
```

Workflow Of EMO

[1] K. Deb, "Multi-objective optimisation using evolutionary algorithms: an introduction." *Multi-objective evolutionary optimisation for product design and manufacturing*. Springer, London, 2011. 3-34.



(follows from previous slide)

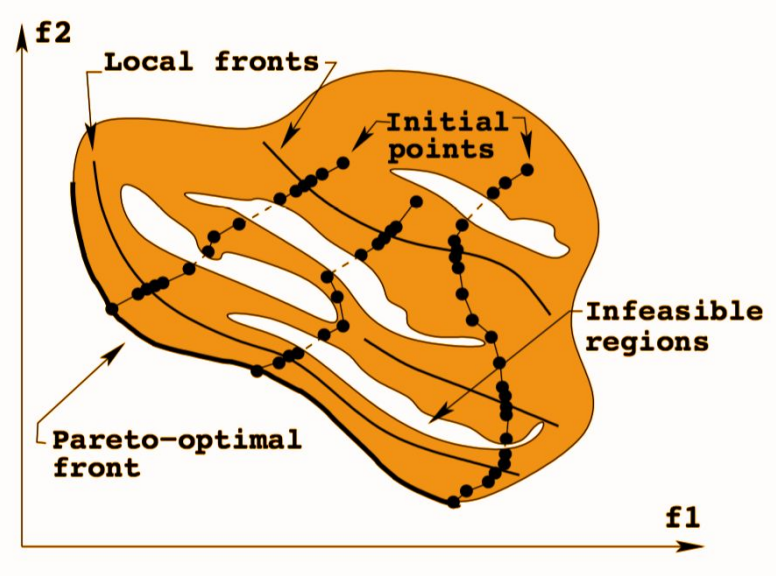
Step 1: Find multiple non-dominated points as close to the Pareto-optimal front as possible, with a wide trade-off among objectives.

Step 2 Choose one of the obtained points using higher-level information.

N.b.: This scheme is particularly useful also for single-objective optimization when multiple global optima are present

MCDM vs EMO

[1] K. Deb, "Multi-objective optimisation using evolutionary algorithms: an introduction." *Multi-objective evolutionary optimisation for product design and manufacturing*. Springer, London, 2011. 3-34.



In an EMO, multiple Pareto-optimal solutions are attempted to be found in a single simulation.

In the Fig., you can imagine instead to have Multiple Criteria Decision Making (MCDM), i.e., independent single-objective optimization which find different Pareto-optimal solutions. The Pareto front corresponds to several scalarized objectives (an example of scalarization is the weighted-sum approach $f(x) = \sum w_i \cdot f_i(x)$, for a given set of weights).

During the optimization task, algorithms must overcome a number of difficulties, to converge to the global optimum (e.g., there could be infeasible regions, local optimum solutions, etc.)

These problems can represent a challenge in computational time. EMO, constitutes an inherent parallel search, and when a population member overcomes these difficulties and make a progress towards the Pareto-optimal front, its variable values and their combination reflect this fact, and information get shared through variable exchange.

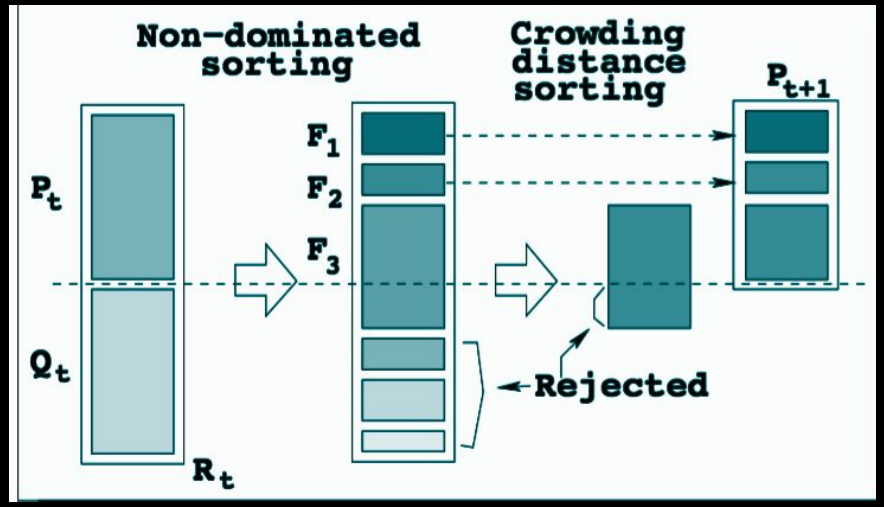
Finding multiple trade-off solutions is a parallelly processed task.

Elitist Non-Dominated Sorting GA or NSGA-II

[1] Deb, K., et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE transactions on evolutionary computation* 6.2 (2002): 182-197.

NSGA-II is one of the most popular EMO (>34k citations on google scholar), characterized by:

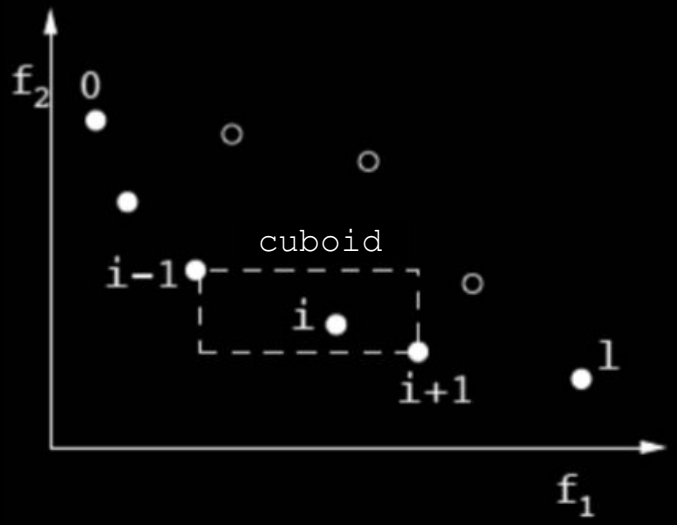
- ❑ Use of an elitist principle,
- ❑ Explicit diversity preserving mechanism
- ❑ Emphasis in non-dominated solutions.



- At any generation t, the offspring population (Q_t) is first created from the parent population (P_t) with GA. The two are combined to form a new population (R_t) of size 2N.
- The population R_t is classified into different non-dominated classes. The new population is filled with points from different non-domination fronts, one at a time
- Not all fronts can be accomodated in N slots available for the new population (P_{t+1}). Some fronts will be deleted, the first ones will be included. The last front to be considered may need to have some members trimmed.

NSGA-II and Crowding Distance

[1] Deb, Kalyanmoy, et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE transactions on evolutionary computation* 6.2 (2002): 182-197.



- Instead of arbitrarily discarding some members from the last front, the points which will make the diversity of the selected points the highest are chosen.
- The crowded-sorting of the points of the last front which will not be accommodated fully is achieved according to the descending order of their crowding distance values.
- The crowding distance d_i of point *i* is a measure of the objective space around *i* which is not occupied by any other solution in the population. A possible metric is the perimeter of the cuboid in Figure, formed by using the neighbors in the objective space as vertices.

Constraint-Handling in EMO

- The binary tournament selection can be modified by the constraints. In presence of constraints, each solution can be either feasible or infeasible.
- There can be three situations:
  - Both solutions are feasible
  - One is feasible, the other not
  - Both are infeasible

A redefinition of the dominion principle is done (called constrained-domination):

Definition 5.1 *A solution $\mathbf{x}^{(i)}$ is said to 'constrained-dominate' a solution $\mathbf{x}^{(j)}$ (or $\mathbf{x}^{(i)} \prec_c \mathbf{x}^{(j)}$), if any of the following conditions are true:*

1. *Solution $\mathbf{x}^{(i)}$ is feasible and solution $\mathbf{x}^{(j)}$ is not.*
2. *Solutions $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are both infeasible, but solution $\mathbf{x}^{(i)}$ has a smaller constraint violation, which can be computed by adding the normalized violation of all constraints:*

$$\mathrm{CV}(\mathbf{x}) = \sum_{j=1}^{J} \langle \bar{g}_j(\mathbf{x}) \rangle + \sum_{k=1}^{K} \mathrm{abs}(\bar{h}_k(\mathbf{x})),$$

3. *Solutions $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are feasible and solution $\mathbf{x}^{(i)}$ dominates solution $\mathbf{x}^{(j)}$ in the usual sense*

This implies that the first non-domination front consists of the "best" (that is, non-dominated and feasible) points from the population and any feasible point lies on a better non-domination front than an infeasible point

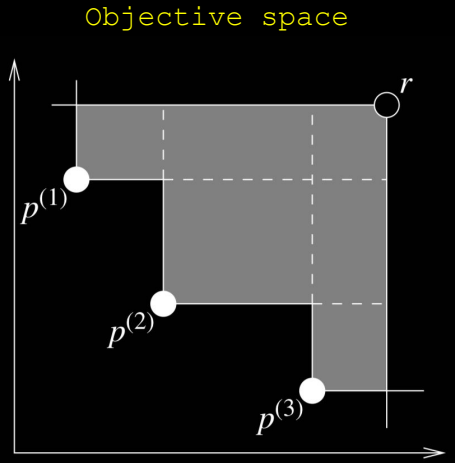
Performance Measures

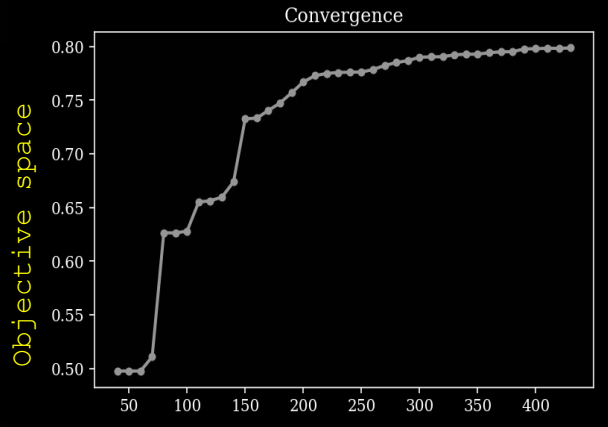
[1] *Zitzler E, Thiele L, Laumanns M, Fonseca CM, Fonseca VG. Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation. 2003;7(2):117– 132.*

- There are two conflicting goals in an EMO procedure: (i) a good convergence to the Pareto-optimal front and (ii) good diversity in obtained solutions:
  - Metrics evaluating convergence to the known Pareto-optimal front (such as error ratio, distance from reference set, etc.),
  - Metrics evaluating spread of solutions on the known Pareto-optimal front (such as spread, spacing, etc.), and
  - Metrics evaluating certain combinations of convergence and spread of solutions (such as hypervolume, coverage, R-metrics, etc.).

- For Hypervolume (see right) only a reference point (r) needs to be provided. Pymoo uses the same implementation of DEAP. https://deap.readthedocs.io

- A study has argued that convergence and diversity cannot be measured by a single metric [1]...





Decision Making

[1] K. Deb, "Multi-objective optimisation using evolutionary algorithms: an introduction." *Multi-objective evolutionary optimisation for product design and manufacturing*. Springer, London, 2011. 3-34.

- Finding a set of representative Pareto-optimal solutions using an EMO procedure is half the task; choosing a single preferred solution from the obtained set is an equally important task.
- Only the integration of the decision-making procedure with the EMO procedure makes the multi-objective optimization a complete procedure.
- There are three main directions of developments that we mention here briefly (details in [1]):
  - A priori approach: i.e., focus the search effort into a part of the Pareto-optimal front, instead of the entire frontier (e.g., reference point approach, reference direction approach, etc.).
  - A posteriori approach: preference information used after a set of representative Pareto optimal solutions are found.
  - Interactive approach: the decision maker is called after every τ generations diversified solutions from the non-dominated front are chosen, and DM is asked to rank them according to preference (utility function). This drives NSGA-II search procedure.

DM in pymoo

[1] Blank, Julian, and Kalyanmoy Deb. "pymoo: Multi-objective Optimization in Python." *IEEE Access* 8 (2020): 89497-89509.

- Pymoo for example offers different approaches for DM, after obtaining a set of non-dominated solutions (in post-processing) [1].
  - Compromise Programming: One way of making a decision is to compute value of a scalarized and aggregated function and select one solution based on minimum or maximum value of the function.
  - Pseudo-Weights: a more intuitive way, where pseudo-weights are defined as:

$$w_i = \frac{(f_i^{\max} - f_i(x)) / (f_i^{\max} - f_i^{\min})}{\sum_{m=1}^{M} (f_m^{\max} - f_m(x)) / (f_m^{\max} - f_m^{\min})}$$

Namely, the normalized distance to the worst solution regarding each objective `i` is calculated.

  - High Trade-Off Solutions: these are usually of interest (the “knees” of the Pareto front). The metric is given by (1). The trade-off measure for each solution is (2).

Aggregated sacrifice

(1) $$T(x_i, x_j) = \frac{\sum_{m=1}^{M} \max[0, f_m(x_j) - f_m(x_i)]}{\sum_{m=1}^{M} \max[0, f_m(x_i) - f_m(x_j)]}$$

Aggregated gain

(2) $$\mu(x_i, S) = \min_{x_j \in S} T(x_i, x_j)$$

Practical EMO

[1] V. Khare, X. Yao, K. Deb. Performance Scaling of Multi-objective Evolutionary Algorithms. In: Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference (LNCS 2632); 2003. p. 376–390.

- EMO is an established successful procedure since many years with 2 and 3 objectives.
- It's well known that problems with many objectives (e.g., O(10)) can present challenges. [1]
  - As the number of objectives increases, most members in a randomly created population become non-dominated to each other. E.g., if N=200, and M=3, ~10% members are non-dominated; if N=200 and M=10, ~90% are non-dominated. An exponentially large population size is needed to represent a large-dimensional Pareto optimal front.
  - This causes a stagnation in the performance of an EMO algorithm.
- Two typical approaches to tackle large objective-problems are:
  - Finding only a part of the Pareto-optimal front: there are many means to indicate a preference information (e.g., distributed computing environment with a unique "cone" for defining domination) see [1]. This worked well up to ~20 objectives.
  - Identifying and Eliminating Redundant Objectives: objectives causing positively correlated relationship between each other on the obtained NSGA-II solutions are identified and declared as redundant using PCA. The EMO-PCA is continued until no further reduction in the objective space is found. Test studies have been done with $O(10^2)$ objectives.

Uncertainties

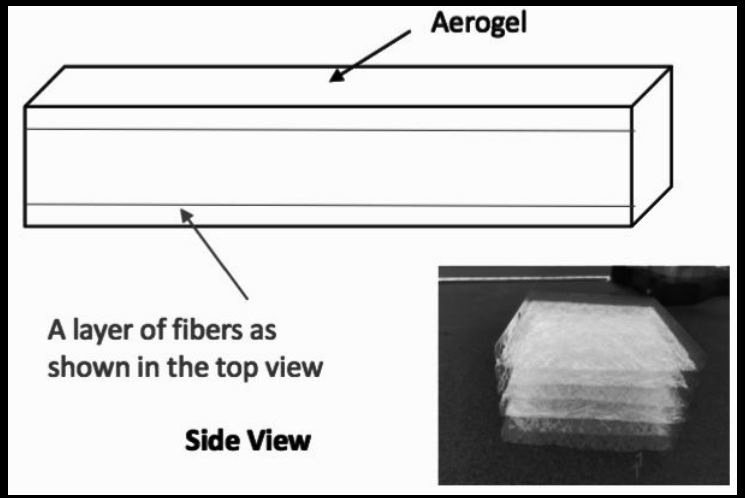
[1] Deb K, Gupta S, Daum D, Branke J, Mall A, Padmanabhan D. Reliability-based optimization using evolutionary algorithms. IEEE Trans on Evolutionary Computation, 2012
[2] Basseur M, Zitzler E. Handling Uncertainty in Indicator-Based Multiobjective Optimization. International Journal of Computational Intelligence Research. 2006;2(3):255–272

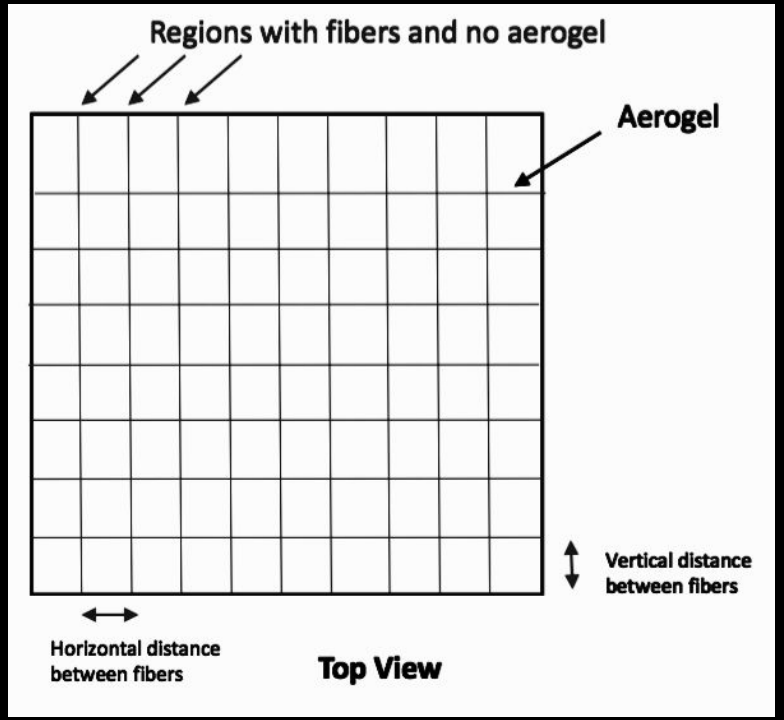
- As already discussed (see, *e.g.*, d-RICH), detector simulation is affected by uncertainty.
- In some cases, optimization problems are full of uncertainties and almost no parameter, dimension, or property can be guaranteed to be fixed at a value it is aimed at, and the evaluation of a solution can become not precise: in this case, the resulting objectives and constraint function values can be treated as probabilistic quantities.
- Some techniques in the literature (see, *e.g.*, nested optimization [1]) take into account these effects: the resulting solution is usually different from the optimum solution and is known as a 'robust' solution. A robust frontier in MOO can be different from a globally Pareto-optimal frontier, but each point is guaranteed to be less sensitive to uncertainties [2].
- For example, regarding deterministic constraints, they become stochastic and involve a reliability index (R) to handle the constraints, such that $\mathbb{P}(g(x)\geq 0)>R$, and separate optimization methodology could be needed to satisfy the stochastic constraint.
- It's worth reminding that in detector design the evaluation of our objectives at each design point is done by typically simulating a relatively large sample of events. This has the advantage to work with average quantities (and be less sensitive to outliers).
- Also remind that the "solution" of a MOO is a set of points corresponding to the Pareto front in the objective space. Further analysis in the post-processing phase allows to study and characterize uncertainties in the design space (e.g., using additional simulations). Coarse-to-fine-grained approaches are also possible.

Novel Material for Aerogel-Based Detector

- Aerogels with low refractive indices are very fragile - tiles break during production and handling, and their installation in detectors.
- To improve the mechanical strength of aerogels, Scintilex developed a strategy: the general concept consists of introducing fibers into the aerogel that increase mechanical strength, but do not affect the optical properties of the aerogel.

The team: V. Berdnikov, J. Crafts, E. Cisbani, C. Fanelli, T. Horn, R. Trotta





Simulation of Aerogel with block of Fibers

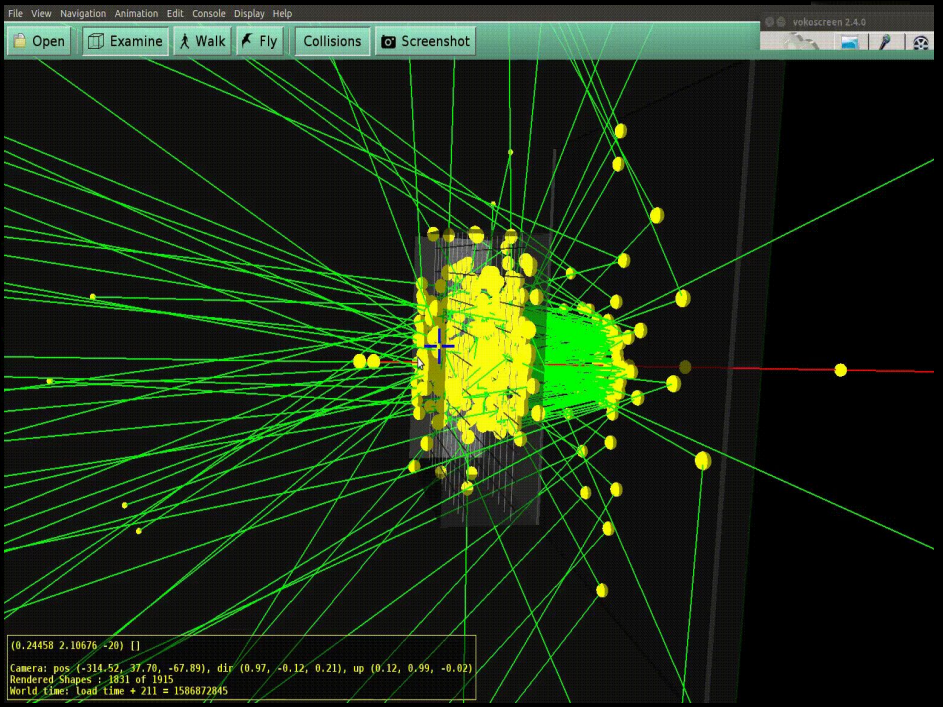
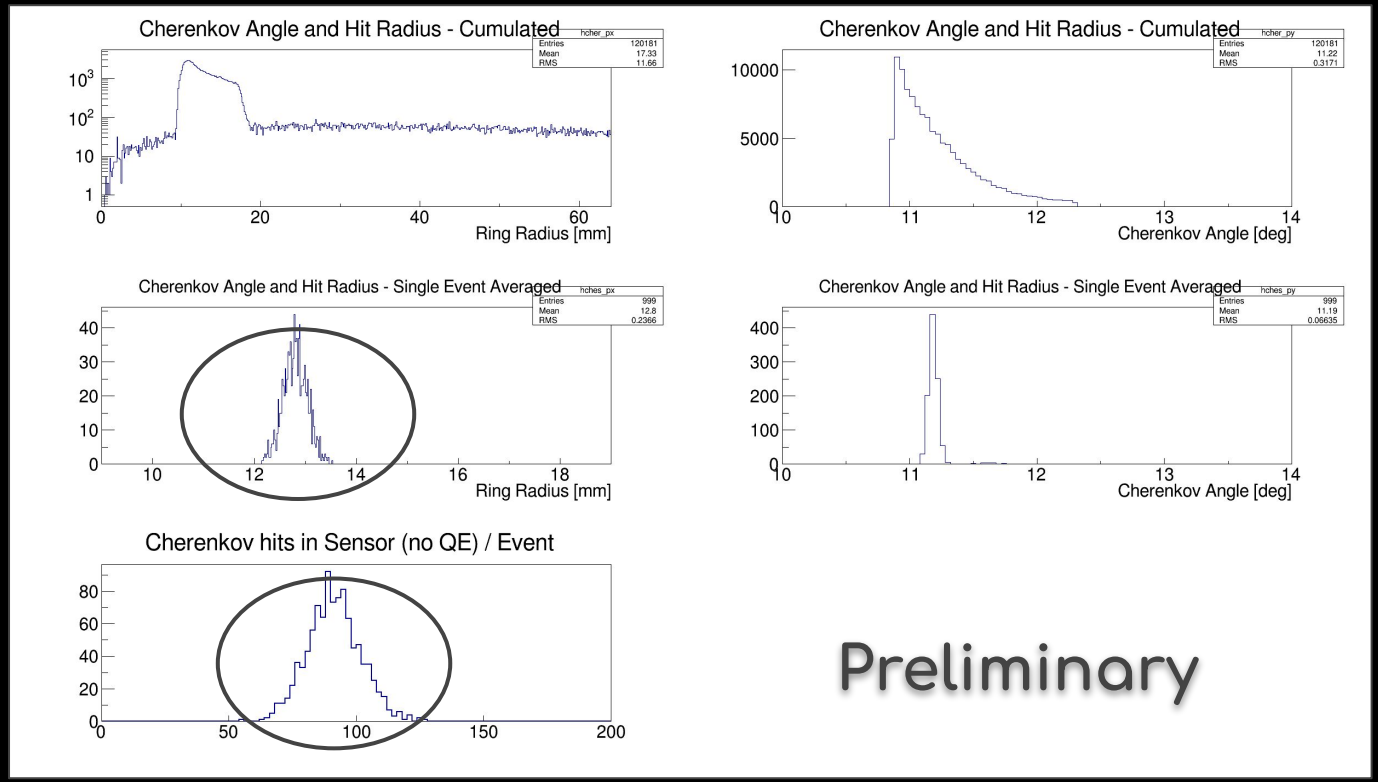


Table of sensitive parameters (fiber and aerogel) used for BO

Variable	Description	# pars	Range
Rigid rotation of tiles f_rotx, f_roty	all fibers rotating by same angle along x, y (z along aerogel thickness)	2	(-5,5), (-5,5) deg
Single fibers rotation f_sthx, f_sthy	used to estimate tolerances on single fiber angles x, y	2	(0.1,1.0) deg
Single fibers shifts f_x, f_y, f_z	to estimate tolerances on single fiber positioning x, y, z	3	(0.5, 3.0) mm
Fiber diameter	Fixed to 50um		
Fiber pitch f_pitch	distance between fibers	1	(5,15) mm
Fiber gap	distance between planes of fibers fixed to 25 mm		
Aerogel thickness	Fixed to 6 cm		
Aerogel width a_width	Side of a square, orthogonal to thickness	1	(8,12) cm
Aerogel refractive index a_n	Allowed to vary	1	(1.01,1.05)

Analysis of Extracted Features and Optimization

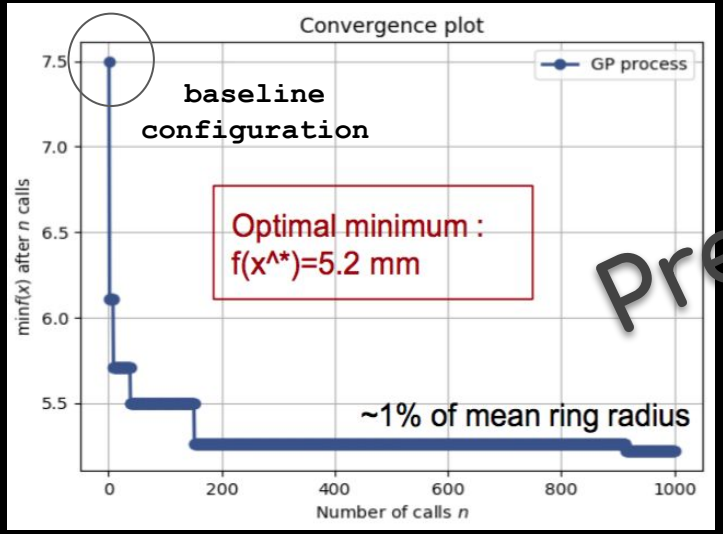
- Resolution from Ring Radius
- Requiring photon yield larger than a threshold

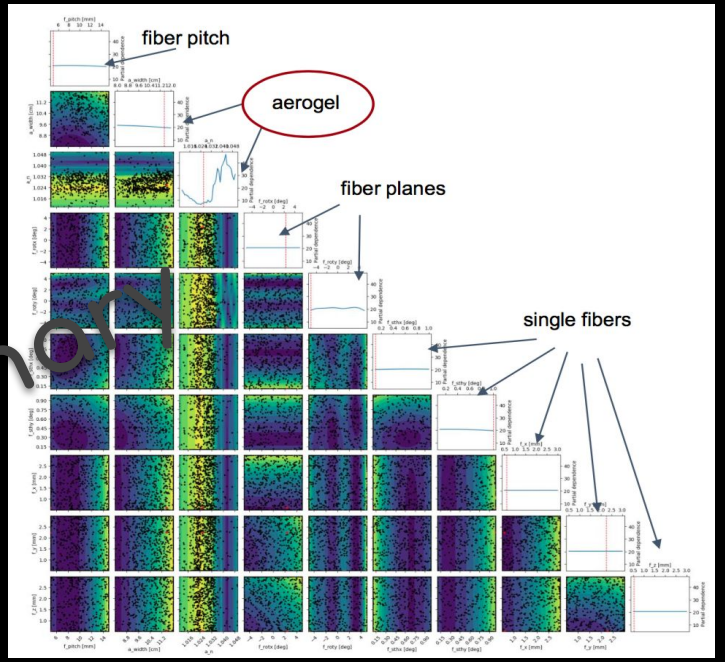


Preliminary Studies based on single objective

- The dominant contribution comes from the aerogel refractive index
- Another optimization suggests a thickness of 6 cm, a fiber gap of 25 mm, and diameter of 50um; with these values the optimal refractive index and width of the aerogel tile are 1.026 and 11.5 cm
- Resolution does not seem to depend critically on fibers --- we have tolerances of up to 1 deg in (θ_x, θ_y) and 3 mm in (x,y,z)

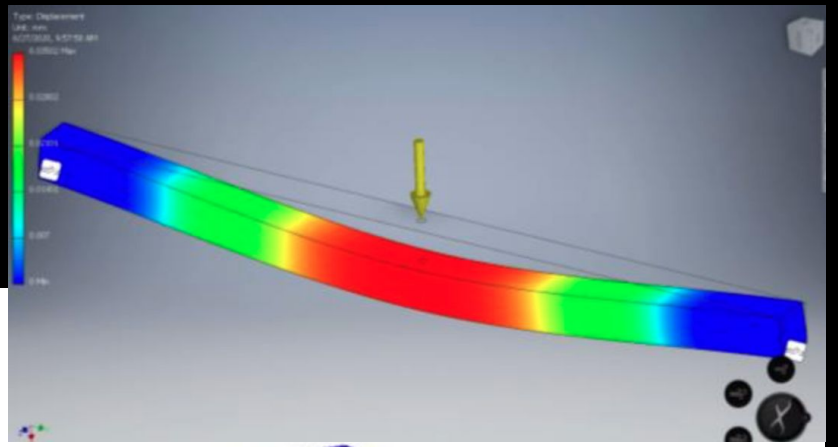
1 single objective:
Resolution

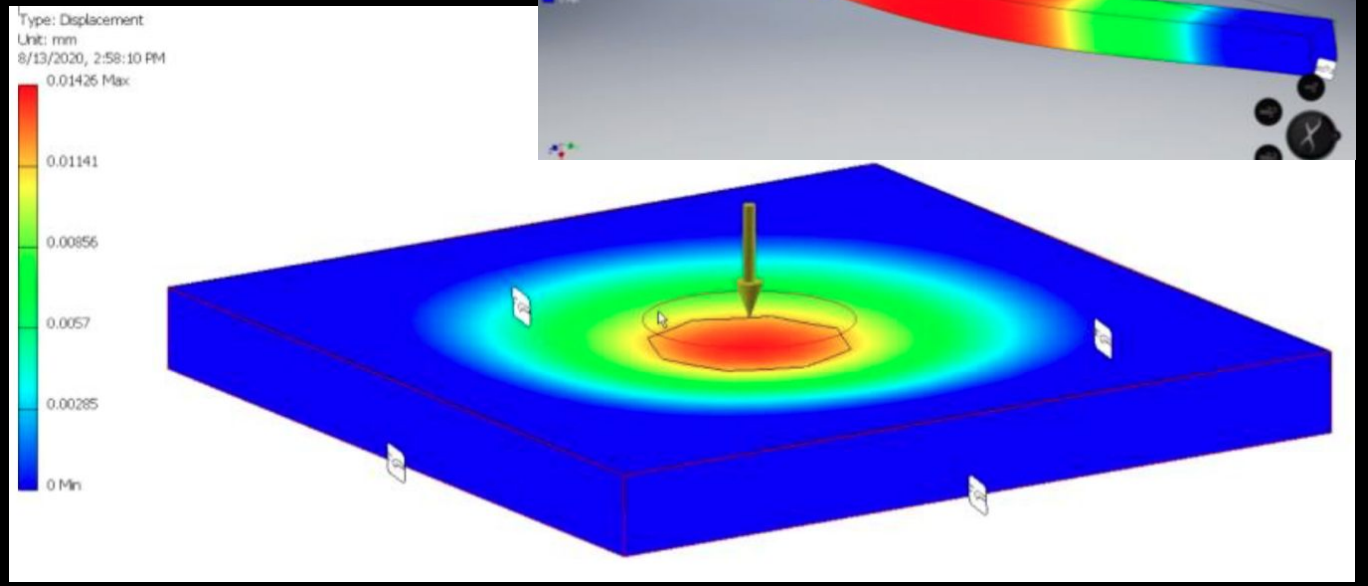




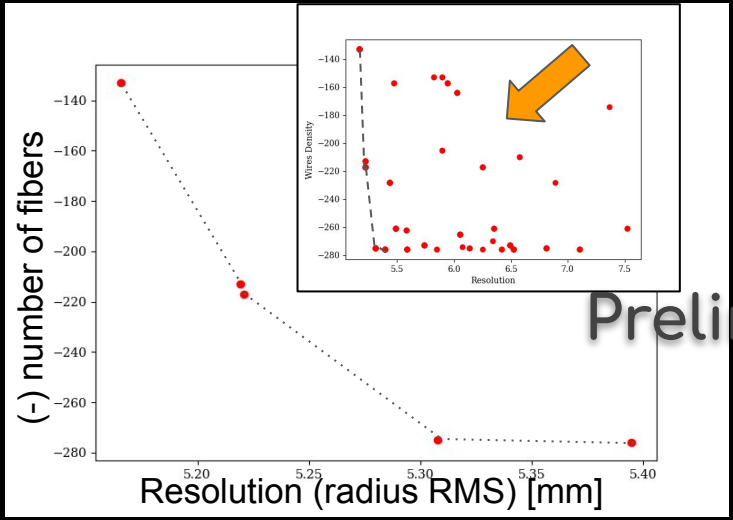
Consider multiple objective functions

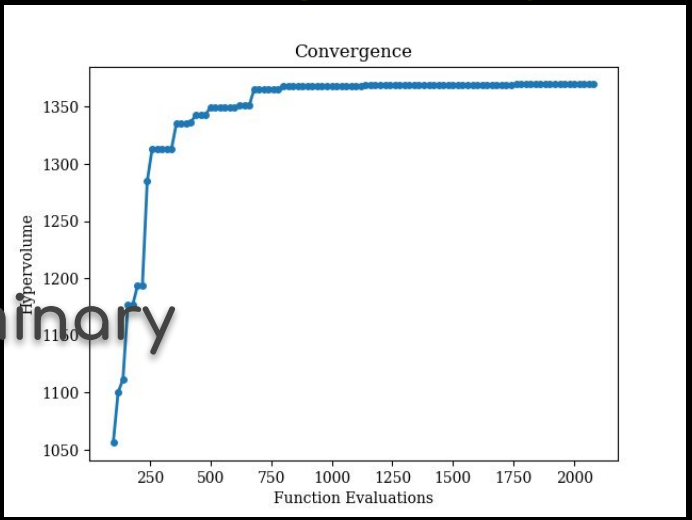
- Objective functions: mechanical strength and resolution (perhaps add cost later as well)
- To develop the mechanical strength function stress simulations in Autodesk Inventor are being developed





Candidate Pareto-front for aer-fib (in progress)





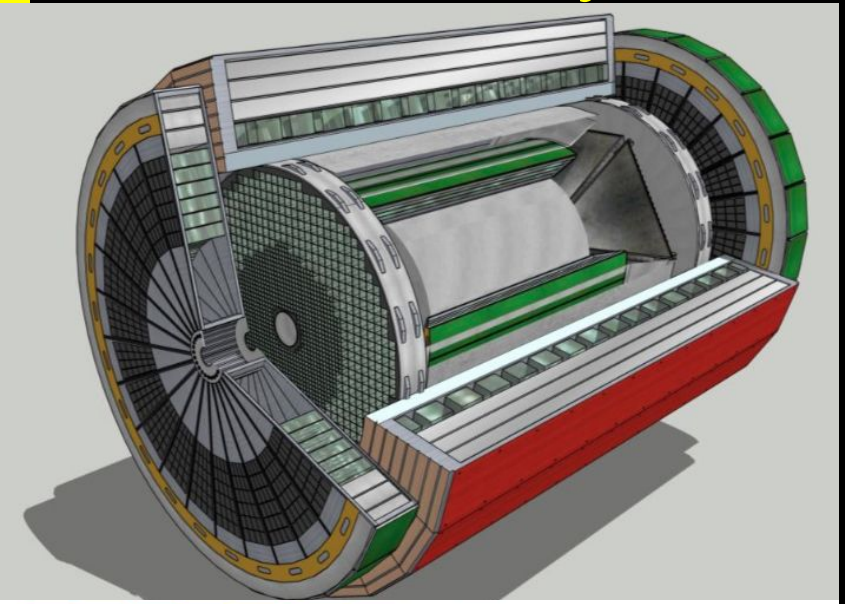
Fiber pitch (5,15) mm	Rot(x) tile (-5,5) deg	Rot(y) tile (-5,5) deg	Rot(x) fiber (0.1,1.0) deg	Rot(y) fiber (0.1,1.0) deg	Shift(x) fiber (0.5, 3.0) mm	Shift(y) fiber (0.5, 3.0) mm	Shift(z) fiber (0.5, 3.0) mm	Diam. fiber (0.05,0.200) mm	Gap planes (10,30) mm	Reso [mm]	Density (# fibers)
10.2	-3.0	3.5	0.3	0.7	1.85	2.9	2.1	0.18	10.2	5.166	133
5.2	-0.3	-1.7	0.7	0.9	2.2	1.5	1.6	0.06	12.1	5.221	217
5.0	-3.2	-3.2	0.2	0.2	2.2	1.3	1.2	0.09	10.0	5.308	275
5.0	0.9	3.4	0.9	0.1	2.3	1.6	3.0	0.06	10.0	5.394	276
5.9	1.1	3.6	0.9	0.2	1.3	1.6	3.6	0.09	11.0	5.219	213

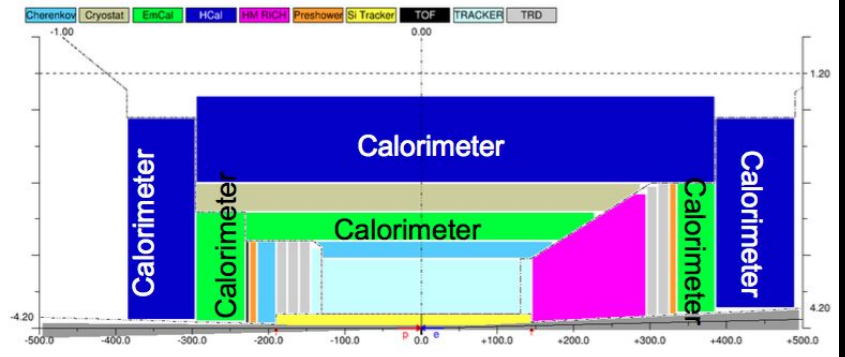
Kept fixed the aerogel thickness to 6 cm, the width to 11.5 cm, and the refractive index to 1.026.

Compared to BO, we obtained a ~similar resolution but could increase the stability by a factor 2 (using a smaller gap between planes).

The proxy used for the stability is the number of fibers.

EIC Calorimetry



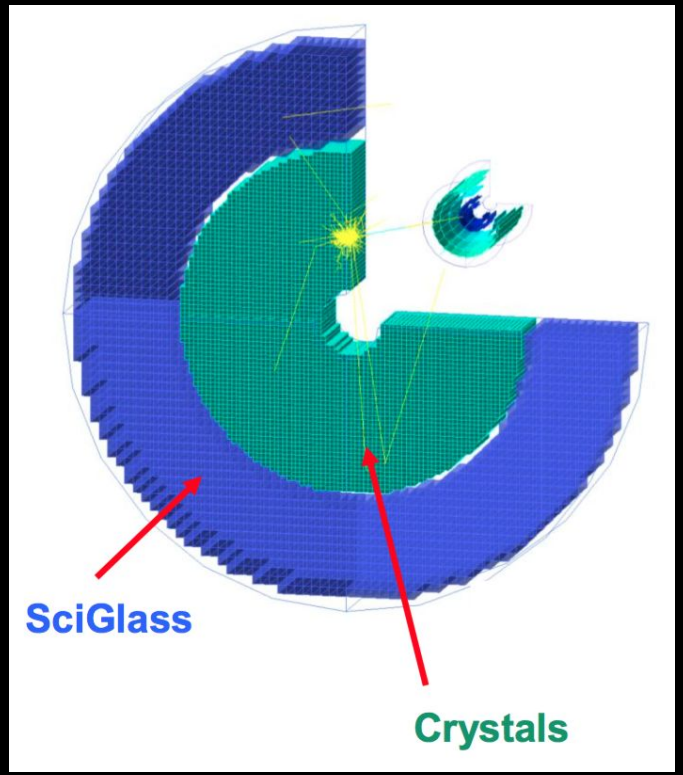


The team: V. Berdnikov, M. Bondi', C. Fanelli, Y. Furletova, T. Horn, I. Larin, D. Romanov, R. Trotta

- EIC central detector requires EM and EM/hadronic calorimetry in the barrel and the two endcaps
- Material selection is important for balancing calorimeter performance and cost

EIC EMCal Electron Endcap

Use MOO to optimize glass/crystal material selection in shared rapidity regions

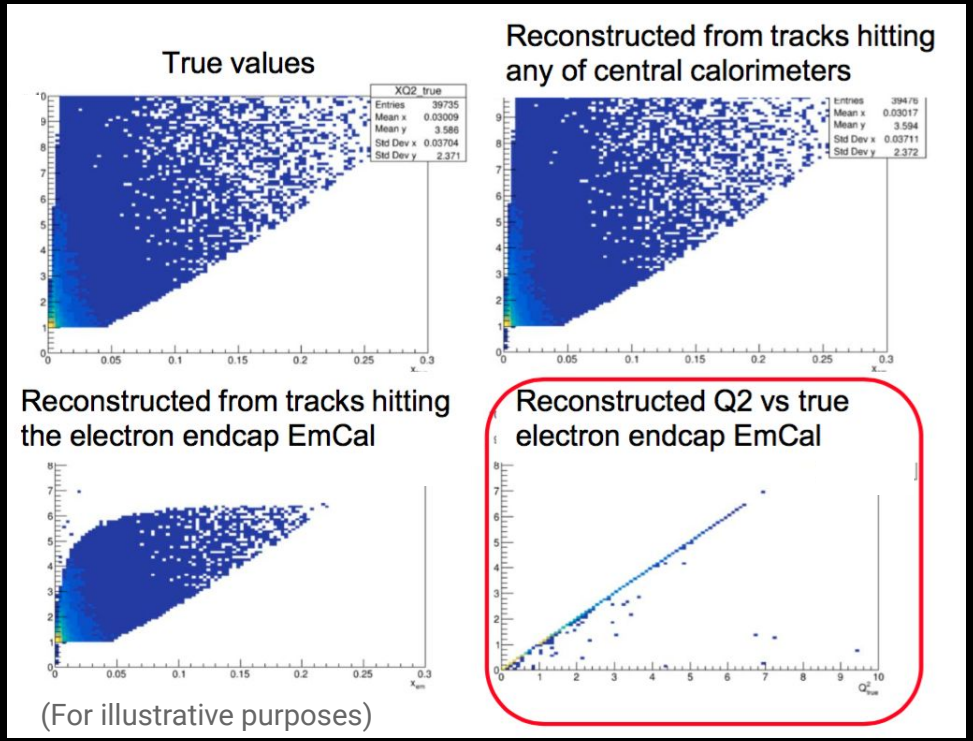


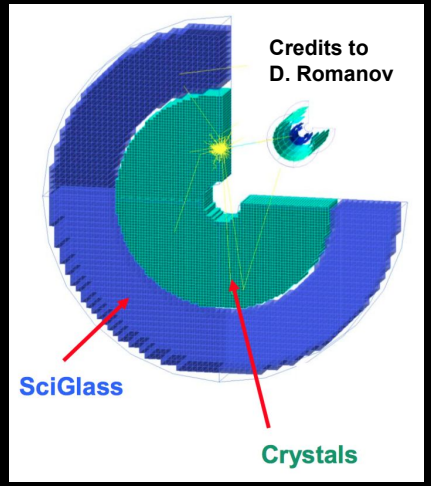
- EIC Electron Endcap requires an inner part (crystal) with high resolution and an outer part (glass) with less stringent requirements
  - Crystals have been used in homogeneous calorimeters but their production is slow and expensive.
  - As an alternative Scintilex develops SciGlass that is much simpler and less expensive to produce and thus offers great potential for both cost reduction and wider application if competitive performance parameters can be achieved.

Goal: maintain the resolution needed by the physics processes (*e.g.*, DIS) while reducing the number of crystals, taking into account constraints in the detector.

EIC EMCal Electron Endcap

Similar to the aerogel approach,
but also introduce constraints of the detector.



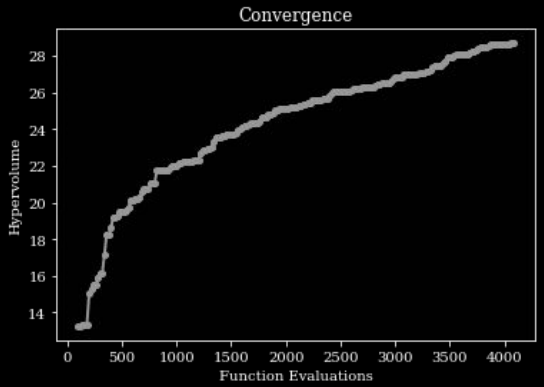


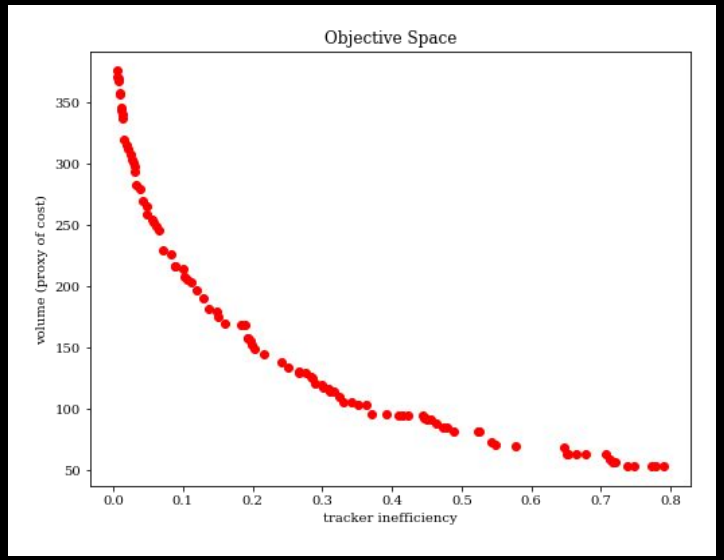
- Include FOMs based on physics processes (DIS)
- Hybrid calorimeter reconstruction algorithm being developed. Other solutions (hierarchical clustering etc.) will be investigated.
- Analysis of impact of EMCal resolution on reconstructed quantities

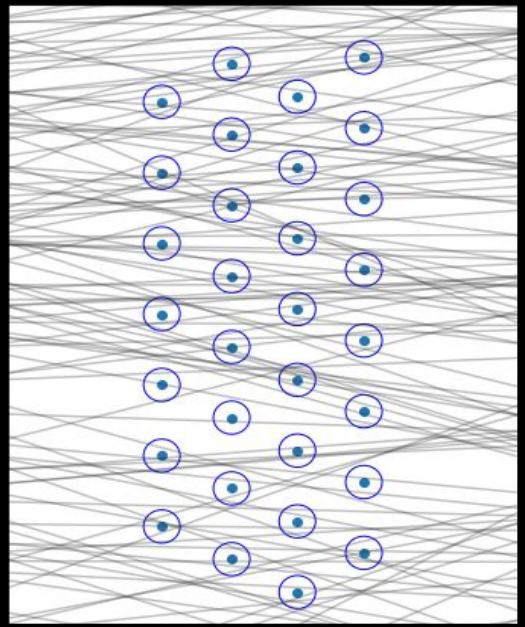
Toy Model: MOO - 2 objectives

Objectives:

- max Tracking Efficiency
- min Tot Material/Cost (N_{wires} * A)





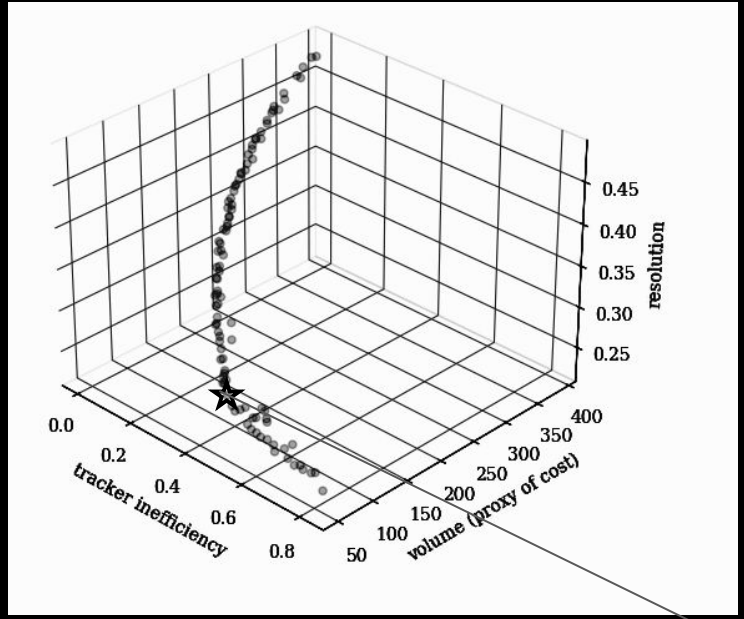


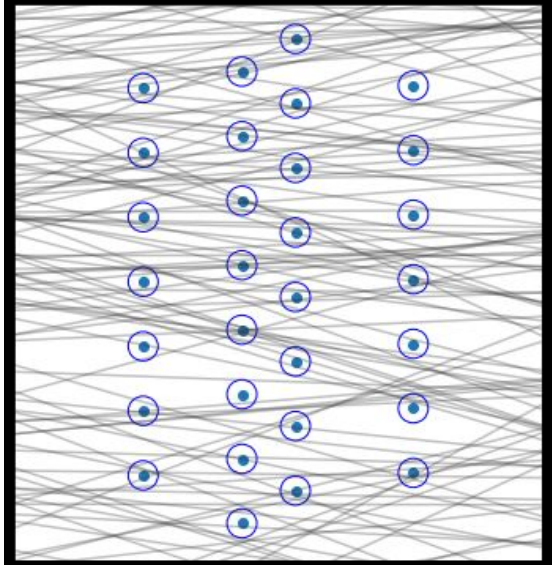
E.g.,
fraction of tracks detected: 0.79
volume: 137.8

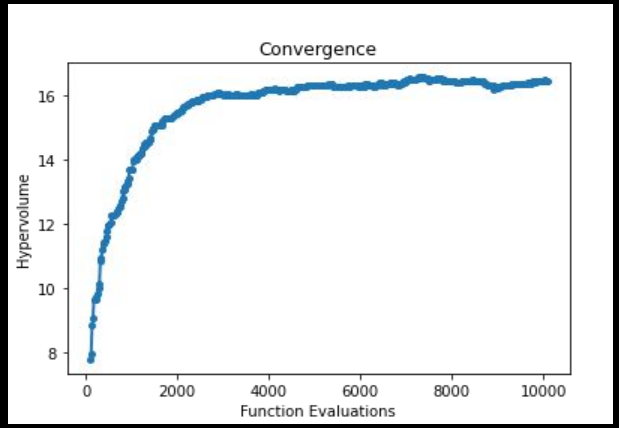
Toy Model: MOO - 3 objectives

Objectives:

- max Tracking Efficiency
- min Tot Material/Cost (N_{wires} * A)
- min Resolution (residual along y with respect to center)







E.g.,
fraction of tracks detected: 0.72
volume: 126.87
resolution: 0.27

Conclusions

- Detectors are essential tools for NP. They have been designed for years using standard driving criteria (objective functions) for each individual subsystem (e.g., the resolution characterizing the intrinsic detector response) considering the constraints from the full detector, with the full detector design being studied after the subsystem prototypes are ready.

- In NP we just started exploring AI for optimal designs in multidimensional space with single objectives. Most of the problems are multi-objectives though.

- None ever accomplished a multi-dimensional / multi-objective optimization of the performance of detectors when operating together. This is a high-dimensional combinatorial problem (with many parameters) that can be solved with AI.

- Likely future detectors will be designed with AI achieving optimal performance and cost reduction.