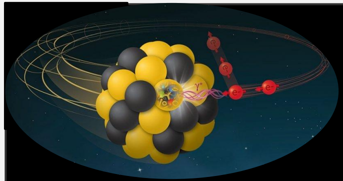


AI Assisted Detector Design for EIC

Karthik Suresh, for AID(2)E Collaboration
Department of Data Science
College of William and Mary

WILLIAM & MARY
CHARTERED 1693

AI4EIC

Outline

- Multi Objective Optimization
- Need for AI in detector design – The AID(2)E project
- Closure Test and Project workflows
- Selected works and future studies

Multi Objective Optimization

Design space spanned by ‘x’

$$\min / \max \;\; \mathbf{f_m}(\mathbf{x}), m = 1, \ldots, M$$

$$\text{s.t.} \;\; \mathbf{g_j}(\mathbf{x}) \leq 0, \; j = 1, \ldots, J$$

$$\mathbf{h_k}(\mathbf{x}) = 0, \; k = 1, \ldots, K$$

$$x_i^L \leq x_i \leq x_i^U, \; i = 1, \ldots, N$$

Guo, Kai, et al. *Materials Horizons* 8.4 (2021): 1153-1172.

Table 1 Popular ML methods in design of mechanical materials

ML method	Characteristics	Example applications in mechanical materials design
Linear regression; polynomial regression	Model the linear or polynomial relationship between input and output variables	Modulus[112] or strength[123] prediction
Support vector machine; SVR	Separate high-dimensional data space with one or a set of hyperplanes	Strength[123] or hardness[125] prediction; structural topology optimization[159]
Random forest	Construct multiple decision trees for classification or prediction	Modulus[112] or toughness[130] prediction
Feedforward neural network (FFNN); MLP	Connect nodes (neurons) with information flowing in one direction	Prediction of modulus,[97,112] strength,[93] toughness[130] or hardness;[97] prediction of hyperelastic or plastic behaviors;[143,145] identification of collision load conditions;[147] design of spinodoid metamaterials[163]
CNNs	Capture features at different hierarchical levels by calculating convolutions; operate on pixel-based or voxel-based data	Prediction of strain fields[104,105] or elastic properties[102,103] of high-contrast composites, modulus of unidirectional composites,[136] stress fields in cantilevered structures,[137] or yield strength of additive-manufactured metals;[121] prediction of fatigue crack propagation in polycrystalline alloys;[140] prediction of crystal plasticity;[120] design of tessellate composites;[107–109] design of stretchable graphene kirigami;[155] structural topology optimization[156–158]
Recurrent neural network (RNN); LSTM; GRU	Connect nodes (neurons) forming a directed graph with history information stored in hidden states; operate on sequential data	Prediction of fracture patterns in crystalline solids;[114] prediction of plastic behaviors in heterogeneous materials;[142,144] multi-scale modeling of porous media[173]
Generative adversarial networks (GANs)	Train two opponent neural networks to generate and discriminate separately until the two networks reach equilibrium; generate new data according to the distribution of training set	Prediction of modulus distribution by solving inverse elasticity problems;[138] prediction of strain or stress fields in composites;[139] composite design;[164] structural topology optimization;[165–167] architected materials design[115]
Gaussian process regression (GPR); Bayesian learning	Treat parameters as random variables and calculate the probability distribution of these variables; quantify the uncertainty of model predictions	Modulus[122] or strength[123,124] prediction; design of supercompressible and recoverable metamaterials[110]
Active learning	Interacts with a user on the fly for labeling new data; augment training data with post-hoc experiments or simulations	Strength prediction[124]
Genetic or evolutionary algorithms	Mimic evolutionary rules for optimizing objective function	Hardness prediction;[126] designs of active materials;[160,161] design of modular metamaterials[162]
Reinforcement learning	Maximize cumulative awards with agents reacting to the environments.	Deriving microstructure-based traction-[illegible]on laws[174]
Graph neural networks (GNNs)	Operate on non-Euclidean data structures; applicable tasks include link prediction, node classification and graph classification	Hardness prediction;[127] architected materials design[168]

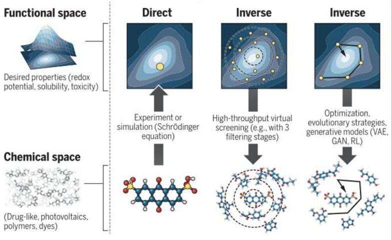


Fig. 2. Schematic of the different approaches toward molecular design. Inverse design starts from desired properties and ends in chemical space, unlike the direct approach that leads from chemical space to the properties.

B. Sanchez-Lengeling, A. Aspuru-Guzik. *Science* 361.6400 (2018): 360-365.

Multiobjective genetic algorithm approach to optimize beam matching and beam transport in high-intensity hadron linacs

M. Yarmohammadi Satri,[1,2,*] A. M. Lombardi,[2] and F. Zimmermann[2]

[1]*School of Particles and Accelerators, Institute for Research in Fundamental Sciences (IPM), P.O. Box 19395-5531, Tehran, Iran*

[2]*CERN, 1211 Geneva 23, Switzerland*

BoTorch
Bayesian Optimization in PyTorch

PyGMO

nextorch
NEXTorch
latest

Ax

Multi-objective Optimization in Python
pymoo

GPflow

jMetalPy

Multi Objective Optimization

Design space spanned by ‘x’

BoTorch
Bayesian Optimization in PyTorch

PyGMO

nextorch
NEXTorch
latest

Ax

Multi-objective Optimization in Python
pymoo

GPflow

jMetalPy

$$\min / \max \ \mathbf{f_m}(\mathbf{x}), m = 1, \ldots, M$$ Objectives to optimize

$$\text{s.t. } \ \mathbf{g_j}(\mathbf{x}) \leq 0,, j = 1, \ldots, J$$

$$\mathbf{h_k}(\mathbf{x}) = 0,, k = 1, \ldots, K$$

$$x_i^L \leq x_i \leq x_i^U,, i = 1, \ldots, N$$

Multi Objective Optimization

BoTorch
Bayesian Optimization in PyTorch

PyGMO

nextorch
NEXTorch
latest

Ax

Multi-objective Optimization in Python
pymoo

GPflow

jMetalPy

Design space spanned by ‘x’

$$\min / \max \ \mathbf{f_m}(\mathbf{x}), m = 1, \ldots, M$$

$$\text{s.t.} \ \mathbf{g_j}(\mathbf{x}) \leq 0,, j = 1, \ldots, J$$

$$\mathbf{h_k}(\mathbf{x}) = 0,, k = 1, \ldots, K$$

Constraints

$$x_i^L \leq x_i \leq x_i^U,, i = 1, \ldots, N$$

Multi Objective Optimization

Design space spanned by ‘x’

$$\min / \max \ \mathbf{f_m}(\mathbf{x}), m = 1, \ldots, M$$

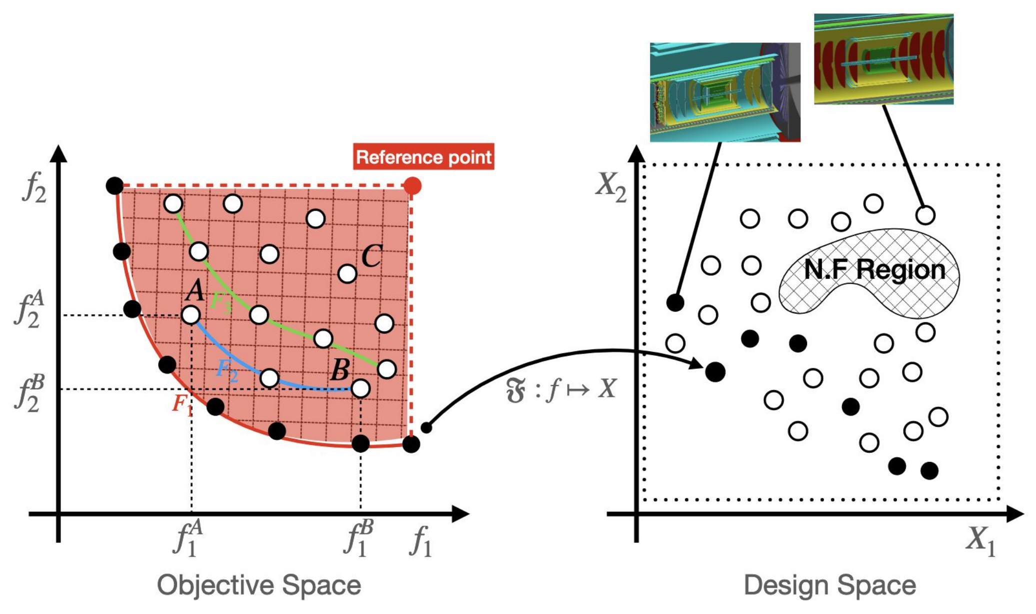
$$\text{s.t.} \ \mathbf{g_j}(\mathbf{x}) \leq 0,, j = 1, \ldots, J$$

$$\mathbf{h_k}(\mathbf{x}) = 0,, k = 1, \ldots, K$$

$$\boxed{x_i^L \leq x_i \leq x_i^U,, i = 1, \ldots, N \quad \text{Bounded Design Space}}$$

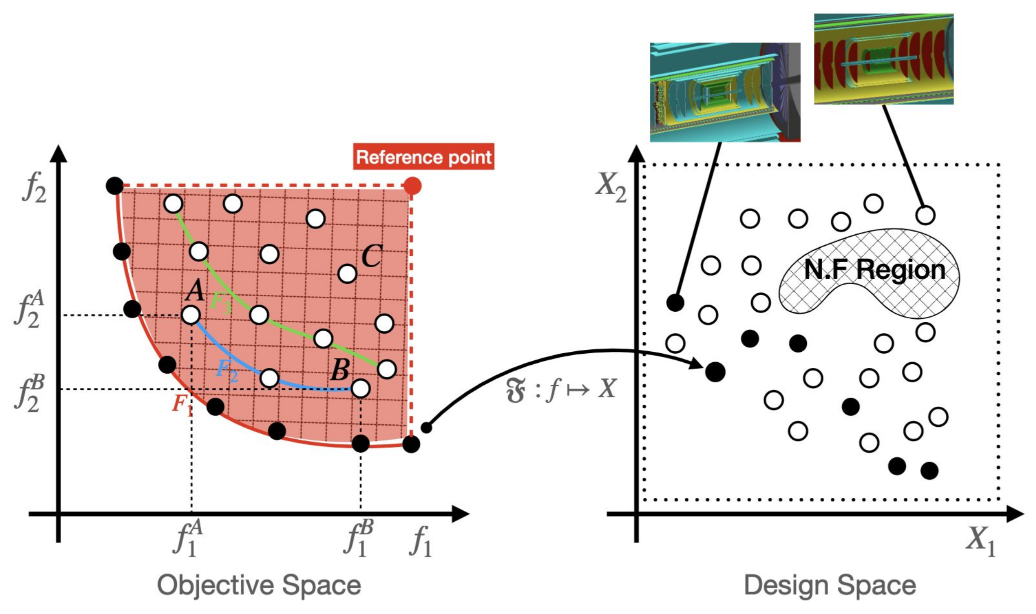
Multi Objective Optimization : Visual Intro

- Multiple "objectives"
  - Momentum resolution
  - θ resolution
  - KF efficiency
  - projected θ resolution @ PID
- Goal : "Optimize" these Objectives
- Map: "Design" space – "Objective" Space
- Non-Feasible region to be avoided



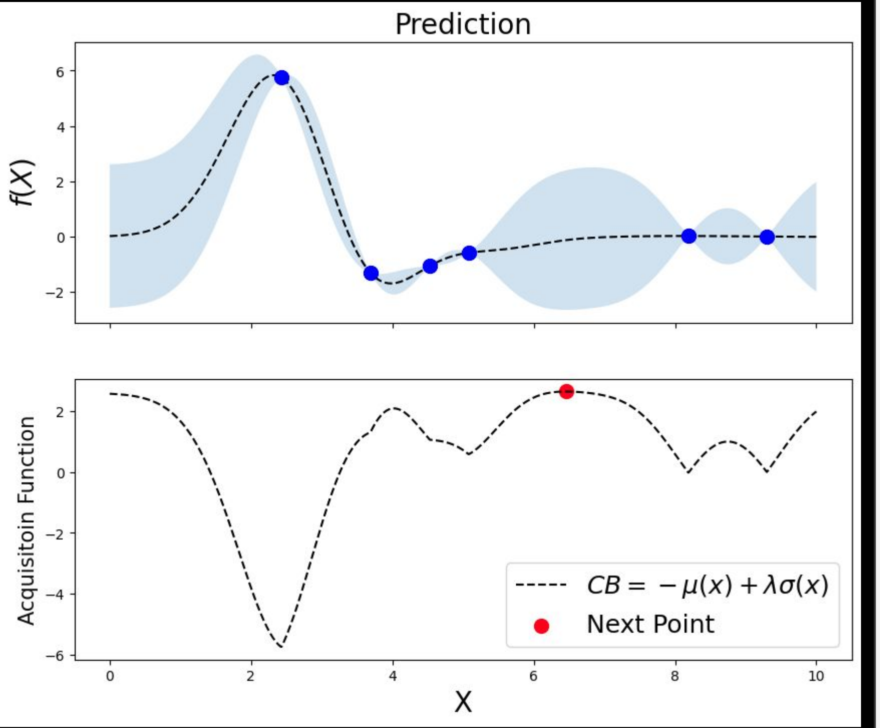
Multi Objective Optimization : Visual Intro

- What is “Optimal”?
  - Non-dominated (Pareto) Solutions
- How to rank solutions?
  - “Fronts” of solutions
- Methods of MOO
  - Evolutionary
  - Bayesian
  - Preferential Learning, etc.



Multi Objective Optimization through surrogate modelling

- Surrogate Model – A model that will be able to successfully approximate the true function.
- Acquisition Model – A quick evaluator to choose the next point to be computed
  - Based on Exploration and Exploitation in the search space.
  - Critically important, since, this is key in convergence.

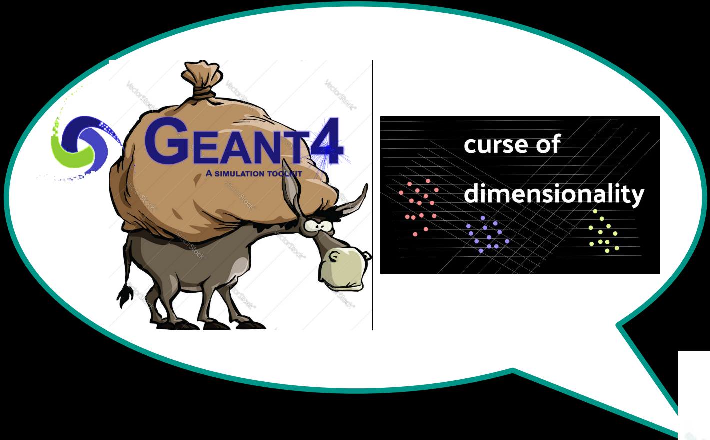


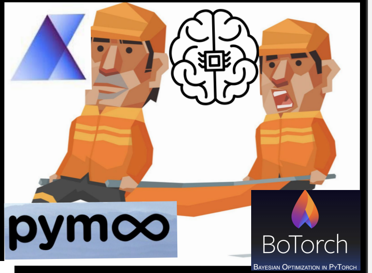
Large Scale Experiments : An Ideal MOO problem

GEANT4 – computation intensive.

Curse of dimensionality due to multiple Objectives and multidimensional design space

Each Design point requires multiple physics studies and hence increased computational needs





Estimated simulation requirements based on observed performance in 2021. https://arxiv.org/pdf/2205.08607.pdf

Year	Number of Events [$\times 10^6$]	Storage [TB]	CPU-core hours [Mcore-hrs]
2022	200	50	45
2023 - 2024	100	25	22.5
2025 - 2028	50	12.5	11
2029 - 2030	500	125	110
Total	1600	400	354

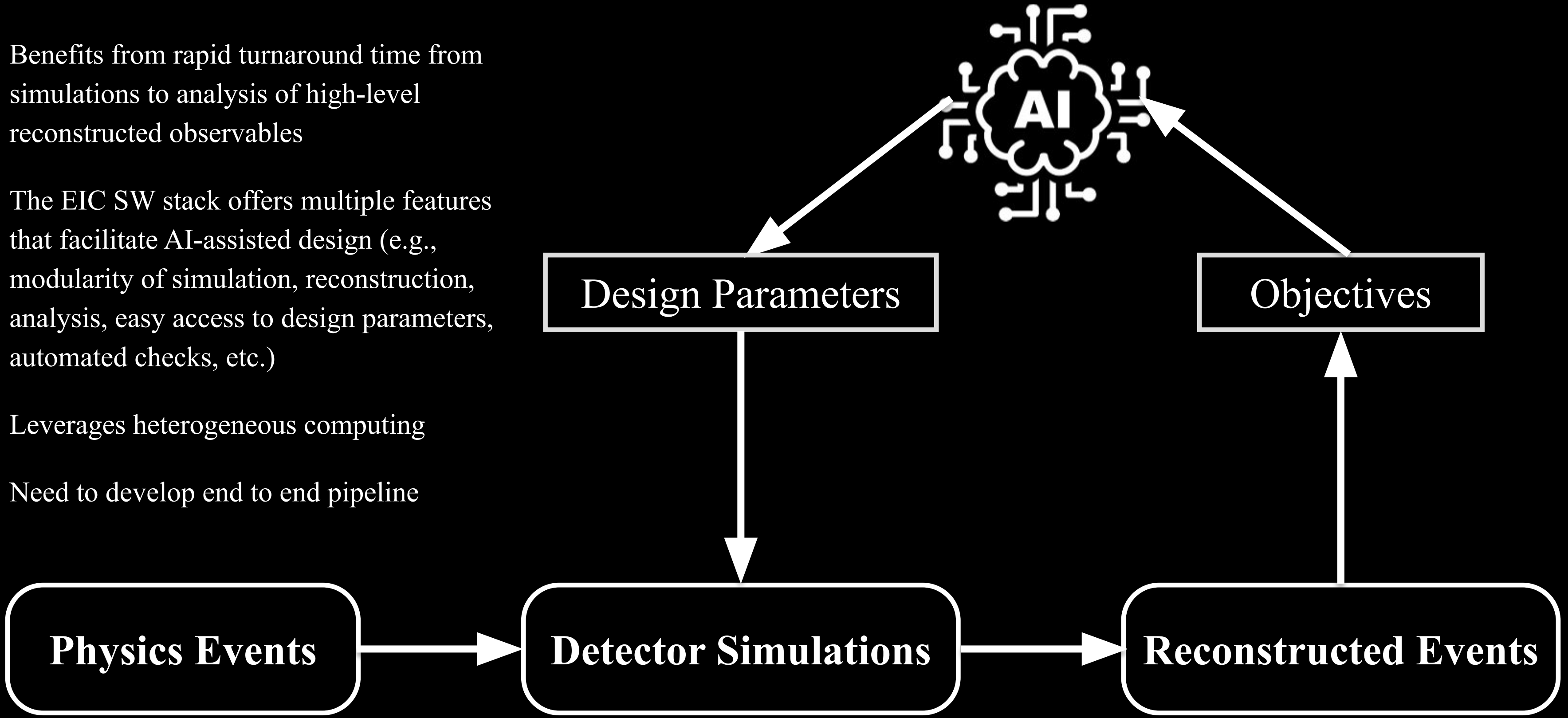
Workflow for AI Assisted detector design

Benefits from rapid turnaround time from simulations to analysis of high-level reconstructed observables

The EIC SW stack offers multiple features that facilitate AI-assisted design (e.g., modularity of simulation, reconstruction, analysis, easy access to design parameters, automated checks, etc.)

Leverages heterogeneous computing

Need to develop end to end pipeline



Desired kinematic range

The AID(2)E Project

[2405.16279] AI-Assisted Detector Design for the EIC (AID(2)E) (2024)

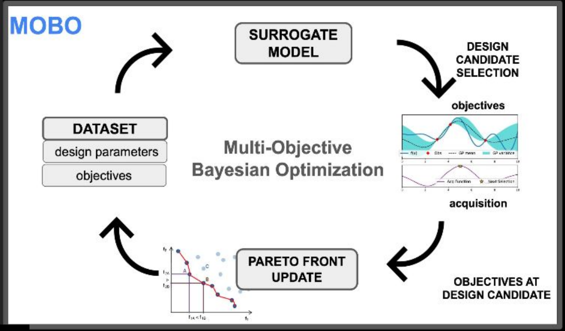
AID(2)E: AI-Assisted Detector Design at EIC

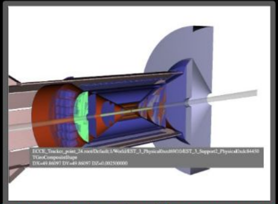
DE-FOA-0002785

BNL, T. Wenaus
CUA, T. Horn
Duke, A. Vossen
JLab, M. Diefenthaler
W&M, CF

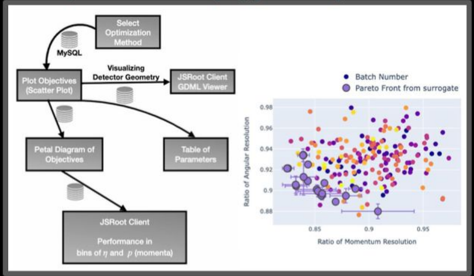
Brookhaven National Laboratory · The Catholic University of America · Duke University · Jefferson Lab · William & Mary (Chartered 1693)

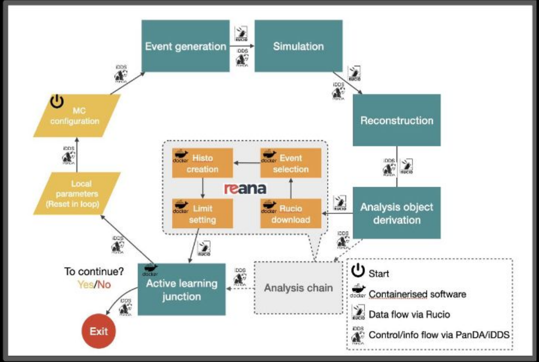
The AID(2)E Project





https://ai4eicdetopt.pythonanywhere.com/





(i) Will contribute to advance the boundaries of MOBO complexity to accommodate a large number of objectives and will explore usage of physics-inspired approaches

(ii) Development of suite of data science tools for interactive navigation of Pareto front (multi-dim design with multiple objectives)

(iii) Will leverage cutting-edge workload management systems capable of operating at massive data and handle complex workflows

Examining solutions on the Pareto front of ePIC at different values of the budget can have great cost benefits

A fractional improvement in the objectives translates to a more efficient use of beam time which will make up a majority of the cost of the EIC over its lifetime

Project Workflow

AID(2)E Wrapper

AID(2)E Pipeline

Thrusts of development

Distributed Computing

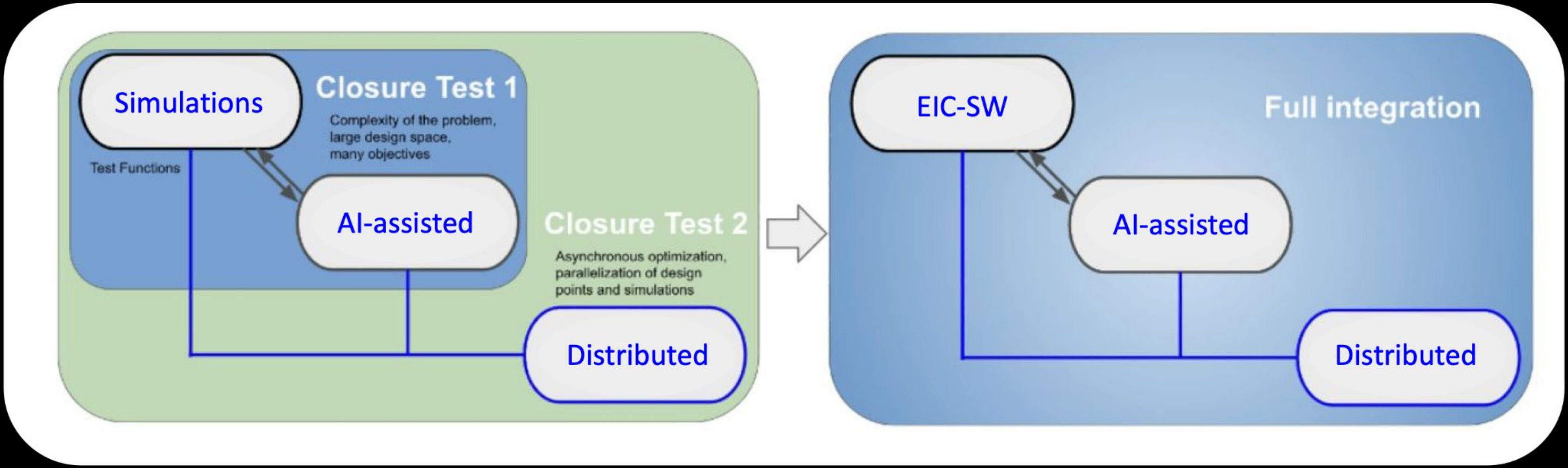
Optimizer
eg. MOBO Algorithm

- Computations during an iteration
- Total number of iterations to converge

ePIC Software
Heavy simulations

- Detector Simulations
  EIC-SIM-RECON
- EIC Analysis
  Physics/Detector response

Project workflow



Closure Test 1 – Stress testing SoTA MOBO

Gaussian Process
$O(n^3)$

- The PDF prior distribution, that describes the Design space to objective. This is the surrogate model.
- SAAS[1] priors have been proven to be successful upto 388 design dimensions.
- Assumes several design variables has increased importance compared to others
- Computational expensive as iteration increases
- Benefit from GPU hardware acceleration

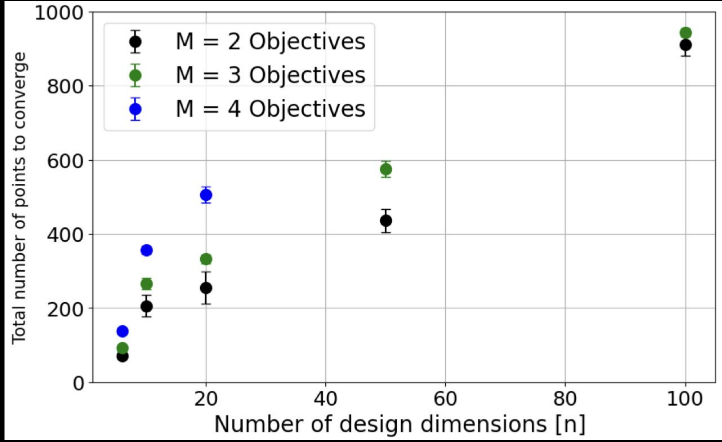
Bayesian Sampling from posteriors
NUTS – $O(Md^{5/4})$[NUTS]

- Sample L points from the posterior distribution.
- HMC is a popular algorithm
- Mainly depends on the Number of objectives and design space dimensions
- Has minimal dependence on iteration.
- GPU acceleration through JAX backend.

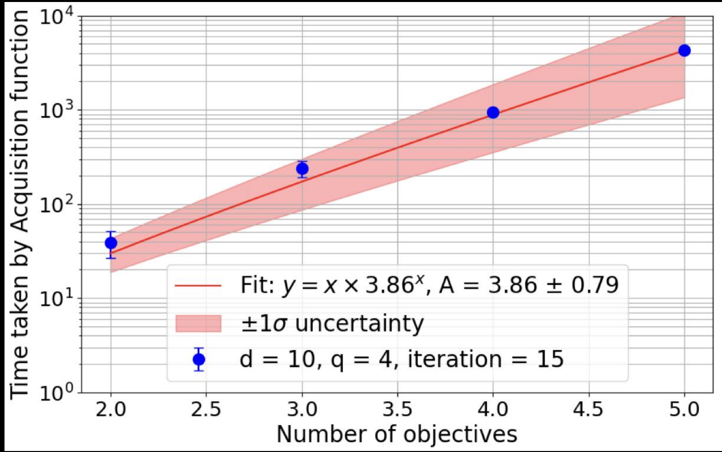
Acquisition function
qNEHVI – $O(M(n + i)^M)$[2]

- A "cheaper" function to evaluate as a proxy for the black box function
- Identifies points of maximum improvements hence, the name
- Scales nonlinearly with iteration, total points explored, design space and objective space.
- Partially benefitted by GPU acceleration.

Closure Test 1 – Stress testing MOBO



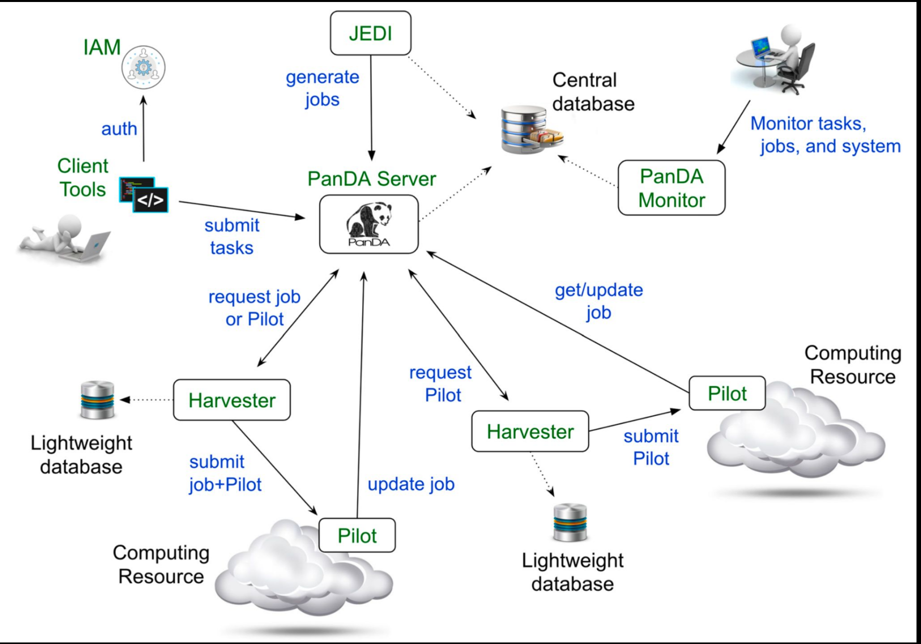
- Stress test the SoTA algorithm used for optimization
- MOBO stress-testing for problems with increasing complexity (design and objectives) and known Pareto



arXiv:2405.16279

Closure Test 2: PanDA/iDDS integration

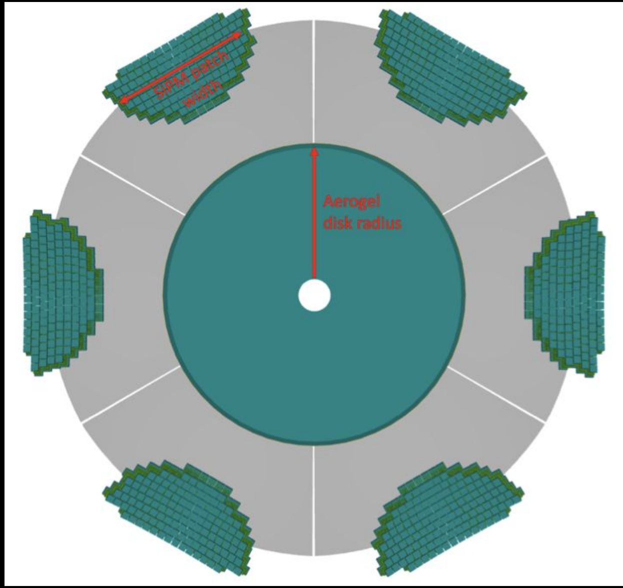
- Stress test scalability across distributed resources
- Integrate PanDA/iDDS AI/ML service to support MOBO workflow for design optimization



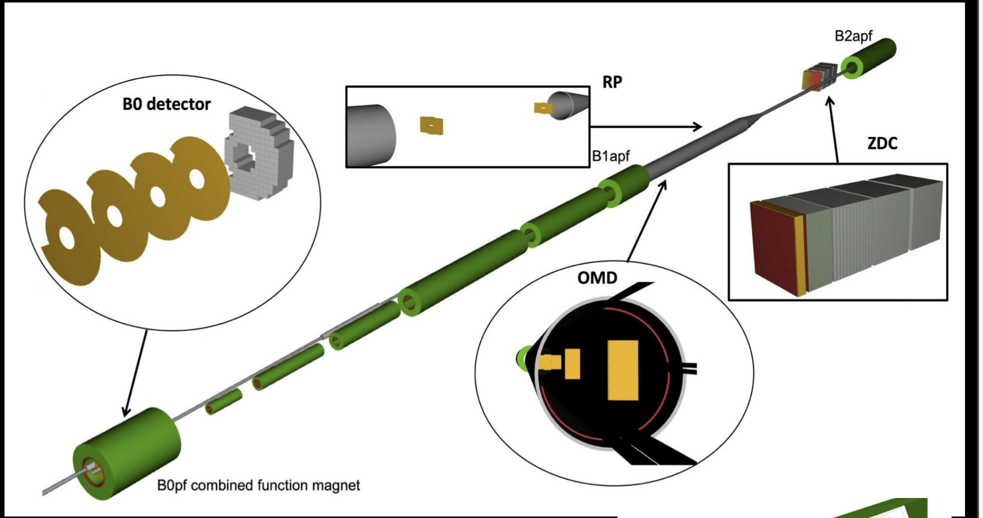
[PanDA: Production and Distributed Analysis System. *Comput Softw Big Sci* **8**, 4 (2024)]

Current Detector Subsystems for optimization in ePIC

d-RICH detector at EIC

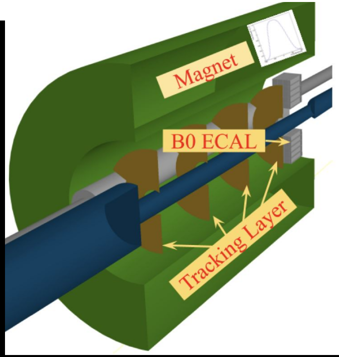


Design params: Mirror, sensor placement, gas, mirror material
Objectives: PID performance in bins of momentum, cost



Far Forward – B0 System

Design params: z positions of disks
Objectives: Momentum resolution, Acceptance



Summary and Conclusion

- Coupling the MOBO to EIC is done. Closure test 1 nearly done.
- Working on code base for a common framework for distributed optimization using PanDA and SLURM.
- EIC can be the first large-scale experiment to be realized with assistance of AI
- Ultimately, we can realize a framework that can optimize holistically a large-scale detector, and that is scalable and distributed. The Detector-2 at EIC an ideal candidate
- Exploring solutions on EIC detector Pareto front across budget values yields significant cost advantages during construction phases.
- Efficient objectives = cost-effective EIC beam time.
- This framework inherently offers broader impacts, can be adapted in various experiments and suitable for compute-intensive applications that necessitate MOO (e.g., calibrations, alignments, etc)

Backups

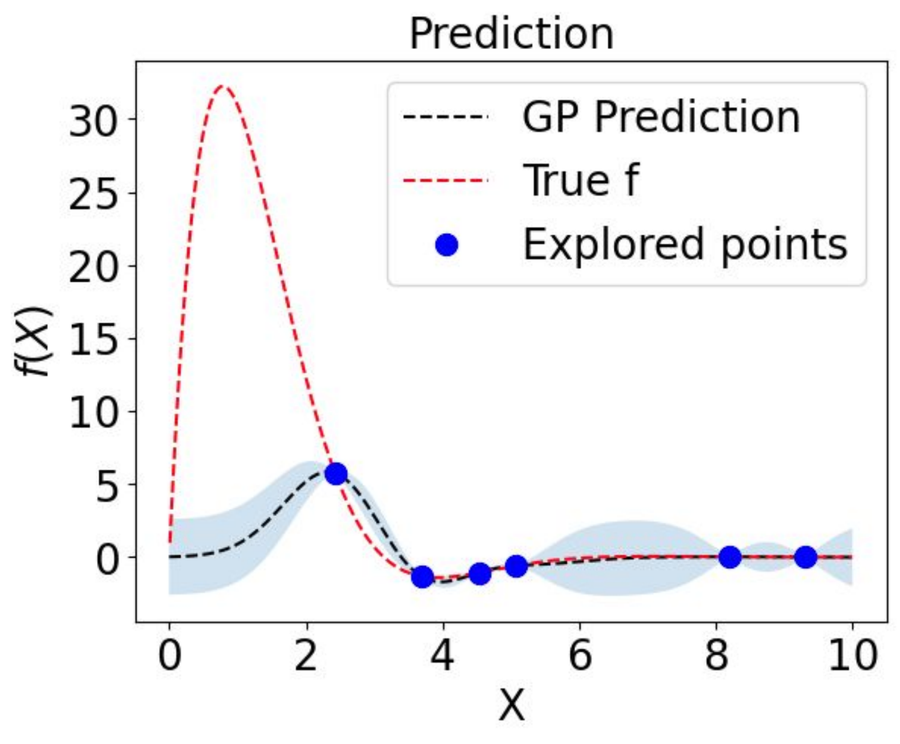
GP as a Surrogate Model

Optimization problem:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_j(x) = 0, \quad j = 1, \ldots, p \end{aligned}$$

Question: What would be the next point to explore from this?

Choose a region?

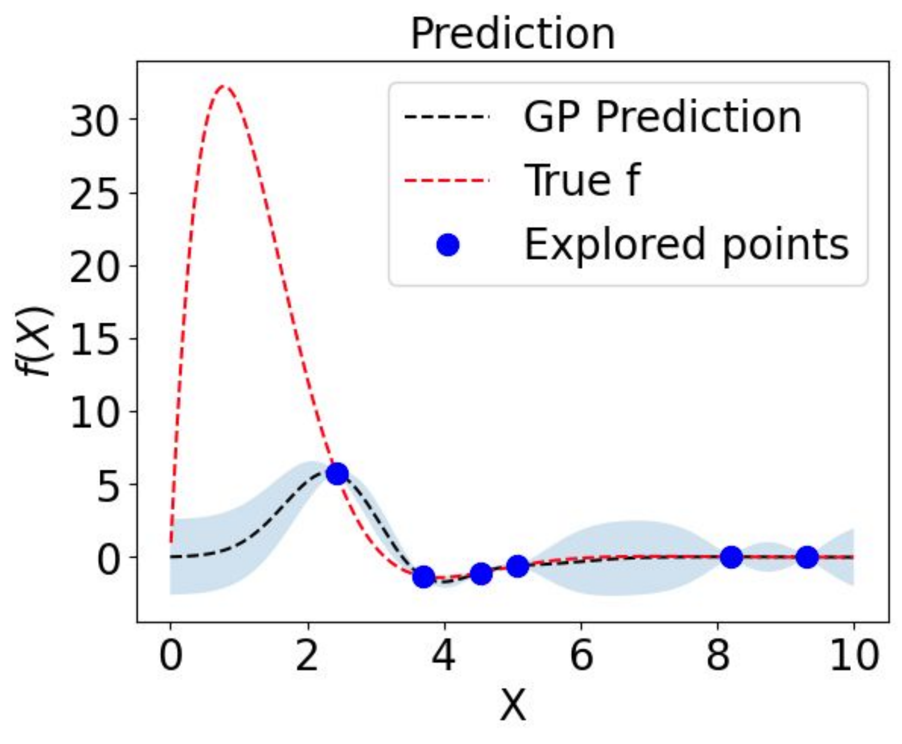


In practice we do not know the True f.

GP as a Surrogate Model

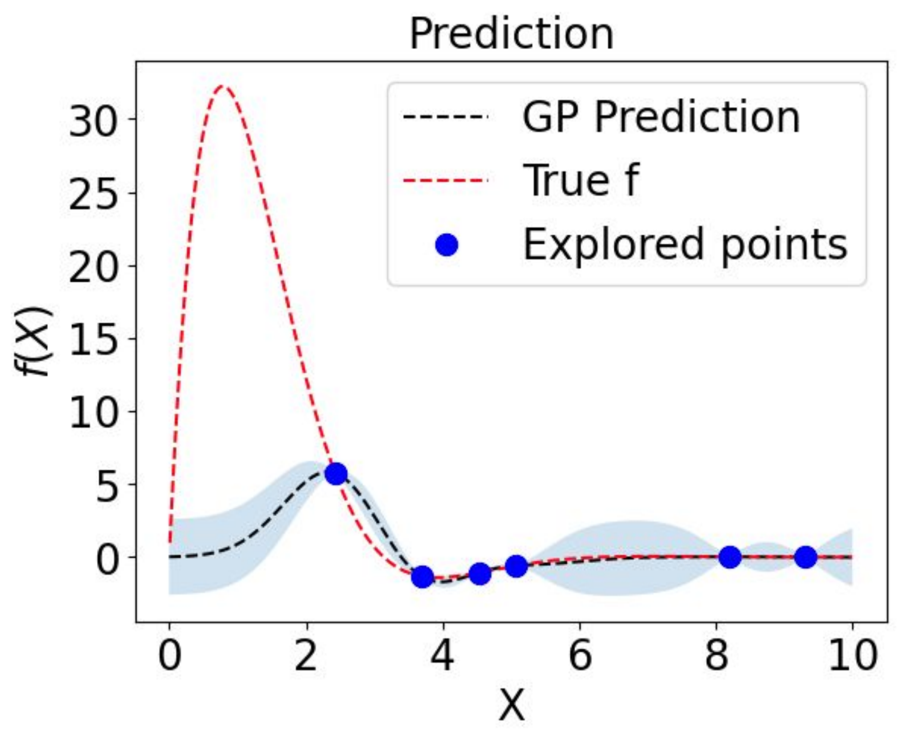
$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \le 0, \quad i = 1, \dots, m \\ & h_j(x) = 0, \quad j = 1, \dots, p \end{aligned}$$

The task: To minimize. So should we even care on regions which are not minimum?



The Acquisition function

- Define a function that scans through the search space for values of `f(x)` using the built GP.
- Much faster than evaluations.
- Carefully choose the next point to evaluate*.
- Model inaccurate in region out of interest



Widely used Acquisition functions

- Confidence Bound
- Probability of Improvement
- Expected Improvement

*Since evaluations are supposed to be very costly

Confidence Bound

Hyper parameter

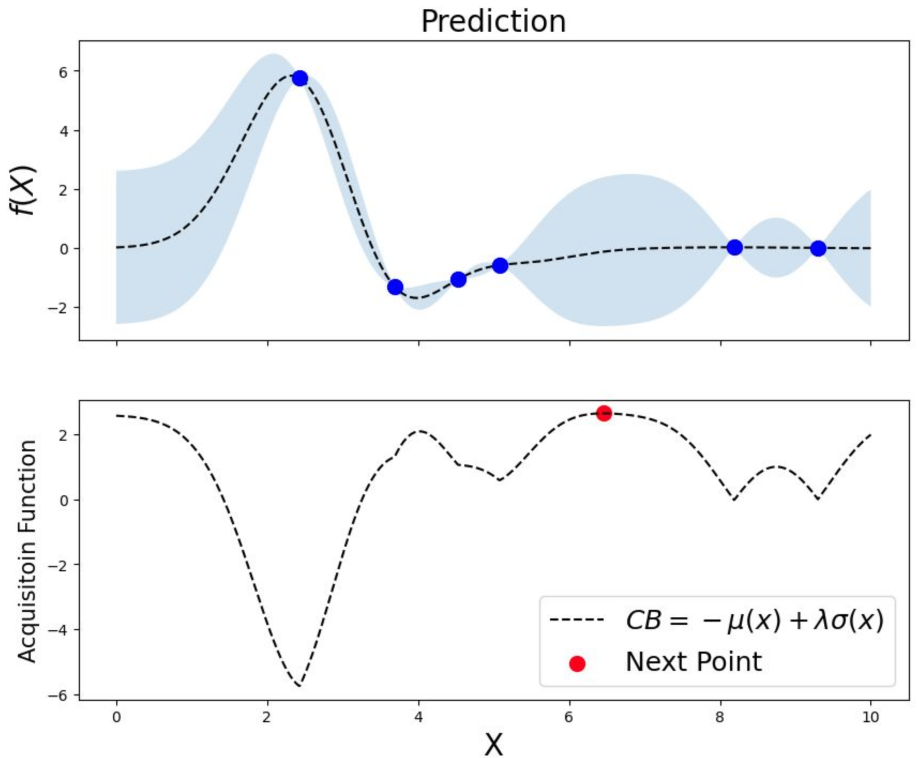
$$\text{LCB} = -\mu(x) + \lambda\sigma(x)$$

Exploitation

Exploration

Can now control where the search will happen in subsequent iterations.

Usually, bias is towards the mean. Since it is optimization



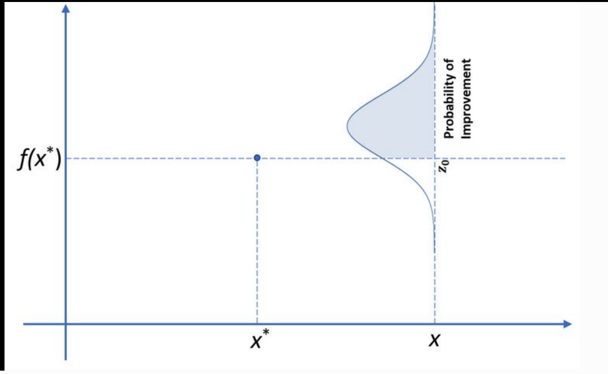

```
def UCB_Acq(input_range, gp_model, Lambda = 1.0):
    xs = np.linspace(start = input_range[0], stop = input_range[-1], num = 1000)
    y, y_std = gp_model.predict(xs.reshape(-1, 1), return_std = True)
    UCB = -y + Lambda*y_std
    return UCB
```

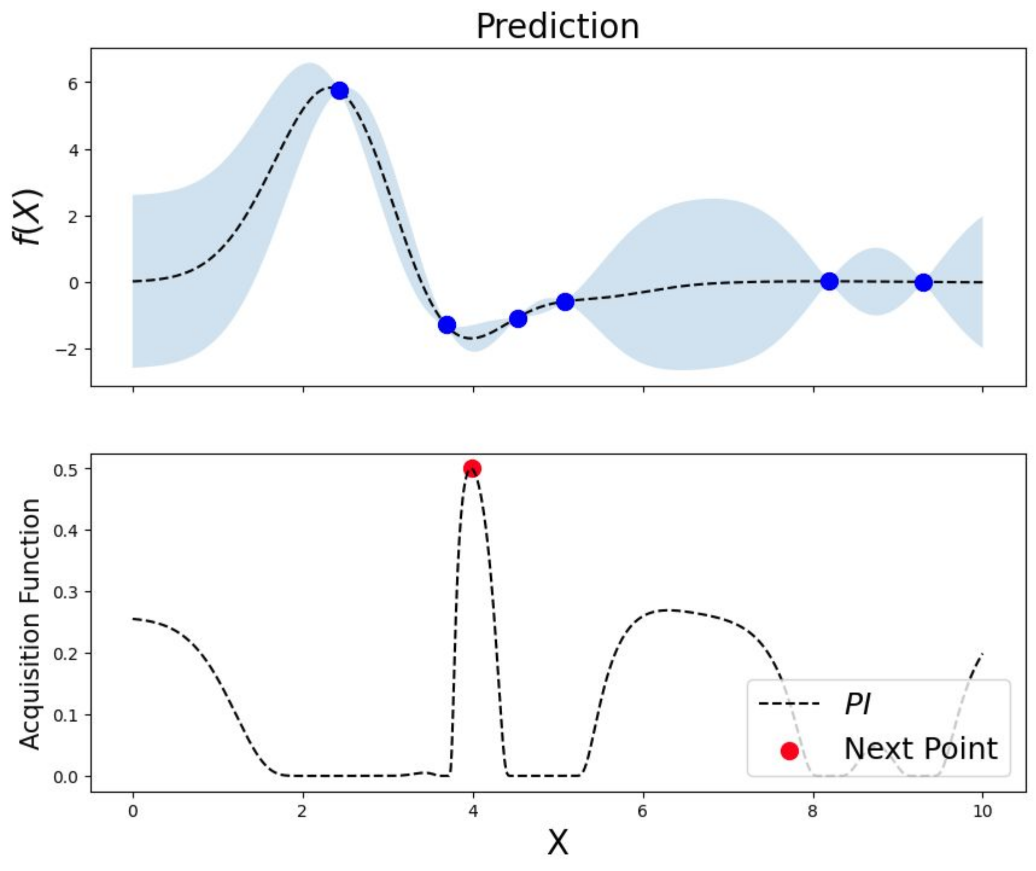
Probability of Improvement

$$\text{PI} = \text{CDF}\left(\frac{min(f(x)) - f(x*)}{\sigma(x) + \epsilon}\right)$$

Choose the point, that has the maximum probability of improvement.

Note: NO consideration of actual value.



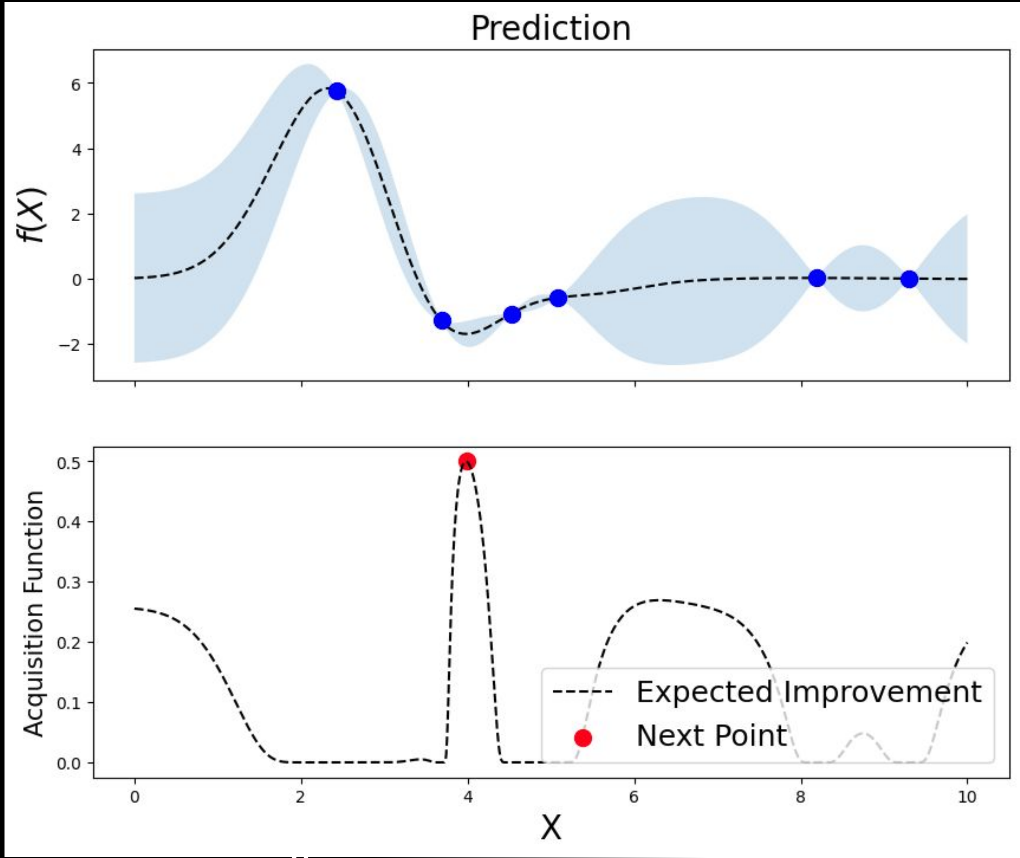


Expected Improvement

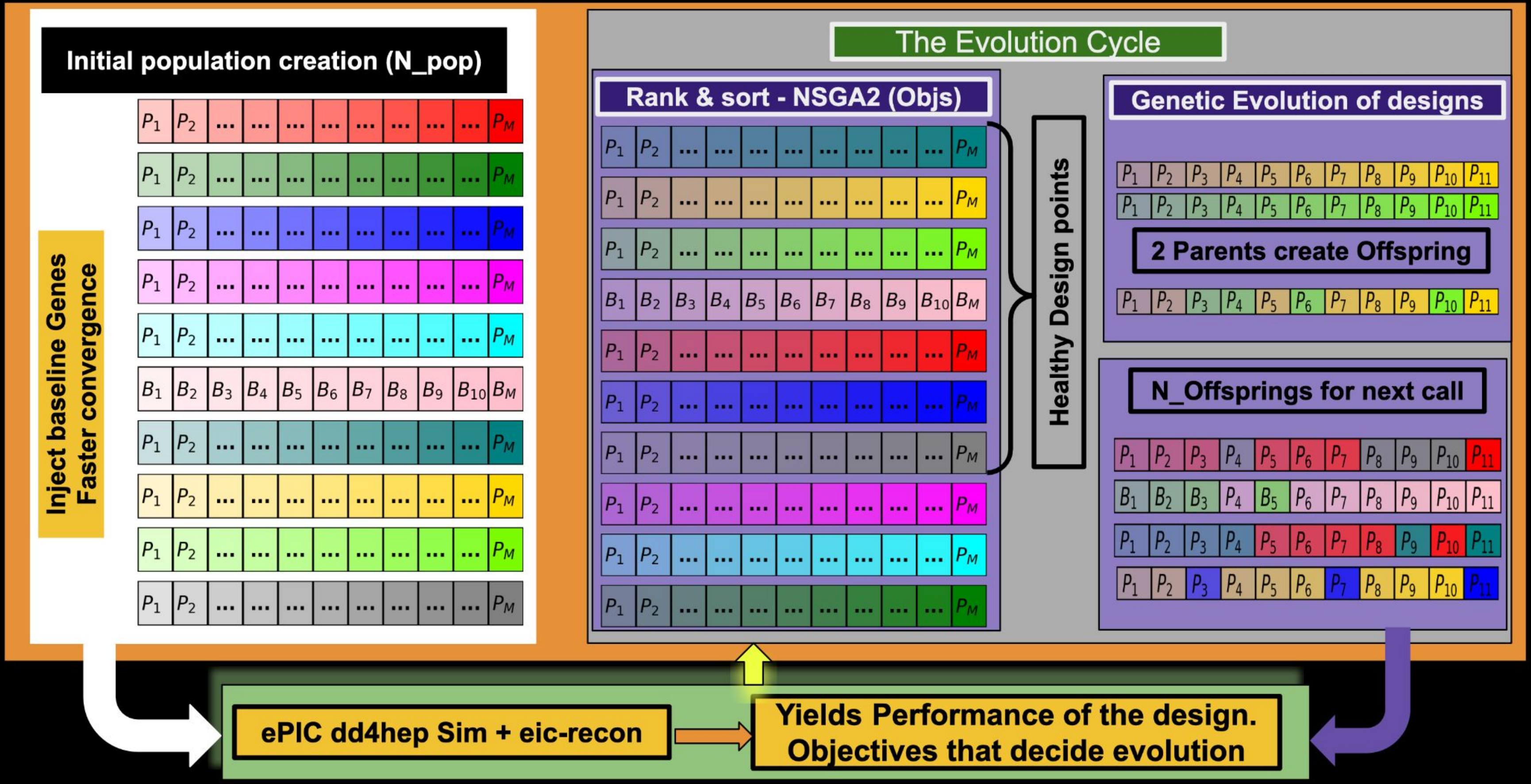
$$\text{EI} = (min(f(x) - f(x*)))\text{CDF}(\frac{min(f(x))-f(x*)}{\sigma(x)+\epsilon}) + \sigma(x)\text{PDF}(\frac{min(f(x))-f(x*)}{\sigma(x)+\epsilon})$$

Considers the Magnitude of improvement along with its probability[1]

- Now, With the suggested point, Start running iterations. Choose the first q points suggested by the Acquisition function.
- Run for N iterations
- Implement early stopping criterion if necessary



The Summary of MOGA Pipeline



Multi Objective Evolutionary Algorithms

- Inspired by Biological Systems.
- Semi heuristic in nature.
- Quite successful in solving MOO problems.
- Embedding constraints relatively easier

Swarm Algorithms
- Ant Colony optimization
- Bees algorithm
- Particle swarm optimization
- Cuckoo search

Genetic Algorithms
- Default Genetic Algorithm
- NSGA
- NSGA-II
- U-NSGA-III

Differential Evolution

Cellular Automata

PyGMO

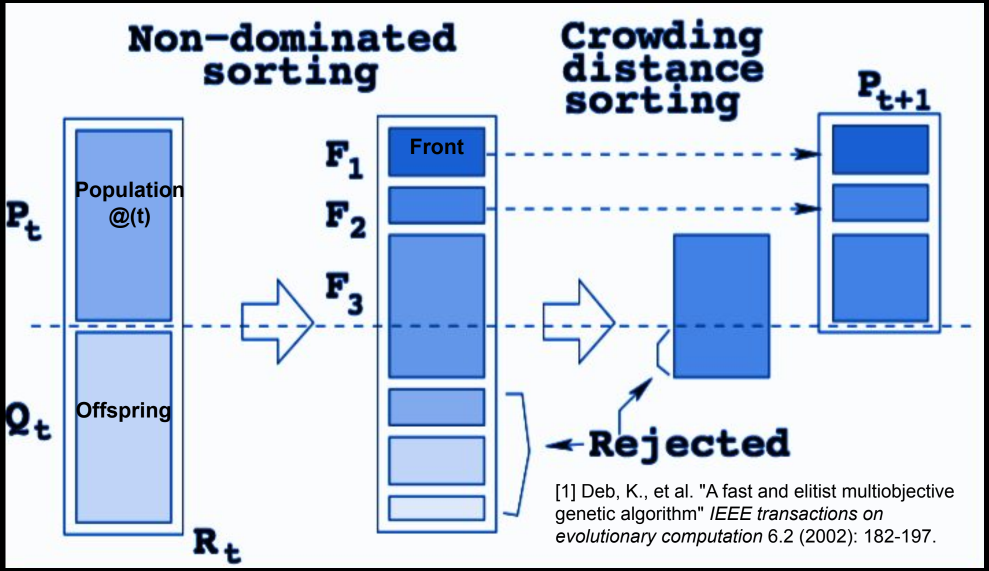
Multi-objective Optimization in Python
pymoo

jMetalPy

TMVA

Elitist Non-Dominated Sorting Genetic (NSGA)

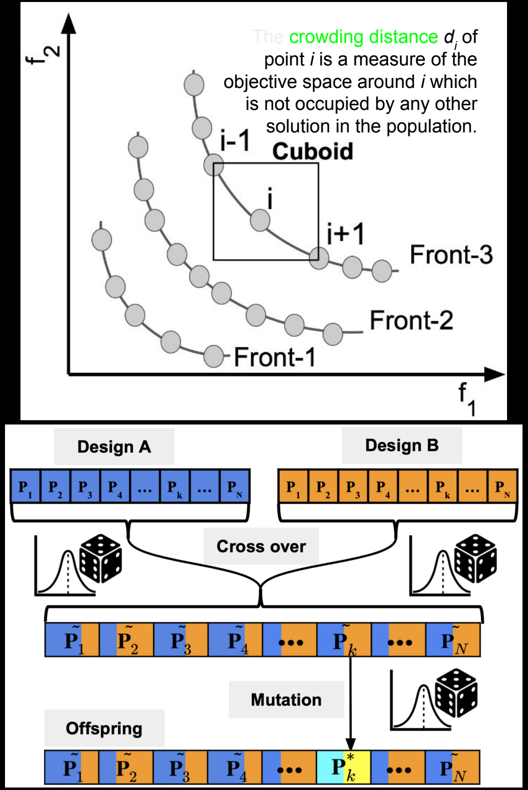
The population R_t is classified in non-dominated fronts. Not all fronts can be accommodated in the N slots of available in the new population P_{t+1}. We use **crowding distance** to keep those points in the last front that contribute to the highest diversity.



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

This is one of the most popular approach (>35k citations on google scholar), characterized by:

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



This is to illustrate Binary Cross-over

MOEA or MOBO ?

MOEA

- Has been widely used for solving MOO problems
- population /off spring — diversity —
- Relatively easier to implement
- Complexity relatively easy to compute
- Ideal — Cost of computing “cheap”
- Successful with large Design and Objective parameters
- No Map : “Design” “Objectives”

MOBO

- Has been around for a while, gaining popularity
- Sequential Strategy — global minimization
- Relatively harder to implement
- Complexity relatively easy to compute
- Ideal — simulations can be heavily parallelized
- Currently, Not recommended beyond 4-5 Objective parameters
- Can Map : “Design” “Objectives” — Fast simulator can be built

A roadmap for scalable optimization

AI Optimization block

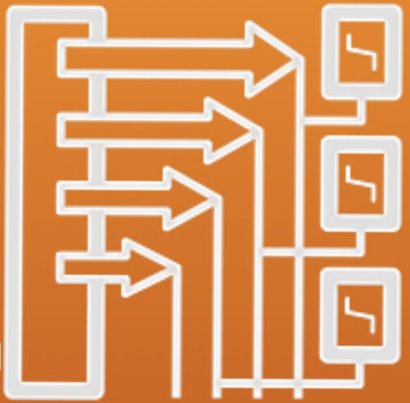
AI Suggested Design points

Checks

Evaluation of the Design points

Sort solutions
Approximate Pareto front
Suggest next set of design points

Parallelization



- Condition for parallelization

Single node

- Less queue time overhead
- Simulation, relatively cheap. (FastSim, Single particle tracking only)
- Queue time > Sim time
- Use Joblib to parallelize across a massive single node
- JLab's ifarm1900s machine (128 cores)

Multiple nodes

- Distribute across nodes in a given site.
- Moderate queue time overhead.
- Moderately heavy simulations(Calorimetry, PID etc)
- Queue time ≈ Sim time
- Use dask, MPI for scheduling.

Multiple sites

- Queue time << Sim time
- Full Sim, Physics-driven objectives.
- Across different OSG sites

Monitoring — MLOps

Need for better visualizations
Beyond 3D Pareto visualizations

Far Forward Updates

Problem	Optimize the momentum resolution subject to the non-homogenous Magnetic field and to increase occupancy at B0 ECAL.		
Objective Space = 2	**Objective Parameter**	**Remarks**	
	Momentum resolution (p_T)	Momentum range of 80 - 100 GeV/c is of interest and specifically proton tracks	
	B0 ECAL acceptance	Ratio of number of tracks before 1st tracking disk to the number of showers detected by B0ECAL	
Design Space = 4	**Design Parameter**	**Range [cm]**	**Least count for variation [cm]**
	Z_1	583.0 - 630.0	1.0
	$\Delta Z_2, \Delta Z_3, \Delta Z_4$	10.0 - 40.0	1.0
Constraints = 2	$Z_1 + \sum_{i=2,3,4} \Delta Z_i \leq 685.5$ cm		
	$\lvert Z_{i+1} + Z_i \rvert \geq 10.0$ cm		