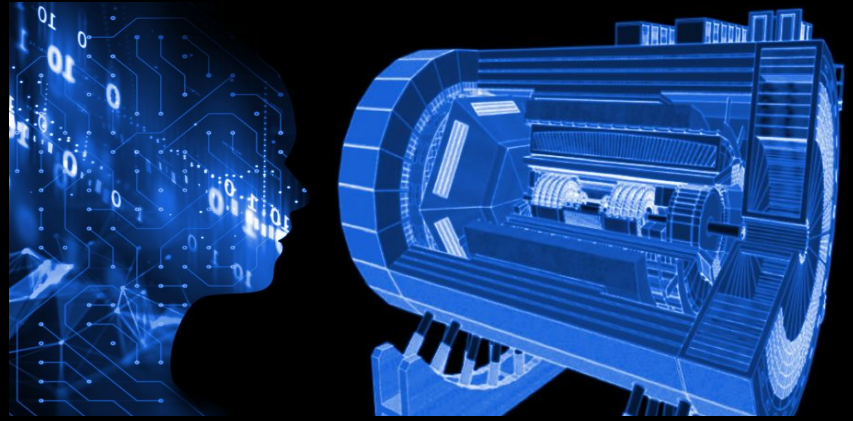


AI for Detector Design

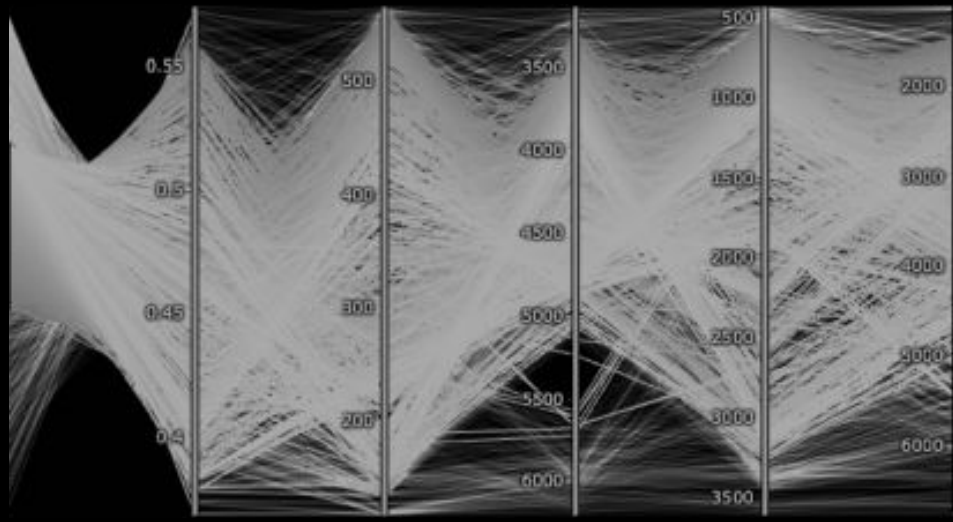
National Nuclear Physics Summer School
MIT, 2022

Cristiano Fanelli

EIC Center at Jefferson Lab

Lecture 2

Multiple (Competing) Objectives



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

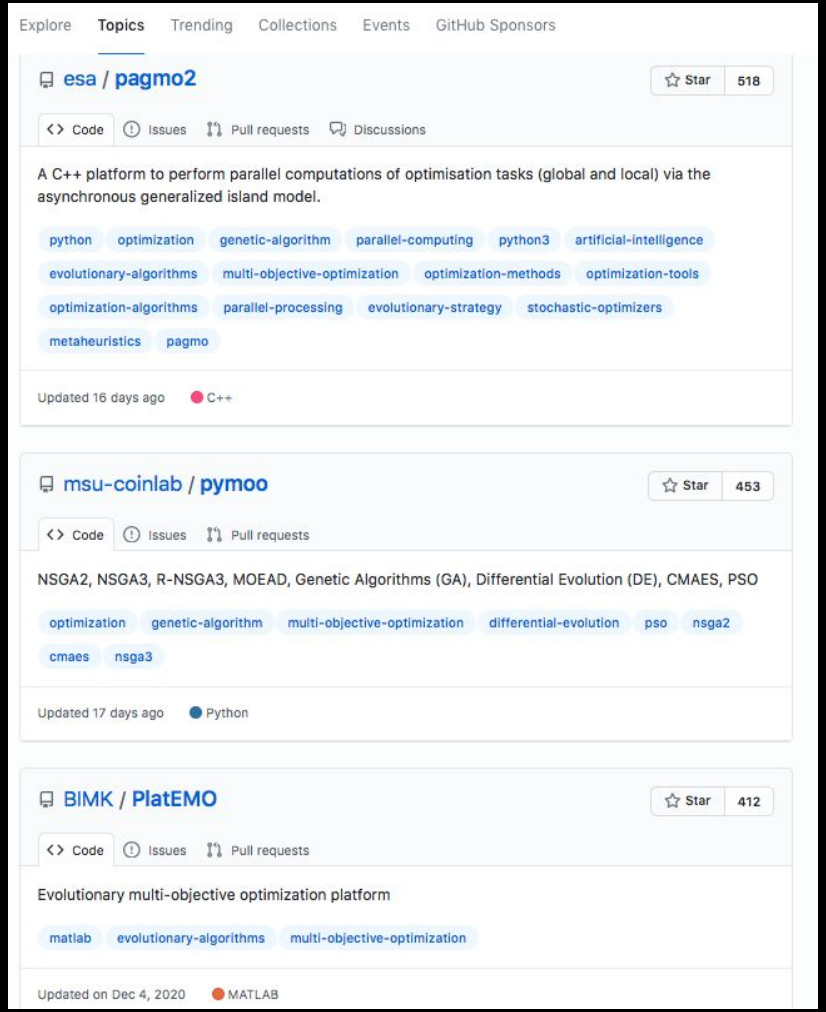
Frameworks

- 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

[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

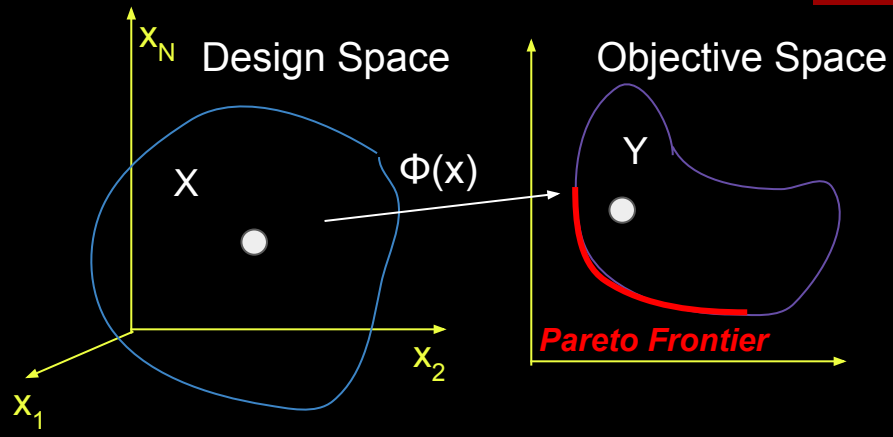
- In the following we will refer to the multi-objective optimization based on evolutionary algorithms [1], and in particular pymoo [2], written in Python, which also includes visualization and decision making tools.
- The definition of a generic MOO problem can be formulated as:

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

- 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.

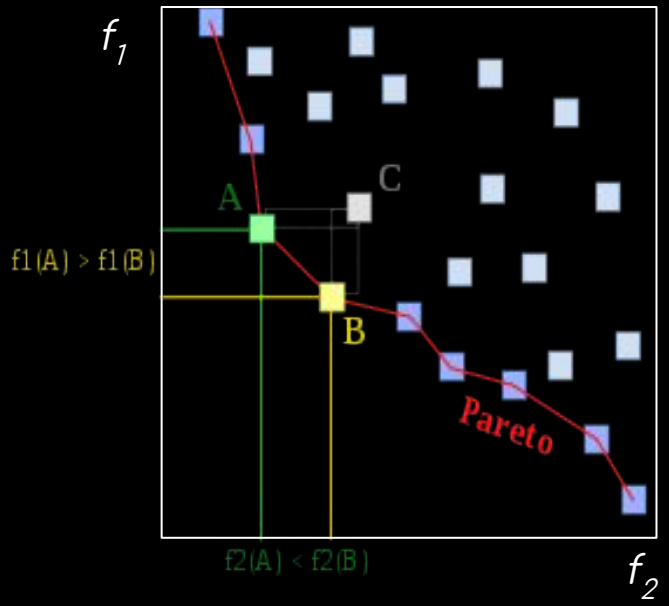
MOO

- 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).

Multi-Objective Optimization



- 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.
- In solving such problems, with or without constraints, they give rise to a trade-off optimal solutions, popularly known as Pareto-optimal solutions.
- 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.
- MO-based solutions are helping to reveal important hidden knowledge about a problem – a matter which is difficult to achieve otherwise

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

EMO: Evolutionary Multi-Objective 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.

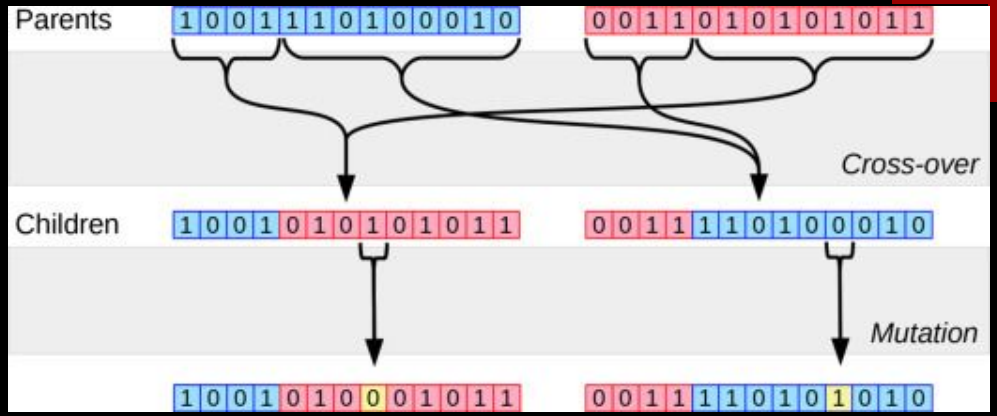
The initialization usually involves a random creation of solutions. it is highlighted that for solving complex real-world optimization problems, a customized initialization is helpful in achieving a faster search.

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

- 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.

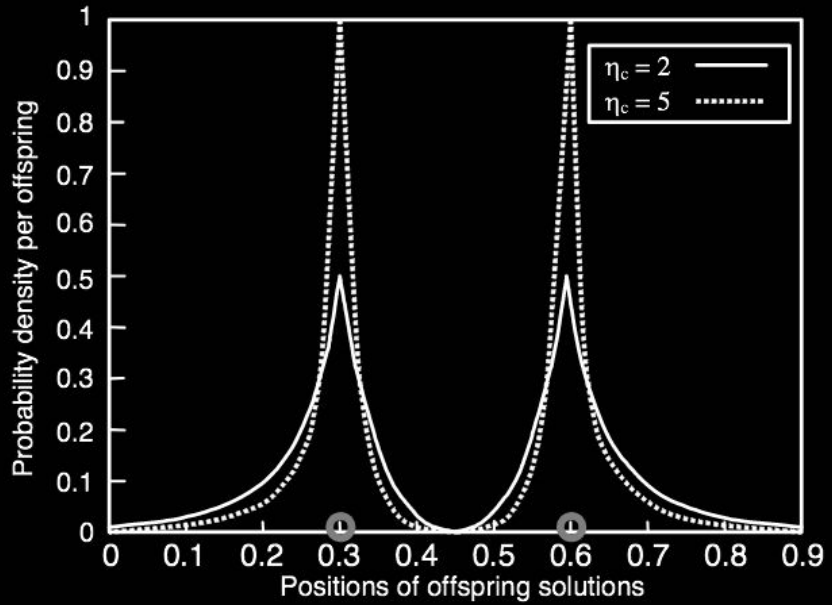
SBX Algorithm:

- 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:

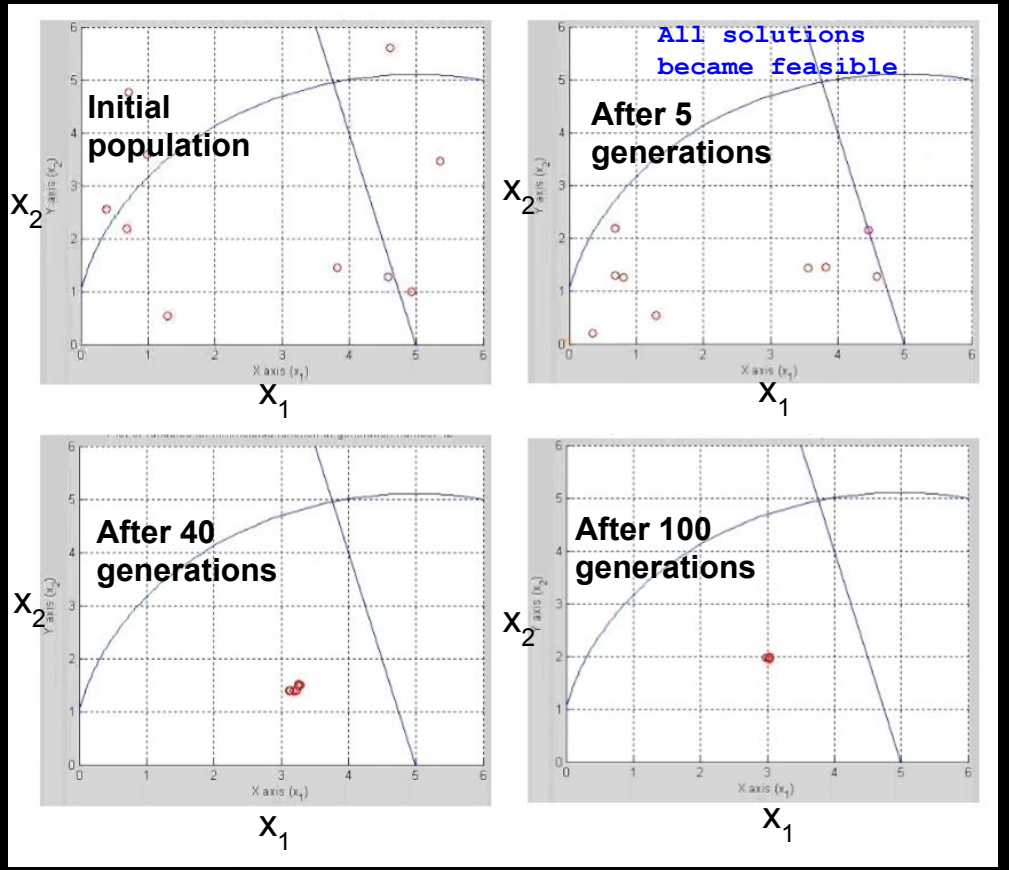
$$x_1^{new} = 0.5[(1+\beta)x_1 + (1-\beta)x_2]$$
$$x_2^{new} = 0.5[(1-\beta)x_1 + (1+\beta)x_2]$$



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

Phases of Evolution

- 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

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

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 typically scales as MN^2 — faster versions of NSGA-II reduce it to O(N log N) for 2 and 3 objectives, and $O(N (\log N)^{M-2})$ for $M > 3$ objectives for an improved NSGA-II [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

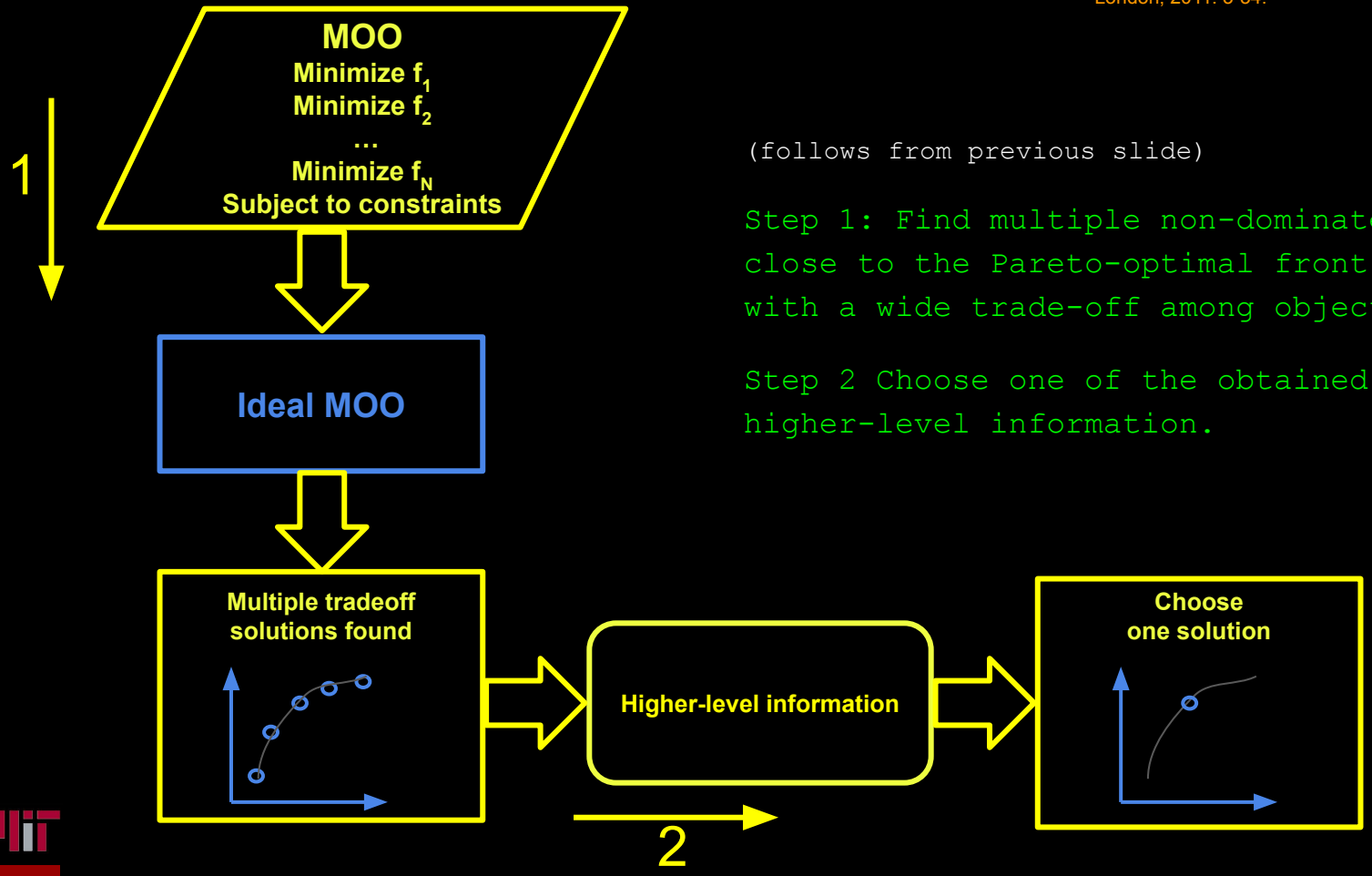
- Since a number of solutions are optimal, the obvious question arises: which of these optimal solutions must one choose?
- Answers this typically involves higher-level information which is often non-technical, qualitative and experience-driven. One has to evaluate the pros and cons of each of these 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

[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.

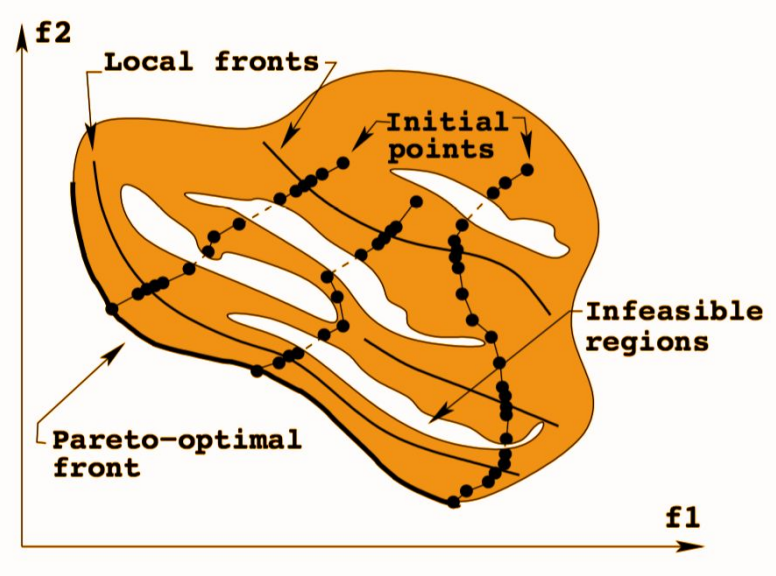


(follows from previous slide)

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.

Single objective(s) VS MOO



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…

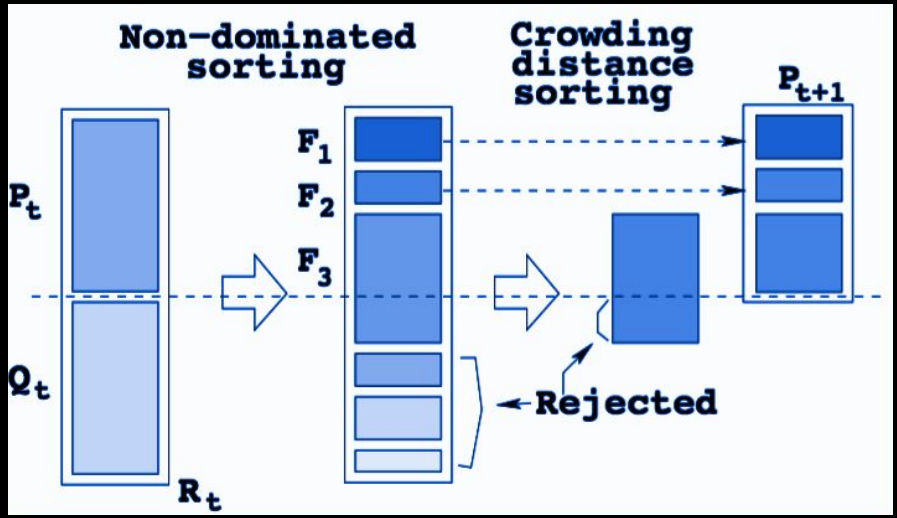
Finding multiple trade-off solutions is a parallelly processed task.

Elitist Non-Dominated Sorting GA (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:

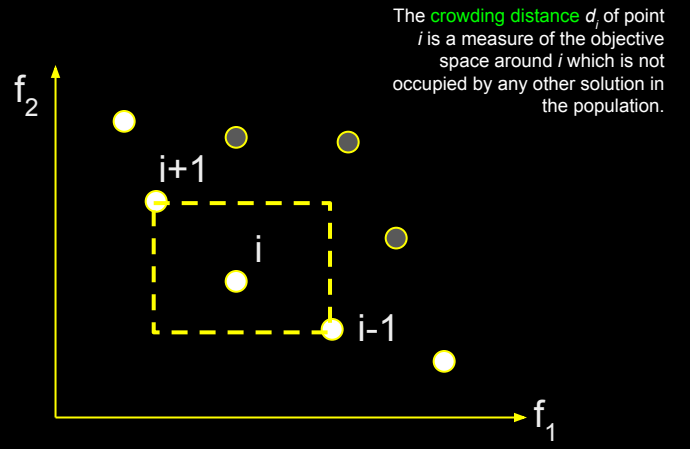
- ❑ Use of an elitist principle,
- ❑ Explicit diversity preserving mechanism
- ❑ Emphasis in non-dominated solutions.



- 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

- 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.

Handling Constraints 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* $\mathbf{x}^{(i)} \preceq_c \mathbf{x}^{(j)}$*), 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

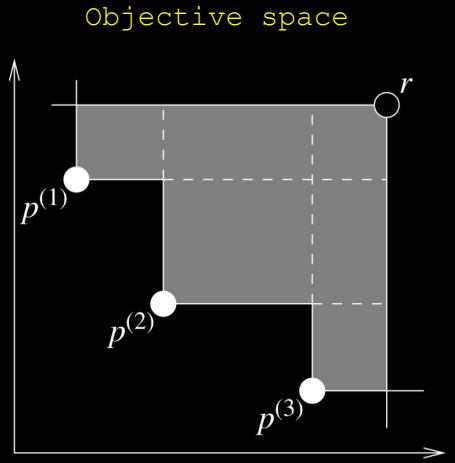
Performance Measures

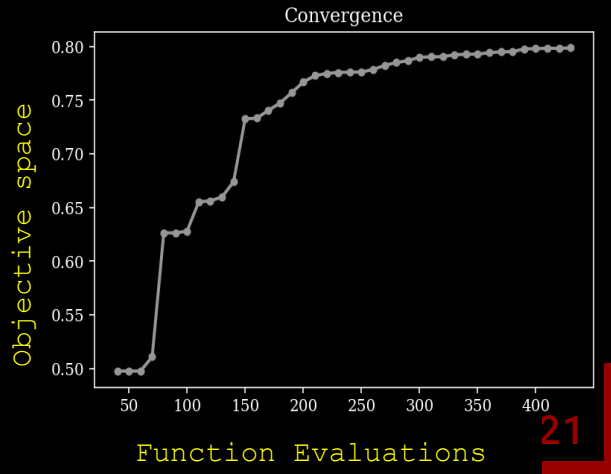
[1] *Zitzler E, Thiele L, Laumanns M, Fonseca CM, Fonseca VG. Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation. 2003;7(2):117– 132.*

- 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.)
  - 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. — *e.g., importance wrt reference

  - Pseudo-Weights: a more intuitive way, where pseudo-weights are defined as:

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

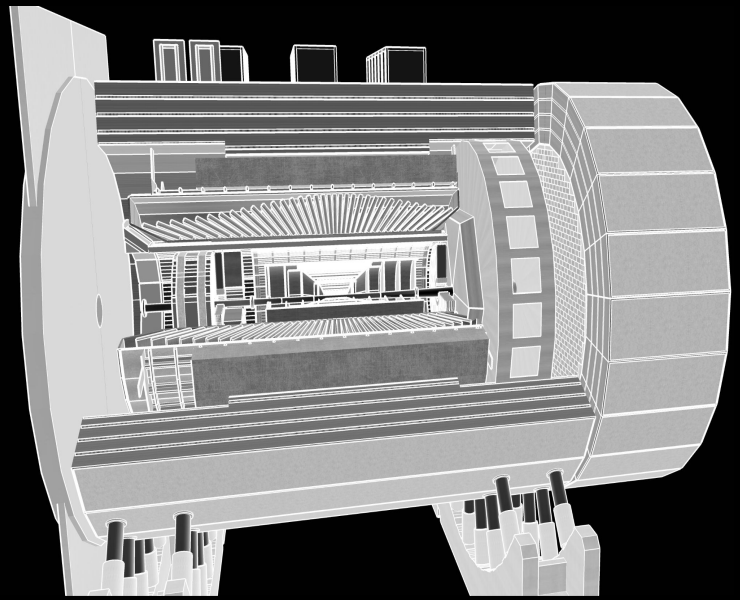
Namely, the normalized distance to the worst solution regarding each objective `i` is calculated.

Some Practical Aspects

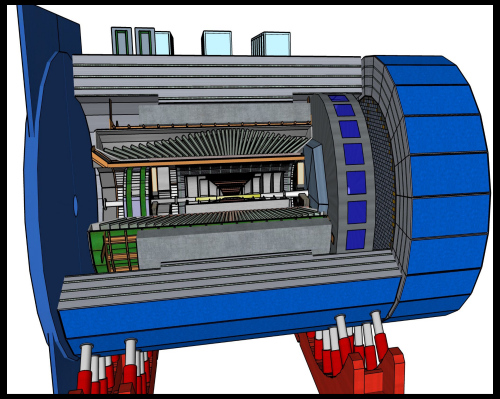
[1] V. Khare, X. Yao, K. Deb. Performance Scaling of Multi-objective Evolutionary Algorithms. In: Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference (LNCS 2632); 2003. p. 376–390.

- EMO is an established successful procedure since many years.
- 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.

The ECCE Example



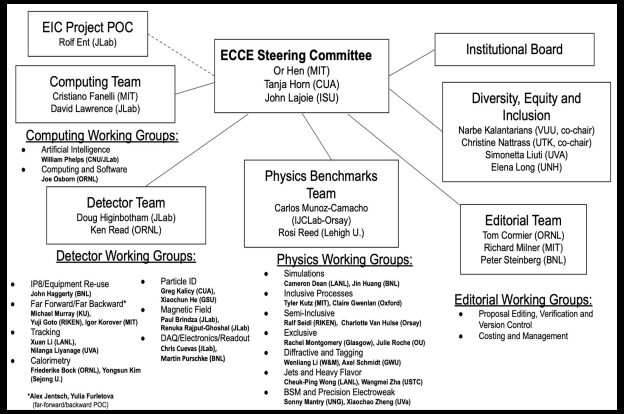
EIC Comprehensive Chromodynamics Experiment



98 institutions

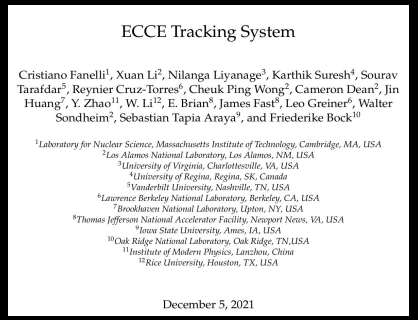
Develop low-risk, cost-effective, flexible and optimized EIC detector

Detector concept based on a 1.5 T magnet



A proposal (60+ pages) [end of 2021] recommended by the DPAP as the reference detector

Used AI during the proposal for the design of the detector concept

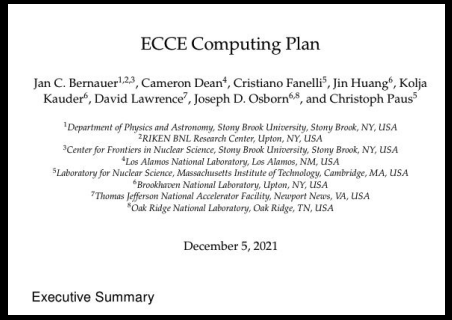


ECCE Tracking System

Cristiano Fanelli[1], Xuan Li[2], Nilanga Liyanage[3], Karthik Suresh[4], Sourav Tarafdar[5], Reynier Cruz-Torres[6], Cheuk Ping Wong[2], Cameron Dean[2], Jin Huang[7], Y. Zhao[11], W. Li[12], E. Brian[8], James Fast[8], Leo Greiner[6], Walter Sondheim[2], Sebastian Tapia Araya[9], and Friederike Bock[10]

[1]*Laboratory for Nuclear Science, Massachusetts Institute of Technology, Cambridge, MA, USA*
[2]*Los Alamos National Laboratory, Los Alamos, NM, USA*
[3]*University of Virginia, Charlottesville, VA, USA*
[4]*University of Regina, Regina, SK, Canada*
[5]*Vanderbilt University, Nashville, TN, USA*
[6]*Lawrence Berkeley National Laboratory, Berkeley, CA, USA*
[7]*Brookhaven National Laboratory, Upton, NY, USA*
[8]*Thomas Jefferson National Accelerator Facility, Newport News, VA, USA*
[9]*Iowa State University, Ames, IA, USA*
[10]*Oak Ridge National Laboratory, Oak Ridge, TN,USA*
[11]*Institute of Modern Physics, Lanzhou, China*
[12]*Rice University, Houston, TX, USA*

December 5, 2021



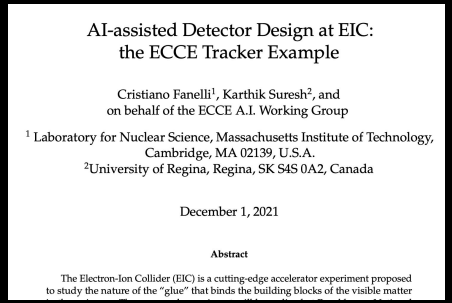
ECCE Computing Plan

Jan C. Bernauer[1,2,3], Cameron Dean[4], Cristiano Fanelli[5], Jin Huang[6], Kolja Kauder[6], David Lawrence[7], Joseph D. Osborn[6,8], and Christoph Paus[5]

[1]*Department of Physics and Astronomy, Stony Brook University, Stony Brook, NY, USA*
[2]*RIKEN BNL Research Center, Upton, NY, USA*
[3]*Center for Frontiers in Nuclear Science, Stony Brook University, Stony Brook, NY, USA*
[4]*Los Alamos National Laboratory, Los Alamos, NM, USA*
[5]*Laboratory for Nuclear Science, Massachusetts Institute of Technology, Cambridge, MA, USA*
[6]*Brookhaven National Laboratory, Upton, NY, USA*
[7]*Thomas Jefferson National Accelerator Facility, Newport News, VA, USA*
[8]*Oak Ridge National Laboratory, Oak Ridge, TN, USA*

December 5, 2021

Executive Summary



AI-assisted Detector Design at EIC:
the ECCE Tracker Example

Cristiano Fanelli[1], Karthik Suresh[2], and
on behalf of the ECCE A.I. Working Group

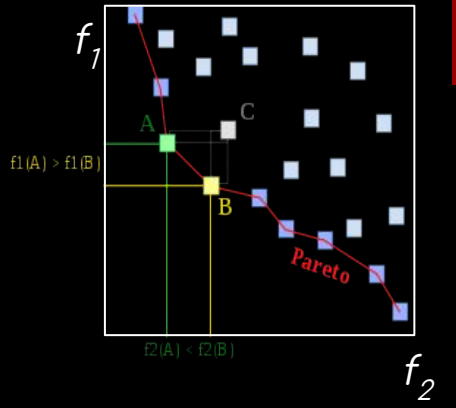
[1] Laboratory for Nuclear Science, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.
[2]University of Regina, Regina, SK S4S 0A2, Canada

December 1, 2021

Abstract

The Electron-Ion Collider (EIC) is a cutting-edge accelerator experiment proposed to study the nature of the "glue" that binds the building blocks of the visible matter

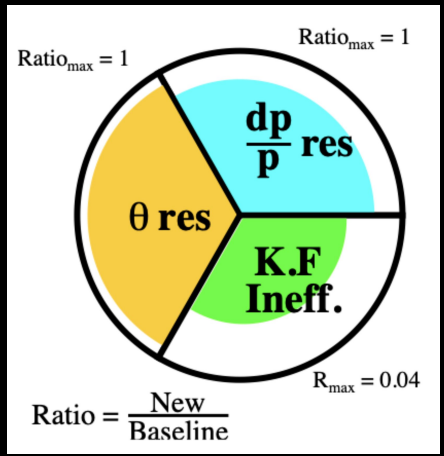
Multi-Objective Optimization



- During the proposal we used both evolutionary Multi-Objective Optimization

The ECCE Inner Tracker Design Optimization considers simultaneously:

- momentum resolution
- angular resolution
- Kalman filter efficiency
- (pointing resolution)
- Mechanical constraints

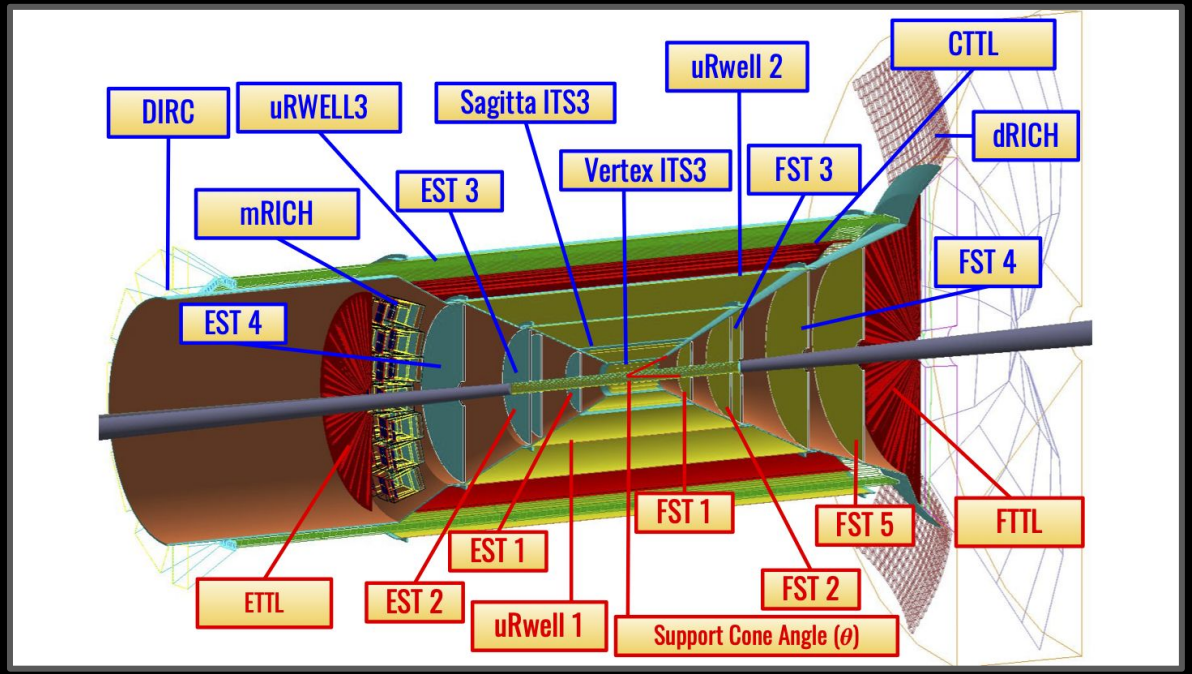


EIC Detector Tracker

CF, K. Suresh, Z. Papandreou et al (ECCE)

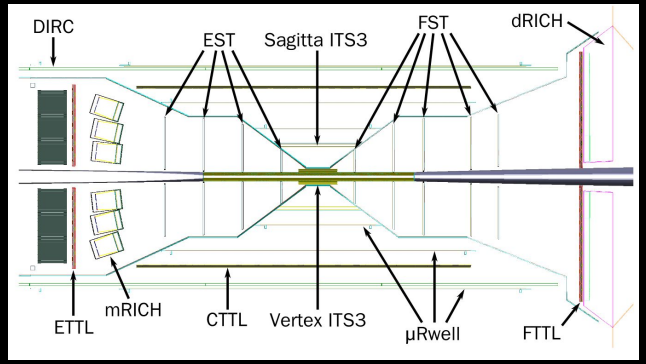
AI-assisted Optimization of the ECCE Tracking System at the Electron Ion Collider

arXiv:2205.09185



EIC Detector Tracker

Sub Detector System	No Of Layers	Technology	Pitch/res [μm]	Thickness [X/X0]	Description
Vertex Barrel	3	MAPS-ITS3	10	0.05	Monolithic Active Pixel Sensor; EIC R&D eRD111. High precision tracking.
Sagitta Barrel	2	MAPS-ITS3	10	0.05	Monolithic Active Pixel Sensor; EIC R&D eRD11. High precision tracking.
Outer Barrel	3	μRwell	55	0.2	μRwell is a gaseous based tracker. EIC R&D ERD6. Low Cost tracking solution
CTTL (TOF)	1	AC-LGAD	30	~0.1	Low Gain Avalanche Detectors (ACLGAD): EIC R&D ERD112. High precision tracking and Timing.
EST	4	MAPS-ITS3	10	0.3	Monolithic Active Pixel Sensor; EIC R&D eRD11. High precision tracking.
FST	5	MAPS-ITS3	10	0.3	Monolithic Active Pixel Sensor; EIC R&D eRD111. High precision tracking.
ETTL	1	AC-LGAD	30	~0.1	Low Gain Avalanche Detectors (ACLGAD): EIC R&D ERD112. High precision tracking and timing
FTTL	1	AC-LGAD	30	~0.1	Low Gain Avalanche Detectors (ACLGAD): EIC R&D ERD112. High precision tracking and timing

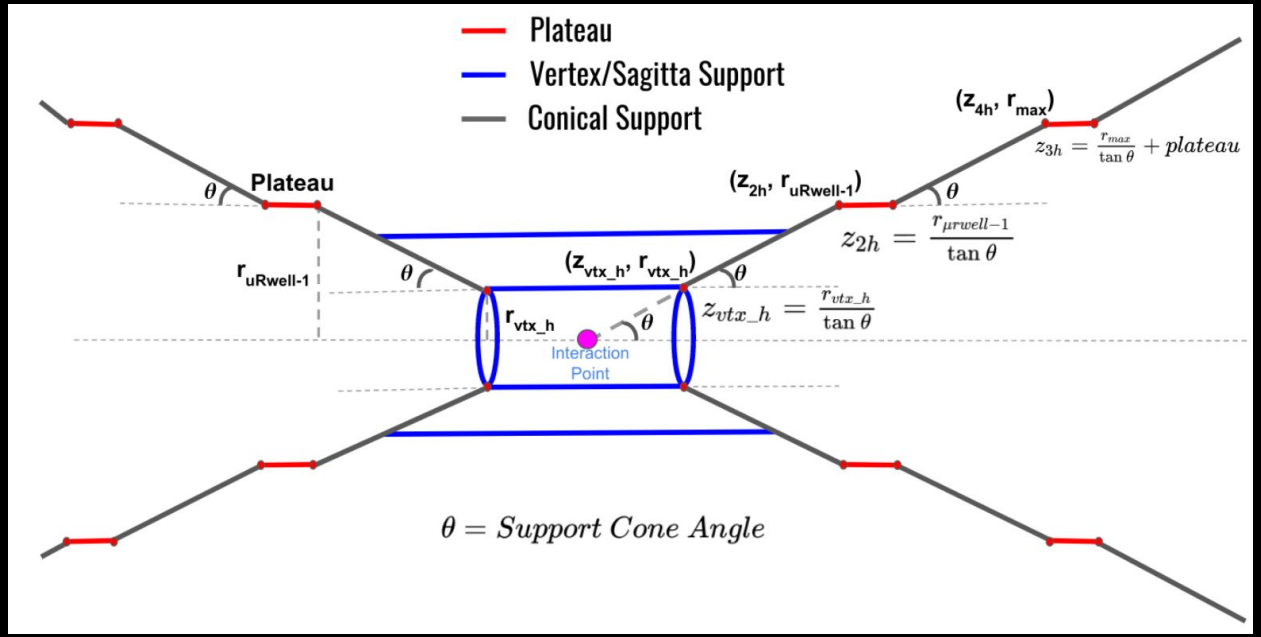


ECCE design (non-projective)	
Design Parameter	**Range**
μRWELL 1 (Inner) (r) Radius	[17.0, 51.0 cm]
μRWELL 2 (Inner) (r) Radius	[18.0, 51.0 cm]
EST 4 z position	[-110.0, -50.0 cm]
EST 3 z position	[-110.0, -40.0 cm]
EST 2 z position	[-80.0, -30.0 cm]
EST 1 z position	[-50.0, -20.0 cm]
FST 1 z position	[20.0, 50.0 cm]
FST 2 z position	[30.0, 80.0 cm]
FST 3 z position	[40.0, 110.0 cm]
FST 4 z position	[50.0, 125.0 cm]
FST 5 z position	[60.0, 125.0 cm]
ECCE ongoing R&D (projective)	
Design Parameter	**Range**
Angle (Support Cone)	[25.0°, 30.0°]
μRWELL 1 (Inner) Radius	[25.0, 45.0 cm]
ETTL z position	[-171.0, -161.0 cm]
EST 2 z position	[45, 100 cm]
EST 1 z position	[35, 50 cm]
FST 1 z position	[35, 50 cm]
FST 2 z position	[45, 100 cm]
FST 5 z position	[100, 150 cm]
FTTL z postion	[156, 183 cm]

Parametrization

arXiv:2205.09185

Parametrization of the support structure



Parametrization of disks radii and TTL

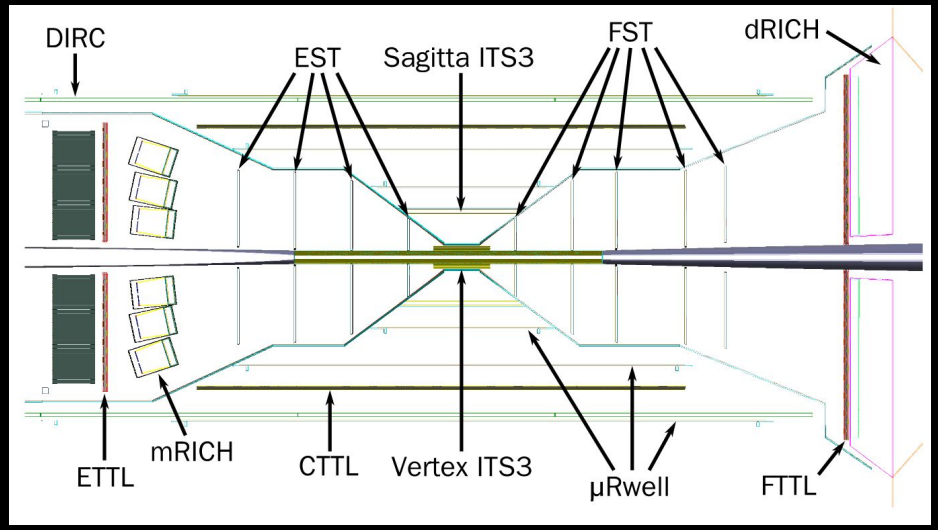
Implementation of Geometric Constraints

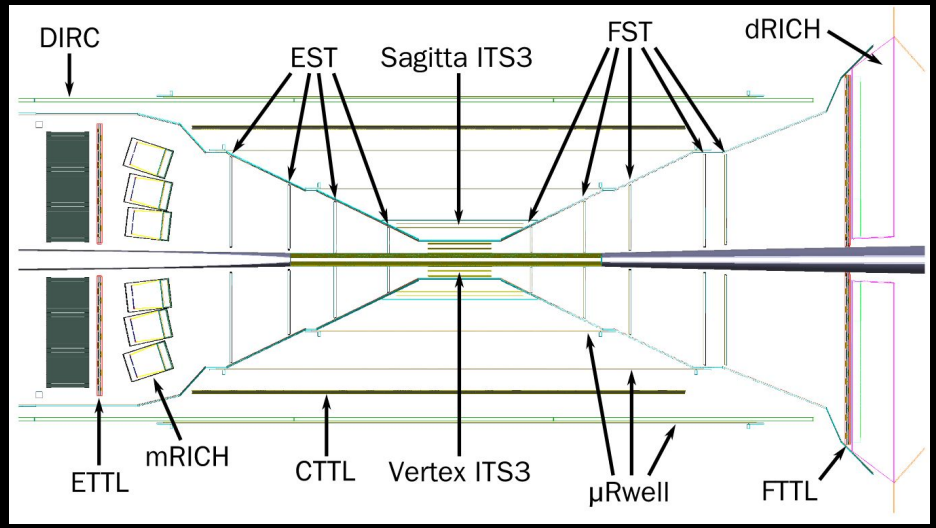
RMax and RMin of the disks are then calculated based on the support structure.

Sagitta Length fixed and Radius changed based on the cone angle.

Parametrization underlies the AI-assisted design and can explore non-projective as well as projective

Reference VS Projective (R&D)





Parametrization underlies the AI-assisted design and can explore non-projective as well as projective

Reference VS Projective (R&D)

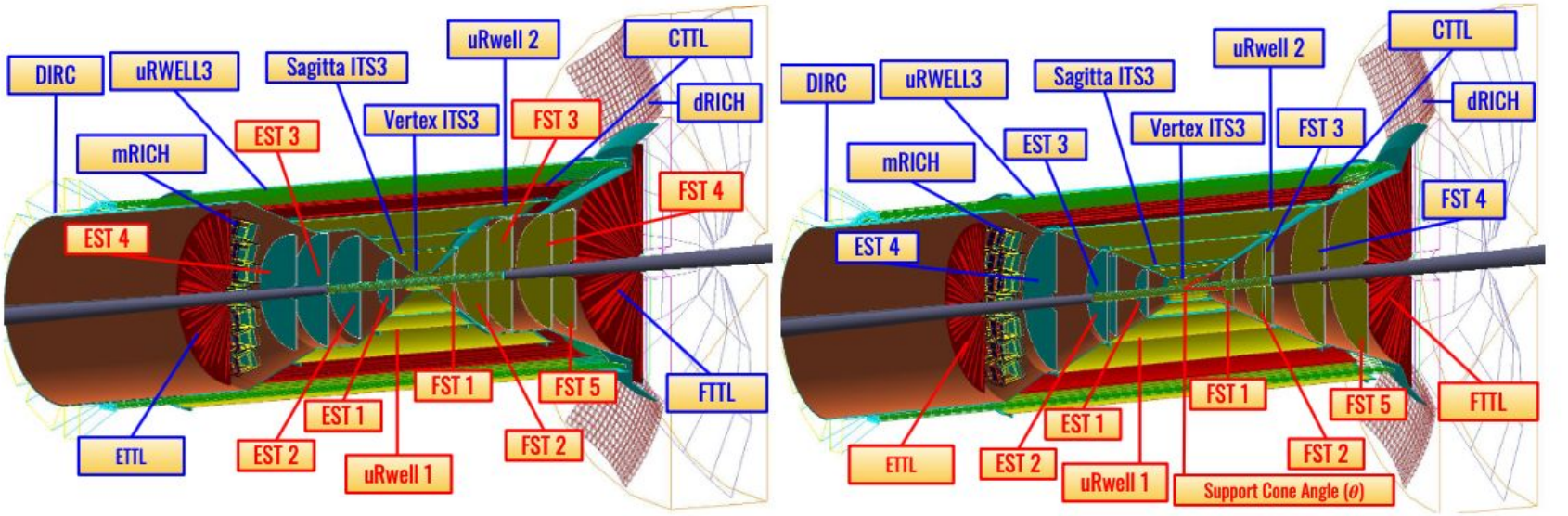
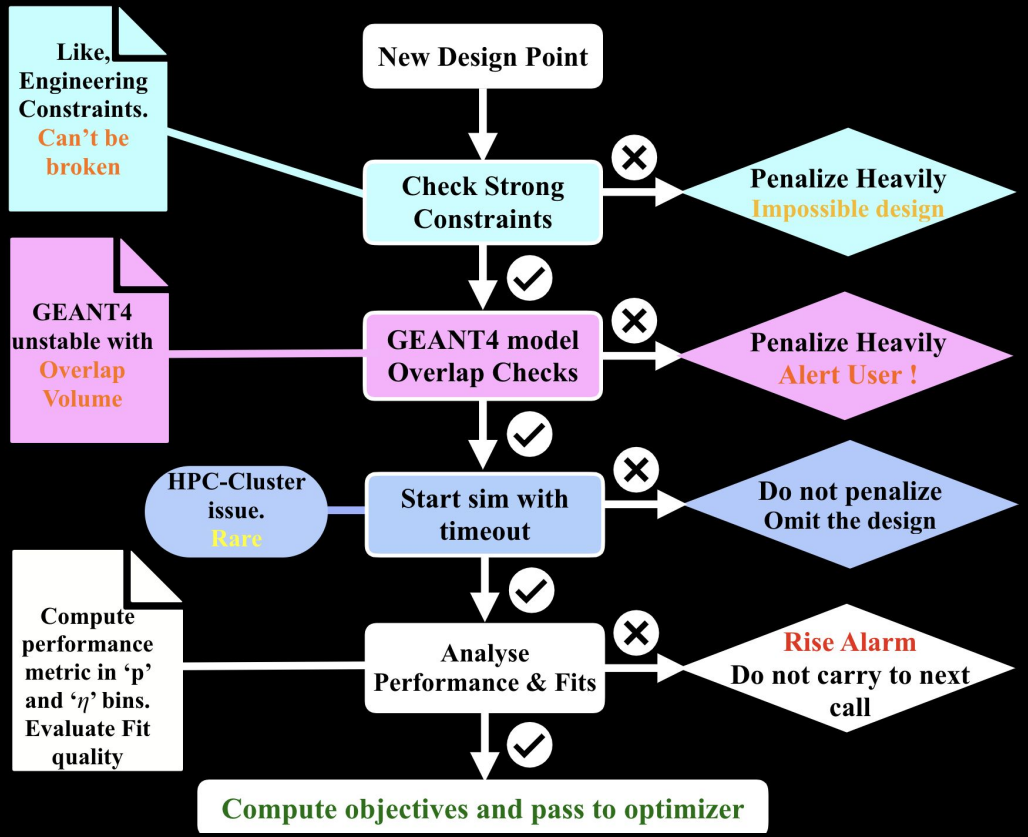


Figure 5: **Tracking and PID system in the non-projective (left) and the ongoing R&D projective (right) designs:** the two figures show the different geometry and parametrization of the ECCE non-projective design (left) and of the ongoing R&D projective design to optimize the support structure (right). Labels in red indicate the sub-detector systems that were optimized, while the labels in blue are the sub-detector systems that were kept fixed due to geometrical constraint. The non-projective geometry (left) is a result of an optimization on the inner tracker layers (labeled in red) while keeping the support structure fixed, The angle made by the support structure to the IP is fixed at about 36.5°. The projective geometry (right) is the result of an ongoing project R&D to reduce the impact of readout and services on tracking resolution.

Soft and Hard Constraints, Overlaps & Other

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

sub-detector	constraint	description
EST/FST disks	$min\left\{\sum_i^{disks}\left\|\frac{R_{out}^i - R_{in}^i}{d} - \left\lfloor\frac{R_{out}^i - R_{in}^i}{d}\right\rfloor\right\|\right\}$	**soft constraint**: sum of residuals in sensor coverage for disks; sensor dimensions: d = 17.8 (30.0) mm
EST/FST disks	$z_{n+1} - z_n >= 10.0$ cm	**strong constraint**: minimum distance between 2 consecutive disks
sagitta layers	$min\left\{\left\|\frac{2\pi r_{sagitta}}{w} - \left\lfloor\frac{2\pi r_{sagitta}}{w}\right\rfloor\right\|\right\}$	**soft constraint**: residual in sensor coverage for every layer; sensor strip width: w = 17.8 mm
μRWELL	$r_{n+1} - r_n >= 5.0$ cm	**strong constraint**: minimum distance between μRwell barrel layers

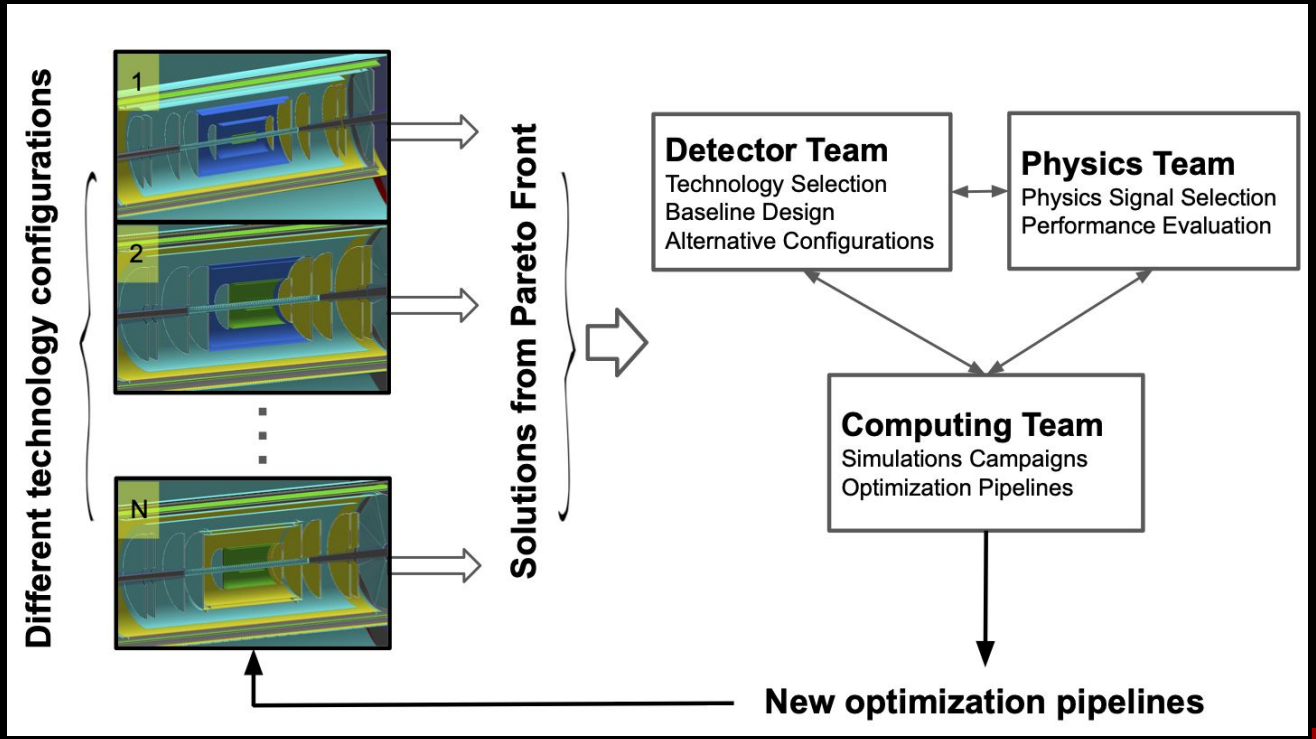


Integration during the EIC Detector Proposal

AI-"Optimization" does not necessarily mean "fine-tuning"

- We want to use these algorithms to: (1) steer the design and suggest parameters that a "manual"/brute-force optimization will likely miss to identify; (2) further optimize some particular detector technology (see d-RICH paper, e.g., optics properties)
- AI allows to capture hidden correlations among the design parameters.
- All "steps" (physics, detector) involved in the AI optimization, strong interplay between working groups

Light/smart optimization pipelines ran during the "explorative" phase of the detector proposal



Implementation

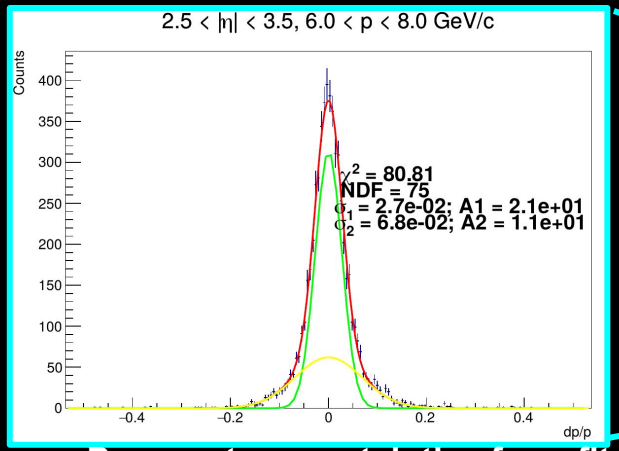
- **Objective functions** Average of Weighted Averages (n_obj ≥ 3)
  - **Momentum resolution dp/p**
  - **Theta resolution dθ/θ**
  - **Projected dθ/θ at PID location.**
  - **Kalman Filtering inefficiency** (improving the tracking reconstruction ability of the algorithm)

- **Validation** of the solutions
  - Validate by comparing optimal vs baseline dφ resolution, vertex resolution and reconstruction efficiency

Weighted sum with errors

Weighted sum with errors

Implementation



Propagate uncertainties from fits

$$\bar{x}_\eta = \frac{\sum_p x_p w_p}{\sum_p w_p}$$

Average objective in a η bin

Sum in bins of P
14 bins

$$\bar{x} = \frac{\sum_\eta^{N_\eta} \bar{x}_\eta}{N_\eta}$$

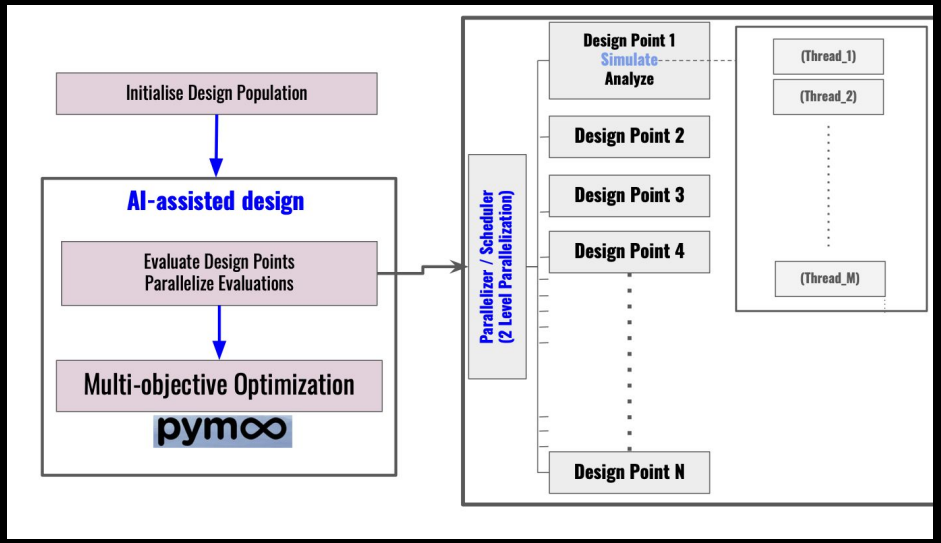
$$R(f) = \frac{1}{N_\eta} \sum_\eta \left(\frac{\sum_p w_{p,\eta} \cdot R(f)_{p,\eta}}{\sum_p w_{p,\eta}} \right)$$

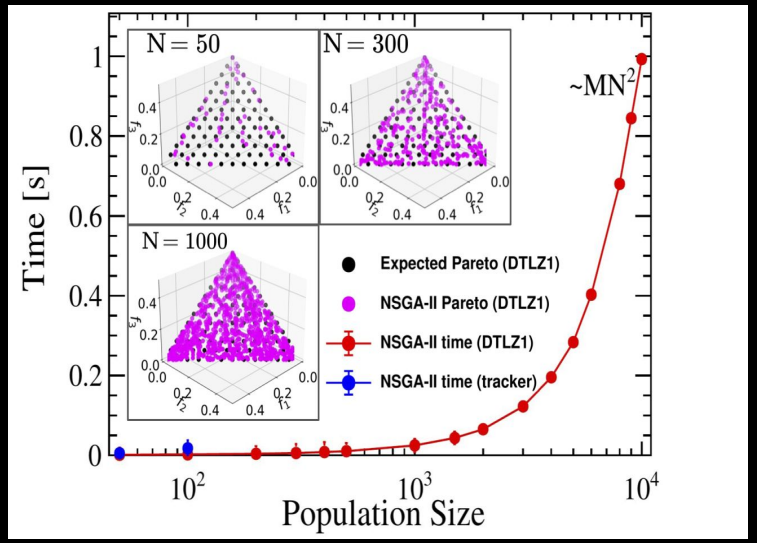
Weighted sum with errors

Weighted sum with errors

Computational Resources

time taken by GA + sorting

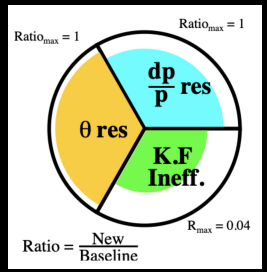




description	symbol	value
population size	N	100
# objectives	M	3
offspring	O	30
design size	D	11 (9)
# calls (tot. budget)	–	200
# cores	–	same as offspring
# charged π tracks	N_{trk}	120k
# bins in η	N_η	5
# bins in p	N_p	10

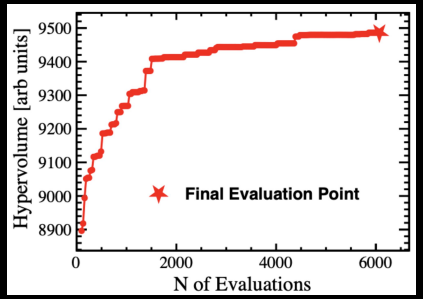
- Used a test problem DTLZ1
- Verified scaling following MN^2 and convergence to true front
- ~1s/call with 10^4 size!
- For 11 variables and 3 objectives needs ~ 10000 evaluations to converge

~10k CPUhours / pipeline

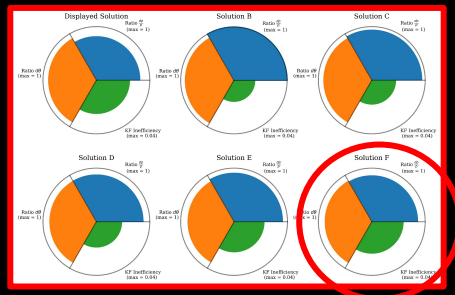


“Navigate” Pareto Front

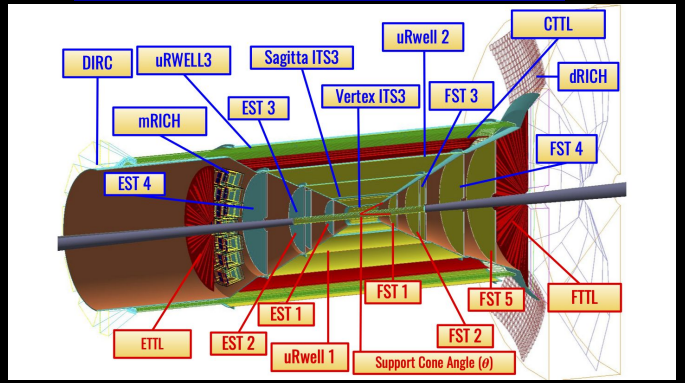
1 Can take a snapshot any time during evaluation



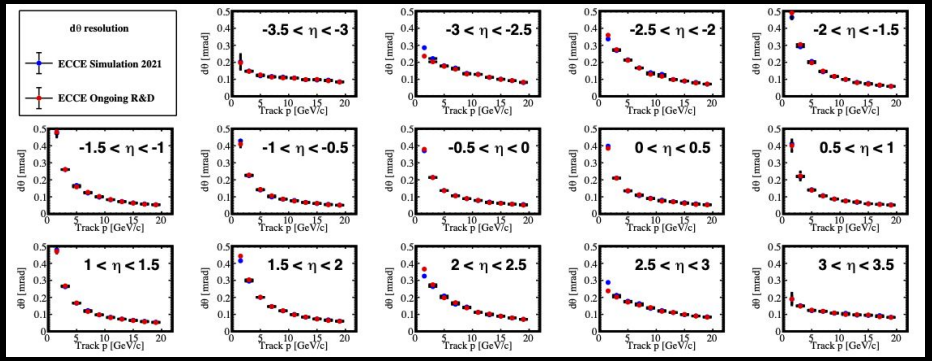
2 Updated Pareto Front at time t

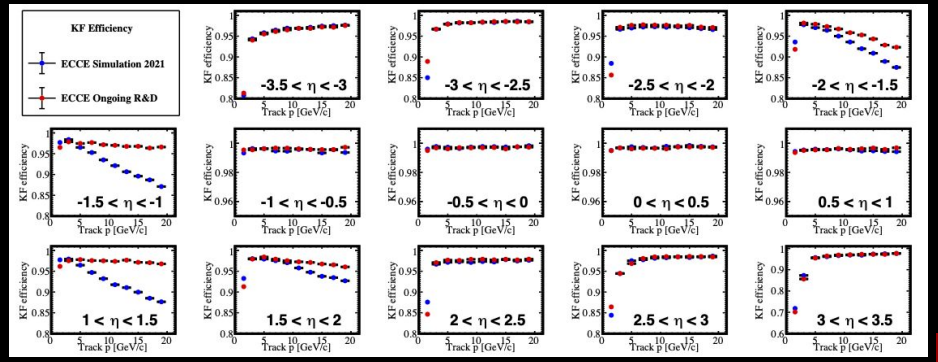


3 At each point in the Pareto front corresponds a design

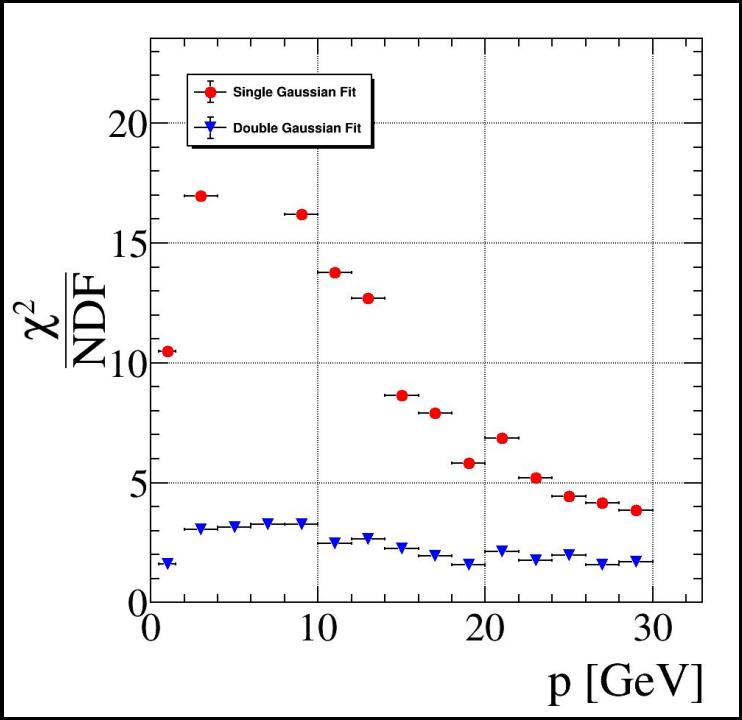


4 Analysis of Objectives (momentum resolution, angular resolution, KF efficiency)





Single VS Double Gaussian



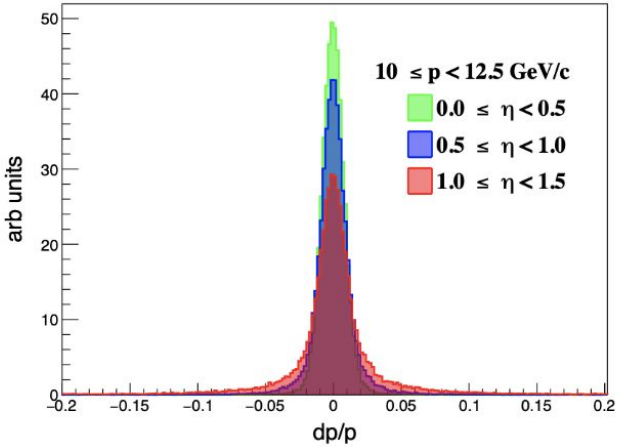
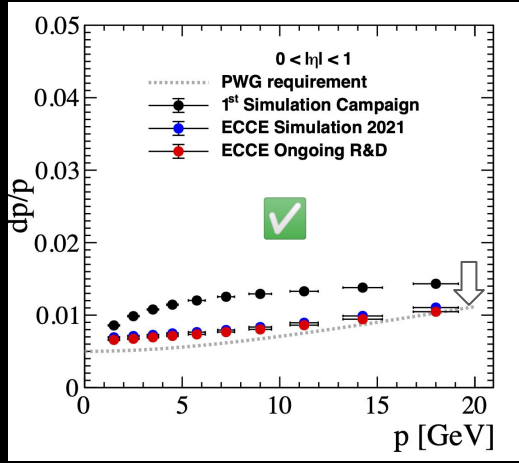
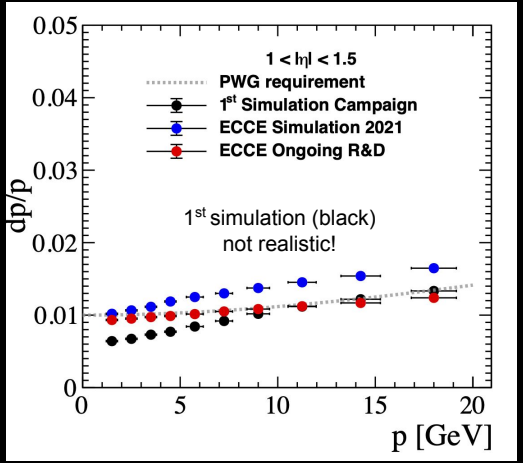


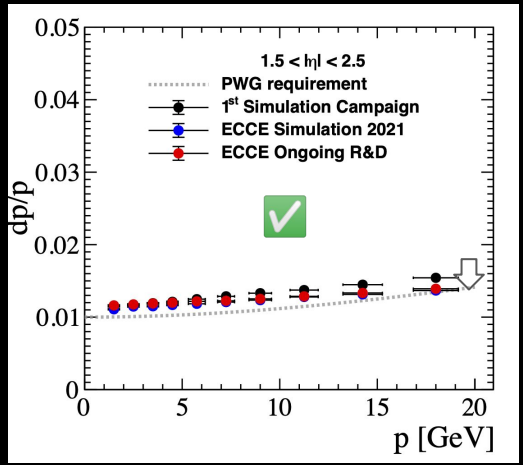
Figure 6: **Fit strategy:** a double-Gaussian fit function is utilized to extract the resolutions. Such a fit function provided good reduced χ^2 and more stable extractions compared to single-Gaussian fits. The resolution is obtained as an average of the two σ's weighted by the relative areas of the two Gaussians according to Eq. (3). The figure represents the results corresponding to a particular bin in η and p.

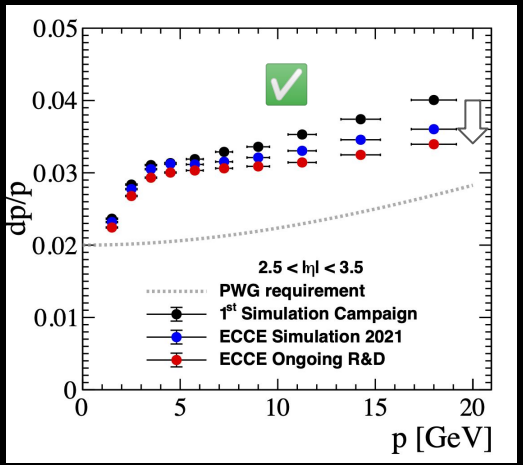
Evolution

- Black points represent the first simulation campaign, and a preliminary detector concept in phase-I optimization which did not have a developed support structure;
- Blue points represent the fully developed simulations for the final ECCE detector proposal concept; red points the ongoing R&D for the optimization of the support structure.
- Compared to black, there is an improvement in performance in all η bins with the exception of the transition region, an artifact that depends on the fact that black points do not include a realistic simulation of the material budget in the transition region!
- In the transition region, it can be also appreciated the improvement provided by the projective design



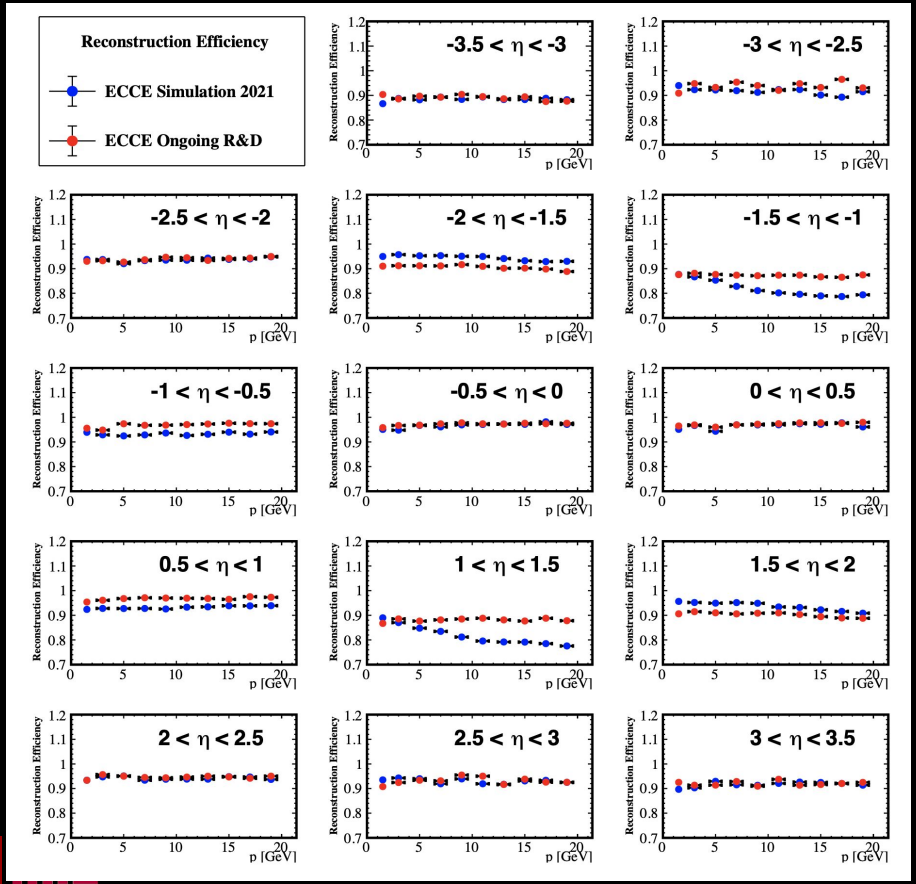






Validation

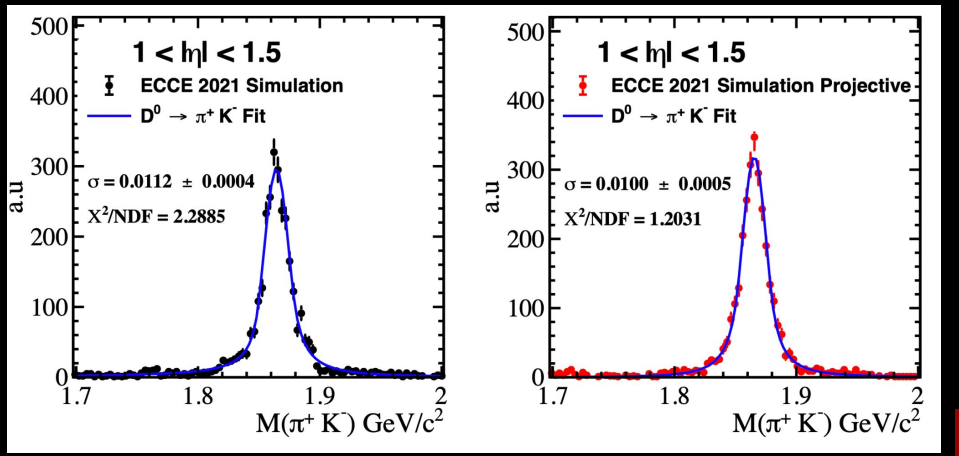
Reconstruction Efficiency



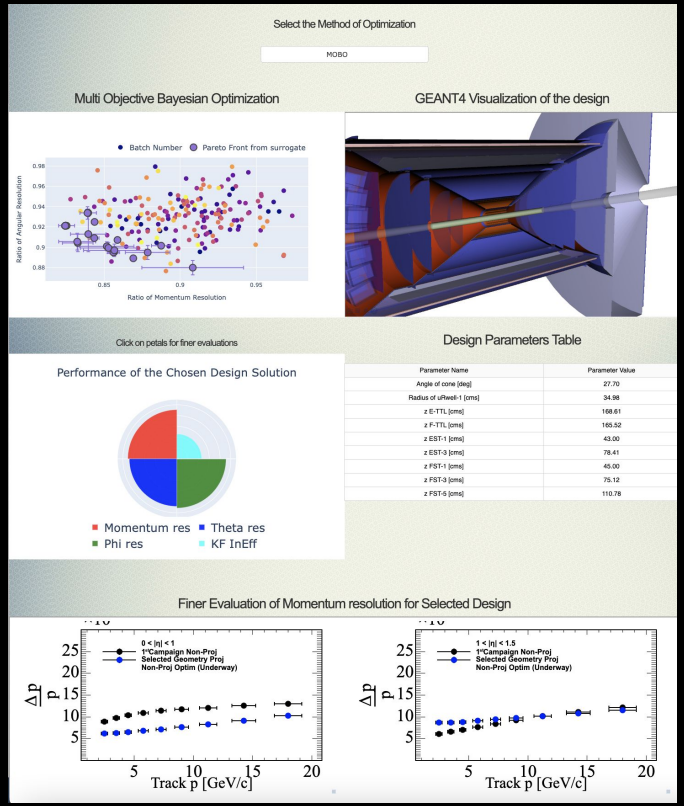
Performance evaluated after optimization process (both designs).

Notice red points are related to an ongoing project R&D with a projective support structure for the ECCE tracker.

D0 invariant mass from semi-inclusive deep inelastic scattering



Navigate Pareto front interactively



- Visualization of results from approximated Pareto front
- Exploration in a multiple objective space
- Facilitate study/comparison of tradeoff solutions
- Here MOBO is used using BoTorch/Ax (benefit from strong community support — Facebook)

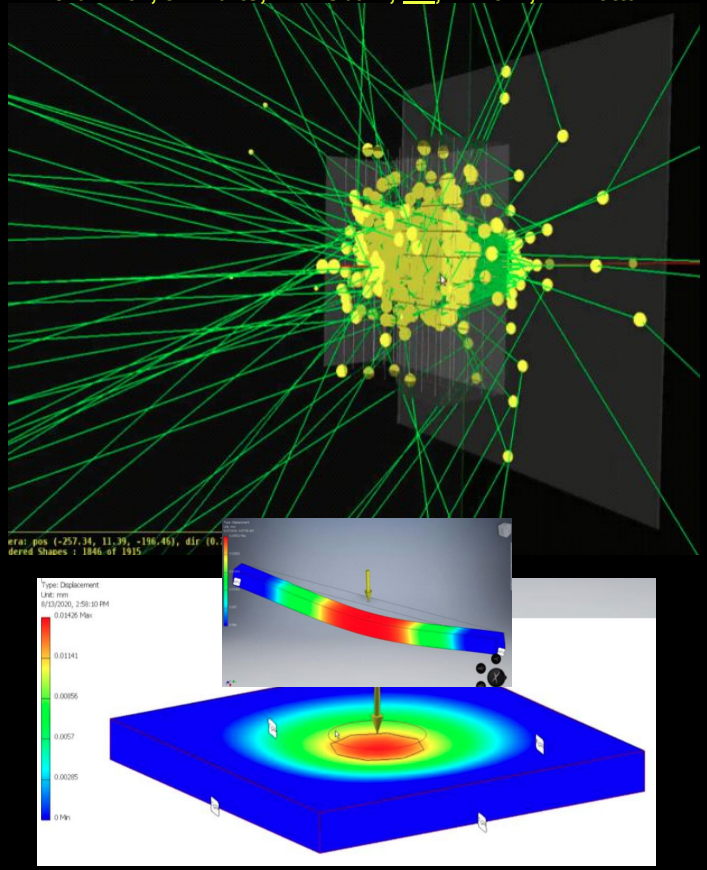
K. Suresh (U. of Regina) https://ai4eicdetopt.pythonanywhere.com

CF, Z. Papandreou, K. Suresh, *Designing EIC with the assistance of AI: strategies and perspectives* (in progress)

Other Applications: novel aerogel material

- 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 is introducing fibers into the aerogel that increase mechanical strength, but do not affect the optical properties.
- We are designing the aerogel+fibers optimizing mechanical stability and resolution.
- Paper in preparation.

V. Berdnikov, J. Crafts, E. Cisbani, CF, T. Horn, R. Trotta



Backup